GX1110/ GX1120
GtWave/WaveEasy
User's Guide

Arbitrary Waveform Function Generator
PXI Boards
GtWave and WaveEasy Software

User's Guide

Last updated February 6, 2015

MARVIN TEST SOLUTIONS

Safety and Handling

Each product shipped by Marvin Test Solutions 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 CD for installation. Store the original CD 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

Marvin Test Solutions products are warranted against defects in materials and workmanship for a period of 12 months. Marvin Test Solutions 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 Marvin Test Solutions Customer Technical Services Department via http://www.marvintest.com/magic/ - the Marvin Test Solutions on-line support system.

If You Need Help

Visit our web site at http://www.marvintest.com more information about Marvin Test Solutions 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 questions please use the following link: http://www.marvintest.com/magic/

You can also use Marvin Test Solutions technical support phone line (949) 263-2222. This service is available between 8:30 AM and 5:30 PM Pacific Standard Time.

Disclaimer

In no event shall Marvin Test Solutions 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 Marvin Test Solutions has been advised of the possibility for such damages.

Copyright

Trademarks

ATEasy®, CalEasy, DIOEasy®, DtifEasy, WaveEasy	Marvin Test Solutions, Inc., Geotest – Marvin Test Systems, Inc (prior company name)
C++ Builder, Delphi	Embarcadero Technologies Inc.
LabView, LabWindowstm/CVI	National Instruments
Microsoft Developer Studio, Microsoft Visual C++, Microsoft Visual Basic, .NET, Windows 95, 98, NT, ME, 2000, XP, VISTA, Windows 7 and 8	Microsoft Corporation

All other trademarks are the property of their respective owners.

Table of Contents

Chapter 1 - Introduction

Manual Scope and Organization

Manual Scope

This manual provides all the information necessary for installation, operation, and maintenance of the **GX1110** and **GX1120** (referred as **GX1100** or **GX11X0** in this manual), PXI Arbitrary Waveform Function Generator. The manual also covers the **GtWave** software package that includes the GX11X0 driver and the **WaveEasy** software for creating and editing waveform files. This manual assumes the reader has a general knowledge of PC based computers, Windows operating systems, and a general knowledge of modular test equipment.

Manual Organization

The GX11X0 manual is organized in the following manner:

Chapter	Content
Chapter 1 – Introduction	Introduces the GX11X0 manual and shows warning conventions used in the manual.
Chapter 2 – Overview	Provides the GX11X0 list of features, description of the board, architecture, specifications and the virtual panel description and operation.
Chapter 3 – Installation and Connections	Provides instructions about how to install a GX11X0 board and the GtWave software.
Chapter 4 –WaveEasy	Provides instructions about how to use the WaveEasy software used to create and edit waveform files.
Chapter 5 – Programming the Board	Provides a list of the GtWave software driver files, general purpose and generic driver functions, and programming methods. Discusses supported application development tools and programming examples.
Chapter 6 – Functions Reference	Provides a list of the GtWave driver functions. Each function description provides syntax, parameters, and any special programming comments.

Conventions Used in this Manual

Symbol Convention	Meaning
	Static Sensitive Electronic Devices. Handle Carefully.
	Warnings that may pose a personal danger to your health. For example, shock hazard.
	Cautions where computer components may be damaged if not handled carefully.
	Tips that aid you in your work.

Formatting Convention	**Meaning**
Monospaced Text	Examples of field syntax and programming samples.
Bold type	Words or characters you type as the manual instructs. For example: function or panel names.
Italic type	Specialized terms. Titles of other references and information sources. Placeholders for items you must supply, such as function parameters

Chapter 2 - Overview

This chapter describes **GX1110** and **GX1120** boards features, applications, description and specification. It also describes the **virtual panel application** that is used to control and display the board settings. The virtual panel application is part of the **GtWave** software that is supplied with the board and provides a driver, programming support to various language. Marvin Test Solutions' **WaveEasy software** is supplied with the GtWave software and supports the creation and / or importing of waveforms (WaveEasy requires purchase of a license to run in non-trial mode).

GX1110 Introduction

The GX1110 is a high performance, single-channel, 3U PXI waveform generator that offers a direct digital synthesis (DDS) based function generator and arbitrary waveform generation (AWG) functionality within one instrument. Built-in waveforms are available for use with either the DDS or AWG modes of operation and include Sine, Square, Triangle, Ramp up, DC and Noise. The output stage for DDS and AWG modes has different attenuators which are automatically set by the instrument software driver when amplitude and offset voltages are set in order to produce the best possible resolution.

GX1110 Features

The GX1110 provides the capabilities and features associated with both a DDS-based function generator and an AWG-based generator. When operating in DDS mode, the GX1110 offers the following features:

- 160 MHz sampling rate, 12-bit D to A converter
- 10 uHz frequency resolution
- Sine wave generation to 30 MHz
- Multi-pole, 35 MHz elliptical low pass filter
- Programmable output amplitude (3 digits resolution) up to 8V peak to peak
- Programmable offset 0 to +/- 4 volts, independent of amplitude range

When operating in AWG mode, the GX1110 offers the following features:

- Programmable sample rate from 10 mS/s to 100 MS/s
- 2 M sample waveform memory
- Looping capability and defined start / stop memory locations for waveform generation
- Programmable output amplitude and offset
- Programmable output amplitude (3 digits resolution) up to 8V peak to peak
- Programmable offset 0 to +/- 4 volts, independent of amplitude range

GX1110 Applications

- Video
- Navigation
- Radar
- Sonar
- Communications
- Converter Testing
- Filter Design & Test
- Computer Peripherals

GX1110 Board Description

The GX1110 contains a PXI interface device as well as an FPGA which is configured to support DDS or AWG operation which essentially provides all of the digital logic for waveform generation. Configuration of the FPGA is performed at power up or via a driver function call prior to programming the module's waveform generation parameters.

Analog components include a 12 bit D to A converter as well as a programmable offset generator, a programmable output amplifier, and multi-pole low pass filters. The output stage is common for both DDS and AWG operation modes with different filters to optimize noise and spectral performance, based on specific mode of operation.

The AWG includes sequence and a waveform memory configured as 2M by 12 bits. A PLL based clock generator provides a programmable sample clock rate from 10 mHz to 100 MHz with a resolution of 4 digits or .01 Hz - whichever is lower. The minimum waveform sequence length is 4 points. Waveform sequencing includes the ability to loop continuously, loop once, sequence between defined start and stop addresses or loop (burst) N times. The output is filtered by a 3-pole, 35 MHz, low pass Bessel filter.

The DDS mode of operation requires that the FPGA be reconfigured to support 2K by 12 bits waveform memory. Standard waveforms are loaded into the waveform memory when the FPGA is configured, eliminating memory load times at run-time. Standard waveforms include sine, triangle, ramp (up or down), pulse, square and noise. The DDS generator operates at a fixed 160 MS/s and generates frequencies from 10uHz to 30 MHz with resolution of 10 uHz. The output is filtered by a 6-pole, 35 MHz, low pass elliptical filter. Figure 2-1 shows the GX1110 with its front panel connectors.

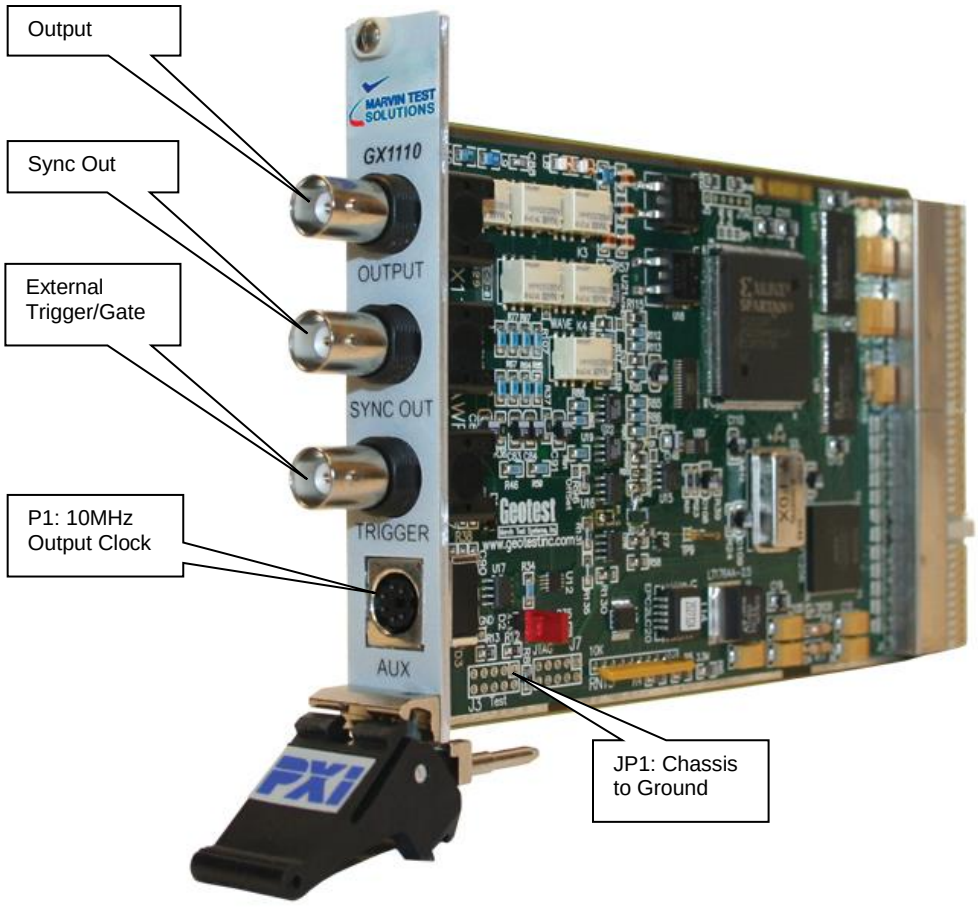


Figure 2-1: GX1110 Board Side View

GX1110 Architecture

A block diagram of the GX1110 is shown below in Figure 2-2 and 2-3 illustrates the GX1110's architecture in AWG and DDS modes. The board communicates with the host computer using the PCI interface. The software driver and application software configures board resources including the configuration / loading of the board's FPGA for AWG or DDS mode. The board has 3 BNC connectors located on the front panel for generator output, sync output, and trigger / gate input. A 6 pin PS2 connector located on the front panel provides connections for an external AM / FM modulation input, an external clock reference input, a marker output, and a clock reference output.

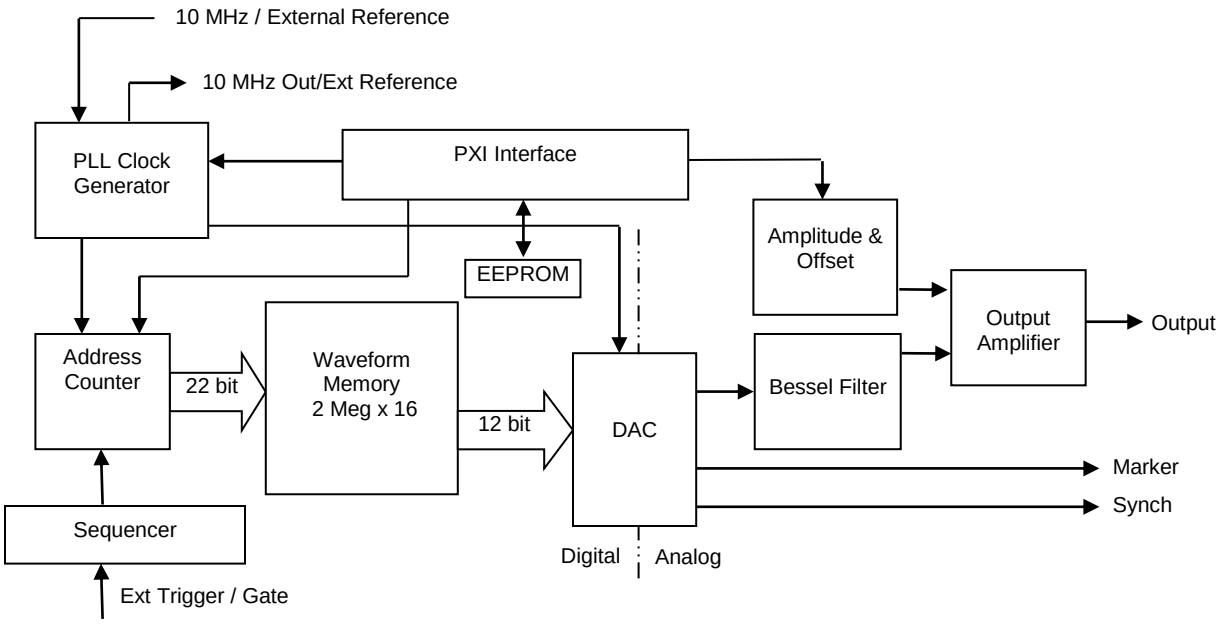


Figure 2-2: GX1110 Block Diagram - AWG Mode

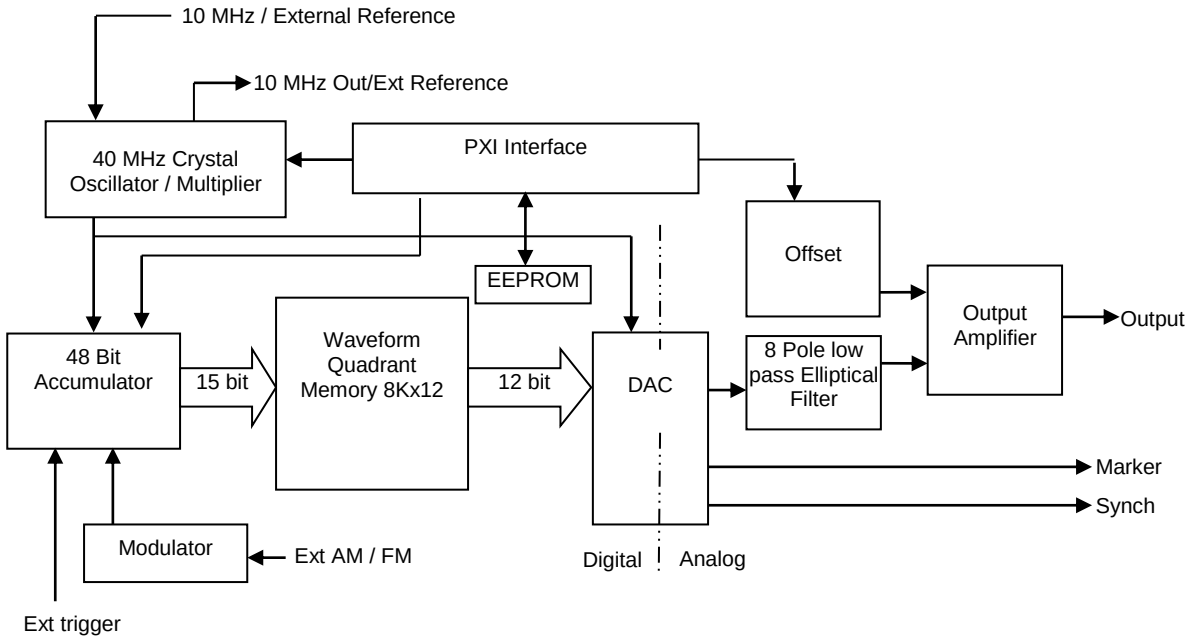


Figure 2-3: GX1110 Block Diagram - DDS Mode

GX1110 Specifications

The following table outlines the specifications of the GX1110.

Standard Waveform			
Sine, triangle, square, pulse, ramp up, ramp down, sinc (sin(x)/x), Gaussian pulse, noise			
Maximum Frequency (FS=160 MHZ)	Waveform	DDS Mode	AWG Mode
	Sine	30 MHz	6.25 MHz
	Triangle	1 MHz	6.25 MHz
	Ramp	1 MHz	6.25 MHz
	Pulse		6.25 MHz
	Square	30 MHz	50 MHz
	Noise		1 MHz
Arbitrary Generator Mode			
Sample Rate	10 mS/s to 100 MS/s		
Sample Rate Resolution & Accuracy	4 digits or .01 Hz, whichever is lower .01% accuracy		
Vertical Resolution	12 bits		
Waveform Memory	2 M samples		
DDS Mode			
Frequency Range, Resolution, And Accuracy	10 uHz to 30 MHz 10 uHz, resolution 50 ppm accuracy (OCXO option available for higher accuracy)		
Non-Harmonic Spurious Components	< 60 dBc (DC to 1 MHz)		
Distortion (2nd Harmonic Relative To Carrier)	< -60 dBc at < 200 KHz < -35 dBc at < 2.0 MHz		

	< -20 dBc at < 20 MHz
Modulation	AM: Int./Ext. 0 -100%, DC to 20 KHz FM: Int./ Ext. FM Deviation: 0-100% (2 x Fc) FM Modulation: DC to 20 KHz
Programmable Phase Offset	Range: 0-360 degrees Resolution: 0.1 degrees
Main Output	
Connector	Front panel BNC
Output Mode	On / Off
Output Impedance	50 Ω, +-/1%
Protection	Protected against shorts to ground and over-voltage
Amplitude Range	100 mV to 8Vpk-pk into 50 Ω; double into an open circuit. 0.01, 0.03, 0.1, 0.3, 1, 3, and 8 volt p-p FS ranges
Amplitude Resolution	3 digits
Amplitude Accuracy	+/- (2% of programmed value + 5 mV) <1V +/-(1% of programmed value + 10 mV) =>1V
Offset Range	0 to ±4 V, amplitude dependent
Offset Resolution	3 digits
Offset Accuracy	±(1% of programmed value + 20 mV)
Rise Time	< 15 ns into a 50 ohm load (10% to 90% full scale step)
Aberrations	< 5% of p-p amplitude, +/- 50 mV
Flatness	+/- .5 dB to 10 MHz, +/- 1 dB to 30 MHz
Filters	35 MHz, 6-pole elliptical LPF; DDS mode 35 MHz, 3-pole Bessel LPF; AWG mode

Sync Output	
Connector	Front Panel BNC. Output is synchronous with output waveform
Output Impedance	50 Ω, ±1%
Level	TTL Compatible

Marker Output	
Positive pulse, user programmable and synchronous with the waveform output.	
Output	Front Panel PS2 connector One Of 8 PXI Trigger Lines
PS2 Output Impedance	50 Ω, ±1%
PS2 Output Level	TTL Compatible

Trigger Modes	
Continuous	Output Is Continuously Generated
Triggered	Output waveform is triggered by an external or software triggered event. The external trigger signal edge may be a rising or falling edge.
Start / Stop	Output waveform is triggered by a trigger signal edge. Waveform is continuously generated until the occurrence of a trigger edge. The trigger signal may be external or software generated. The external trigger signal edge may be ther rising and or falling edge.
Gated	Same as triggered mode except that the waveform is generated for as long as the gate signal stays high when the gate signal goes low the output goes quiescent.
Gated Hold	Same as gated mode except when the gate signal goes high the waveform freezes until the gate signal goes low.
Burst	Output waveform will become active on the occurrence of a trigger edge. The selected waveform is generated for a preset number of cycles between 1 and 1,048,576. Output will then disable.

Trigger Sources	
Software Trigger	Generate A Trigger Via A Function Call
Internal Trigger Generator	Programmable Trigger Generator Frequency Range: 0.01 Hz– 50 MHz
External Trigger	External Trigger Input Via The External Trigger / Gate Input. Active Trigger Edge Is Programmable.
PXI Star Trigger	Select PXI Star Trigger As A Trigger Source
PXI Trigger	Select One Of 8 PXI Triggers For Trigger Functionality Note: – The PXI Trigger Can Be Used With Other Trigger Source Selections.
External Trigger / Gate Input	
Connector	Front Panel BNC
Impedance	10K ohm nominal
Threshold Level	TTL
Repetition Rate	DC To 10 MHz
Min Pulse Width	50 ns
Slope	Positive Or Negative Going Edge
Reference Clock	
Internal Reference	40 Mhz Oscillator Accuracy: 50 ppm
External Reference Input	PXI 10 MHz External Clock Reference Input (10 MHz)
External Clock Reference Input	Front panel PS2 connector Input impedance: 10k Ohm Nominal Threshold level: TTL

Reference Clock Output	
Connector	Front Panel PS2
Frequency	10 MHz
Output Level	TTL
Output Impedance	50 Ohms

Waveform Modulation	
Modulation	AM or FM
Modulation Source	Internal or External
Internal Modulation Waveform	Sine, square, triangle, ramp, noise, Frequency range: 0.01 Hz to 20 KHz
Modulation Range	0 to 100%
External Modulation Input	Front panel PS2 Input voltage range: 5 volts p-p for 100% modulation Input impedance: 10 K ohms nominal Bandwidth: DC to 20 KHz

General	
Power Requirements	15 W (Max.)
Current Consumption (Max)	+5 V @ 1 A +12 V @ 300 mA -12 V @ 300 mA +3.3 V @ 1 A
Weight	Approx. 12 oz.
Size	3U, single slot
Temperature Operating Storage	0° C to 50° C -20° C to 70° C

Humidity (Non-Condensing)	5% to 95%, < 70° C
Safety	Designed to meet IEC 1010-1, UL 3111-1, and CSA 22.2#1010
Note: Specifications are subject to change without notice	

GX1120 Introduction

The GX1120 is a 16-bit high performance dual-channel 3U form factor PXI board that offers function generator and arbitrary waveform generator functionality within one instrument. Each channel is fully independent and can be programmatically configured as a function generator or arbitrary waveform generator. Each channel's sample clock, output level, offset level, and waveform setting can be programmed independently in arbitrary waveform generator mode both channels share 32M samples of waveform data with 16 bits of vertical resolution. Maximum sample rate is 250 MHz for each channel. In addition a third mode of operation offers a 400MHz arbitrary waveform generator by combining both channels into one. Built-in waveforms are available for use with both the function generator and arbitrary waveform generator modes of operation and include Sine, Triangle, Ramp, Noise and pulse waveforms.

GX1120 Features

The GX1120 provides the capabilities and features associated with both a DDS-based function generator and an AWG-based generator. When operating in DDS mode, the GX1120 offers the following features:

Arbitrary Generator Mode

- Built-in waveforms: Sine, Triangle, Ramp up, Ramp down, Pulse, Square, Noise
- Arbitrary Generator Sample Rate 100 mS/s to 250 MS/s Arbitrary Generator Multiplexed Channel Sample Rate 400 MS/s
- Sample Rate Resolution: 4 digits or 10 ps (whichever is smaller)
- Vertical Resolution: 16 bits
- Waveform Memory: 32 M samples
- Waveform Sequencing: Loop in a defined segment of memory or Loop N Times
- Sequencing Rate: 0.01 Hz to 10 MHz (max)
- Filter: 500 MHz, 7 pole low pass Bessel filter

DDS Mode

- Built-in waveforms: Sine, Triangle, Ramp up, Ramp down, Pulse, Square,
- Frequency Range 1 uHz to 100 MHz
- Resolution 12 digits
- Output Filter: 100 MHz, 9 pole elliptical LPF
- Waveform Memory: 8K samples

Main Output

- Protected against shorts to ground, over-voltage and over-current.
- Output Impedance: 50 Ω, +-/1%
- Programmable Amplitude Range: 10 mV to 10Vpk-pk into 50 Ω (double into an open circuit)
- Amplitude Resolution: 4 digits (9.999 V)
- Programmable Offset Range of 0 to ±5 V (independent of programmed output level), resolution of 1 mV or 4 digits Sweep Modes: Linear or logarithmic, up or down

Modulation

- AM: 0.01 Hz to 20 KHz (internal), Sine, square and triangle modulation, 0% to 100% modulation, DC to 50 KHz (external input)
- FM: 0.01 Hz to 20 KHz (internal), Deviation: 0 to 50% of the carrier frequency, DC to 50 KHz, (external)
- FSK: 0.01 Hz to 1 MHz (internal), DC to 10 MHz (external input)
- Phase Modulation: 0.01 Hz to 20 KHz, Sine, square and triangle modulation, Deviation: 0 to 360 degrees, DC to 50 KHz (external input)

Trigger

- Modes: Continuous, Triggered, Gated, Burst
- Sources: Internal, external or PXI trigger.
- Internal Programmable Trigger Generator: 1uSec to 100 Sec

Connectors

- Sync Output (one per channel)
- Marker Output (one per channel)
- External Trigger(one per channel)
- External Input Clock
- External Output Clock

GX1120 Applications

- Video
- Navigation
- Radar
- Sonar
- Communications
- Converter Testing
- Filter Design & Test
- Computer Peripherals

GX1120 Board Description

The GX1120 contains a PXI interface device as well as an FPGA which is configured to support DDS or AWG operation which essentially provides all of the digital logic for waveform generation. Configuration of the FPGA is performed at power up or via a driver function call prior to programming the module's waveform generation parameters.

Analog components include a 12-bit A to D converter that supports external modulation functionality as well as a programmable offset generator, a programmable output amplifier, and multi-pole low pass filters. The output stage is common for both DDS and AWG operation modes with two different filters to optimize noise and spectral performance, based on specific mode of operation.

The AWG includes sequence and waveform memory configured as 32M by 16 bits. A PLL based clock generator provides a programmable sample clock rate from 10 mHz to 250 MHz with a resolution of 4 digits or .01 Hz - whichever is lower. The minimum waveform sequence length is 2 points. Waveform sequencing includes the ability to loop continuously, loop once, sequence between defined start and stop addresses or loop (burst) N times.

The DDS mode utilizes 8K by 16 bits of waveform memory. Standard waveforms are loaded into the waveform memory when the FPGA is configured, eliminating memory load times at run-time. Standard waveforms include sine, triangle, ramp (up or down), pulse, square and noise.

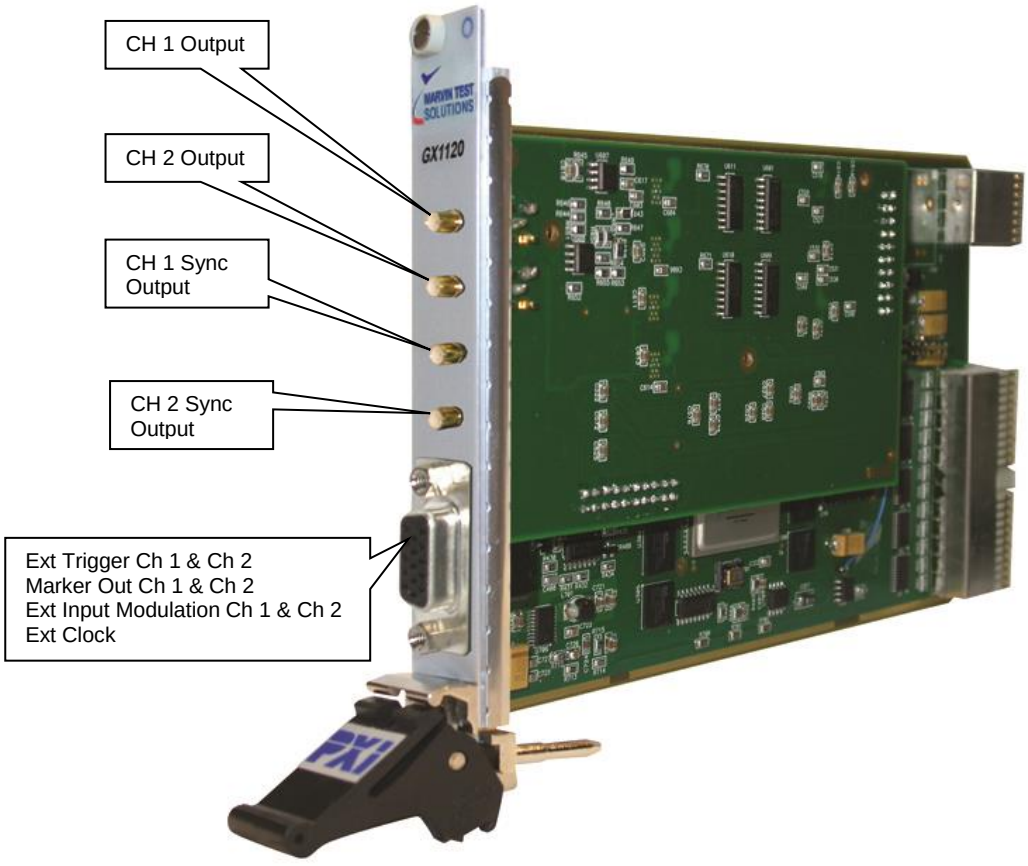


Figure 2-4: GX1120 Board Side View

GX1120 Specifications

The following table details the specifications of the GX1120.

Standard Waveforms			
Sine, triangle, square, pulse, ramp up, ramp down, noise			
Maximum Frequency (FS=250 MHZ)	Waveform	DDS Mode	AWG Mode
	Sine	100 MHz	
	Triangle	20 MHz	
	Ramp	20 MHz	
	Pulse		125 MHz
	Square	100 MHz	
	Noise		10 MHz

Arbitrary Waveform Generator Mode	
Sample Rate	100 mS/s to 250 MS/s
Multiplexed Channel Sample Rate	400 MS/s
Sample Rate Resolution & Accuracy	4 digits or 10 ps, whichever is lower 1 ppm, 15° C– 35° C
Vertical Resolution	16 bits
Waveform Memory	32 M samples

DDS Mode	
Frequency Range, Resolution, And Accuracy	1 uHz to 100 MHz 12 digits, resolution 1 ppm accuracy
Non-Harmonic Spurious Components	< 60 dBc (DC to 1 MHz) < 50 dBc to 20 MHz
Distortion (2nd Harmonic Relative	< -65 dBc at < 20 KHz

To Carrier)	< -60 dBc, 20 KHz to 100 KHz < -50 dBc, 100 KHz to 5 MHz < -30 dBc, 5 MHz to 80 MHz
Phase Noise	< -100 dBc / Hz (typical) at 1 MHz, 10 KHz offset from carrier
Modulation	AM: Int./Ext. 0 -100%, DC to 20 KHz FM: Int./ Ext. DC to 50 KHz
Programmable Phase Offset	Phase offset: 0-360 degrees, 0.1 degree resolution
Main Output	
Connector	Front panel SMB, one per channel
Output Mode	On / Off
Output Impedance	50 Ω, +-/1%
Protection	Protected against shorts to ground, over-voltage and over-current.
Amplitude Range	10 mV to 10Vpk-pk into 50 Ω; double into an open circuit.
Amplitude Resolution	4 digits (9.999 V)
Amplitude Accuracy (at 1 KHz)	+/- (1% of programmed value + 20 mV), 1 – 10 volt p-p output +/- (2% of programmed value + 5 mV), 10 mV to 1 V p-p output
Flatness	+/- 1% (0.1 dB) to 1 MHz +/- 1 dB to 50 MHz +/- 3 dB to 100 MHz
Offset Range	0 to ±5 V, independent of programmed output level
Offset Resolution	1 mV, or 4 digits, whichever is less.
Offset Accuracy	±(1% of programmed value + 10 mV)
Rise / Fall Time	< 6 ns into a 50 ohm load (10% to 90% full amplitude step)
Abberations	< 5% of p-p amplitude, +/- 20 mV

Asymmetry (Square Wave)	< 1 % of period, +/- 5 ns
Filters	DDS Mode: 100 MHz, 9 pole elliptical LPF ARB mode: 7 pole Bessel filter
Channel Phase Locking	Channels may be phase locked (0 to +/- 360 degrees) Resolution: 0.1 degree
Channel To Channel Skew	< 200 ps (phase locked mode)
Sync Output (one per channel)	
Connector	Front panel SMB output (one per channel), synchronous with output waveform
Output Impedance	50 Ω
Level	TTL Compatible
Marker Output (one per channel)	
Positive pulse, user programmable and synchronous with the waveform output.	
Connector	Front Panel PS2 connector One Of 8 PXI Trigger Lines
Impedance	50 Ω
Level	TTL Compatible
Modulation	
AM	0.01 Hz to 20 KHz (internal) Sine, square and triangle modulation 0% to 100% modulation DC to 50 KHz (external input)
FM	0.01 Hz to 20 KHz (internal) Deviation: 0 to 50% of the carrier frequency DC to 50 KHz, (external)
FSK	0.01 Hz to 1 MHz (internal) Deviation (Hi or Lo frequency shift): 1 u Hz to 100 MHz DC to 10 MHz (external input)

Phase Modulation	0.01 Hz to 20 KHz Sine, square and triangle modulation Deviation: 0 to 360 degrees DC to 50 KHz (external input)
Sweep Characteristics	
Sweep Modes	Linear or logarithmic, up or down
Sweep Time	1 ms to 500 sec
Sweep Trigger	Continuous, triggered or burst, internal, external or PXI trigger.
Waveform Sequencing (ARB Mode)	
Loop	Loop in a defined segment of memory
Loop N Times	Loop in a defined segment of memory N times, N is programmable from 1 to 999999 or loop continuously
Sequencing Rate	0.01 Hz to 10 MHz (max)
Trigger Modes	
Continuous	Output is continuously generated
Triggered	Output waveform triggered by external or software triggered event. The external trigger signal edge may be a rising or falling edge. One waveform cycle generated. 50 MHz trigger rate for Arb mode 20 MHz trigger rate for DDS mode
Gated	Same as Triggered mode except that the waveform is generated for as long as the gate signal stays true (logic one). When the gate signal goes false (logic zero) the output goes quiescent.
Burst	Output waveform will become active on the occurrence of a trigger edge. The selected waveform is generated for a preset number of cycles between 1 and 999,999. Output will then disable.
Trigger Sources	Internal, external or PXI trigger.

Internal Trigger	
Internal Trigger Generator Rate	Programmable Trigger Generator 1uSec to 100 Sec
Resolution	4 digits
Accuracy	+/- 0.01%
External Trigger Input	
Connector	Front panel DB-15 connector
Impedance	10K ohm nominal
Threshold Level	Variable from -5V to +5V, with 10 mV resolution
Repetition Rate	DC to 50 MHz
Min Pulse Width	10 ns
Slope	Positive or negative going edge.
Trigger Delay	0 – 15 sec, with 4 ns resolution
Trigger Hold	0 – 15 sec, with 4 ns resolution
External Input Clock	
Connector	Front panel DB-15
Input	External 10 MHz reference clock
Threshold Level	TTL
External Output Clock	
Connector	Front panel DB-15
Output	10 MHz reference clock
Level	TTL
Internal Reference Clock	
Time Base	50 MHz, +/- 1ppm

	PXI 10 MHz clock External 10 MHz clock
External Modulation Input	
Input Voltage Range	5 volts p-p for 100% modulation
Input Impedance	10K ohm nominal
Bandwidth	DC to 50 KHz DC to 10 MHz (FSK modulation)
Connector	Front panel DB-15
Input Voltage Range	5 volts p-p for 100% modulation
General	
Power Requirements	15 W (Max.)
Current Consumption (Max)	+5 V @ 0.3 A +12 V @ 0.5 A -12 V @ 0.2 A +3.3 V @ 2 A
Weight	Approx. 12 oz.
Size	3U, single slot
Temperature Operating Storage	0° C to 50° C -20° C to 70° C
Humidity (Non-Condensing)	5% to 95%, < 70° C
Safety	Designed to meet IEC 1010-1, UL 3111-1, and CSA 22.2#1010
CE Labeled	Yes
Note: Specifications are subject to change without notice	

Virtual Panel

The **GtWave** software includes a virtual panel program, which provides full access to the various configuration settings and operating modes. To fully understand the front panel operation, it is best to become familiar with the functionality of the board.

To open the virtual panel application, select **GtWave Panel** from the **Marvin Test Solutions**, **GtWave** menu under the **Start** menu. The GtWave Panel opens as shown here:

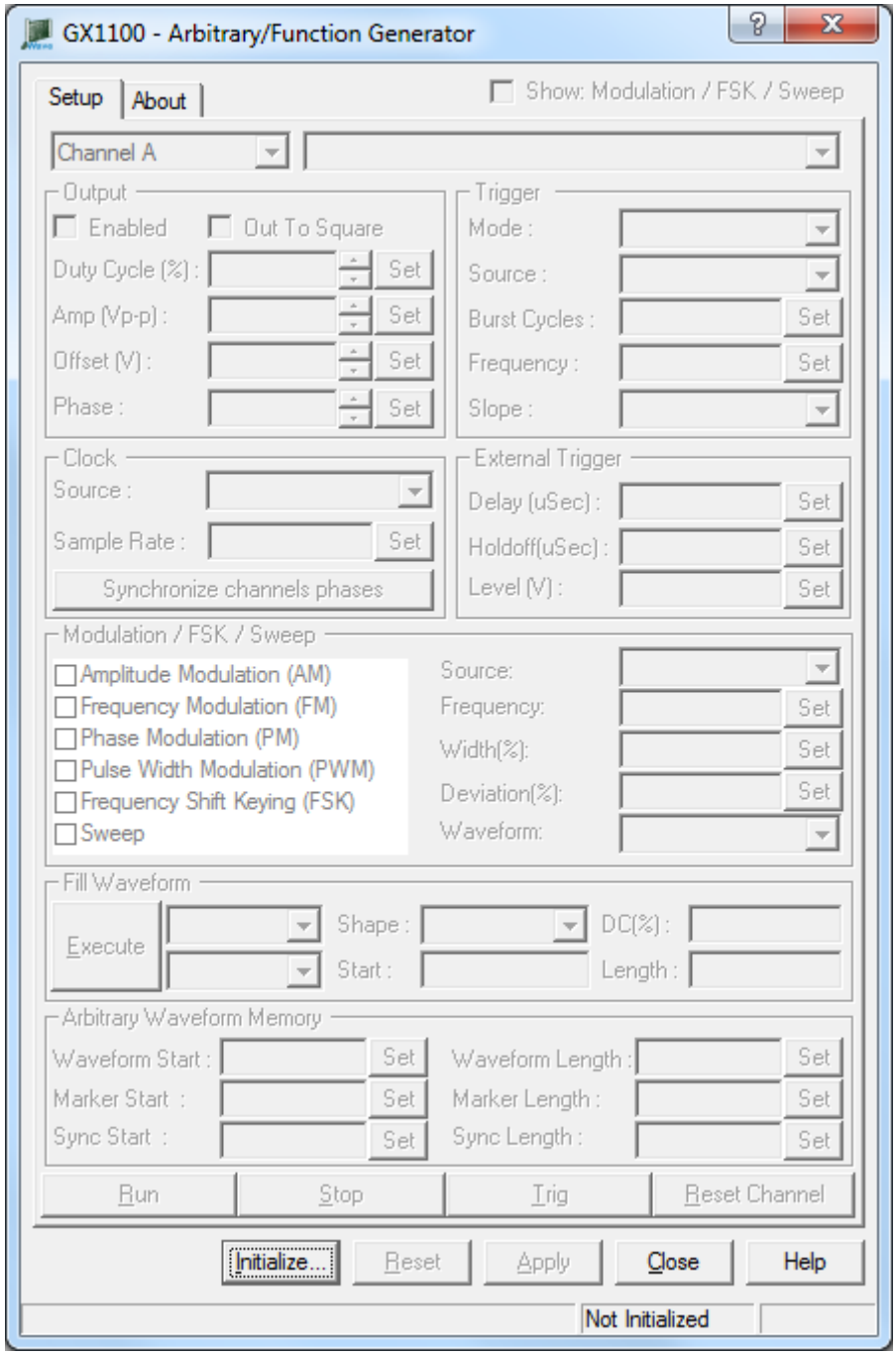


Figure 2-5: GtWave Virtual Panel (not Initialized)

The functionality of the panel button controls are described below:

Initialize – Opens the Initialize Dialog (see Initialize Dialog paragraph) in order to initialize the board driver. The current settings of the selected board will **not change after calling initialize**. The panel will reflect the current settings of the board after the Initialize dialog closes.

Reset – Resets the PXI board settings to their default state and clears the reading.

Apply – Applies changed settings to the board.

Close – Closes the panel. Closing the panel **does not affect** the board settings.

Help – Opens the on-line help window. In addition to the help menu, the caption shows a **What's This Help** button (?) button. This button can be used to obtain help on any control that is displayed in the panel window. To displays the What's This Help information click on the (?) button and then click on the control – a small window will displays the information regarding this control.

Virtual Panel Initialize Dialog

The Initialize dialog initializes the driver for the selected counter board. The counter settings **will not change** after initialize is called. Once initialize, the panel will reflect the current settings of the counter.

The Initialize dialog supports two different device drivers that can be used to access and control the board:

1. **Use Marvin Test Solutions' HW** – this is the device driver installed by the setup program and is the default driver. When selected, the **Slot Number** list displays the available counter boards installed in the system and their slots. The chassis, slots, devices and their resources are also displayed by the HW resource manager, **PXI/PCI Explorer** applet that can be opened from the Windows Control Panel. The PXI/PCI Explorer can be used to configure the system chassis, controllers, slots and devices. The configuration is saved to PXISYS.INI and PXIeSYS.INI located in the Windows folder. These configuration files are also used by VISA. The following figure shows the slot number 0x10D (chassis 1 Slot 13). This is the slot number argument (*nSlot*) passed by the panel when calling the driver **GtWaveInitialize** function used to initialize driver with the specified board.

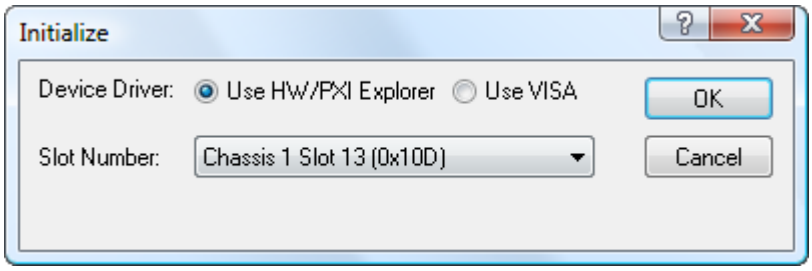


Figure 2-6: Initialize Dialog Box using Marvin Test Solutions' HW driver

2. **Use VISA** – this is a third party device driver usually provided by National Instrument (NI-VISA). When selected, the **Resource** list displays the available boards installed in the system and their VISA resource address. The chassis, slots, devices and their resources are also displayed by the VISA resource manager, **Measurement & Automation** (NI-MAX) and in Marvin Test Solutions's **PXI/PCI Explorer**. The following figure shows PXI7::10::INSTR as the VISA resource (PCI bus 7 and Device 10). This is VISA resource string argument (*szVisaResource*) passed by the panel when calling the driver **GtWaveInitializeVisa** function to initialize the driver with the specified board.

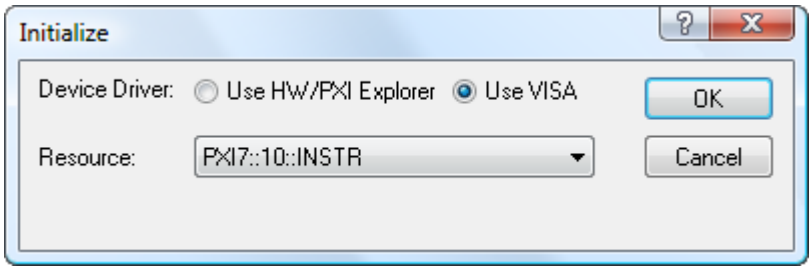


Figure 2-7: Initialize Dialog Box using VISA resources

Virtual Panel: Setup Page in Function Generator Mode

After the board is initialized, the panel is enabled and will display the current setting of the board. The following picture shows the **Setup** page settings when the board is in **Function Generator** mode**:**

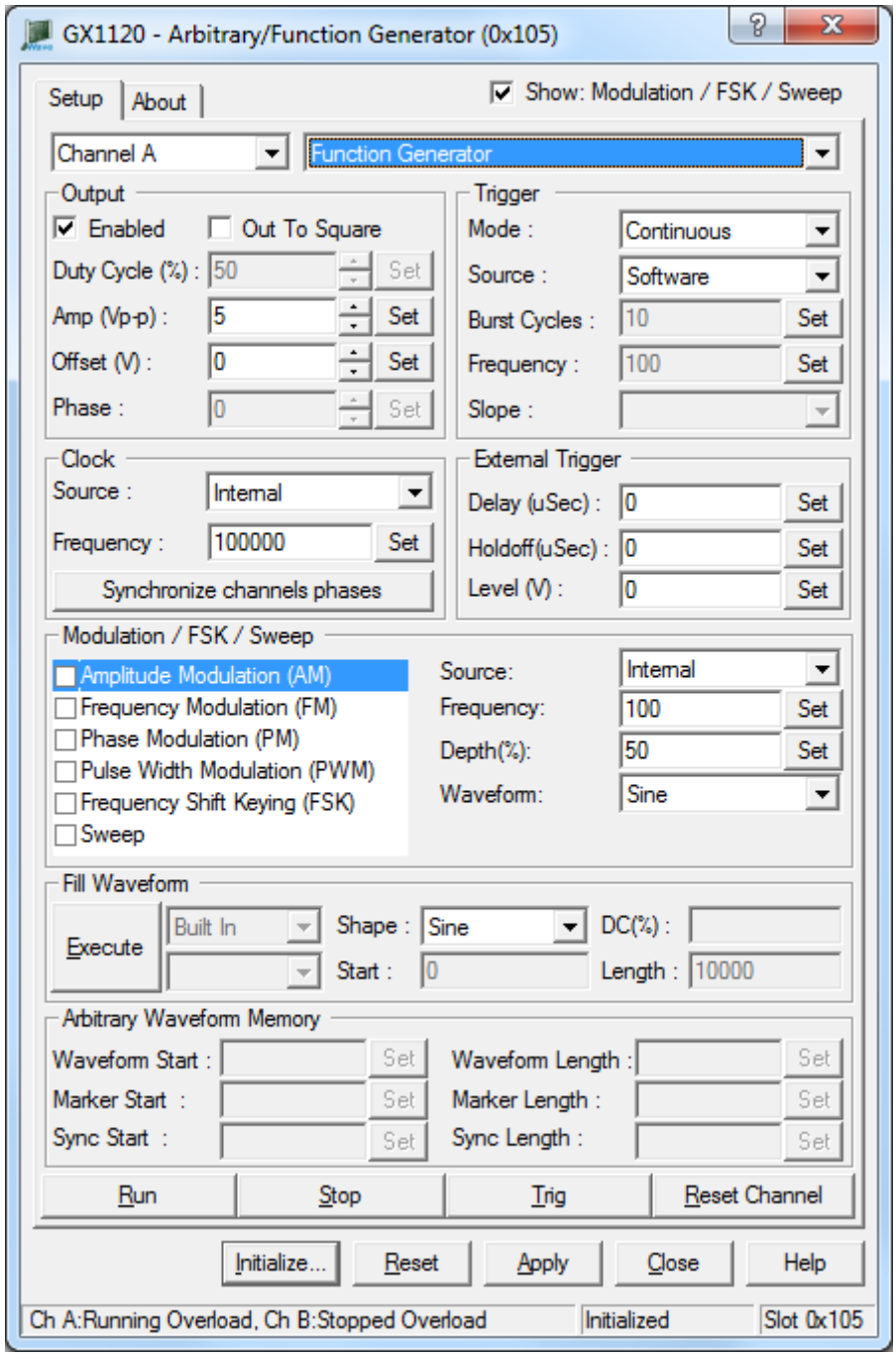


Figure 2-8: GX1120 Virtual Panel (Initialized) in Function Generator mode

The following controls are shown in the Setup page:

Channel (dropdown list): – Sets/displays the specified channel's settings.

Operation Mode (dropdown list): – Sets/displays the specified channel's operation mode, either Function Generator or Arbitrary waveform Generator, each channel can be programmed as follows:

- Function Generator mode
- Arbitrary Waveform Generator mode
- High speed Arbitrary Waveform Generator mode. In this mode both channels A and B are combined into one channel (Channel A) to produce a high speed Arbitrary Waveform Generator that can run as high as 400 MS/s (Gx1120 only).

Relevant API functions: **GtWaveSetOperationMode** .

Output (group box)

- **Output Enable** (check box) – Sets/displays the specified channel's output relay state.

Relevant API functions: GtWaveSetOutputState .

Output to square (check box) – Sets/displays the output to convert from a Output to square output. Enabled only in Function Generator mode. The function is only applicable when generating a sine waveform. When enabled, a comparator is connected to the sine waveforms and convert the output to a square waveform. The duty cycle of that square waveform can be programmed by calling the GtWaveFuncSetOutToSquareDutyCycle API. The output square wave is much cleaner waveform since it was derived directly from a sine wave.

Relevant API functions: **GtWaveFuncSetOutToSquareState** .

- **Duty Cycle (%)** (edit box) – Sets/displays the Output to square Duty Cycle. Output to square duty cycle, range is from 20% to 80%.

This function is only active when the Output to square state is enabled. The function is only applicable when generating a sine waveform. When enabled, a comparator is connected to the sine waveforms and convert the output to a square waveform. The duty cycle of that square waveform can be programmed by calling the GtWaveFuncSetOutToSquareDutyCycle API. The output square wave is much cleaner waveform since it was derived directly from a sine wave.

Relevant API functions: GtWaveFuncSetOutToSquareDutyCycle .

- **Amplitude** (edit box) – Sets/displays the waveform's peak-to-peak amplitude voltage.

GX1110: 0.1V to 8V (p-p) into 50 Ohm, 0.2V to 16V (p-p) into open circuit.

GX1120: 0.01V to 10V (p-p) into 50 Ohm, 0.02V to 20V (p-p) into open circuit.

Offset may be applied to the output to shift the signal either positive or negative. Offset and amplitude are inter-related, i.e. the maximum positive or negative peak voltage at any time cannot be more then +/-4V (GX1110) and +/-5V (GX1120).

Relevant API functions: GtWaveSetAmplitude .

- **Offset** (edit box) – Sets/displays the output's DC offset voltage.

GX1110: Output offset voltage ranges from -4.0V to +4.0V. The output level may be programmed from 0.1V to 8V p-p into 50 Ohm (0.2V to 16V p-p into an open circuit).

GX1120: Output offset voltage ranges from -5.0V to +5.0V. The output level may be programmed from 0.01V to 10V p-p into 50 Ohm (0.2V to 20V p-p into an open circuit).

Offset may be applied to the output to shift the signal either positive or negative. Offset and amplitude are inter-related, i.e. the maximum positive or negative peak voltage at any time cannot be more then +/- 4V (GX1110) and +/- 5V (GX1120).

Relevant API functions: GtWaveSetOffset .

- **Phase** (edit box) – Sets/displays the function generator's output start phase in units of degrees. Function Generator mode only, Output start phase in units of degrees, -360 to 360.

Relevant API functions: GtWaveFuncSetPhase .

Clock (group box)

- **Ref Source** (dropdown list) – Sets/displays the clock source, options are: internal 10MHz clock, external clock, PXI backplane 10MHz clock.

Relevant API functions: GtWaveSetReferenceClockSource .

- **Frequency** (edit box) – Sets/displays the waveform frequency. In Arbitrary Waveform Generator mode sets/displays the data point sampling rate.

GX1110: Function generator frequency, frequency range is 10uHz to 30MHz

GX1120: Function generator frequency, frequency range is 10uHz to 100MHz

Relevant API functions: GtWaveFuncSetFrequency .

Trigger (group box)

- **Mode** (dropdown list) – Sets/displays the trigger mode: Continuous, Triggered, Gated and Burst.

Continuous: The waveform generates continuously by repeatedly cycling through the waveform table using the programmed waveform parameters.

Triggered:Output is quiescent until triggered by an internal or external trigger, and then one waveform cycle is generated using the programmed waveform parameters.

Gated;The waveform generates continuously by repeatedly cycling through the waveform table as long as the gate is active

Burst: After a trigger is received, waveform generation will be executed for the number of cycles that were defined via the GtWaveGetTriggerBurstCount function

- Relevant API functions: GtWaveSetTriggerMode .
- **Source** (dropdown list) – Sets/displays the trigger source: Trigger source are as follows:

Software: Trigger source is software (immediate).

Internal: Trigger source is the internal programmable trigger generator.

External: Trigger source is an external input.

PXI Star Trig: Trigger source is the star trigger.

Alternate Channel: Trigger source is the alternate channel.

Relevant API functions: GtWaveSetTriggerSource .

- **Burst cycles** (dropdown list/edit box) – Sets/displays the trigger burst count. Trigger burst count can range from 1 to 9999999. When the trigger mode (by calling **GtWaveSetTriggerMode** function) is set to burst mode, which is an extension of the triggered mode, the generator can be programmed to output a pre-determined number of waveforms. The sources to trigger a burst are the same as for the trigger mode.

Relevant API functions: GtWaveSetTriggerBurstCount .

- **Frequency** (dropdown list /edit box) – Sets/displays the trigger generator's internal frequency.

GX1110: Trigger internal frequency range can be from 0.01Hz to 50MHz.

GX1120: Trigger internal frequency range can be from 0.01Hz to 1 MHz.

Relevant API functions: GtWaveSetTriggerInternalFrequency .

- **Edge** (dropdown list box) – Sets/displays the External Trigger Slope: 0-Positive edge, 1-Negative edge.

Relevant API functions: GtWaveSetTriggerEdge .

External Trigger (group box)

- **Delay (uSec)** (edit box) – Sets/displays the external trigger delay in uSec. External trigger delay range is 0 - 15 sec, with 4 ns resolution. Units are in uSec, e.g. if pdDelay is equal 10 then the delay is 10uSec.

Relevant API functions: GtWaveSetOperationMode .

- **Holdoff (uSec)** (edit box) – Sets/displays the external trigger hold off in uSec. External trigger holdoff range is 0 - 15 sec, with 4 ns resolution. Units are in uSec, e.g. if pdDelay is equal 10 then the hold off is 10uSec.

Relevant API functions: GtWaveSetTriggerHoldoff .

- **Level (V)** (edit box) – Sets/displays the external trigger level in volts. External trigger level range is -5V to +5V with 10mV of resolution.

Relevant API functions: GtWaveSetTriggerLevel .

Modulation / FSK / Sweep (group box)

Modulation / FSK / Sweep (list): Sets/displays the active Modulation, FSK and Sweep functionality. Check means enabled. **Amplitude Modulation (AM)**

- **Source** (dropdown list) – Sets/displays the Amplitude Modulation source, modulation source can be rather internal or external.

Relevant API functions: GtWaveSetAmSource .

- **Frequency** (edit box) – Sets/displays the AM modulation frequency,

GX1110: AM modulation frequency range is from 0.01Hz to 20KHz

GX1120: AM modulation frequency range is from 0.01Hz to 20KHz.

Relevant API functions: GtWaveSetAmFrequency .

- **Depth** (edit box) – Sets/displays the AM modulation depth in percentage in percentage 0% (no modulation) to 100% (full modulation).

Relevant API functions: GtWaveSetAmDepth .

- **Waveform** (dropdown list) – Sets/displays the AM modulation waveform, Standard waveform can be one of the following: Sinusoidal waveform, Square waveform, Triangular waveform, Positive ramp waveform, Negative ramp waveform, Constant voltage, White noise

Relevant API functions: GtWaveSetAmWaveform .

Frequency Modulation (FM)

- **Source** (dropdown list) – Sets/displays the FM Modulation source, modulation source can be rather internal or external.

Relevant API functions: GtWaveFuncSetFmSource .

- **Frequency** (edit box) – Sets/displays the FM modulation frequency,

GX1110: FM modulation frequency range is from 0.01Hz to 20KHz

GX1120: FM modulation frequency range is from 0.01Hz to 20KHz.

Relevant API functions: GtWaveFuncSetFmFrequency .

- **Deviation** (edit box) – Sets/displays the FM deviation frequency, FM deviation frequency range is 0Hz to 2MHz.

Relevant API functions: GtWaveFuncSetFmDeviation .

- **Waveform** (dropdown list) – Sets/displays the FM modulation waveform, Standard waveform can be one of the following: Sinusoidal waveform, Square waveform, Triangular waveform, Positive ramp waveform, Negative ramp waveform, Constant voltage, White noise

Relevant API functions: GtWaveFuncSetFmWaveform .

Phase Modulation (PM)

- **Source** (dropdown list) – Sets/displays the Phase Modulation source, modulation source can be rather internal or external.

Relevant API functions: GtWaveFuncSetPmSource .

- **Frequency** (edit box) – Sets/displays the Phase modulation frequency,

GX1120: Phase modulation frequency range is from 0.01Hz to 20KHz.

Relevant API functions: GtWaveSetFuncPmFrequency .

- **Deviation** (edit box) – Sets/displays the Phase Modulation deviation, Phase Modulation deviation range is 0 to 360 degrees.

Relevant API functions: GtWaveFuncSetPmDeviation .

- **Waveform** (dropdown list) – Sets/displays the Phase modulation waveform, Standard waveform can be one of the following: Sinusoidal waveform, Square waveform, Triangular waveform, Positive ramp waveform, Negative ramp waveform, Constant voltage, White noise

Relevant API functions: GtWaveFuncSetPmWaveform .

Pulse Width Modulation (PWM)

- **Source** (dropdown list) – Sets/displays the Pulse Width Modulation source, modulation source can be rather internal or external.

Relevant API functions: GtWaveFuncSetPwmSource .

- **Frequency** (edit box) – Sets/displays the Pulse Width modulation frequency,

GX1120: Pulse Width modulation frequency range is from 0.01Hz to 20KHz.

Relevant API functions: GtWaveFuncSetPwmFrequency .

- **Width** (edit box) – Sets/displays the Phase Modulation width, Pulse-width range is 0% to 100%.

Relevant API functions: GtWaveFuncSetPwmWidth .

Deviation (edit box) – Sets/displays the Phase Modulation deviation, Pulse-width modulation deviation (gain) is 0% to 100%.

Relevant API functions: GtWaveFuncSetPwmDeviation .

- **Waveform** (dropdown list) – Sets/displays the Phase modulation waveform, Standard waveform can be one of the following: Sinusoidal waveform, Square waveform, Triangular waveform, Positive ramp waveform, Negative ramp waveform, Constant voltage, White noise

Relevant API functions: GtWaveFuncSetPwmWaveform .

Frequency Shift Keying (FSK)

- **Source** (dropdown list) – Sets/displays the Frequency shift keying source,source can be rather internal or external.

Relevant API functions: GtWaveFuncSetFskSource .

- **Low Frequency** (edit box) – Sets/displays the Frequency shift keying modulation low frequencies,

GX1110: Range from 10.0E-6 to 50.0E+6.

GX1120: Range from 1.0E-6 to 100.0E+6

Relevant API functions: GtWaveFuncSetFskFrequencies .

- **High Frequency** (edit box) – Sets/displays the Frequency shift keying modulation high frequencies,

GX1110: Range from 10.0E-6 to 50.0E+6.

GX1120: Range from 1.0E-6 to 100.0E+6

Relevant API functions: GtWaveFuncSetFskFrequencies .

Rate (edit box) – Sets/displays the FSK switching rate for Frequency shift keying modulation. Rate of switching between the two frequencies for frequency shift keying modulation.

GX1110: The rate can be set between 0.01Hz to 100KHz.

GX1120: The rate can be set between 0.01Hz to 1MHz.

Relevant API functions: GtWaveFuncSetFskRate .

Sweep

- **Mode** (edit box) – Sets/displays the Sweep mode, linear or Logarithmic.

Relevant API functions: GtWaveFuncSetSweep .

- **Start Frequency** (edit box) – Sets/displays the Sweep start freqancy,

GX1120: Range from 10.0E-6 to 100.0E+6

Relevant API functions: GtWaveFuncSetSweep .

- **Stop Frequency** (edit box) – Sets/displays the Sweep stop Frequency,

GX1120: Range from 10.0E-6 to 100.0E+6

Relevant API functions: GtWaveFuncSetSweep .

- **Time** (edit box) – Sets/displays the Sweeptime in seconds,

GX1120: Range from 0.001sec to 500 sec

Relevant API functions: GtWaveFuncSetSweep .

Fill Waveform (group box)

- **Execute** (button) – Executes the specified waveform load/show file settings.
- **Source** (dropdown list) – Sets/displays the fill waveform source, built in, file or WaveEasy.

Relevant API functions: GtWaveArbFileLoad, GtWaveArbFillPredefinedWaveform.

- **Load/Save** (dropdown list) – Select if the waveform should be loaded from the source or saved.

Relevant API functions: GtWaveArbFileLoad, GtWaveArbFileSave.

- **Shape** (dropdown list) – Sets/displays the waveform shape.

Relevant API functions: GtWaveFuncSetWaveform.

- **DC** (**%)(**edit box) – Sets/displays the square wave duty cycle in percentage (pulse generator).

Relevant API functions: GtWaveFuncSetSquareWaveDutyCycle.

- **Start (**edit box) – Sets/displays the start step number to load/save to or from.

Relevant API functions: GtWaveArbFileLoad.

- **Length (**edit box) – Sets/displays the number of steps to load/save.

Relevant API functions: GtWaveArbFileLoad.

Run (group box)

- **Run** (button) – Enables the board for running.

Relevant API functions: GtWaveRun .

- **Stop** (button) – Disables the board from running.

Relevant API functions: GtWaveStop.

- **Trig** (button) – Issue a software trigger.

Relevant API functions: GtWaveTrig.

- **Reset** Channel (button) – Resets the specified channel.

Relevant API functions: GtWaveResetChannel.

Virtual Panel: Setup Page in Arbitrary Waveform Generator Mode

After the board is initialized, the panel is enabled and will display the current setting of the board. The following picture shows the **Setup** page settings when the board is in **Arbitrary Waveform Generator** mode**:**

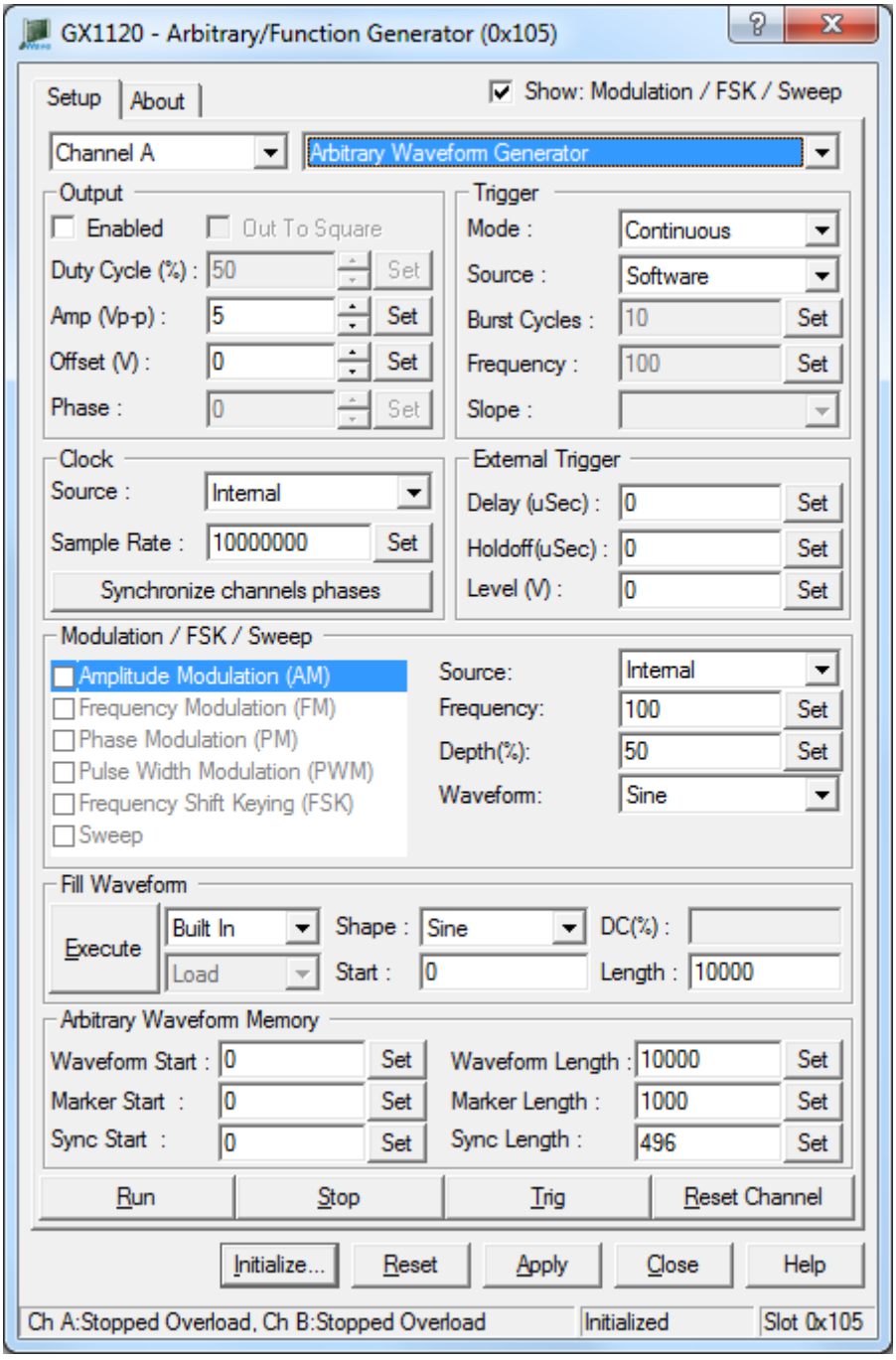


Figure 2-9: GX1120 Virtual Panel (Initialized) in Arbitrary Waveform Generator Mode

The following controls are shown in the Setup page:

Arbitrary Waveform Memory (group box)

- **Start Address (**edit box) – Sets/displays the Arbitrary Waveform Memory start address.

GX1110: Waveform start address, any even value between 0 and 2097148 that can be divided by 4.

GX1120: Waveform start address, any even value between 0 and 33554430 that can be divided by 2.

Relevant API functions: GtWaveArbSetWaveformLength .

- **Length (**edit box) – Sets/displays the Arbitrary Waveform Memory waveform length.

GX1110: Waveform length can be any value between 4 and 2097152 that can be divided by 4. E.g. to run a waveform of 4096 steps starting from step 0, start address will be 0 and length be 4096.

GX1120: Waveform length can be any value between 2 and 33554432 that can be divided by 2.

Relevant API functions: GtWaveArbSetWaveformLength .

- **Marker Start (**edit box) – Sets/displays the Arbitrary Waveform Memory marker start address.

GX1110: Waveform start address, any even value between 0 and 2097151.

GX1120: Waveform start address, any even value between 0 and 33554431.

Relevant API functions: GtWaveArbSetMarker.

- **Marker Length (**edit box) – Sets/displays the Arbitrary Waveform Memory marker length.

GX1110: Waveform length can be any value between 1 and 2097152. If dwLength is -1 will automatically fill all the steps between the start address and the end of the memory.

GX1120: When waveform length is set to be less than 2048, the marker can only start on multiples of 2 points and have length of multiples of 2. Otherwise, marker can only start on multiples of 8 points and have length of multiples of 8 and be between 1 and 33554432.

Relevant API functions: GtWaveArbSetMarker.

- **Sync Start (**edit box) – Sets/displays the Arbitrary Waveform Memory sync start address.

GX1110: sync start address, any even value between 0 and 2097151.

GX1120: sync start address, any even value between 0 and 33554431.

Relevant API functions: GtWaveArbSetSync.

- **Sync Length (**edit box) – Sets/displays the Arbitrary Waveform Memory sync length.

GX1110: sync length can be any value between 1 and 2097152. If dwLength is -1 will automatically fill all the steps between the start address and the end of the memory.

GX1120: When waveform length is set to be less than 2048, the sync can only start on multiples of 2 points and have length of multiples of 2. Otherwise, sync can only start on multiples of 8 points and have length of multiples of 8 and be between 1 and 33554432.

Relevant API functions: GtWaveArbSetSync.

Virtual Panel: About Page

Clicking on the **About** tab will show the **About** page as shown in _____

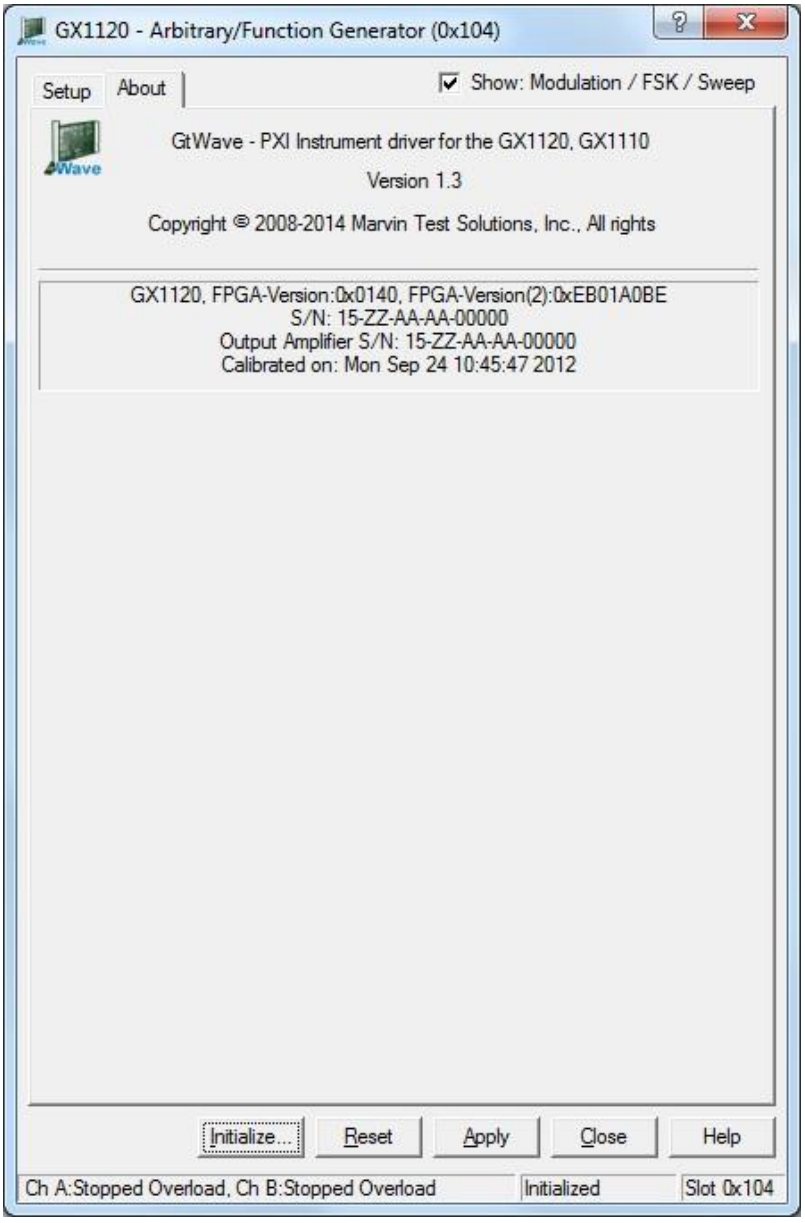


Figure 2-10: GX1110/GX1120 Virtual Panel – About Page

The following controls are shown in the About page:

The top part of the **About** page displays version and copyright of the GtWave driver. The bottom part displays the board summary.

The **About** page also contains a button **Upgrade Firmware…** which is used to upgrade the board's FPGA firmware. This button maybe used only when the board requires upgrade as directed by Marvin Test Solutions support. The upgrade requires a firmware file (.jam) that is written to the board FPGA. After the upgrade is complete, you must shut down the computer to recycle power to the board.

Chapter 3 - Installation and Connections

Getting Started

This section includes general hardware installation procedures for the GX1100 board and installation instructions for the GX1100 (GtWave) software. Before proceeding, please refer to the appropriate chapter to become familiar with the board being installed.

To Find Information on…	Refer to…
Hardware Installation	This Chapter
GX1100 Driver Installation	This Chapter
WaveEasy	Chapter 4
Programming	Chapter 5
GX1100 Function Reference	Chapter 6

Packing List

All GX1100 boards have the same basic packing list, which includes:

1. GX1100 Board
2. A CD that includes the GtWave software

Unpacking and Inspection

After removing the board from the shipping carton:


Caution - Static sensitive devices are present. Ground yourself to discharge static.

1. Remove the board from the static bag by handling only the metal portions.
2. Be sure to check the contents of the shipping carton to verify that all of the items found in it match the packing list.
3. Inspect the board for possible damage. If there is any sign of damage, return the board immediately. Please refer to the warranty information at the beginning of the manual.

System Requirements

All GX11X0 instrument boards are designed for use with a 3U or 6U cPCI or PXI compatible chassis. The software is compatible with any computer system running Windows 98, Windows Me, Windows 2000, Windows XP, Windows VISTA (32-bit) and Windows 7 (32 or 64 –bit) operating systems. In addition, Microsoft Windows Explorer version 4.0 or above is required to view the online help.

Each board requires one unoccupied 3U PXI bus slot.

Installation of the GtWave Software

Before installing the board it is recommended that you install the GtWave software as described in this section. To install the GtWave software follow the instruction described below:

1. Insert the Marvin Test Solutions CD-ROM and locate the **GtWave.EXE** setup program. If you computer's Auto Run is configured, when inserting the CD a browser will show several options. Select the Marvin Test Solutions Files option, then locate the setup file. If Auto Run is not configured you can open the Windows explorer and locate the setup files (usually located under \Files\Setup folder). You can also download the file from Marvin Test Solutions' web site (www.MarvinTest.com).
2. Run the GtWave setup and follow the instruction on the Setup screen to install the GtWave driver.

   Note: When installing under Windows NT/2000/XP/VISTA, you may be required to restart the setup after logging-in as a user with Administrator privileges. This is required in-order to upgrade your system with newer Windows components and to install kernel-mode device drivers (HW.SYS and HWDEVICE.SYS) which are required by the GtWave driver to access resources on your board.
3. The first setup screen to appear is the Welcome screen. Click **Next** to continue.
4. Enter the folder where GtWave is to be installed. Either click **Browse** to set up a new folder, or click **Next** to accept the default entry of `C:\Program Files\Marvin Test Solutions\GtWave` or `C:\Program Files (x86)\Marvin Test Solutions\GtWave`.
5. Select the type of Setup you wish and click **Next.** You can choose between **Typical**, **Run-Time** and **Custom** setups types. The **Typical** setup type installs all files. **Run-Time** setup type will install only the files required for controlling the board either from its driver or from its virtual panel. The **Custom** setup type lets you select from the available components.

The program will now start its installation. During the installation, Setup may upgrade some of the Windows shared components and files. The Setup may ask you to reboot after completion if some of the components it replaced were used by another application during the installation – do so before attempting to use the software.

You can now continue with the installation to install the board. After the board installation is complete you can test your installation by starting a panel program that lets you control the board interactively. The panel program can be started by selecting it from the **Start, Programs, Marvin Test Solutions, GtWave** menu located in the Windows Start menu.

Overview of the GtWave Software

Once the software is installed, the following tools and software components are available:

- **GtWave Panel** – Configures and controls the GX11X0's various features via an interactive user interface.
- **GtWave driver** – A DLL based function library (GtWave.DLL, located in the Windows System folder) is used to program and control the board.
- **Programming files and examples** – Interface files and libraries for support of various programming tools. A complete list of files and development tools supported by the driver is included in subsequent sections of this manual.
- **WaveEasy application and run time libraries** – Waveform development environment and script libraries used to create, import, edit, save and export waveform files. When used with the GtWave Panel application it can upload and download waveforms from / to the GX11X0 arbitrary function generators. A 30 day trial version is installed with GtWave. The trial version enables you to create and edit waveforms but will not allow you to save the waveforms to files. If you have purchased WaveEasy, you will need to obtain a license string from the Marvin Test Solutions support site www.MarvinTest.com/magic/. Once you have received the license, you can install it during the trial period by following the instructions in the WaveEasy Help menu, About dialog box. Alternatively, if your trial period has expired, run WaveEasy and enter the license string when prompted.
- **Documentation** – On-Line help and User's Guide for the GX11X0 board, GtWave driver and panel and WaveEasy.
- **PXI/PCI Explorer applet** – Configures the PXI chassis, controllers and devices. This is required for accurate identification of your PXI instruments later on when installed in your system. The applet configuration is saved to PXISYS.ini and PXIeSYS.ini and is used by Marvin Test Solutions instruments, the VISA provider and VISA based instruments drivers. In addition, the applet can be used to assign chassis numbers, Legacy Slot numbers and instrument alias names.

  VISA is a standard maintained by the VXI Plug & Play System Alliance and the PXI Systems Alliance organizations (http://www.vxipnp.org/, http://www.pxisa.org/). VISA provides a standard way for instrument manufacturers and users to write and use instruments drivers. The VISA resource manager such as National Instruments' **Measurement & Automation** (NI-MAX) displays and configures instruments and their address (similar to Marvin Test Solutions' PXI/PCI Explorer).

Configuring Your PXI System using the PXI/PCI Explorer

To configure your PXI/PCI system using the **PXI/PCI Explorer** applet follow these steps:

1. **Start the PXI/PCI Explorer applet**. The applet can be start from the Windows Control Panel or from the Windows Start Menu, **Marvin Test Solutions**, **HW**, **PXI/PCI Explorer**.
2. **Identify Chassis and Controllers**. After the PXI/PCI Explorer started it will scan your system for changes and will display the current configuration. The PXI/PCI Explorer automatically detects systems that have Marvin Test Solutions controllers and chassis. In addition, the applet detects PXI-MXI-3/4 extenders in your system (manufactured by National Instruments). If your chassis is not shown in the explorer main window, use the Identify Chassis/Controller commands to identify your system. Chassis and Controller manufacturers should provide INI and driver files for their chassis and controllers to be used by these commands.
3. **Change chassis numbers, PXI devices Legacy Slot numbering and PXI devices Alias names.** These are optional steps to be performed if you would like your chassis to have different numbers. Legacy slots numbers are used by older Marvin Test Solutions or VISA drivers. Alias names can provide a way to address a PXI device using your logical name (e.g. "DMM1"). For more information regarding these numbers see the **GxAoInitialize** and **GxAoInitializeVisa** functions.
4. **Save you work**. PXI Explorer saves the configuration to the following files located in the Windows folder: PXISYS.ini, PXIeSYS.ini and GxPxiSys.ini. Click on the **Save** button to save you changes. The PXI/Explorer prompt you to save the changes if changes were made or detected (an asterisk sign ' *' in the caption indicated changes).

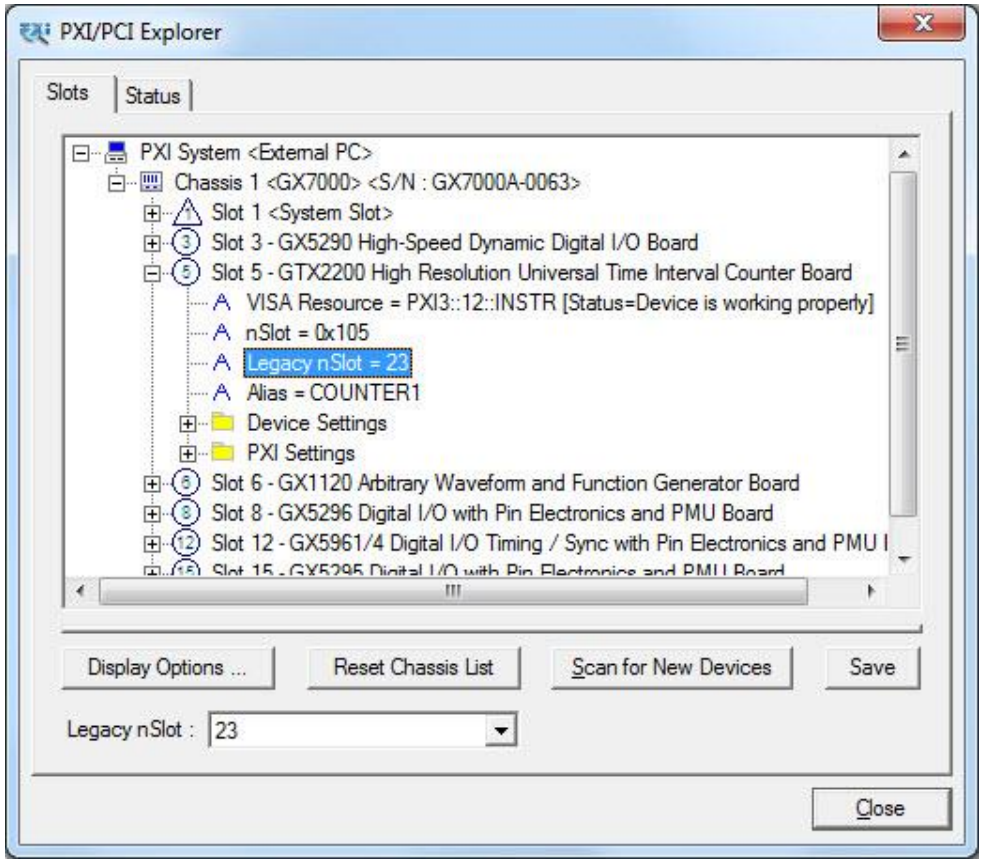


Figure 3-1: PXI/PCI Explorer

Board Installation

Before you Begin

- Install the GtWave driver as described in the prior section.
- Configure your PXI/PC system using **PXI/PCI Explorer** as described in the prior section.
- Verify that all the components listed in the packing list (see previous section in this chapter) are present.

Electric Static Discharge (ESD) Precautions

To reduce the risk of damage to the GX11X0 board, the following precautions should be observed:

- Leave the board in the anti-static bags until installation requires removal. The anti-static bag protects the board from harmful static electricity.
- Save the anti-static bag in case the board is removed from the computer in the future.
- Carefully unpack and install the board. Do not drop or handle the board roughly.
- Handle the board by the edges. Avoid contact with any components on the circuit board.

Caution – Do not insert or remove any board while the computer is on. Turn off the power from the PXI chassis before installation.

Installing a Board

Install the board as follows:

1. Install first the GtWave Driver as described in the next section.
2. Turn off the PXI chassis and unplug the power cord.
3. Locate a PXI empty slot on the PXI chassis (hybrid slot or PXI-1 slot).
4. Place the module edges into the PXI chassis rails (top and bottom).
5. Carefully slide the PXI board to the rear of the chassis, make sure that the ejector handles are pushed **out** (as shown in Figure 3-2).

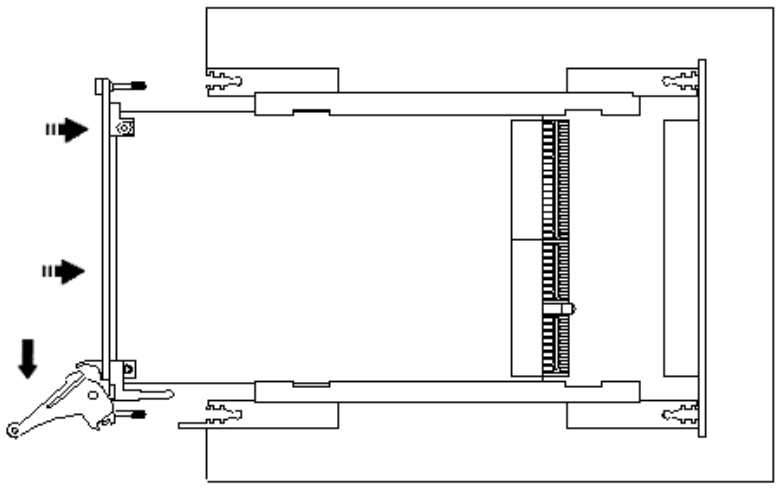


Figure 3-2: Ejector handles position during module insertion

6. After you feel resistance, push in the ejector handles as shown in Figure 3-3 to secure the module into the frame.

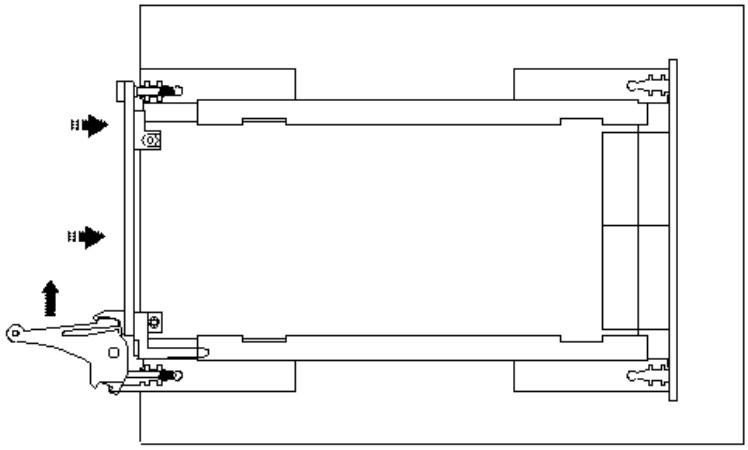


Figure 3-3: Ejector handles position after module insertion

7. Tighten the module's front panel to the chassis to secure the module in.
8. Connect any necessary cables to the board.
9. Plug the power cord in and turn on the PXI chassis.

Plug & Play Driver Installation

Plug & Play operating systems such as Windows 9x, Me, Windows 2000, XP or Windows 7 (Not Windows NT) notify the user that a new board was found using the **New Hardware Found** wizard after restarting the system with the new board.

If another Marvin Test Solutions board software package was already installed, Windows will suggest using the driver information file: HW.INF. The file is located in your Program Files\Marvin Test Solutions\HW folder. Click **Next** to confirm and follow the instructions on the screen to complete the driver installation.

If the operating system was unable to find the driver (since the GtWave driver was not installed prior to the board installation), you may install the GtWave driver as described in the prior section, then click on the **Have Disk** button and browse to select the HW.INF file located in C:\Program File\Marvin Test Solutions\HW.

If you are unable to locate the driver click **Cancel** to the found New Hardware wizard and exit the New Hardware Found Wizard, install the GtWave driver, reboot your computer and repeat this procedure.

The Windows Device Manager (open from the System applet from the Windows Control Panel) must display the proper board name before continuing to use the board software (no Yellow warning icon shown next to device). If the device is displayed with an error you can select it and press delete and then press F5 to rescan the system again and to start the New Hardware Found wizard.

Removing a Board

Remove the board as follows:

1. Turn off the PXI chassis and unplug the power cord.
2. Locate a PXI slot on the PXI chassis.
3. Disconnect and remove any cables/connectors connected to the board.
4. Un-tighten the module's front panel screws to the chassis.
5. Push out the ejector handles and slide the PXI board away from the chassis.
6. Optionally – uninstall the GtWave driver.

GX1110 Connectors

Figure 3-4: GX1110 Front Connector Panel shows the available GX1110 connectors:

Figure 3-4: GX1110 Front Connectors Panel

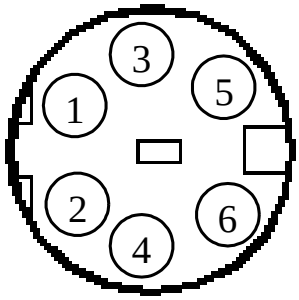


Figure 3-5: GX1110 Front PS2 Connector Pin Number Layout

The following are the connector signals for the GX1110 board:

Pin#	Signal
1	External Amplitude Modulation Input (AM) and Frequency Modulation (FM) Input.
2	GND
3	External Clock Reference Input
4	Marker Out
5	Clock Reference Out.
6	GND

Table 3-1: GX1110 Output Connectors

GX1120 Connectors

Figure 3-6: GX1120 Front Connector Panel shows the available GX1120 connectors:

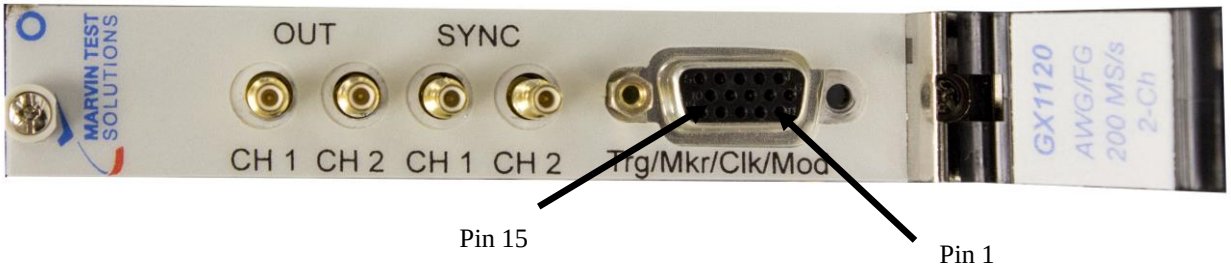


Figure 3-6: GX1120 Front Connectors Panel

The following are the DB-15 connector signals for the GX1120 board:

Pin #	Signal
1	Channel 2 External Trigger input
2	Ground
3	Channel 2 Marker Out
4	Ground
5	No connected
6	Channel 1 External Trigger input
7	No connected
8	Channel 1 Marker Out
9	Ground
10	Channel 2 Modulation input
11	Ground
12	Reference Clock out
13	Reference Clock input
14	Ground
15	Channel 1 Modulation input

Table 3-2: GX1120 Output Connectors

Installation Directories

The GtWave driver files are installed in the default directory `C:\Program Files\Marvin Test Solutions\GtWave or C:\Program Files (x86)\Marvin Test Solutions\GtWave`. You can change the default `GtWave` directory to one of your choosing at the time of installation.

During the installation, `GtWave` Setup creates and copies files to the following directories:

Name	Purpose / Contents
…\Marvin Test Solutions\GtWave	The GX11X0 directory. Contains panel programs, programming libraries, interface files and examples, on-line help files and other documentation.
…\Marvin Test Solutions\HW	HW device driver. Provide access to your board hardware resources such as memory, IO ports and PCI board configuration. See the README.TXT located in this directory for more information.
…\ATEasy\Drivers	ATEasy drivers directory. GtWave Driver and example are copied to this directory only if ATEasy is installed to your machine.
…\Windows\System (Windows 9x/Me), or …\Windows\System32 when running Windows NT/2000/XP	Windows System directory. Contains the GtWave DLL driver and some upgraded system components, such as the HTML help viewer, etc.

GtWave Driver Files Description

The Setup program copies the GtWave driver, a panel executable; the GtWave help file, the README.TXT file, and driver samples. The following is a brief description of each installation file:

WaveEasy Files

- WaveEasy.exe – Windows application used to create and edit waveform files.
- WaveEasyRT.dll – Scripting library (using Microsoft Component Object Model) used to create and modify waveform files via any programming language that supports Microsoft ActiveX/COM standard.
- Examples.WaveEasy – WaveEasy file which contains many examples for various formulas.
- BandFilter.WaveEasy – WaveEasy file which contains several segments with FIR and FFT Band Pass Filter formulas.

Driver and Virtual Panel

- GtWave.dll – Windows DLL for 32 bit applications running under Windows (32/64 bit).
- GtWave64.dll – Windows DLL for 64 bit applications running under Windows (64 bit).
- GtWavePanel.exe – A 32 bit instrument front panel application for all GtWave supported boards running under Windows (32/64 bit).
- GtWavePanel64.exe – A 64 bit instrument front panel application for all GtWave supported boards running under Windows (64 bit).

Programming Interface Files

The following GtWave interface files are used to support the various development tools:

- GtWave.h – header file for accessing the DLL functions using the C/C++ programming language. The header file is compatible with the following 32-bit development tools:

  - Microsoft Visual C++, Microsoft Visual C++ .NET
  - Borland C++

- GtWave.lib – Import library for GtWave.DLL (used when linking 32 bit C/C++ application that uses GtWave.dll).
- GtWave64.lib – Import library for GtWave.DLL (used when linking 64 bit C/C++ application that uses GtWave64.dll).
- GtWaveBC.lib – Import library for GtWave.DLL (used when linking Borland C/C++ application that uses GtWave.dll).
- GtWave.pas – interface file to support Borland Pascal Borland Delphi.
- GtWave.bas – Supports Microsoft Visual Basic 4.0, 5.0 and 6.0.
- GtWave.cs – Supports Microsoft Visual C#.
- GtWave.vb – Supports Microsoft Visual Basic .NET.
- GX1110.drv – ATEasy driver File for GX1110
- GX1120.drv – ATEasy driver File for GX1120
- GtWave.llb – LabView library.

Documentation

- GtWave.chm – On-line version of the GX11X0 User's Guide. The help file is provided in a Windows Compiled HTML help file (.chm). The file contains information about the GX11X0 board, programming reference and panel operation.
- GtWaveUG.pdf – On line, printable version of the GX11X0 and WaveEasy User's Guide in Adobe Acrobat format. To view or print the file you must have the reader installed. If not, you can download the Adobe Acrobat reader (free) from http://www.adobe.com.
- ReadMe.txt – Contains important last minute information not available when the manual was printed. This text file covers topics such as a list of files required for installation, additional technical notes, and corrections to the GtWave manuals. You can view and/or print this file using the Windows NOTEPAD.exe or other text file editors.

GtWave Example Programs

The sample programs include a C/C++ sample compiled with various development tools, Visual Basic example and an ATEasy example. Other examples may be available for other programming tools. The examples are installed to the GtWave Examples subfolder.

Microsoft Visual C++ .NET example files:

- GtWaveExampleC.cpp – Source file
- GtWaveExampleC.ico – Icon file
- GtWaveExampleC.rc – Resource file
- GtWaveExampleC.vcproj – VC++ .NET project file
- GtWaveExampleC.exe – Example executable

Microsoft Visual C++ 6.0 example files:

- GtWaveExampleC.cpp – Source file
- GtWaveExampleC.ico – Icon file
- GtWaveExampleC.rc – Resource file

- GtWaveExampleC.dsp – VC++ project file
- GtWaveExampleC.exe – Example executable

Borland C++ example files:

- GtWaveExampleC.cpp – Source file
- GtWaveExample.ico – Icon file
- GtWaveExampleC.rc – Resource file
- GtWaveExampleC.bpr – Borland project file
- GtWaveExampleC.exe – Example executable

Microsoft Visual C# .NET example files:

- GtWaveExampleCS.cs – Source file
- GtWaveExampleCS.csproj – Project file
- GtWaveExampleCS.exe – Example executable

Microsoft Visual Basic .NET example files:

- GtWaveExampleVB.vb – Example form.
- GtWaveExampleVB.Designer.vb – Example form designer file.
- GtWaveExampleVB.resx – Example form resource.
- GtWaveExampleVB.vbproj – Project file
- My Project subfolder – Example project settings files
- GtWaveExampleVB.exe – Example executable

Microsoft Visual Basic 6.0 example files:

- GtWaveExampleVB6.frm – Example form
- GtWaveExampleVB6.frx – Example form binary file
- GtWaveExampleVB6.vbp – Project file
- GtWaveExampleVB6.exe – Example executable.

ATEasy driver and examples files (ATEasy Drivers directory):

- GtWave.wsp – workspace
- GX1110.prj – example project
- GX1120.sys – example system
- GX1110.prg – example program
- GX1120.prj – example project
- GX1120.sys – example system
- GX1120.prg – example program

LabView example:

- GtWaveExample.vi

WaveEasy Run-Time / GtWave Example Programs

Microsoft Visual Basic .NET example files:

- WaveEasyRTExampleVB.vb – Example form.
- WaveEasyRTExampleVB.Designer.vb – Example form designer file.
- WaveEasyRTExampleVB.resx – Example form resource.
- WaveEasyRTeExampleVB.vbproj – Project file
- My Project subfolder – Example project settings files
- WaveEasyRTExampleVB.exe – Example executable

Setup Maintenance Program

You can run the Setup again after GtWave has been installed from the original disk or from the Windows Control Panel – Add Remove Programs applet. Setup will be in the Maintenance mode when running for the second time. The Maintenance window show below allows you to modify the current GtWave installation. The following options are available in Maintenance mode:

- **Modify.** When you want to add or remove GtWave components.
- **Repair.** When you have corrupted files and need to reinstall.
- **Remove.** When you want to completely remove GtWave.

Select one of the options and click **Next** and follow the instruction on the screen until Setup is complete.

Chapter 4 - WaveEasy

This chapter contains information about how to use **WaveEasy** – a waveform development tool for creating, importing, analyzing, editing, saving and exporting waveform files via the WaveEasy application environment or via your programming language using the WaveEasy run-time ActiveX libraries. When used with the GtWave Panel application you can upload and download waveforms to the GX11X0 arbitrary function generator board.

Features

The WaveEasy application offers the following features:

- Create, edit, print and save waveform files (.WaveEasy file format).
- Mathematical formula based storage and editing for waveform segments and waveform segment items supports incremental changes / edits and compact storage of waveform files.
- Formula features include:
  - Operators: +, -, *, /, ^, (,).
  - Mathematical functions: Abs, Ceil, Exp, Floor, Ln, Log, Log10, Rnd and Sqrt.
  - Trigonometric functions: ArcCos, ArcCosh, ArcSinh, ArcTan, ArcTanh, Cos, Cosh, Cot, Sin, Sinc, Sinh, Tan and Tanh.
  - Filtering functions: FIR and FFT Band Pass.
  - Pseudo functions: Container, Clip, Noise, SawTooth, Sketch, Square, and Triangle.
- Editing functions include: Insert, Delete, Cut, Copy and Paste for range, segment items and segments selections, unlimited Undo/Redo.
- Drawing functions: Freehand Draw and Line Draw.
- Zoom operations: Zoom In, Zoom Out, Area Zoom.
- Development environment display includes: Segments window, Waveform window, Preview window, Properties window for properties of Waveform, Segment and Segment Item, Status bar for display of waveform cursor, waveform range selection and mouse cursor positions.
- Waveform window display supports multiple horizontal and vertical axes including user defined axes ranges, units with grid, ticks, labels, name and a cursor. Each window displays the waveform segment wave and the waveform frequency spectrum that is used to analyze the waveform frequencies.
- Development environment supports customized menu, toolbar, keyboard shortcuts and tools menu.
- Import and export NI-HWS Files (National Instruments Files .hws), CSV Files (Comma Delimited Files .csv), Text Files (.txt) and PRN Files (Space Delimited Files .prn).
- Run time library used to create, import, analyze and export waveform files. DSP functions can be use to further analyze your waveform to perform band filtering, FFT (Fast Fourier Transform) and more.
- When used with the GtWave Panel application, waveforms can be upload and downloaded directly from the GX11X0 family of arbitrary function generators to the WaveEasy waveform development environment.

Starting WaveEasy

To start WaveEasy, click on the **WaveEasy** icon on your desktop or select **Marvin Test Solutions**, **GtWave**, **and WaveEasy** from the Windows Start menu.

Trial Period and License Installation

WaveEasy is provided with the GtWave software package. A limited feature, license free, 30 day trial version is included with GtWave software. The trial version enables you to create, view and edit waveforms but will not allow you to save the waveforms to files. If you have purchased WaveEasy separately, you will need to obtain a license string from the Marvin Test Solutions support site www.MarvinTest.com/magic/. You will need to supply the Computer ID that is displayed at the bottom of the WaveEasy License Setup dialog. Once you have received the license string, you can install it during the trial period from the WaveEasy Help menu, About dialog box. If the trial period has expired, run WaveEasy and enter the string when prompted.

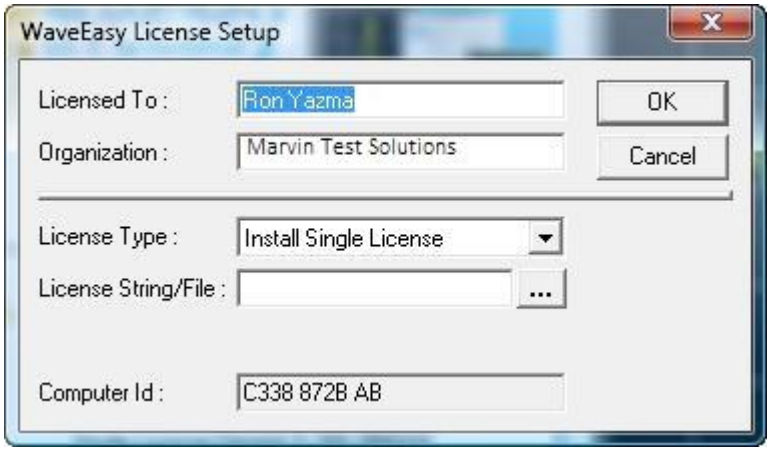


Figure 4-1: WaveEasy License Setup Dialog

WaveEasy Main Window

Once WaveEasy is started the main window is displayed as shown below:

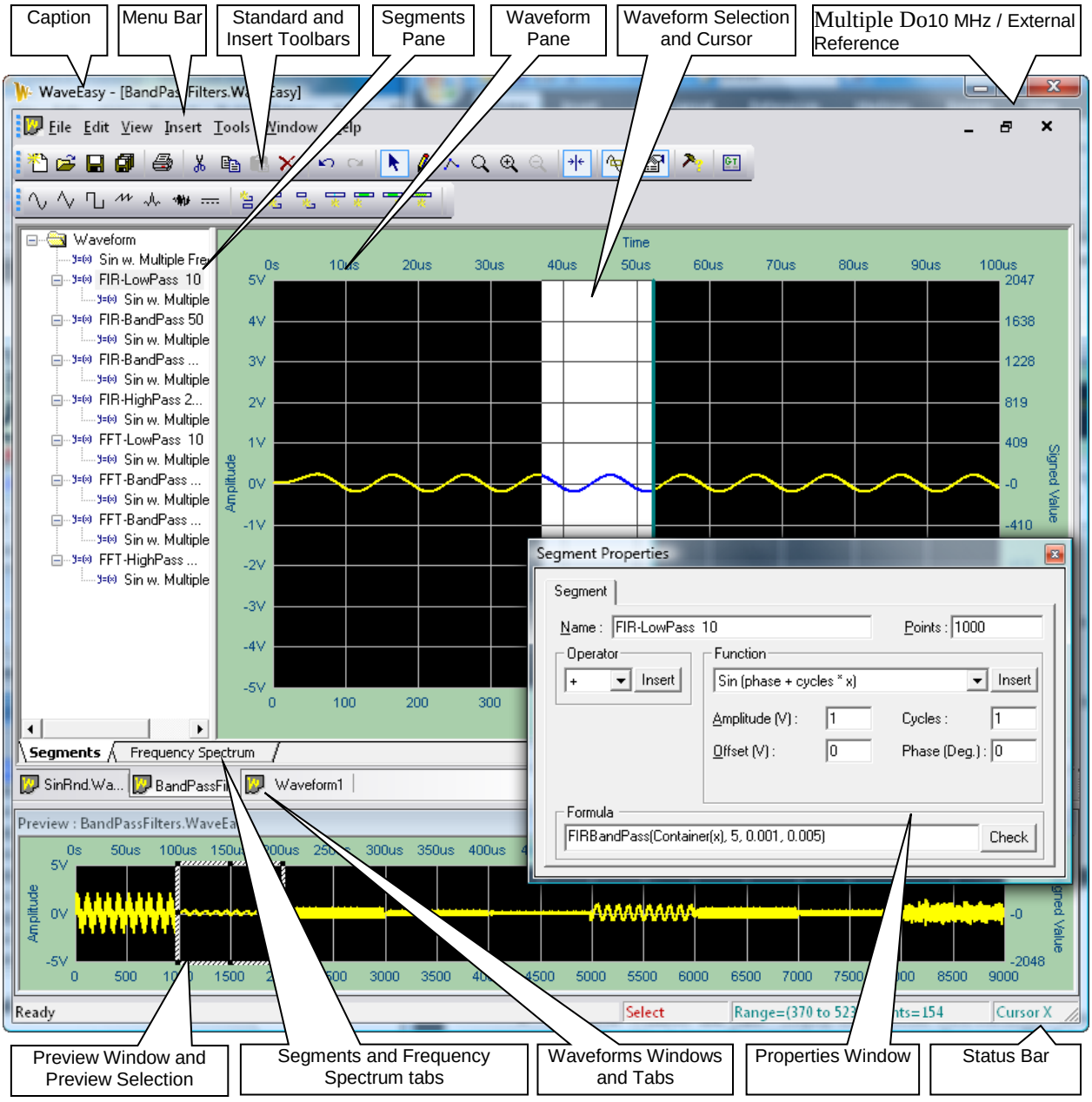


Figure 4-2: WaveEasy Main Window

The following items are displayed:

- **Caption** – Displays the active waveform file name being edited. If the waveform is not saved to a file yet the waveform name is displayed with no file extension.
- **Menu Bar** – The application menu bar.

- **Standard and Insert Toolbars** – Displays two toolbars which are used for quick access to invoke commands that are also available from the menu.
- **Waveform Windows and Tabs** – Display the waveform in time domain or frequency domain. The Segments view is divided with a moveable splitter between the **Segments Pane** and the **Waveform Pane** which can display the waveform in time domain – **Segments tab**, or an analysis of the time domain - **Frequency Spectrum tab**.
- **Segments Pane** – Part of the Waveform window. Displays the waveform **segments** and **segment items** in a tree. Each waveform is divided into segments which can be further divided into segment items. Segment and segment items can be edited using drawing commands or from the Properties Window by a mathematical formula associated with the segment or the segment item.
- **Waveform Pane** – Part of the Waveform window. Displays the waveform. The pane can display the whole waveform or selected segment or segment item as selected in the segments pane or in the waveform preview window. The waveform is displayed in yellow. Additional waveforms axes and axes values, grid (gray) and the waveform selection (white) and cursor (blue) can also be displayed. You can customize many of the elements displayed in the pane including the color of these elements, horizontal and vertical axes range and name, grid and more from the Tools, Options menu.
- **Waveform Selection and Cursor** – Selected area inside the waveform pane (white) and the cursor (blue). The status bar displays the selected area's start and end position, the cursor position and voltage.
- **Multiple Document Interface (MDI) controls** – The main window can display multiple documents, each in its own window. Each of the document views can be minimized, maximized or displayed in normal mode (in the above example the window is maximized).
- **Preview Window and Preview Selection** – The preview window displays the whole waveform. Users can draw a selection rectangle which is used to select a portion of the waveform – this will cause the active Waveform Pane to show the selected waveform portion.
- **Segments and Frequency Spectrum tabs** – Each Waveform window displays the waveform or its segments in time or frequency domain.
- **Waveform Window and Tabs** – Displays a list of open waveform documents windows. Clicking on the specific tab will show the associated document window.
- **Properties Window** – Displays the properties of the Segments Pane selected item which can be a waveform, waveform segment or waveform segment item. These properties are different for each selection. However, this window is mostly used to edit and enter mathematical formulas for a specific segment or segment item.
- **Status Bar** - Displays the menu/toolbar description, working mode (Select, Freehand Draw, LineDraw, and Zoom), selection area points (start, end), cursor position and voltage level, and mouse position (x, y).

Working with WaveEasy

Follow this tutorial to learn how to create and edit basic waveform files and understand the various WaveEasy concepts, operations and commands.

Creating a New Waveform

Start a new waveform file and save your work:

Click **File** menu, **New**. A new Waveform Window is displayed with Sin waveform. By default, WaveEasy creates a waveform with one segment. The waveform has 1000 points with 12 bit resolution and the waveform amplitude is set to +/- 5V. 10 M/Sec is the frequency displayed on the horizontal axis of the waveform window.

The Waveform Properties

To view or change the Waveform properties:

Right click on the **Waveform** from the Segments pane and select Properties. The Waveform properties window will open as shown here:

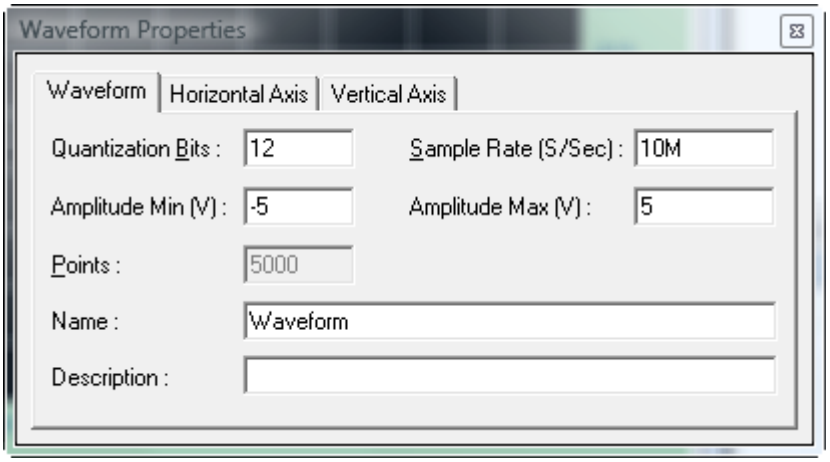


Figure 4-3: Waveform Properties

At this time you can change the waveform's number of bits per point, sample rate, amplitude, number of points in the file (read-only, to change this - see the next paragraph), waveform name and description. The other two tabs Horizontal and Vertical Axis are used to change the waveform's X and Y axes. Each tab can set multiple axes on the right, left, top and bottom of the Waveform Window. In addition, you can set the axis to different units as required for your specific application.

Segment and Segment Item Properties and the Formula Editor

To view or change the Waveform Segment or Segment Item properties:

Select the first segment **Segment1** the properties will reflect the properties of that first segment as shown here:

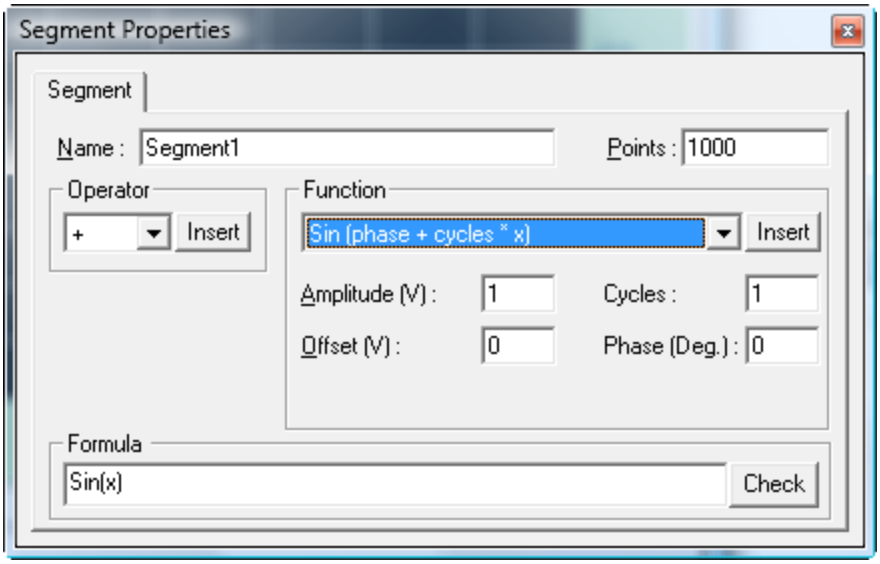


Figure 4-4: Waveform Segment Properties

Using the **Segment properties** window you can adjust the segment name, number of segment points (shown here as 1000), and the mathematical formula used to create the segment wave. The formula control allows you to enter the formula directly by typing text or by using the formula editor controls that allows you to insert operators such as: +, -, *, /, ^ (power of). Opening and closing parentheses are used to change the order of evaluation of the formula expression.

The following formula **Sin (2 * x) * 3 + 1.5** – will create a wave with two cycles of a Sine, at 3V amplitude with a 1.5 positive offset – and will result in the following waveform:

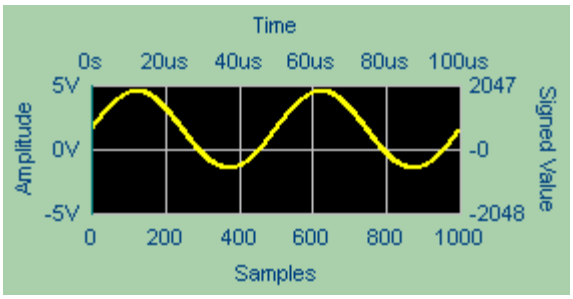


Figure 4-5: Sin (2 * x) * 3 + 1.5 Waveform

WaveEasy formulas are very powerful. Here are some additional examples:

- **Clip ((Sin (2 * x) * 3 + 1.5) , -5, 3)** will result in a clipped Sine with a peak amplitude of 3V:

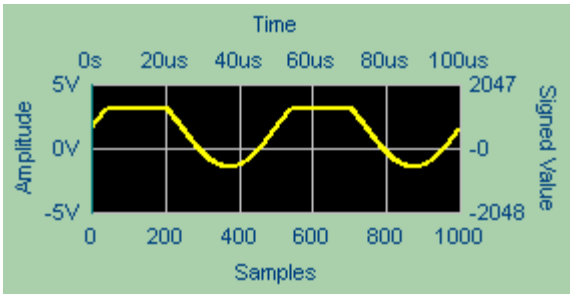


Figure 4-6: Clip ((Sin (2 * x) * 3 + 1.5) , -5, 3) Waveform

- **Clip((Sin (2 * x) * 3 + 1.5) , -5, 3) + 0.4 * Noise(x)** will add 0.4V of noise to the above waveform:

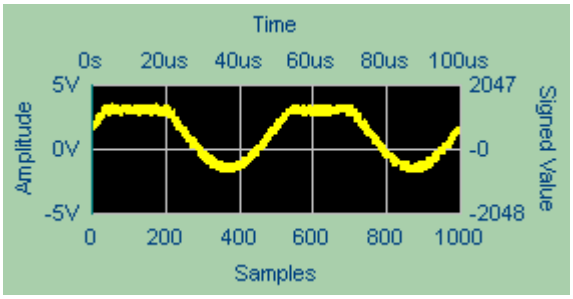


Figure 4-7: Clip ((Sin (2 * x) * 3 + 1.5) , -5, 3) Waveform

You can experiment with the formula editor using the different operators and functions.

Waveform Editing Modes and Tools

The WaveEasy editor can be set to one of the following edit modes:

- **Selection Mode** – This is the default editing mode and can be activated using the Selection tool from the toolbar, by pressing the ESC key or by Selecting Select from the Edit menu. This mode is used to move the cursor or select an area in the waveform. The status bar always displays the current cursor location and the selected area position (if an area is selected). Many of WaveEasy's Edit, Insert or View menu commands use the cursor position or the selected area. These command allow you to Copy an area, Cut or Delete a selected area, Paste at the cursor position another area, Split a segment or a segment item at the cursor position or create a new segment item from selected area, Insert a segment item at the cursor position, or Zoom-In to a selected area.
- **Hand Draw Mode** – In this mode, you can free hand draw to create your own waveform. To draw, select the Hand-Draw tool from the toolbar or from the Edit menu, then, move the cursor to the waveform area, the cursor will display a pen; press the left mouse button down to start drawing and release to stop drawing. Once you are done drawing you can select the Select button from the toolbar or press the ESC key to return to Selection mode. Drawing on the segment or segment item will split the segment to segment items (unless the area that you draw on already had a free hand or a line drawing).
- **Line Draw Mode** – In this mode, you can draw straight lines to create your own waveform. To line draw, select the Line-Draw tool from the toolbar or from the Edit menu, then, move the cursor to the waveform area, the cursor will display a cross, click the left mouse button to start drawing and click again to the next line and so on. To stop drawing, press the ESC key or click on the selection tool. Drawing on the segment or segment item will split the segment into segment items (unless the area you have drawn on already has a free hand or a line drawing).
- **Zoom Mode** – In this mode, you can zoom in or out to be able to see more (or less) of the waveform. To zoom to an area, click on the Zoom tool from the toolbar or from the View menu, than draw a rectangle on the waveform area (click the left mouse button down and then move the mouse cursor and release the button to form a rectangle). The waveform will now display the rectangle area. To zoom out, click on the Zoom Out button from the toolbar.

Working with Segments and Segment Items

The Waveform that you create using WaveEasy is divided into segments. Each segment can be further divided into sub-segments that are also called segment items. The Segments Pane displays the waveform objects in a tree structure. These objects are displayed as part of the Waveform root and include segments below the waveform root and segment items below the segments. Selecting an object will cause its wave to be displayed in the Waveform Pane and in the Properties window (if open). Once the object is selected you can rename it using the F2 key or by clicking on the name of the object in the tree. The Segment Pane, can also be used to move or copy segments or segment items across the waveform. You can drag and drop using the mouse for a segment or a segment item. Pressing the CTRL key down when you drop will copy the segment or segment item instead of moving it. You can also use the Edit menu commands to **Cut**, **Copy**, **Paste** or **Delete** a segment or segment item. Other available commands are from the **Insert** menu where you can insert a segment or segment item relative to the selected object in the Segment Pane.

Saving your Work and WaveEasy File Formats

Once you are done editing the waveform you can use the File menu Save command to Save you work to one of the supported WaveEasy file formats. The following file formats are supported by WaveEasy:

- **WaveEasy file format** (*.WaveEasy) – this is the default file format. In this format all the information you entered when you created the waveform is preserved including segment and segment item names and properties, and the data points or mathematical formulas used to create the waveform. WaveEasy file format is supported by the GtWave driver and can be used to load the waveform to the board. When you save a file in this format, you can later reopen the file and modify the segments and segment items as when you initially created them. This file format also requires less disk space since segments that have mathematical formulas are saved as a formula instead of an expanded data point array. When you load the file using the GtWave driver to the board it will expand the mathematical formulas and the line or free hand drawing formulas to create the required waveform points.
- **NI-HWS File format** (*.hws) – This is the file format used by National Instruments waveform editor and boards. Use this file format if you intend to load the waveform to a NI board. This file format is binary and contains the waveform points. Segments and formulas are not maintained in this file format.
- **Comma Delimited File format** (*.csv) – ASCII (text) file format contains all of the point values separated by commas.
- **Text File format** (*.txt) – ASCII (text) file format that contains all the points value separated by commas.
- **PRN (Space Delimited) File format** (*.prn) – ASCII (text) File format that contains all the points value separated by spaces.

WaveEasy can open and save all the supported file formats. However, to maintain the formulas and segments as entered and to allow re-editing of these files, it is recommended that you save your work in the WaveEasy file format and use the Save As… command to retain your original file when you intend to use other file formats.

Building Formula Expressions

WaveEasy formulas can be entered from the Segment properties window. The properties window has a formula editor which displays function parameters in a friendly manner. The user can then populate the function arguments using the formula's controls. Clicking on Insert Function will insert the formula at the current cursor position into the formula text box. If text is selected in the formula text box prior to clicking Insert button, the selected text will be used as the argument for the function. User can also enter one of the supported operators to complete the formula.

The following operators are supported:

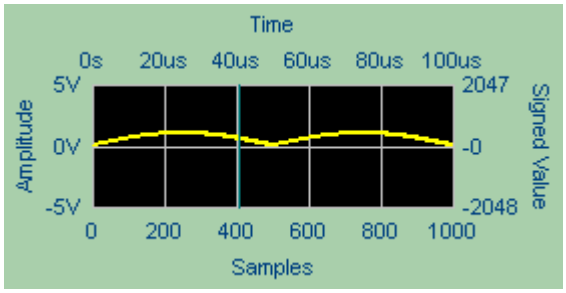
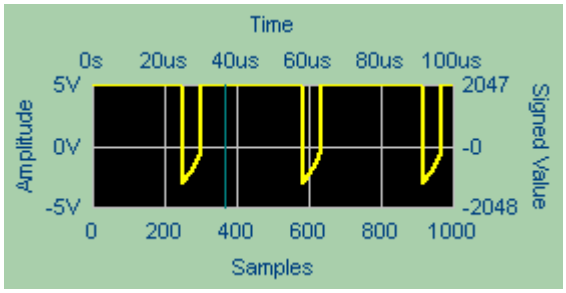
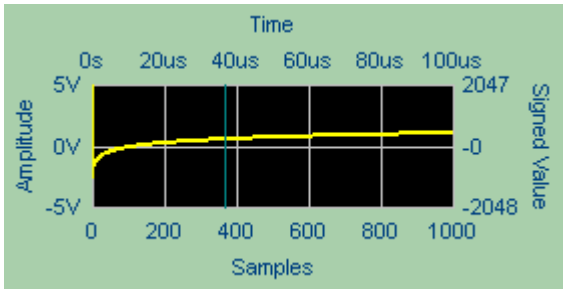
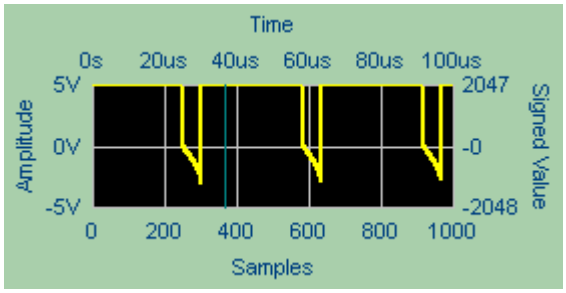
- Addition (+), usually used for amplitude offset.
- Subtraction (-), usually used for amplitude offset.
- Multiplication (*), usually used to multiply the amplitude for trigonometric function or to specify number of cycles.
- Division (/), usually used to multiply the amplitude for trigonometric function or to specify number of cycles.
- Power (^), used to increase amplitude
- Open and Close Parentheses (( ))

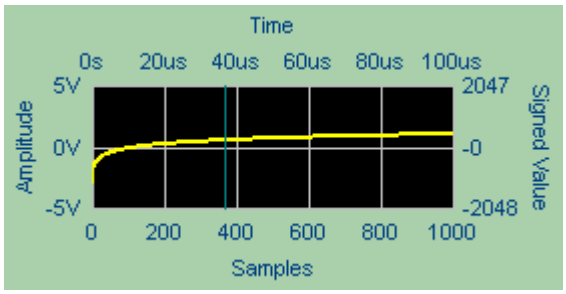
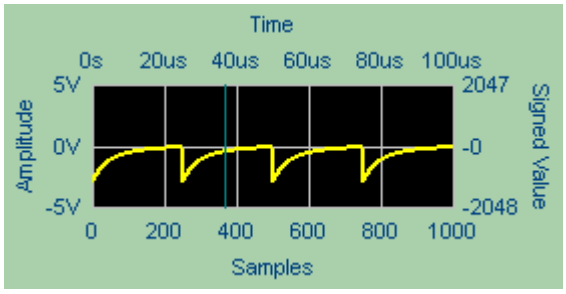
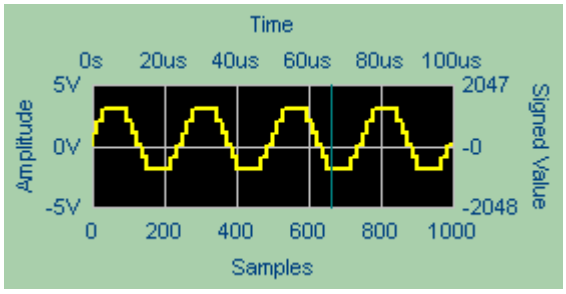
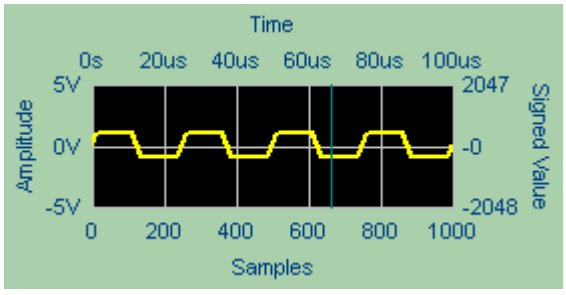
The *x* is used to specify the time value when the formula is calculated.

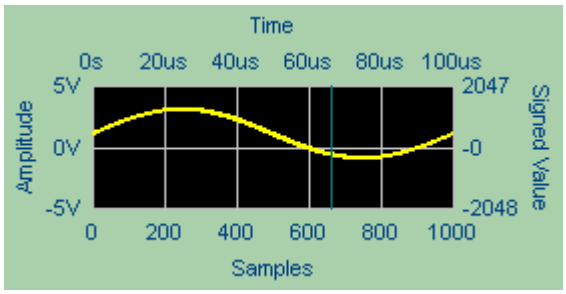
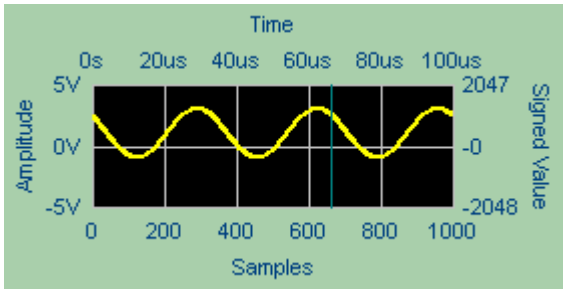
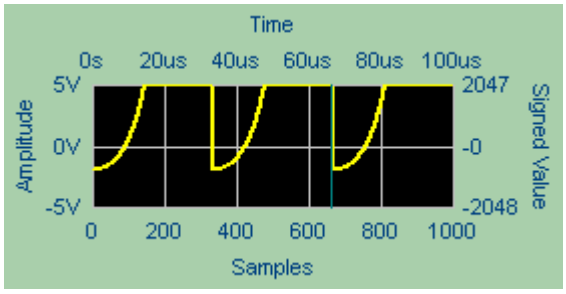
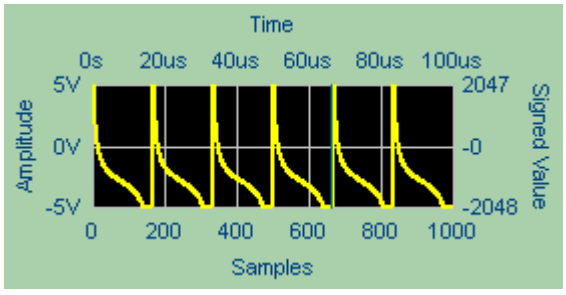
The section below shows the available WaveEasy formula functions and their parameters as selected or entered in the properties window.

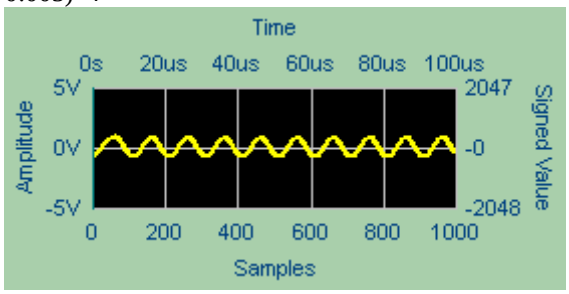
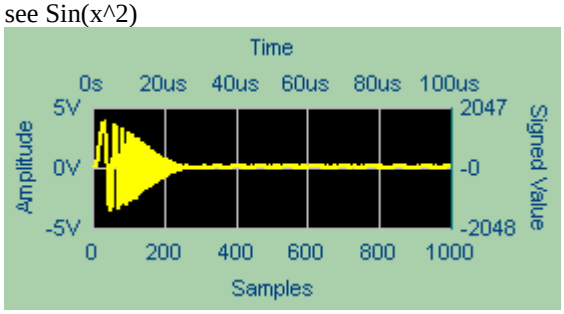
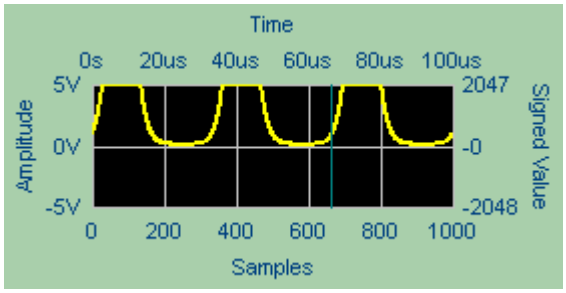
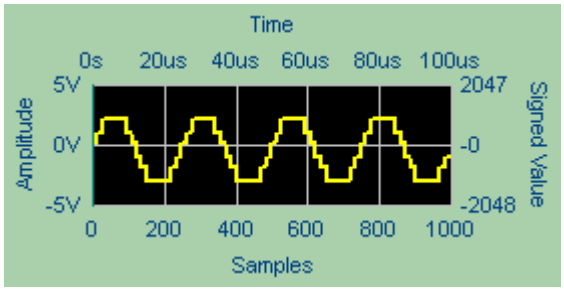
WaveEasy Formula Functions

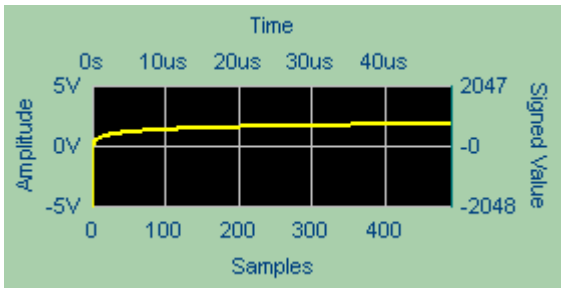
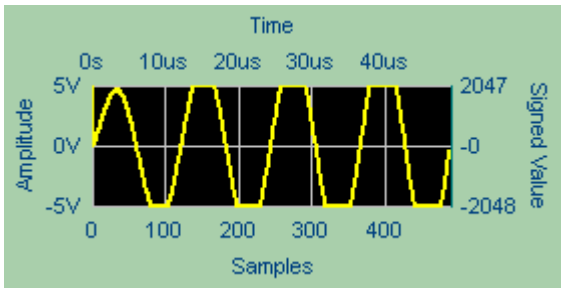
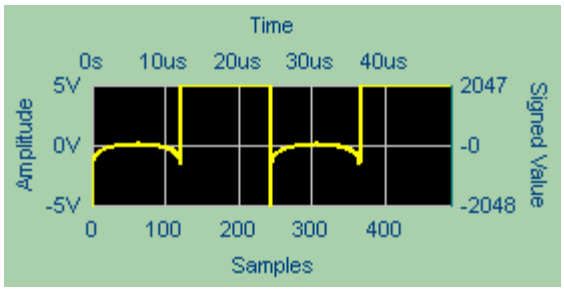
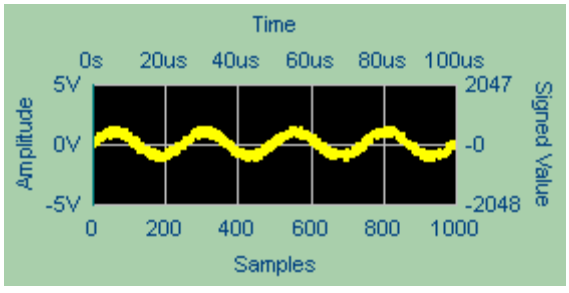
The following mathematical functions are available for use when constructing the segment formula. Many of these examples are included in the WaveEasy folder (.WaveEasy files).

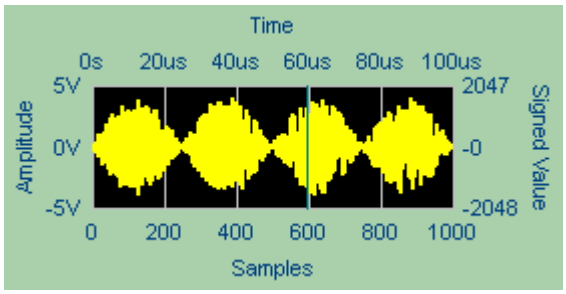
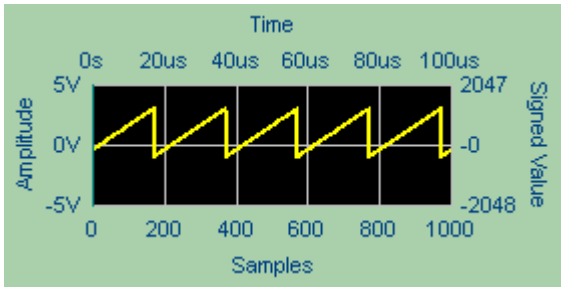
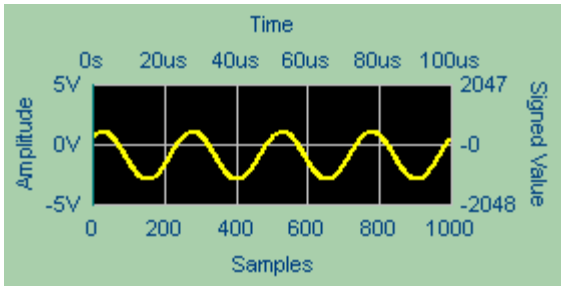
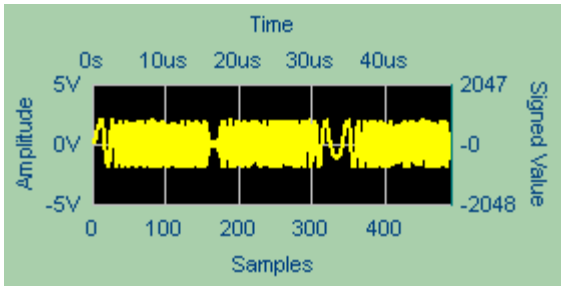
Function	Description	Example
Abs(*x*)	Absolute value of *x*.	Abs(Sin(x)) 
ArcCos(*phase* + *cycles* * x)	Inverse cosine of *x*.	-2*ArcCos(x*3+90) 
ArcCosh(*phase* + *cycles* * *x*)	Inverse hyperbolic cosine of *x*.	-3 + 0.5 * ArcCosh(4 * x) 
ArcSin(*phase* + *cycles* * *x*)	Inverse sine of *x*.	-2*ArcSin(x*3+90) 

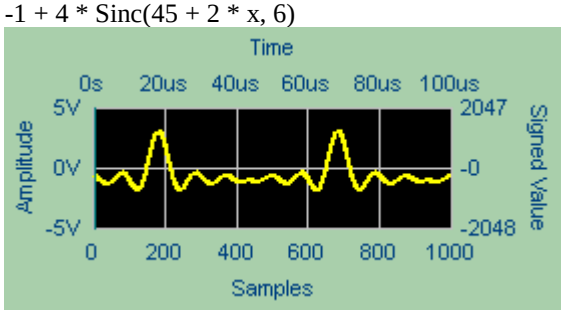
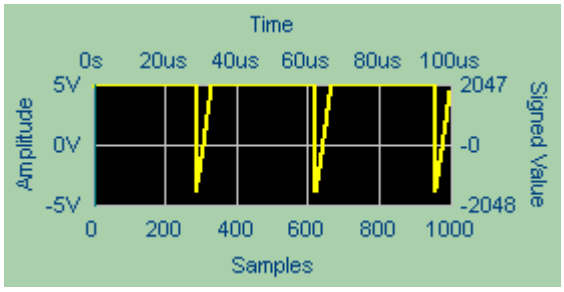
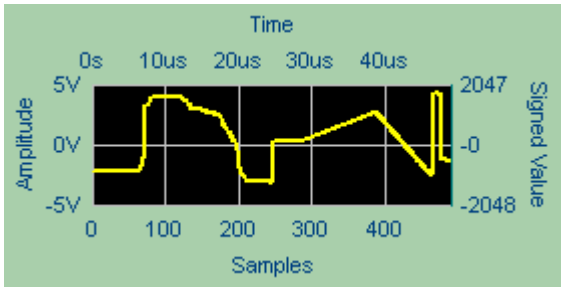
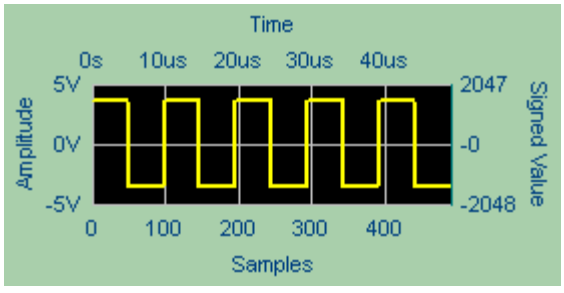
Function	Description	Example
ArcSinh(*phase* + *cycles* * *x*)	Inverse hyperbolic sine of *x*.	-3 + 0.5 * ArcSinh(4 * x) 
ArcTan(*phase* + *cycles* * *x*)	Inverse tangent of *x*.	-3 + 2* ArcTan(4 * x) 
Ceil(*x*)	Rounds the elements of *x* to the nearest integers greater than or equal to *x*.	Ceil(3*sin(4*x)) 
Clip(x, amplitude low, amplitude high)	Clip the elements of *x* to the high/low amplitude.	Clip(3*sin(4*x), -1, 1) 

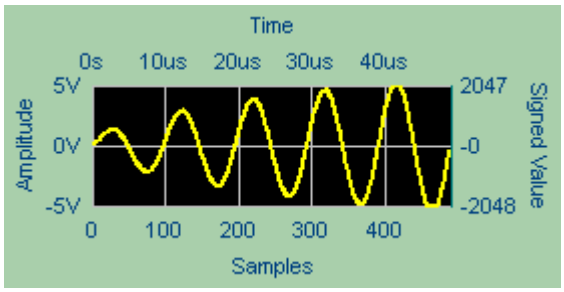
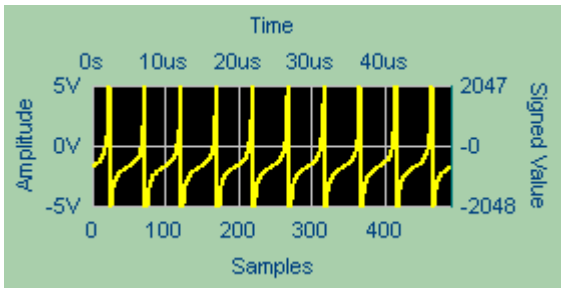
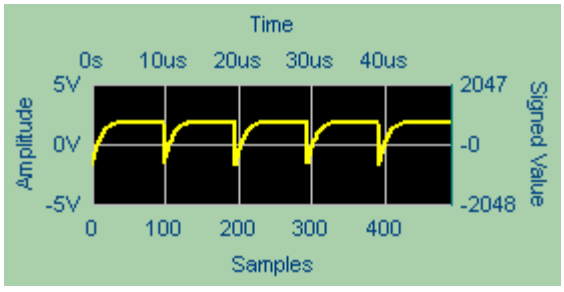
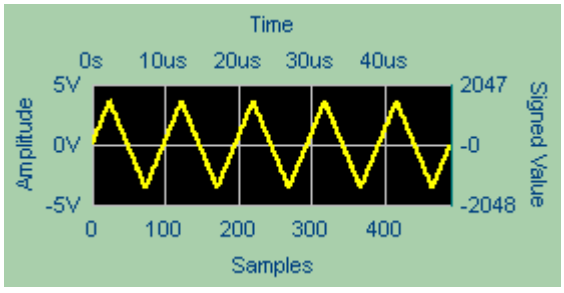
Function	Description	Example
Container(x)	Used in a formula for a segment that has sub-segments. Represents a collection of segment sub segments.	1+2*Container(x) and sub segment is Sin(x) 
Cos(phase + cycles * x)	Cosine of *x*.	1+2*Cos(x*3+45) 
Cosh(*phase + cycles * x*)	Hyperbolic cosine of *x*.	Cosh(x*3)-3 
Cot(*phase + cycles* * x)	Cotangent of *x*.	Cot(x*3)-3 

Function	Description	Example
FFTBandPass(x, filter order, cutoff low freq, cutoff high freq)	Filter that passes frequencies within a certain range and rejects (attenuates) frequencies outside that range. BandPass uses a frequency domain convolution. Higher filter orders produce better results however computation is slower.	FFTBandPass(Sin(10 * x) + 0.8 * Sin(50 * x) + 0.5 * Sin(100 * x) + 0.2 * Sin(200 * x), 5, 0.001, 0.005)*4 
FIRBandPass(x, filter order, cutoff flow freq, cutoff high freq)	A filter that passes low frequencies well, but attenuates (reduces the amplitude of) frequencies higher than the cutoff frequency. LowPass uses a time domain convolution. Higher filter orders produce better results however computation is slower.	FIRBandPass(Sin(x^2), 1, 0.001, 0.1), see Sin(x) to see Sin(x^2) 
Exp(*x*)	Exponent of *x*.	Exp(3*Sin(3*x)) 
Floor(*x*)	Rounds the elements of *x* to the nearest integers smaller than or equal to *x*.	Floor(3*Sin(4*x)) 

Function	Description	Example
Ln(*x*)	Natural logarithm of *x*.	Ln(x)*0.3 
Log(*x*, *base*)	The logarithm of *x* to the *base*.	Log(x, 2)*Sin(4*x) 
Log10(*x*)	Common logarithm (base 10) of *x*.	Log10(sin(2* x)) 
Noise(*x*)	Noise, generate a number in the range -1 to 1.	Sin(4*x)+0.5*Noise(x) 

Function	Description	Example
Rnd(*x*)	Randomize, random number between -1 and 1.	Sin(2*x)*Rnd(x)*4 
SawTooth(*phase* + *offset* * *x*)	Wave that resembles the teeth of a saw.	1 + 2 * SawTooth(45 + 5 * x) 
Sin(*phase* + *cycles* * *x*)	Sine of *x*.	-1+2*Sin(4*x+45)  Sin(x^2)*2 

Function	Description	Example
Sinc(phase + cycles * x, zero crossing)	Sine cardinal of *x*.	-1 + 4 * Sinc(45 + 2 * x, 6) 
Sinh(*phase* + *cycles* * *x*)	Hyperbolic sine of *x*.	-4 + 10 * Sinh(45 + 3 * x) 
Sketch(x)	Free Hand or line drawing using WaveEasy tools. Use a formula to manipulate and change the amplitude, add offset, etc.	Sketch(x)*2-1 
Square(phase + cycles * x , duty cycle)	Square of *x*.	3.6 * Square(5 * x, 50) 

Function	Description	Example
Sqrt(*x*)	Square root of *x*.	Sqrt(x) * 0.3 * sin(5 * x) 
Tan(*phase* + *cycles* * *x*)	Tangent of *x*.	-1.8 + 0.6 * Tan(5 * x) 
Tanh(*phase* + *cycles* * *x*)	Hyperbolic tangent of of *x*.	-1.8 + 3.6 * Tanh(5 * x) 
Triangle(*phase* + *cycles* *)	Triangle of *x*.	3.6 * Triangle(5 * x) 

Using the Frequency Spectrum Tab

WaveEasy can display the waveform or a segment in frequency domain. This is used to analyze the waveform frequency spectrum. The following is a screen capture of a waveform containing multiple frequencies using the following formula:: 4 * Sin(10 * x) + 3 * Sin(50 * x) + 3 * Sin(100 * x) +1 * Sin(200 * x) with 10 Ms/Sec sampling rate. The resultant frequency spectrum shows 4 frequencies (100 KHz, 500KHz, 1MHz, 2MHz). The upper axis shows a scale in fraction of the sampling rate (0.05 of 10M s/Sec which is 500 KHz). You can also use the zoom tool to view the spectrum in more details.

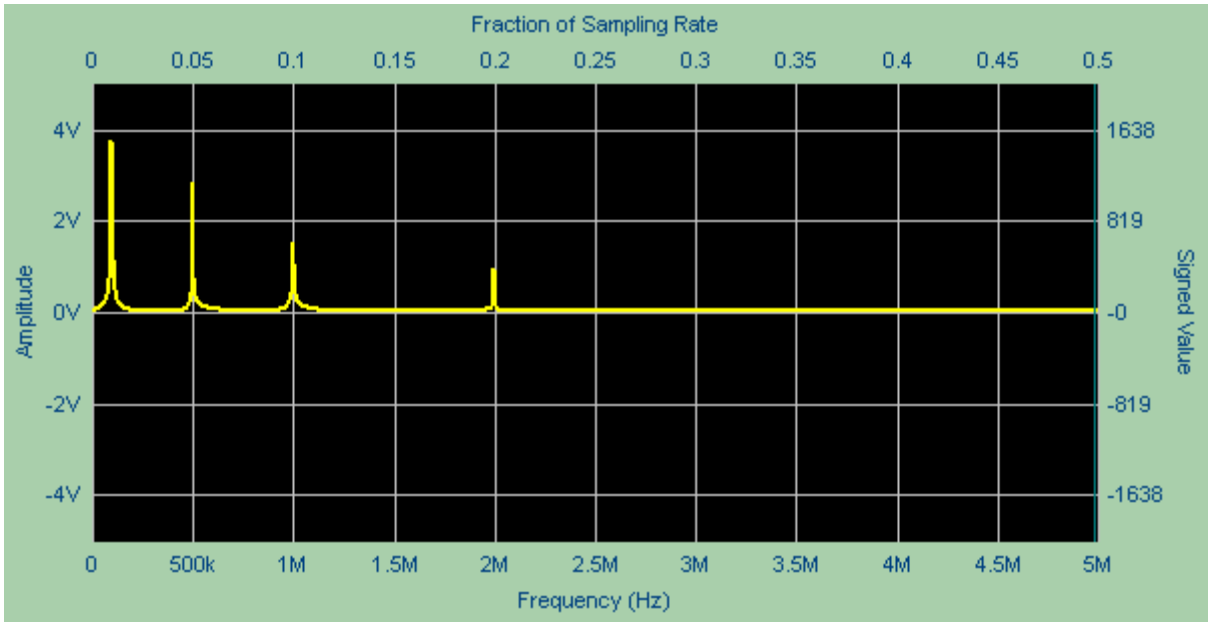


Figure 4-8: Frequency Spectrum Tab

WaveEasy Run-Time Objects

WaveEasy is supplied with a library that allows applications to read, create and modify waveform files. The scripting library is an ActiveX/COM library. The following section describes the available classes and their properties and methods.

Data types and the variable name prefixes used in this reference:

- Bool – Boolean data types, 16 bit, 0 for FALSE and other (then 0) for TRUE (usually -1).
- Word – 16 bit unsigned integer, use "w" as prefix.
- Long – 32 bit signed integer, use "l" as prefix.
- Double – 64 bit double precision floating point, use "d" as prefix.
- String – Buffer contains array of characters, use "s" as prefix.
- Variant – Data type that can hold various data types an arrays, use "v" as prefix, if expected array use "a" followd by the expected data type prefix ("ad" array of doubles) .
- [in] [out], [in, out] – input, output or both parameter, output parameter uses "p" for prefix (pointer).
- Arrays – passed in a variant, use a as "a" as prefix followed by the basic type ("aw", array of words).

Enum enumWPointsDataType

Constants used for **GetPoint**, **SetPoints** methods and the **Points** properties.

Constants:

- **WPointsDataTypeAmplitudeDouble = 0** – Amplitude Voltage (double precision floating point)
- **WPointsDataTypeSigned Short = 1** – Signed Value (short, 16 bit integer)
- **WPointsDataTypeUnsigned Word = 2** – Unsigned Value (word, 16 bit integer)

Class WWaveform

Waveform is a collection of segments that can be saved and loaded to/from a wave file.

Properties:

- **AmplitudeMax** : Double – Returns or sets the waveform maximum amplitude value.
- **AmplitudeMin** : Double – Returns or sets the waveform minimum amplitude value.
- **Description** : String – Returns or sets the waveform description.
- **Name** : String – Returns or sets the waveform identifier name.
- **Points**([in] Long lPoint, [in] enumWPointsDataType enPointsDataType = WPointsDataTypeSignedShort]) : Variant – Returns the waveform data point at the specified index.
- **PointsCount** : Long – Returns the waveform number of data points. For the GX1100 this number can be (1 to 2M points).
- **QuantizationBits** : Long – Returns or sets the waveform quantization bits. For the GX1110 use 12 bits, for the GX1120 use 16 bits .
- **SamplingRate** : Double – Returns or sets the waveform generator sampling rate. For GX1100 use a number up from 0.01 to 100,000,000 (10 mS/s – 100 MS/s).
- **SegmentsCount** : Long – Returns the waveform number of segments.
- **Segment** (Variant vrNameOrIndex) : WSegment – Returns the specified waveform segment.

Methods:

- **GetPoints**([in] Long lPointStart, [in, out] Long * plPoints, [out] Variant * pvaPoints, [enumWPointsDataType enPointsDataType = WPointsDataTypeSignedShort]) – Returns the waveform data points array.
- **SetPoints**([in] Long lPointStart, [in] Long lPoints, [in] Variant vaPoints, [in] enumWPointsDataType enPointsDataType = WPointsDataTypeSignedShort]) – Sets the waveform data points to the specified array.
- **DeleteSegment**([in] Variant vrNameOrIndex): Bool – Deletes the specified waveform segment.
- **InsertSegment**([in] Variant vrNameOrIndex, [in] String sName=”Segment1”, [in] Long lPoints=1000, [in] String sFormula=”Sin(x)) : WSegment – Inserts a new segment into the waveform.
- **Open**([in] String sFileName) : Bool – Opens and reads a waveform file.
- **Save**() : Bool – Saves the waveform to a file.
- **SaveAs**([in] String sFileName) : Bool – Save the waveform to a new file.

Class WSegment

A waveform is divided to segments. Each top level segment can also have segments below it.

Properties:

- **Formula** : String – Returns or sets the waveform formula.
- **ItemsCount** : Long – Returns the waveform number of sub-segments.
- **Items** (Variant vrNameOrIndex) : WSegment – Returns the specified waveform sub-segment.
- **Name** : String – Returns or sets the segment identifier name.
- **Parent** : WSegment – Returns the parent segment of the current segment.
- **Points**([in] Long lPoint, [in] enumWPointsDataType enPointsDataType = WPointsDataTypeSignedShort]) : Variant – Returns the waveform data point at the specified point index.
- **PointsCount** : Long – Returns or sets the segment number of data points.
- **Waveform** : WWaveform – Returns the segment waveform object.

Methods:

- **GetPoints**([in] Long lPointStart, [in, out] Long * plPoints, [out] Variant * pvaPoints, [in] enumWPointsDataType enPointsDataType = WPointsDataTypeSignedShort]) – Returns the segment data points array.
- **SetPoints**([in] Long lPointStart, Long lPoints, [in] Variant vaPoints, [in] enumWPointsDataType enPointsDataType = WPointsDataTypeSignedShort]) – Sets the segment data points to the specified array.

WDSP Module

DSP functions are used from your application to perform Fast Fourier Transform (FFT) and its inverse analysis.

Procedures:

- **FFT**([in] Variant vaSignal, [out] Variant * pvadFFTReal, [out] Variant * pvadFFTImaginary) - Calculates Fast Fourier Transform for the specified array of time-domain signals and return arrays of real and imaginary numbers.
- **FFTInvert**([in] Variant vadFFTReal, [in] Variant vadFFTImaginary, [out] Variant * pvadSignal) - Converts arrays of Fast Fourier Transform data back to an array of time-domain signals.
- **FFTComplexToPolar**([in] Variant vadFFTReal, [in] Variant vadFFTImaginary, [out] Variant * padMagnitude, [out] Variant * padPhase) - Converts Fast Fourier Transform complex real and imaginary data to magnitude and phase in polar coodinates.
- **FFTPolarToComplex**([in] Variant vadFFTMagnitude, [in] Variant vadFFTPhase, [in] Variant * pvadFFTReal, [in] Variant * pvadFFTImaginary) - Converts Fast Fourier Transform magnitude and phase polar coodinates to complex real and imaginary data.
- **FFTBandPassFilter**([in] Variant vadTimeDomainSignal, [out] VARIANT * pvadFilteredSignal, , [in] Double dFilterOrder=5, [in] Double dCutoffFrequencyLow=0, [in] Double dCutoffFrequencyHigh=0.5) - Implements a BandPass filter using the Fast Fourier Transform.

WaveEasy/GtWave Run-Time Example

The following example shows how to use the WaveEasy run-time along with the GtWave driver. The example is written in VB .NET. The example opens a form that displays several controls as shown below:

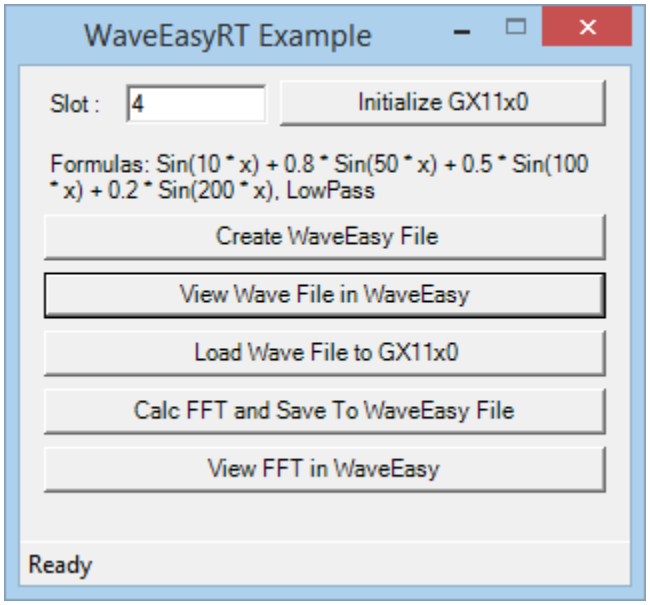


Figure 4-9: WaveEasyRT/GX11X0 Example Form

The form controls can be used as follows:

1. **Initialize GX11X0** – The GX1100 to the slot specified in the Slot edit box. Use this button if you have a GX11X0 board installed.
2. **Create WaveEasy File** – This will create a WaveEasy file that contains two segments. The first segment will contain a formula with multiple formulas and the second will perform a **FIRBandPass** filter on the same formula that is associated with the first segment.
3. **View Wave File in WaveEasy** – This will display the waveform created in the step 2 in WaveEasy.
4. **Load Wave File to GX11X0** – This will load the file created in step 2 to the GX11X0. Use this button if you have a GX11X0 board installed.
5. **Calc FFT and Save to WaveEasy File** – This will calculate an FFT on the multiple frequencies formula using the DSP module.
6. **View FFT in WaveEasy** – This will display the waveform created in the step 5 in WaveEasy.

The example does not require a GX11X0 board except for steps 1 and 4. A WaveEasy license is required in order to run the example.

The WaveEasyEasyRTExampleForm.vbproj in the WaveEasy folder contains the complete sources for this example. The following is the code used for handling the example form:

```
Imports WaveEasyRtLib
Imports System.Math

Public Class WaveEasyRtExampleForm

    Dim m_nSlot As Int16
    Dim m_nHandle As Int16
    Dim m_sWaveFileName1 As String = "WaveEasyRTExampleSinLowPass.WaveEasy"
    Dim m_sWaveFileName2 As String = "WaveEasyRTExampleSinFFT.WaveEasy"
    Dim m_wwave1 As New WWaveform
    Dim m_wwave2 As New WWaveform


    Private Sub WaveEasyViewFile(ByVal sWaveFileName As String)

        Dim iProcessID As Integer = 0
        Dim sWaveEasyPath As String = ""

        ' launch WaveEasy to view the specified file
        Try
            sWaveEasyPath =
             My.Computer.Registry.GetValue("HKEY_LOCAL_MACHINE\Software\Micro
             soft\Windows\CurrentVersion\App Paths\WaveEasy.exe", Nothing,
             Nothing)
            iProcessID = Shell(sWaveEasyPath + " " + sWaveFileName)
        Catch

        End Try
        If iProcessID = 0 Then
            MsgBox("Unable to Launch WaveEasy: '" + sWaveEasyPath + " '" +
             sWaveFileName + "'")
        Else
            'AppActivate(iProcessID)
        End If
    End Sub


    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
             System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub btnInitialize_Click(ByVal sender As System.Object, ByVal e As
             System.EventArgs) Handles btnInitialize.Click

        Dim nStatus As Int16
        Dim s As New String(" ", 256)


        m_nSlot = Val(tbSlot.Text)
        GTWAVE.GtWaveInitialize(m_nSlot, m_nHandle, nStatus)
        If nStatus < 0 Then
            GTWAVE.GtWaveGetErrorString(nStatus, s, 256, nStatus)
```

```
        MsgBox("Unable to Initialize the GX1110 on Slot " + Str(m_nSlot)
         + " : " + s)
    End If
    tssStatus.Text = "Board Initialized Successfully"

End Sub

Private Sub btnCreateWaveFile_Click(ByVal sender As System.Object, ByVal
         e As System.EventArgs) Handles btnCreateWaveFile.Click
    Dim wseg As WSegment

    If IsNothing(m_wwave1) = False Then
        ' delete content
        m_wwave1 = New WWaveform
    End If
    ' setup the wave
    m_wwave1.QuantizationBits = 12
    m_wwave1.SamplingRate = 1000000
    m_wwave1.Name = "MyWaveform"
    m_wwave1.AmplitudeMax = 5.0
    m_wwave1.AmplitudeMin = -5.0
    m_wwave1.Description = "Waveform created using WaveEasyRT library"

    ' setup the existing first segment
    wseg = m_wwave1.Segments(0)
    wseg.Name = "Sin w. Multiple Freq."
    wseg.PointsCount = 2000
    wseg.Formula = "Sin(10 * x) + 0.8 * Sin(50 * x) + 0.5 * Sin(100 * x)
         + 0.2 * Sin(200 * x)"

    ' insert second segment
    wseg = m_wwave1.InsertSegment(-1, "LowPass", 2000,
         "FIRBandPass(Sin(10 * x) + 0.8 * Sin(50 * x) + 0.5 * Sin(100 *
         x) + 0.2 * Sin(200 * x), 5, 0.0001, 0.005)")

    ' save the file - require waveeasy license
    If m_wwave1.SaveAs(m_sWaveFileName1) = False Then
        MsgBox("Unable to Save : '" + m_sWaveFileName1 + "',
         WSegment.SaveAs require a WaveEasy license.")
    End If

End Sub

Private Sub btnViewWaveFile_Click(ByVal sender As System.Object, ByVal e
         As System.EventArgs) Handles btnViewWaveFile.Click

    ' create the wave if not exist
    If IsNothing(m_wwave1) Then
        btnCreateWaveFile_Click(sender, e)
    End If

    ' display in WaveEasy
    WaveEasyViewFile(m_sWaveFileName1)
End Sub

Private Sub btnLoadWaveFile_Click(ByVal sender As System.Object, ByVal e
         As System.EventArgs) Handles btnLoadWaveFile.Click
```

```
        Dim nStatus As Int16
        Dim acChar(512) As Char

        ' initialize the card if not exist
        If m_nHandle <= 0 Then
            btnInitialize_Click(sender, e)
        End If
        If m_nHandle <= 0 Then
            Return
        End If

        ' create the wave if not exist
        If IsNothing(m_wwave1) Then
            btnCreateWaveFile_Click(sender, e)
        End If

        ' another way of initializing the board with data - require licence
        Dim lPoints As Int32 = m_wwave1.PointsCount
        Dim awPoints(lPoints) As UInt16

        ' load points from array
        m_wwave1.GetPoints(0, lPoints, awPoints)
        GTWAVE.GtWaveArbWriteWaveformData(m_nHandle, 0, 0, lPoints, awPoints,
             nStatus)
        If nStatus < 0 Then
            GTWAVE.GtWaveGetErrorString(nStatus, acChar, 512, nStatus)
            MsgBox("Unable to load points to the GX1110 board")
        End If

        ' load points from a file
        GTWAVE.GtWaveArbFileLoad(m_nHandle, 0, m_sWaveFileName1, 0, 0, -1,
             nStatus)
        If nStatus < 0 Then
            GTWAVE.GtWaveGetErrorString(nStatus, acChar, 512, nStatus)
            MsgBox("Unable to load WaveEasy file '" + m_sWaveFileName1 + "':"
             + acChar)
        End If
    End Sub


    Private Sub btnCalcFFT_Click(ByVal sender As System.Object, ByVal e As
             System.EventArgs) Handles btnCalcFFT.Click
        Dim wseg As WSegment
        Dim dsp As New WDSP
        Dim lPoints As Long
        Dim l As Long
        Dim dMaxAmplitude As Double
        Dim vSignal As Object = Nothing
        Dim vFFTReal As Object = Nothing
        Dim vFFTPhase As Object = Nothing
        Dim vFFTMagnitude As Object = Nothing
        Dim vFFTImaginary As Object = Nothing

        If IsNothing(m_wwave2) = False Then
            ' delete content
            m_wwave2 = New WWaveform
```

```
End If
' setup the wave
m_wwave2.QuantizationBits = 12
m_wwave2.SamplingRate = 1000000
m_wwave2.Name = "FFT, Frequency Spectrum Examples"
m_wwave2.AmplitudeMax = 5.0
m_wwave2.AmplitudeMin = -5.0
m_wwave2.Description = "Calc FFT and display freq spectrum on 'Sin(10
     * x) + 0.8 * Sin(50 * x) + 0.5 * Sin(100 * x) + 0.2 * Sin(200 *
     x)'"

' *** Segment 1: setup the first segment
wseg = m_wwave2.Segments(0)
wseg.Name = "Sin w. Multiple Freq."
wseg.PointsCount = 2000
wseg.Formula = "Sin(10 * x) + 0.8 * Sin(50 * x) + 0.5 * Sin(100 * x)
     + 0.2 * Sin(200 * x)"

' *** segment 2
' retrieve the first segments points data
lPoints = wseg.PointsCount
wseg.GetPoints(0, lPoints, vSignal,
     enumWPointsDataType.WPointsDataTypeAmplitudeDouble)

' use FFT to transform time domain signal into frequency domain
     spectrum
dsp.FFT(vSignal, vFFTReal, vFFTImaginary)
' convert FFT complex data into easier to understand polar coordinate
     (Magnitude + Phase)
dsp.FFTComplexToPolar(vFFTReal, vFFTImaginary, vFFTMagnitude,
     vFFTPhase)

' scale magnitude to within +-5, since we use the same axis to
     display the spectrum (instead of Hz)
lPoints = UBound(vFFTMagnitude) + 1
For l = 0 To lPoints - 1
    If Abs(vFFTMagnitude(l)) > dMaxAmplitude Then
        dMaxAmplitude = Abs(vFFTMagnitude(l))
    End If
Next
For l = 0 To lPoints - 1
    vFFTMagnitude(l) = vFFTMagnitude(l) * 5 / dMaxAmplitude
Next

' save Magnitude data into another segment which can be viewed using
     WaveEasy
' use only first half of the mag. array since the second half is a
     reverse image of the first
wseg = m_wwave2.InsertSegment(, "Frequency Spectrum Magnitude using
     FFT", lPoints / 2)
wseg.SetPoints(0, lPoints / 2, vFFTMagnitude,
     enumWPointsDataType.WPointsDataTypeAmplitudeDouble)

' *** segment 3
' now use the invert FFT to restore the original wave
dsp.FFTInvert(vFFTReal, vFFTImaginary, vSignal)
lPoints = UBound(vSignal) + 1
```

```
        wseg = m_wwave2.InsertSegment(, "Invert FFT", lPoints)
        wseg.SetPoints(0, lPoints, vSignal,
             enumWPointsDataType.WPointsDataTypeAmplitudeDouble)

        ' save the file
        If m_wwave2.SaveAs(m_sWaveFileName2) = False Then
            MsgBox("Unable to Save : '" + m_sWaveFileName1 + "',
             WSegment.SaveAs require a WaveEasy license.")
        End If

        ' *** segment 3
        ' FFT band filter
        dsp.FFTBandPassFilter(vSignal, vFFTMagnitude, 5, 0.005, 0.015)
        lPoints = UBound(vFFTMagnitude) + 1
        wseg = m_wwave2.InsertSegment(, "FFT Band Pass Filter", lPoints)
        wseg.SetPoints(0, lPoints, vFFTMagnitude,
             enumWPointsDataType.WPointsDataTypeAmplitudeDouble)

        ' save the file
        If m_wwave2.SaveAs(m_sWaveFileName2) = False Then
            MsgBox("Unable to Save : '" + m_sWaveFileName1 + "',
             WSegment.SaveAs require a WaveEasy license.")
        End If

    End Sub

    Private Sub btnViewFFT_Click(ByVal sender As System.Object, ByVal e As
             System.EventArgs) Handles btnViewFFT.Click

        ' create the wave if not exist
        If IsNothing(m_wwave2) Then
            btnCalcFFT_Click(sender, e)
        End If

        ' display in WaveEasy
        WaveEasyViewFile(m_sWaveFileName2)
    End Sub
End Class
```

Chapter 5 - Programming the Board

This chapter contains information about how to program the GX11X0 using the GtWave driver. The GtWave driver contains functions to initialize, reset, and control the GX11X0 instrument. A brief description of the functions, as well as how and when to use them, is included in this chapter. Chapter 5 and the specific instrument User's Guide contain a complete and detailed description of the available programming functions.

The driver supports many development tools. Using these tools with the driver is described in this chapter. In addition, the GX11X0 directory contains examples written for these development tools. Refer to Chapter 3 for a list of the available examples.

An example using the DLL driver with Microsoft Visual C++ 6.0 is included at the end of this chapter. Since the driver functions and parameters are identical for all operating systems and development tools, the example can serve as a framework when using other programming languages, programming tools, and other GX11X0 driver types.

The GtWave Driver

The GtWave driver is a 32 and a 64-bit Windows DLL files: GtWave.DLL and GTWAVE64.DLL. The DLLs used with 32 and 64 bit applications running under Windows. The DLL uses a device driver to access the board resources. The device driver HW.SYS (on Windows NT/2000/XP/Vista/7) or HW.VXD (on Windows 9x/Me) is installed by the setup program and is shared by other Marvin Test Solutions products (ATEasy, GXDIO, etc).

The DLLs can be used with various development tools such as Microsoft Visual C++, Borland C++ Builder, Microsoft Visual Basic, Borland Pascal or Delphi, ATEasy and more. The following paragraphs describe how to create an application that uses the driver with various development tools. Refer to the paragraph describing the specific development tool for more information.

Programming Using C/C++ Tools

The following steps are required to use the GX11X0 driver with C/C++ development tools:

- Include the GtWave.H header file in the C/C++ source file that uses the GX11X0 function. This header file is used for all driver types. The file contains function prototypes and constant declarations which are used by the compiler for the application.
- Add the required .LIB file to the projects. This can be the imported library GtWave.LIB/GtWaveLib64.LIB for 32/64 bit applications developed on Microsoft Visual C++ and GtWaveBC.LIB for Borland C++. Windows based applications that explicitly load the DLL by calling the Windows **LoadLibrary** API should not include the .LIB file in the project.
- Add code to call the GX11X0 as required by the application.
- Build the project.
- Run, test, and debug the application.

Programming Using Visual Basic

To use the driver with Visual Basic 4.0, 5.0 or 6.0 (for 32-bit applications), the user must include the GtWave.BAS to the project. For Visual Basic .NET use the GtWave.VB.

The file can be loaded using *Add File* from the Visual Basic *File menu*. The GtWave.BAS/.VB contains function declarations for the DLL driver.

Programming Using C#

To use the driver with Visual C#, the user must include the GtWave.CS to the project. The GtWave.CS contains function declarations for the DLL driver.

Programming Using Pascal/Delphi

To use the driver with Borland Pascal or Delphi, the user must include the GtWave.PAS to the project. The GtWave.PAS file contains a **unit** with function prototypes for the DLL functions. Include the GX11X0 unit in the **uses** statement before making calls to the GX11X0 functions.

Programming GtWave Boards Using ATEasy®

The ATEasy driver (GX11X0 .drv) uses the GtWave.DLL to program the board. In addition, each driver is supplied with an example that contains a program and a system file pre-configured with the ATEasy driver. Use the driver shortcut property page from the System Drivers sub-module to change the PCI slot number before attempting to run the example.

Using commands declared in the ATEasy driver are easier to use than using the DLL functions directly. The driver commands will also generate exception that allows the ATEasy application to trap errors without checking the status code returned by the DLL function after each function call.

The ATEasy driver contains commands that are similar to the DLL functions in name and parameters, with the following exceptions:

- The *nHandle* parameter is omitted. The driver handles this parameter automatically. ATEasy uses driver logical names instead i.e. PRES1, PRES2 for GX11X0.
- The *nStatus* parameter was omitted. Use the Get Status commands instead of checking the status. After calling a DLL function the ATEasy driver will check the returned status and will call the error statement (in case of an error status) to generate exception that can be easily trapped by the application using the **OnError** module event or using the **try-catch** statement.

Some ATEasy drivers contain additional commands to permit easier access to the board features. For example parameters for a function may be omitted by using a command item instead of typing the parameter value. The commands are self-documented. Their syntax is similar to English. In addition, you may generate the commands from the code editor context menu or by using the ATEasy's code completion feature instead of typing them directly.

Programming Using LabView

To use the driver with LabView, the user must include the GtWave.LLB to the project. The GtWave.llb file contains function prototypes for the DLL functions.

Using the GtWave Driver Functions

The GtWave driver contains a set of functions for the GX11X0. Function names that start with the **GtWave** prefix apply to all GtWave boards (i.e. **GtWaveGetDriverSummary**). The GtWave functions are designed with a consistent set of arguments and functionality. All boards have a function that initializes the GtWave driver for a specific board, resets the board, and displays the virtual panel. All the functions use handles to identify and reference a specific board and all functions return status and share the same functions to handle error codes.

Initialization, HW Slot Numbers and VISA Resource

The GtWave driver supports two device drivers HW and VISA which are used to initialize, identify and control the board. The user can use the **GtWaveInitialize** to initialize the board's driver using HW and **GtWaveInitializeVisa** to initialize using VISA. The following describes the two different methods used to initialize:

1. **Marvin Test Solutions' HW** – This is the default device driver that is installed by the GtWave driver. To initialize and control the board using the HW use the **GtWaveInitialize**(*nSlot, pnHandle, pnStatus*) function. The function initializes the driver for the board at the specified PXI slot number (*nSlot*) and returns boards handle. The **PXI/PCI Explorer** applet in the Windows Control Panel displays the PXI slot assignments. You can specify the *nSlot* parameter in the following way:

   - A combination of chassis number (chassis # x 256) with the chassis slot number, e.g. 0x105 for chassis 1 and slot 5. The chassis number can be set by the **PXI/PCI Explorer** applet.
   - Legacy nSlot is used by earlier versions of HW/VISA. The slot number contains no chassis number and can be changed using the **PXI/PCI Explorer** applet: 23 in this example.

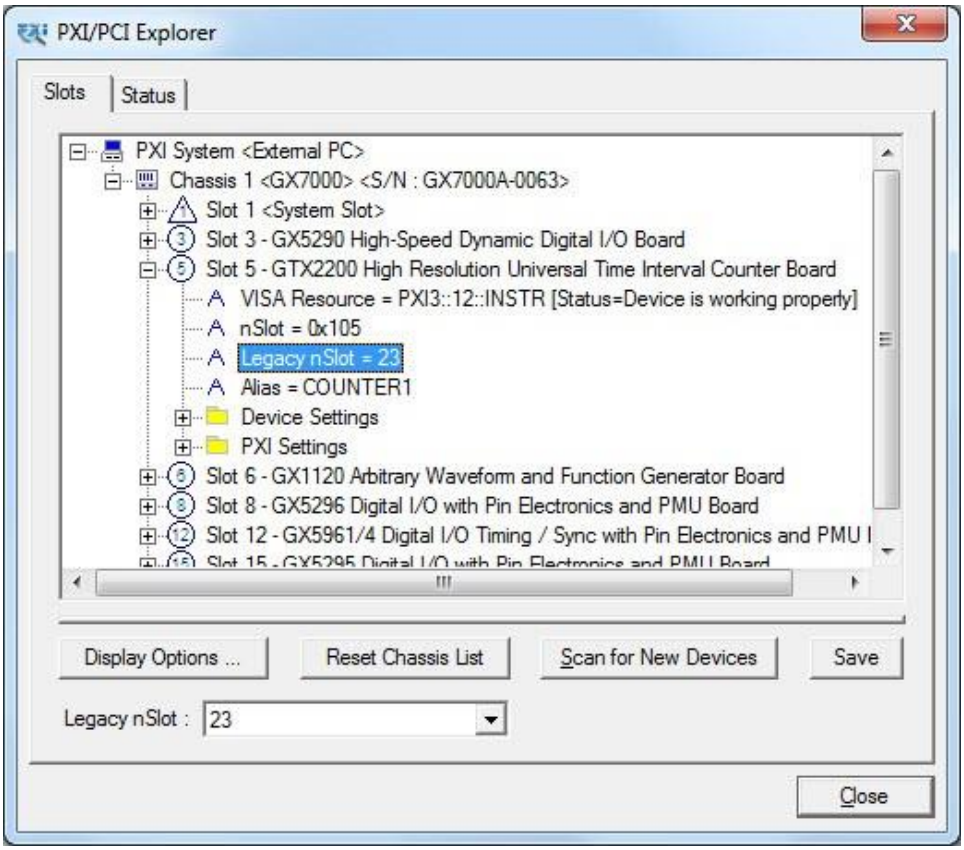


Figure 5-1: PXI/PCI Explorer

2. **VISA** – This is a third party library usually supplied by National Instruments (NI-VISA). You must ensure that the VISA installed supports PXI and PCI devices (not all VISA providers supports PXI/PCI). GtWave setup

installs a VISA compatible driver for the GtWave board in-order to be recognized by the VISA provider. Use the GtWave function **GtWaveInitializeVisa** (*szVisaResource, pnHandle, pnStatus*) to initialize the driver's board using VISA. The first argument *szVisaResource* is a string that is displayed by the VISA resource manager such as NI **Measurement and Automation** (NI_MAX). It is also displayed by Marvin Test Solutions **PXI/PCI Explorer** as shown in the prior figure. The VISA resource string can be specified in several ways as the following examples demonstrate:

- Using chassis, slot: "PXI0::CHASSIS1::SLOT5"
- Using the PCI Bus/Device combination: "PXI9::13::INSTR" (bus 9, device 9).
- Using the alias: "ARB1". Use the PXI/PCI Explorer to set the device alias.

Information about VISA is available at http://www.pxisa.org.

The **GtWaveInitialize** function returns a handle that is required by other driver functions in order to program the board. This handle is usually saved in the program as a global variable for later use when calling other functions. The initialize function does not change the state of the board or its settings.

Board Handle

The board handle argument, *nHandle* , is passed (by reference) to the parameter *pnHandle* of the **GtWaveInitialize** or the **GtWaveInitializeVisa** functions as a short integer (16 bits) number. It is used by the GtWave driver functions to identify the board being accessed by the application. Since the driver supports many boards at the same time, the *nHandle* argument is required to uniquely identify which board is being programmed.

The *nHandle* is created when the application calls the **GtWaveInitialize** function. There is no need to destroy the handle. Calling **GtWaveInitialize** with the same slot number will return the same handle.

Once the board is initialized the handle can be used with other functions to program the board.

Reset

The Reset function causes the driver to change all settings to their default state. The application software issue a Reset after the initializing the board, but a Reset can be issued at any time. All GX11X0 boards have the **GtWaveReset**(*nHandle, nStatus*) function. See the Function Reference section for more information regarding specific board functionality.

Error Handling

All GtWave functions pass a fail or success status – *pnStatus* – in the last parameter. A successful function call passes zero in the status parameter upon return. If the status is non-zero, then the function call fails. This parameter can be later used for error handling. When the status is error, the program can call the **GtWaveGetErrorString** function to return a string representing the error. The **GtWaveGetErrorString** reference contains possible error numbers and their associated error strings.

Driver Version

The **GtWaveGetDriverSummary** function can be used to return the current GtWave driver version. It can be used to differentiate between the driver versions. See the Function Reference section for more information.

Panel

Calling the **GtWavePanel** will display the instrument's front panel dialog window. The panel can be used to initialize and control the board interactively. The panel function may be used by the application to allow the user to directly interact with the board.

The **GtWavePanel** function is also used by the GtWavePANEL.EXE panel program that is supplied with this package and provides a stand-alone Windows application that displays the instrument panel.

Distributing the Driver

Once the application is developed, the driver files (GtWave.DLL and the HW device driver files located in the HW folder) can be shipped with the application. Typically, the GtWave.DLL should be copied to the Windows System directory. The HW device driver files should be installed using a special setup program HWSETUP.EXE that is provided with GtWave driver files. Alternatively, you can provide the GtWave.EXE setup program to be installed along with the board. The setup program can be invoked in silent mode (with /s) to install the driver using the command line specified settings with no user interface. Marvin Test Solutions provides permission to re-distribute the driver software providing it is in support of an application using the target hardware.

Sample Program

The following example demonstrates how to program the board using the C programming language under Windows. The example shows how to initialize the GX11X0 , set it up for measurement or trigger settings and get the reading.

To run, enter the following command line:

GtWaveExample <*Slot*> <function> <value >

Where:

<Slot>	PXI Explorer slot number where the board reside.
< function >	Set Function: • GTWAVE_FUNCTION_NOT_CONNECTED=Disconnect all Disconnect all functions • GTWAVE_FUNCTION_4WIRE=Connect 4-wire • GTWAVE_FUNCTION_2WIRE=Connect 2-wire • GTWAVE_FUNCTION_POSITIVE_VDC=Connect positive voltage GTWAVE_FUNCTION_NEGATIVE_VDC=Connect negative voltage
<value >	Set Function value

Sample Program Listing

```
/*************************************************************
   FILE     : GtWaveExampleC.cpp
   PURPOSE  : WIN32 example program for GX11X0 board
            using the GtWave driver.
   CREATED  : Mar 2008
   COPYRIGHT   : Copyright 2008 MARVIN TEST SOLUTIONS, Inc.
   COMMENTS : To compile the WIN32 example:
*************************************************************/
#include "windows.h"
#include "GtWave.h"
#include "stdio.h"

// Borland C++ Builder compat. block
#if defined(__BORLANDC__)
#pragma hdrstop
#include <condefs.h>
USELIB("GtWaveBC.lib");
USERC("GtWaveExampleC.rc");
//---------------------------------------------------------------------------
#endif // defined(__BORLANDC__)

//*************************************************************
//    DisplayMsg
//*************************************************************
void DisplayMsg(PSTR lpszMsg)
{
   MessageBeep(0);
   MessageBox(0, lpszMsg,"GtWave example program", MB_OK);
   return;
}
```

```
//*********************************************************
//    DisplayUsage
//*********************************************************
void DisplayUsage(void)
{
   DisplayMsg(
   "This example shows how to use the GX11X0:\r\n"

   "Usage:\r\n"
   "GtWaveExampleC <slot> <command> <channel> <value>"

   "\r\n\r\nWhere :"
   "<slot> - PCI/PXI slot number as shown by the PXI explorer\r\n"

   "<command> - one of the followings:\r\n"
   "\tSET_OP_MODE_FUNC = set operating mode to Function Generator\r\n"
   "\tSET_OP_MODE_ARB = set operating mode to Arbitrary Waveform
   Generator\r\n"
   "\tRUN = Run the waveform\r\n"
   "\tSTOP = Stop running the waveform\r\n"
   "\tSET_AMPLITUDE = set output Amplitude (Peak-to-Peak)\r\n"
   "\tSET_OFFSET = set output Offset\r\n"

   "<channel>: Channel A or B\r\n"

   "<value>: voltage if command is offset or amplitude\r\n"

   "\r\nTo change command line under Windows:\r\n"
   "\tRight click on the example shortcut from the start menu\r\n"
   "\tand type the new command line"
   );
exit(1);
}

//*********************************************************
//    CheckStatus
//*********************************************************
void CheckStatus(SHORT nStatus)
{
   CHAR  sz[512];

   if (!nStatus) return;
   GtWaveGetErrorString(nStatus, sz, sizeof sz, &nStatus);
   DisplayMsg(sz);
   DisplayMsg("Aborting the program...");
   exit(nStatus);
}

//*********************************************************
// MAIN
//
// This main function receives the following parameters
//
// GX11X0 GtWave board slot number (e.g. 1)
//
// <command> - one of the followings
// SET_OP_MODE_FUNC = set operating mode to Function Generator
```

```
// SET_OP_MODE_ARB = set operating mode to Arbitrary Waveform Generator
// RUN = Run the waveform
// STOP = Stop running the waveform
// SET_AMPLITUDE = set output Amplitude (Peak-to-Peak)
// SET_OFFSET = set output Offset
//
// <channel>: Channel A or B
//
// <value>: voltage if command is offset or amplitude
//
//*********************************************************
int main(int argc, char **argv)
{
short nSlotNum;      // Board slot number
char* sCommand;      // Board command
short nHandle;    // Board handle
short nStatus;    // Returned status
short nChannel;      // channel
double dValue;    // value

// Parse command line parameters
nSlotNum=(SHORT)strtol(*(++argv), NULL, 0);
if (nSlotNum<0) DisplayUsage();
GtWaveInitialize(nSlotNum, &nHandle, &nStatus);
CheckStatus(nStatus);

sCommand = strupr(*(++argv));

if(!strcmp(sCommand, "SET_OP_MODE_FUNC"))
{GtWaveSetOperationMode(nHandle, GTWAVE_OPERATING_MODE_FUNC, &nStatus);
   CheckStatus(nStatus);
   printf("Set operation mode to Function Generator\r\n");
}
else if (!strcmp(sCommand, "SET_OP_MODE_ARB"))
{GtWaveSetOperationMode(nHandle, GTWAVE_OPERATING_MODE_ARB, &nStatus);
   CheckStatus(nStatus);
   printf("Set operation mode to Arbitrary Waveform Generator\r\n");
}
else if (!strcmp(sCommand, "RUN"))
{
   nChannel=(SHORT)strtol(*(++argv), NULL, 0);
   GtWaveRun(nHandle, nChannel, &nStatus);
   CheckStatus(nStatus);
   printf("Run waveform\r\n");
}
else if (!strcmp(sCommand, "STOP"))
{  nChannel=(SHORT)strtol(*(++argv), NULL, 0);
   GtWaveStop(nHandle, nChannel, &nStatus);
   CheckStatus(nStatus);
   printf("Stop waveform\r\n");
}
else if (!strcmp(sCommand, "SET_AMPLITUDE"))
{  nChannel=(SHORT)strtol(*(++argv), NULL, 0);
   dValue =strtod(*(++argv), NULL);
   GtWaveSetAmplitude(nHandle, nChannel, dValue, &nStatus);
   CheckStatus(nStatus);
   printf("Set output Amplitude to %f\r\n", dValue);
```

```
}
else if (!strcmp(sCommand, "SET_OFFSET"))
{  nChannel=(SHORT)strtol(*(++argv), NULL, 0);
   dValue =strtod(*(++argv), NULL);
   GtWaveSetOffset(nHandle, nChannel, dValue, &nStatus);
   CheckStatus(nStatus);
   printf("Set output Offset to %f\r\n", dValue);
}
else
   DisplayUsage();

return 0;
}
//*********************************************************
// End Of File
//*********************************************************
```

Chapter 6 - Functions Reference

Introduction

The functions reference chapter organizes the list of GtWave driver functions in an alphabetical order. Each function description contains the function name; purpose, syntax, parameters description and type followed by Comments, an Example (written in C), and a See Also section.

All function and parameter syntaxes follow the same rules:

- Strings are ASCIIZ (null or zero character terminated).
- The first parameter of most functions is *nHandle* (16-bit integer). This parameter is required for operating the board and is returned by the **GtWaveInitialize** function. The *nHandle* is used to identify the board when calling a function for programming and controlling the operation of that board.
- All functions return a status with the last parameter named *pnStatus*. The *pnStatus* is zero if the function was successful, or non-zero on error. The description of the error is available using the **GtWaveGetErrorString** function or by using a predefined constant, defined in the driver interface files: GtWave.H, GtWave.BAS, GtWave.VB, GtWave.PAS or GX1100.DRV.
- Parameter name are prefixed as follows:

Prefix	Type	Example
a	Array – prefix this before the simple type.	*anArray* (Array of Short)
b	BOOL – Boolean, 0 for FALSE; <>0 for TRUE	bUpdate
d	DOUBLE – 8 bytes floating point	dReading
dw	DWORD – double word (unsigned 32-bit)	dwTimeout
hwnd	Window handle (32-bit integer).	hwndPanel
l	LONG – (signed 32-bit)	lBits
n	SHORT – (signed 16-bit)	nMode
p	Pointer – Usually used to return a value. Prefix this before the simple type.	pnStatus
sz	Null – (zero value character) terminated string	szMsg
uc	BYTE – (8 bits) unsigned.	ucValue
w	WORD – Unsigned short (unsigned 16-bit)	wParam

Table 6-1: Parameter Name Prefixes

GtWave Functions

The following list is a summary of functions available for the GX11X0:

Driver Functions	Description
General	
GtWaveInitialize	Initializes the driver for the specified slot using the HW device driver.
GtWaveInitializeVisa	Initializes the driver for the specified slot using VISA.
GtWavePanel	Opens a virtual panel to interactively control the GX11X0.
GtWaveReset	Resets the GX11X0 board to its default settings.
GtWaveGetBoardSummary	Returns the board's information.
GtWaveGetDriverSummary	Returns the driver name and version.
GtWaveGetErrorString	Returns the error string associated with the specified error number.
ARB Functions	
GtWaveArbClearWaveformMemory	Clears all of the Arb's waveform memory.
GtWaveArbFileLoad	Loads the specified file to the board.
GtWaveArbFileSave	Saves the specified board data to a file
GtWaveArbFillPredefinedWaveform	Fills the waveform memory with the specific predefined waveform.
GtWaveArbGetMarker	Returns the waveform marker bit state to the specified waveform memory locations.
GtWaveArbGetSampleRate	Returns the data point sampling rate.
GtWaveArbGetSync	Returns the waveform sync bit state to the specified waveform memory locations.
GtWaveArbGetWaveformLength	Returns the waveform start address and length.
GtWaveArbReadWaveformData	Reads an array of data from the board's waveform memory.
GtWaveArbSetMarker	Sets the waveform marker bit state to the specified waveform memory locations.
GtWaveArbSetSampleRate	Sets the data point sampling rate.
GtWaveArbSetSync	Sets the waveform sync bit state to the specified waveform memory locations.
GtWaveArbSetWaveformLength	Sets the waveform start address and length.
GtWaveArbWriteWaveformData	Writes an array of data to the board's waveform memory.

Driver Functions	Description
Function Generator Functions	
GtWaveFuncGetFmDeviation	Returns the FM deviation frequency.
GtWaveFuncGetFmFrequency	Returns the FM frequency.
GtWaveFuncGetFmSource	Returns the FM source.
GtWaveFuncGetFmState	Returns the FM state.
GtWaveFuncGetFmWaveform	Returns the FM waveform.
GtWaveFuncGetPmFrequency	Returns the PM frequency.
GtWaveFuncGetPmSource	Returns the PM source.
GtWaveFuncGetPmState	Returns the PM state.
GtWaveFuncGetPmWaveform	Returns the PM waveform.
GtWaveFuncGetPwmFrequency	Returns the PWM frequency.
GtWaveFuncGetPwmSource	Returns the PWM source.
GtWaveFuncGetPwmState	Returns the PWM state.
GtWaveFuncGetPwmWaveform	Returns the PWM waveform.
GtWaveFuncGetFrequency	Returns the function generator frequency.
GtWaveFuncGetFskFrequencies	Returns the Frequency shift keying modulation frequencies.
GtWaveFuncGetFskRate	Returns the rate of switching between the two frequencies of the Frequency shift keying modulation.
GtWaveFuncGetFskSource	Returns the Frequency shift keying source.
GtWaveFuncGetFskState	Returns the Frequency shift keying state.
GtWaveFuncGetOutToSquareDutyCycle	Returns the output to square duty cycle.
GtWaveFuncGetOutToSquareState	Returns the output Output to square state.
GtWaveFuncGetPhase	Returns the function generator output start phase in units of degrees.
GtWaveFuncGetPmDeviation	Returns the Phase Modulation deviation frequency.
GtWaveFuncGetPmFrequency	Returns the Phase Modulation frequency.
GtWaveFuncGetPmSource	Returns the Phase Modulation source.
GtWaveFuncGetPmState	Returns the Phase Modulation state.
GtWaveFuncGetPmWaveform	Returns the Phase Modulation waveform.
GtWaveFuncGetPwmDeviation	Returns the pulse-width modulation deviation frequency.
GtWaveFuncGetPwmFrequency	Returns the pulse-width modulation frequency.
GtWaveFuncGetPwmSource	Returns the pulse-width modulation source.

Driver Functions	Description
GtWaveFuncGetPwmState	Returns the pulse-width modulation state.
GtWaveFuncGetPwmWaveform	Returns the pulse-width modulation waveform.
GtWaveFuncGetPwmWidth	Returns the pulse-width modulation width.
GtWaveFuncGetSquareWaveDutyCycle	Returns the square wave duty cycle.
GtWaveFuncGetSweep	Returns the sweep parameters.
GtWaveFuncGetSweepState	Returns the sweep state.
GtWaveFuncGetWaveform	Returns the loaded waveform shape.
GtWaveFuncSetFmDeviation	Sets the FM deviation frequency.
GtWaveFuncSetFmFrequency	Sets the FM frequency.
GtWaveFuncSetFmSource	Sets the FM source.
GtWaveFuncSetFmState	Sets the FM state.
GtWaveFuncSetFmWaveform	Sets the FM waveform.
GtWaveFuncSetPmDeviation	Sets the Phase Modulation deviation
GtWaveFuncSetPmFrequency	Sets the PM frequency.
GtWaveFuncSetPmSource	Sets the PM source.
GtWaveFuncSetPmState	Sets the PM state.
GtWaveFuncSetPmWaveform	Sets the PM waveform.
GtWaveFuncSetPwmDeviation	Sets the pulse-width modulation deviation.
GtWaveFuncSetPwmFrequency	Sets the PWM frequency.
GtWaveFuncSetPwmSource	Sets the PWM source.
GtWaveFuncSetPwmState	Sets the PWM state.
GtWaveFuncSetPwmWaveform	Sets the PWM waveform.
GtWaveFuncSetPwmWidth	Sets the pulse-width modulation width.
GtWaveFuncSetFrequency	Sets the function generator frequency.
GtWaveFuncSetFskFrequencies	Sets the Frequency shift keying modulation frequencies.
GtWaveFuncSetFskRate	Sets the rate of switching between the two frequencies of the Frequency shift keying modulation.
GtWaveFuncSetFskSource	Sets the Frequency shift keying source.
GtWaveFuncSetFskState	Sets the Frequency shift keying state.
GtWaveFuncSetPhase	Sets the function generator output start phase in units of degrees.
GtWaveFuncSetOutToSquareState	Sets the output to square conversion state.

Driver Functions	Description
GtWaveFuncSetOutToSquareDutyCycle	Sets the output to square duty cycle value.
GtWaveFuncSetSquareWaveDutyCycle	Sets the square wave duty cycle.
GtWaveFuncSetSweep	Sets the sweep parameters.
GtWaveFuncSetSweepState	Sets the sweep state.
GtWaveFuncSetWaveform	Loads the specified standard waveform.
GtWaveFuncWriteWaveform	Write an array of data to the board's Function Generator waveform memory.
AM Functions	
GtWaveGetAmDepth	Returns the AM modulation depth in percentage.
GtWaveGetAmFrequency	Returns the AM modulation frequency.
GtWaveGetAmSource	Returns the AM modulation source.
GtWaveGetAmState	Returns the AM modulation state.
GtWaveGetAmWaveform	Returns the AM modulation waveform.
GtWaveSetAmDepth	Sets the AM modulation depth in percentage.
GtWaveSetAmFrequency	Sets the AM modulation frequency.
GtWaveSetAmSource	Sets the AM modulation source.
GtWaveSetAmState	Sets the AM modulation state.
GtWaveSetAmWaveform	Sets the AM modulation waveform.
Function Generator and Arbitrary Waveform Generator Functions	
GtWaveGetAmplitude	Returns the waveform's peak-to-peak amplitude voltage.
GtWaveGetBoardType	Returns the board type.
GtWaveGetFilterMode	Returns the filter mode.
GtWaveGetMarkerToPxiTriggerBusLine	Returns the Marker to PXI trigger bus line and the state.
GtWaveGetOffset	Returns the output offset voltage.
GtWaveGetOperationMode	Returns the board's operation mode.
GtWaveGetOutputState	Returns the specified channel's output state.
GtWaveGetPxiTriggerBusLine	Returns the PXI trigger bus line and state.
GtWaveGetReferenceClockSource	Returns the reference clock source
GtWaveGetStatusRegister	Returns the status register.
GtWaveGetTriggerBurstCount	Returns the trigger burst count.
GtWaveGetTriggerDelay	Returns the external trigger delay.

Driver Functions	Description
GtWaveGetTriggerEdge	Function Generator and Arbitrary Waveform Generator modes.
GtWaveGetTriggerHoldoff	Returns the external trigger hold off delay.
GtWaveGetTriggerInternalFrequency	Returns the trigger internal frequency.
GtWaveGetTriggerLevel	Returns the external trigger level.
GtWaveGetTriggerMode	Returns the trigger mode.
GtWaveGetTriggerSource	Returns the trigger source
GtWaveGetTriggerToPxiTriggerBusLine	Returns the trigger to PXI trigger bus line and state.
GtWaveGetVoltageRangeMode	Returns the Amplitude and Offset voltage mode and range.
GtWaveResetChannel	Sets the board to its default settings
GtWaveRun	Enables the board for running.
GtWaveSetAmplitude	Sets the waveform peak-to-peak amplitude voltage.
GtWaveSetFilterMode	Sets the filter mode.
GtWaveSetMarkerToPxiTriggerBusLine	Sets the Marker to PXI trigger bus line and the state.
GtWaveSetOffset	Sets the output offset voltage.
GtWaveSetOperationMode	Sets the board's operation mode.
GtWaveSetOutputState	Sets the specified channel's output state.
GtWaveSetPxiTriggerBusLine	Sets the PXI trigger bus line and state.
GtWaveSetReferenceClockSource	Sets the reference clock source.
GtWaveSetTriggerBurstCount	Sets the trigger burst count.
GtWaveSetTriggerDelay	Sets the external trigger delay.
GtWaveSetTriggerEdge	Sets the external trigger edge.
GtWaveSetTriggerHoldoff	Sets the external trigger hold off delay.
GtWaveSetTriggerInternalFrequency	Sets the trigger internal frequency.
GtWaveSetTriggerLevel	Sets the external trigger level.
GtWaveSetTriggerMode	Sets the trigger mode.
GtWaveSetTriggerSource	Sets the trigger source.
GtWaveSetTriggerToPxiTriggerBusLine	Sets the trigger to PXI trigger bus line and state.
GtWaveSetVoltageRangeMode	Sets the Amplitude and Offset voltages setting mode and voltage range.
GtWaveStop	Disables the board from running.
GtWaveSynchronizePhases	Synchronize both channels phases.

Driver Functions	Description
GtWaveTestMemory	Runs a memory self-test.
GtWaveTrig	Issues a software trigger.

GtWaveArbClearWaveformMemory

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Clears all the Arb's waveform memory.

Syntax

GtWaveArbClearWaveformMemory(*nHandle, nChannel, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Not used, pass as 0.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1120: Since the memory is shared by both channels, there is no meaning to the channel number.

Example

The following example clears all the Arb's waveform memory:

```
GtWaveArbClearWaveformMemory (nHandle, 0, &nStatus);
```

See Also

GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbClearWaveformMemory, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbFileLoad

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Load the specified file to the board.

Syntax

GtWaveArbFileLoad(nHandle, *nChannel, szFileName, dwFileStart, dwStartAddress, dwSize, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Not used, pass as 0.
szFileName	LPCSTR	File name. The saved data format will be determined by the file extention. The driveer supports the follwing file formats: • WaveEasy Files (*.WaveEasy) • NI-HWS Files (*.hws) • CSV (comma delimited) Files (*.csv) • Text Files (*.txt) • PRN (space delimited) Files (*.prn)
dwFileStart	DWORD	File start data point. GX1110: Range: 0 to 2097151. GX1120: Range: 0 to 33554431.
dwStartAddress	DWORD	Board start address to fill data.
dwSize	DWORD	Specified number of data points to load. If dwSize=-1, then the function will load all the data from the file to the board. If there are more data points in the file than available memory, i.e. max number of steps less start address, the function returns an error without loading any data. GX1110: Range: 1 to 2097152 (GTWAVE_GX1110_ARB_WAVE_MAX_LENGTH). GX1120: Range: 1 to 33554432 (GTWAVE_GX1120_ARB_WAVE_MAX_LENGTH).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110:

Since there is only one channel, there is no meaning to the channel number.

When calling this function, the function will cast all values to 12-bit wide value and will clear all Marker and Sync bits previously set at the specified memory range.

Data point value of 2047 (0x7FF) represents highest output amplitude and -2047 (0x800) represents lowest output amplitude.

The following equation represents the relative output amplitude voltage:

Output voltage = amplitude setting (peak-to-peak)* (data point value)/4096+ offset

GX1120:

Since the memory is shared by both channels, there is no meaning to the channel number.

Data point value of 0x7FFF represents highest output amplitude and 0x8000 represents lowest output amplitude.

Example

The following example loads the "WaveEasy1.WaveEasy" file to the board starting at step 0 in the file, step 0 at the board and all available data in the file:

```
GtWaveArbFileLoad(nHandle, GTWAVE_CHANNEL_A, "WaveEasy1.WaveEasy", 0, 0, -1, &nStatus);
```

See Also

GtWaveArbFileSave, GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData

GtWaveArbFileSave

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Save the specified board data to a file.

Syntax

GtWaveArbFileSave(*nHandle, nChannel, szFileName, dwFileStart, dwStartAddress, dwSize, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Not used, pass as 0.
szFileName	LPCSTR	File name. The saved data format will be determined by the file extension. The driver support the following file formats: • WaveEasy Files (*.WaveEasy) • NI-HWS Files (*.hws) • CSV (comma delimited) Files (*.csv) • Text Files (*.txt) • PRN (space delimited) Files (*.prn)
dwFileStart	DWORD	File start data point. GX1110: Range: 0 to 2097151. GX1120: Range: 0 to 33554431.
dwStartAddress	DWORD	Board starts address to read data.
dwSize	DWORD	Specified number of data points to save. If dwSize=-1, then the function will save all the data from the board. The start address and the size must meet the specification that. Start address + Length <= max number of board's steps. GX1110: Range: 1 to 2097152 (GTWAVE_GX1110_ARB_WAVE_MAX_LENGTH). GX1120: Range: 1 to 33554432 (GTWAVE_GX1120_ARB_WAVE_MAX_LENGTH).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110:

Since there is only one channel, there is no meaning to the channel number.

When calling this function, the function will cast all values to a 12-bit wide value and will clear all Marker and Sync bits previously set at the specified memory range.

A data point value of 2047 (0x7FF) represents the maximum output amplitude and -2047 (0x800) represents the lowest output amplitude.

The following equation represents the relative output amplitude voltage:

Output voltage = amplitude setting (peak-to-peak)* (data point value)/4096+ offset

GX1120:

Since the memory is shared by both channels, there is no meaning to the channel number.

Data point value of 0x7FFF represents highest output amplitude and 0x8000 represents lowest output amplitude.

Example

The following example saves 4096 data points from the board starting at step 0 in the file and, step 0 at the board:

```
GtWaveArbFileSave(nHandle, GTWAVE_CHANNEL_A, "WaveEasy1.WaveEasy", 0, 0, 4096, &nStatus);
```

See Also

GtWaveArbFileLoad, GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData,

GtWaveArbFillPredefinedWaveform

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Fill the waveform memory with the specific predefined waveform.

Syntax

GtWaveArbFillPredefinedWaveform(*nHandle, nChannel, nWaveform, dwStartAddress, dwSize, dParam1, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Not used, pass as 0.
nWaveform	SHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform 5. GTWAVE_WAVEFORM_RAMP_DC: Constant voltage 6. GTWAVE_WAVEFORM_NOISE: White noise
dwStartAddress	DWORD	Start address to fill. GX1110: Range: 0 to 2097151. GX1120: Range: 0 to 33554431.
dwSize	DWORD	Specify the number of data points to fill. If dwSize=-1, then the function will fill all steps between dwStartAddress and the board's last step. The start address and the size must meet the following conditions: Start address + Length <= max number of board's steps. GX1110: Range: 1 to 2097152 (GTWAVE_GX1110_ARB_WAVE_MAX_LENGTH). GX1120: Range: 1 to 33554432 (GTWAVE_GX1120_ARB_WAVE_MAX_LENGTH).
dParam1	DOUBLE	If standard waveform is square, this parameter specifies the duty cycle in percents.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110:

Since there is only one channel, there is no meaning to the channel number.

The function will fill the specified number of steps with the specified Predefined Waveform. The function will fill a waveform whose data values range from -2047 to +2047 and will fit one period.

The frequency of the waveform will be determined as follows: (Sampling Rate)/(Waveform size).

GX1120:

Since the memory is shared by both channels, there is no meaning to the channel number.

Data point value of 0x7FFF represents highest output amplitude and 0x8000 represents lowest output amplitude.

NOTE: The best way to generate waveforms is via the **WaveEasy** application and run time libraries. The WaveEasy development environment and script libraries can be used to create, import, edit, save and export waveform files.

Example

The following example fills the first 4096 steps with a Sine waveform:

```
GtWaveArbFillPredefinedWaveform (nHandle, GTWAVE_CHANNEL_A, GTWAVE_WAVEFORM_SINE, 0, 4096,
&nStatus);
```

See Also

GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbGetMarker

Supported By

Gx1120

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Returns the waveform marker bit state to the specified waveform memory locations.

Syntax

GtWaveArbGetMarker(*nHandle, nChannel, pdwStartAddress, pdwLength, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdwStartAddress	PDWORD	GX1110: Waveform start address, any even value between 0 and 2097151. GX1120: Waveform start address, any even value between 0 and 33554431.
dwLength	DWORD	GX1110: Waveform length can be any value between 1 and 2097152. If dwLength is -1 will automatically fill all the steps between the start address and the end of the memory. GX1120: When waveform length is set to be less than 2048, the marker can only start on multiples of 2 points and have length of multiples of 2. Otherwise, marker can only start on multiples of 8 points and have length of multiples of 8 and be between 1 and 33554432.
pbState	PBOOL	GX1110: Marker to be set to On or Off for specified waveform memory locations. GX1120: not applicable.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function returns the marker output address and length (marker pulse width) of the signal available at the marker output connector.

Example

The following example returns the waveform marker:

```
DWORD   dwStartAddress, dwLength;
BOOL    bState;
GtWaveArbGetMarker (nHandle, GTWAVE_CHANNEL_A, &dwStartAddress, &dwLength, &bState, &nStatus);
```

See Also

GtWaveArbGetMarker, GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbGetSampleRate

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Returns the data point sampling rate.

Syntax

GtWaveArbGetSampleRate(nHandle, *nChannel, pdSampleRate, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdSampleRate	PDOUBLE	GX1110: Returns the data point sampling rate, sampling rate can be between 1S/sec to 100MS/Sec. GX1120: Returns the data point sampling rate, sampling rate can be between 0.01S/Sec to 250MS/Sec.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Frequency accuracy of the output waveform is determined by the clock reference. If higher accuracy and/or stability required, an external frequency reference can be used to replace the internal 10MHz reference clock.

The waveform frequency is given by the following equation: Frequency = 1/(Sample Rate * Wavelength)

Example

The following example returns the data point sampling rate:

```
DOUBLE dSampleRate;
GtWaveArbGetSampleRate (nHandle, GTWAVE_CHANNEL_A, &dSampleRate, &nStatus);
```

See Also

GtWaveArbSetSampleRate, GtWaveArbSetMarker, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbGetSync

Supported By

Gx11200

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Returns the waveform sync bit state to the specified waveform memory locations.

Syntax

GtWaveArbGetSync(*nHandle, nChannel, pdwStartAddress, pdwLength, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdwStartAddress	PDWORD	GX1110: sync start address, any even value between 0 and 2097151. GX1120: sync start address, any even value between 0 and 33554431.
dwLength	DWORD	GX1110: sync length can be any value between 1 and 2097152. If dwLength is -1 will automatically fill all the steps between the start address and the end of the memory. GX1120: When waveform length is set to be less than 2048, the sync can only start on multiples of 2 points and have length of multiples of 2. Otherwise, sync can only start on multiples of 8 points and have length of multiples of 8 and be between 1 and 33554432.
pbState	PBOOL	GX1110: Sync to be set to On or Off to the specified waveform memory locations. GX1120: not applicable.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function returns the sync output address and length (pulse width) of the signal available at the Sync Out connector.

Example

The following example returns the waveform sync:

```
DWORD   dwStartAddress, dwLength;
BOOL    bState;
GtWaveArbSetSync (nHandle, GTWAVE_CHANNEL_A, &dwStartAddress, &dwLength, &bState, &nStatus);
```

See Also

GtWaveArbSetSync , GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbGetWaveformLength

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Returns the waveform start address and length.

Syntax

GtWaveArbGetWaveformLength (*nHandle, nChannel, pdwStartAddress, pdwLength, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdwStartAddress	PDWORD	GX1110: Waveform start address, any even value between 0 and 2097148 that can be divided by 4. GX1120: Waveform start address, any even value between 0 and 33554430 that can be divided by 2.
pdwLength	PDWORD	GX1110: Waveform length can be any value between 4 and 2097152 that can be divided by 4. E.g. to run a waveform of 4096 steps starting from step 0, start address will be 0 and length be 4096. GX1120: Waveform length can be any value between 2 and 33554432 that can be divided by 2.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the waveform start address and length:

```
DWORD dwStartAddress, dwLength;
GtWaveArbGetWaveformLength (nHandle, GTWAVE_CHANNEL_A, &dwStartAddress, &dwLength, &nStatus);
```

See Also

GtWaveArbSetWaveformLength, GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbReadWaveformData

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Read an array of data from the board's waveform memory.

Syntax

GtWaveArbReadWaveformData(*nHandle, nChannel, dwStartAddress, dwSize, pnData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Not used, pass as 0.
dwStartAddress	DWORD	GX1110: Waveform start address, any even value between 0 and 2097151. GX1120: Waveform start address, any even value between 0 and 33554431.
dwSize	DWORD	GX1110: Any value between 1 and 2097152. GX1120: Any value between 1 and 33554432.
pnData	PSHORT	Pointer to an array containing waveform data
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110:

Since there is only one channel, there is no meaning to the channel number.

When calling this function, the function will cast all the array values to a 12-bit wide value and will clear all Marker and Sync bits previously set at the specified memory range.

Data point value of 2047 (0x7FF) represents the highest output amplitude and -2047 (0x800) represents the lowest output amplitude.

The following equation represents the relative output amplitude voltage:

Output voltage = amplitude setting (peak-to-peak)* (data point value)/4096+ offset

GX1120:

Since the memory is shared by both channels, there is no meaning to the channel number.

Data point value of 0x7FFF represents the highest output amplitude and 0x8000 represents the lowest output amplitude.

Example

The following example reads an array of 4096 data points from the board's waveform memory stating at address 0:

```
GtWaveArbReadWaveformData (nHandle, GTWAVE_CHANNEL_A, 0, 4096, pnData, &nStatus);
```

See Also

GtWaveArbWriteWaveformData, GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbClearWaveformMemory, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbSetMarker

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Sets the waveform marker bit state to the specified waveform memory locations.

Syntax

GtWaveArbSetMarker(*nHandle, nChannel, dwStartAddress, dwLength, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dwStartAddress	DWORD	GX1110: Waveform start address, any even value between 0 and 2097151. GX1120: Waveform start address, any even value between 0 and 33554431.
dwLength	DWORD	GX1110: Waveform length can be any value between 1 and 2097152. If dwLength is -1 will automatically fill all the steps between the start address and the end of the memory. GX1120: When waveform length is set to be less than 2048, the marker can only start on multiples of 2 points and have length of multiples of 2. Otherwise, marker can only start on multiples of 8 points and have length of multiples of 8 and be between 1 and 33554432.
bState	BOOL	Marker to be set to On or Off for specified waveform memory locations.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function **s**elects the marker output address and length (marker pulse width) of the signal available at the marker output connector.

Example

The following example sets the waveform marker to On for 20 memory locations stating at address 2048:

```
GtWaveArbSetMarker (nHandle, GTWAVE_CHANNEL_A, 2048, 20, TRUE, &nStatus);
```

See Also

GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbSetSampleRate

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Sets data point sampling rate.

Syntax

GtWaveArbGetSampleRate(nHandle, *nChannel, pdSampleRate, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dSampleRate	DOUBLE	GX1110: Returns the data point sampling rate, sampling rate can be between 1S/sec to 100MS/Sec. GX1120: Returns the data point sampling rate, sampling rate can be between 0.01S/Sec to 250MS/Sec.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Frequency accuracy of the output waveform is determined by the clock reference. If higher accuracy and/or stability is required, an external frequency reference can be used to replace the internal 10MHz reference clock.

The waveform frequency is given by the following equation: Frequency = 1/(Sample Rate * Wavelength)

Example

The following example sets the data point sampling rate to 10MHz:

```
GtWaveArbSetSampleRate (nHandle, GTWAVE_CHANNEL_A, 10e6, &nStatus);
```

See Also

GtWaveArbGetSampleRate, GtWaveArbSetMarker, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbSetSync

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Sets the waveform sync bit state to the specified waveform memory locations.

Syntax

GtWaveArbSetSync(*nHandle, nChannel, dwStartAddress, dwLength, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dwStartAddress	DWORD	GX1110: sync start address, any even value between 0 and 2097151. GX1120: sync start address, any even value between 0 and 33554431.
dwLength	DWORD	GX1110: sync length can be any value between 1 and 2097152. If dwLength is -1 will automatically fill all the steps between the start address and the end of the memory. GX1120: When waveform length is set to be less than 2048, the sync can only start on multiples of 2 points and have length of multiples of 2. Otherwise, sync can only start on multiples of 8 points and have length of multiples of 8 and be between 1 and 33554432.
bState	BOOL	GX1110: Sync to be set to On or Off to the specified waveform memory locations. GX1120: not applicable.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function **s**elects the sync output address and length (pulse width) of the signal available at the Sync Out connector.

Example

The following example sets the waveform sync to on for 20 memory locations stating at address 2048:

```
GtWaveArbSetSync (nHandle, GTWAVE_CHANNEL_A, 2048, 20, TRUE, &nStatus);
```

See Also

GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbSetWaveformLength

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Sets the waveform start address and length.

Syntax

GtWaveArbSetWaveformLength (*nHandle, nChannel, dwStartAddress, dwLength, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dwStartAddress	DWORD	GX1110: Waveform start address, any even value between 0 and 2097148 that can be divided by 4. GX1120: Waveform start address, any even value between 0 and 33554430 that can be divided by 2.
dwLength	DWORD	GX1110: Waveform length can be any value between 4 and 2097152 that can be divided by 4. E.g. to run a waveform of 4096 steps starting from step 0, start address will be 0 and length be 4096. GX1120: Waveform length can be any value between 2 and 33554432 that can be divided by 2.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example runs a waveform of 4096 steps starting from step 4:

```
GtWaveArbSetWaveformLength (nHandle, GTWAVE_CHANNEL_A, 4, 4096, &nStatus);
```

See Also

GtWaveArbGetWaveformLength, GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbReadWaveformData, GtWaveArbWriteWaveformData, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveArbWriteWaveformData

Supported By

Gx1120, Gx1110

Applies To

Arbitrary Waveform Generator mode only.

Purpose

Write an array of data to the board's waveform memory.

Syntax

GtWaveArbWriteWaveformData(*nHandle, nChannel, dwStartAddress, dwSize, pnData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dwStartAddress	DWORD	GX1110: Waveform start address, any even value between 0 and 2097151. GX1120: Waveform start address, any even value between 0 and 33554431.
dwSize	DWORD	GX1110: Any value between 1 and 2097152. GX1120: Any value between 1 and 33554432.
pnData	PSHORT	Pointer to an array containing waveform data
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110:

When calling this function, the function will cast all the array values to 12-bit wide value and will clear all Marker and Sync bits previously set at the specified memory range.

Data point value of 2047 (0x7FF) represents t he highest output amplitude and -2047 (0x800) represents the lowest output amplitude.

The following equation represents the relative output amplitude voltage:

Output voltage = amplitude setting (peak-to-peak)* (data point value)/4096+ offset

GX1120:

Since the memory is shared by both channels, there is no meaning to the channel number.

Data point value of 0x7FFF represents the highest output amplitude and 0x8000 represents the lowest output amplitude.

Example

The following example writes an array of 4096 data points to the board's waveform memory stating at address 0:

```
GtWaveArbWriteWaveformData (nHandle, GTWAVE_CHANNEL_A, 0, 4096, pnData, &nStatus);
```

See Also

GtWaveArbReadWaveformData, GtWaveArbSetMarker, GtWaveArbSetSampleRate, GtWaveArbSetWaveformLength, GtWaveArbSetSync, GtWaveArbFillPredefinedWaveform, GtWaveArbClearWaveformMemory, GtWaveArbFileLoad, GtWaveArbFileSave

GtWaveFuncGetFmDeviation

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the FM deviation frequency.

Syntax

GtWaveFuncGetFmDeviation(*nHandle, nChannel, pdDeviation, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdDeviation	PDOUBLE	GX1120: FM deviation frequency range is 0Hz to 2MHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

When FM source is set to internal then the FM deviation frequency range is 0Hz to 2MHz. When FM source is set to external then the deviation frequency will be 100% at the maximum external modulation signal with range of -5V to +5V. The deviation will increase linearly as a percentage of the max value. E.g. an external signal with peak voltage of 2.5V and deviation value of 1KHz will result in actual deviation of 500Hz (50% of peak input voltage range of 5V).

Example

The following example returns the FM deviation frequency:

```
DOUBLE dDeviation;
GtWaveFuncGetFmDeviation (nHandle, GTWAVE_CHANNEL_A, &dDeviation, &nStatus);
```

See Also

GtWaveFuncSetFmDeviation, GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncSetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncGetFmFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the FM frequency.

Syntax

GtWaveFuncGetFmFrequency(*nHandle*, *nChannel*, *pdFrequency*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdFrequency	PDOUBLE	GX1110: Frequency range is 0.01Hz to 20KHz GX1120: Frequency range is 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the FM modulation frequency:

```
DOUBLE dFrequency;
GtWaveFuncGetFmFrequency (nHandle, GTWAVE_CHANNEL_A, &dFrequency, &nStatus);
```

See Also

GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncSetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncGetFmSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the FM modulation source.

Syntax

GtWaveFuncGetFmSource(*nHandle, nChannel, pnSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSource	PSHORT	FM modulation source can be: GTWAVE_FUNC_FM_SOURCE_INTERNAL: FM source is an internal waveform. GTWAVE_FUNC_FM_SOURCE_EXTERNAL: FM source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the FM source:

```
SHORT nSource;
GtWaveFuncGetFmSource (nHandle, GTWAVE_CHANNEL_A, &nSource, &nStatus);
```

See Also

GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncSetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncGetFmState

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the FM state.

Syntax

GtWaveFuncGetFmState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	FM state: 0=Disabled. 1=Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Note: External AM needs to be disabled before calling this function.

Example

The following example returns the FM state:

```
BOOL   bState;
GtWaveFuncGetFmFrequency (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncSetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncGetFmWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the FM modulation waveform.

Syntax

GtWaveFuncSetFmWaveform(*nHandle, nChannel, pnWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnWaveform	PSHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform 5. GTWAVE_WAVEFORM_RAMP_DC: Constant voltage 6. GTWAVE_WAVEFORM_NOISE: White noise
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the FM modulation waveform:

```
SHORT   nWaveform;
GtWaveFuncGetFmWaveform (nHandle, GTWAVE_CHANNEL_A, &nWaveform, &nStatus);
```

See Also

GtWaveFuncGetFmWaveform, **GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncGetFmState, GtWaveGetErrorString**

GtWaveFuncGetFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the function generator frequency.

Syntax

GtWaveFuncGetFrequency(*nHandle, nChannel, pdFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdFrequency	PDOUBLE	GX1110: Function generator frequency, frequency range is (GTWAVE_GX1110_FUNC_MIN_FREQUENCY) 10uHz to (GTWAVE_GX1110_FUNC_MAX_FREQUENCY) 30MHz GX1120: Function generator frequency, frequency range is (GTWAVE_GX1120_FUNC_MIN_FREQUENCY) 10uHz to (GTWAVE_GX1120_FUNC_MAX_FREQUENCY) 100MHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the function generator's frequency:

```
DOUBLE dFrequency;
GtWaveReset (nHandle, GTWAVE_CHANNEL_A, dFrequency, &nStatus);
```

See Also

GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncGetFskFrequencies

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the Frequency shift keying modulation frequencies.

Syntax

GtWaveFuncGetFskFrequencies(*nHandle, nChannel, pdHighFrequency, pdLowFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdHighFrequency	PDOUBLE	High Frequency shift keying modulation frequency. GX1110: Range from GTWAVE_GX1110_FSK_MIN_FREQUENCY (10.0E-6) to GTWAVE_GX1110_FSK_MAX_FREQUENCY (50.0E+6). GX1120: Range from GTWAVE_GX1120_FSK_MIN_FREQUENCY (1.0E-6) to GTWAVE_GX1120_FSK_MAX_FREQUENCY (100.0E+6)
pdLowFrequency	PDOUBLE	Low Frequency shift keying modulation frequency GX1110: Range from GTWAVE_GX1110_FSK_MIN_FREQUENCY (10.0E-6) to GTWAVE_GX1110_FSK_MAX_FREQUENCY (50.0E+6). GX1120: Range from GTWAVE_GX1120_FSK_MIN_FREQUENCY (1.0E-6) to GTWAVE_GX1120_FSK_MAX_FREQUENCY (100.0E+6)
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the frequency shift keying modulation frequencies:

```
DOUBLE dHighFrequency, &dLowFrequency;
GtWaveFuncGetFskFrequencies (nHandle, GTWAVE_CHANNEL_A, &dHighFrequency, &dLowFrequency,
&nStatus);
```

See Also

GtWaveFuncSetFskFrequencies, GtWaveFuncSetFskRate, GtWaveFuncSetFskSource, GtWaveFuncSetFskState, GtWaveGetErrorString

GtWaveFuncGetFskRate

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the switching rate for the two frequency shift keying frequencies.

Syntax

GtWaveFuncGetFskRate(*nHandle, nChannel, pdRate, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdRate	PDOUBLE	Rate of switching between the two frequencies of the Frequency shift keying modulation. GX1110: The rate can be set between 0.01Hz to 100KHz. GX1120: The rate can be set between 0.01Hz to 1MHz.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the switching rate for the two frequency shift keying frequencies:

```
DOUBLE dRate;
GtWaveFuncGetFskRate (nHandle, GTWAVE_CHANNEL_A, &dRate, &nStatus);
```

See Also

GtWaveFuncSetFskFrequencies, GtWaveFuncSetFskRate, GtWaveFuncSetFskSource, GtWaveFuncSetFskState, GtWaveGetErrorString

GtWaveFuncGetFskSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the Frequency shift keying source.

Syntax

GtWaveFuncGetFskSource(*nHandle, nChannel, pnSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSource	PSHORT	Frequency shift keying source: 0. GTWAVE_FSK_SOURCE_INTERNAL: Frequency shift keying source is internal. 1. GTWAVE_FSK_SOURCE_EXTERNAL Frequency shift keying source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the Frequency shift keying source:

```
SHORT  nSource;
GtWaveReset (nHandle, GTWAVE_CHANNEL_A, &nSource, &nStatus);
```

See Also

GtWaveFuncSetFskFrequencies, GtWaveFuncSetFskRate, GtWaveFuncSetFskSource, GtWaveFuncSetFskState, GtWaveGetErrorString

GtWaveFuncGetFskState

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the Frequency shift keying state.

Syntax

GtWaveFuncGetFskState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	Frequency shift keying state: 0. Disabled. 1. Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the Frequency shift keying state:

```
BOOL    bState;
GtWaveFuncGetFskState (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveFuncSetFskFrequencies, GtWaveFuncSetFskRate, GtWaveFuncSetFskSource, GtWaveFuncSetFskState, **GtWaveGetErrorString**

GtWaveFuncGetOutToSquareDutyCycle

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the output's square wave duty cycle.

Syntax

GtWaveFuncGetOutToSquareDutyCycle (*nHandle, nChannel, pdDutyCycle, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdDutyCycle	PDOUBLE	Returns the output to square duty cycle, range is from 20% to 80%.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

This function is only active when the Output to square wave state is enabled.

Example

The following example returns the output's square wave duty cycle:

```
DOUBLE dDutyCycle;
GtWaveFuncGetOutToSquareDutyCycle (nHandle, GTWAVE_CHANNEL_A, &dDutyCycle, &nStatus);
```

See Also

GtWaveFuncSetOutToSquareDutyCycle, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncGetOutToSquareState

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the output to square state.

Syntax

GtWaveFuncGetOutToSquareState (*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	Returns the output to square state: 0=FALSE: Out to square state not active. 1=TRUE: Out to square state active.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the output to square state:

```
BOOL    bState;
GtWaveFuncGetOutToSquareState (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveFuncSetOutToSquareDutyCycle, **GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase**, **GtWaveGetErrorString**

GtWaveFuncGetPhase

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the function generator's output starting phase in units of degrees.

Syntax

GtWaveFuncGetPhase(*nHandle, nChannel, pdPhase, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdPhase	PDOUBLE	Output start phase in units of degrees, -360 to 360.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The parameter defines the starting phase angle for the sine waveform.

Example

The following example returns the function generator's phase:

```
DOUBLE dPhase;
GtWaveFuncGetPhase (nHandle, GTWAVE_CHANNEL_A, &dPhase, &nStatus);
```

See Also

GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncGetPmDeviation

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the Phase Modulation deviation.

Syntax

GtWaveFuncGetPmDeviation(*nHandle, nChannel, pdDeviation, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdDeviation	PDOUBLE	GX1120: Phase Modulation deviation range is 0 to 360 degrees.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the Phase Modulation deviation:

```
DOUBLE dDeviation;
GtWaveFuncGetPmDeviation (nHandle, GTWAVE_CHANNEL_A, &dDeviation, &nStatus);
```

See Also

GtWaveFuncSetPmDeviation, GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncGetPmFrequency

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the Phase Modulation frequency.

Syntax

GtWaveFuncGetPmFrequency(*nHandle*, *nChannel*, *pdFrequency*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdFrequency	PDOUBLE	GX1120: Frequency range is 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the Phase Modulation frequency:

```
DOUBLE  dFrequency;
GtWaveFuncGetPmFrequency (nHandle, GTWAVE_CHANNEL_A, &dFrequency, &nStatus);
```

See Also

GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncGetPmSource

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the Phase Modulation source.

Syntax

GtWaveFuncGetPmSource(*nHandle, nChannel, pnSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSource	PSHORT	PM modulation source can be: 0. GTWAVE_FUNC_PM_SOURCE_INTERNAL: Phase Modulation source is internal waveform. 1. GTWAVE_FUNC_PM_SOURCE_EXTERNAL: Phase Modulation source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PM source:

```
SHORT nSource;
GtWaveFuncGetPmSource (nHandle, GTWAVE_CHANNEL_A, &nSource, &nStatus);
```

See Also

GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncGetPmState

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the Phase Modulation state.

Syntax

GtWaveFuncGetPmState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	Phase Modulation state: 0. Disabled. 1. Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PM state:

```
BOOL    bState;
GtWaveFuncGetPmState (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncGetPmWaveform

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the Phase Modulation waveform.

Syntax

GtWaveFuncSetPmWaveform(*nHandle, nChannel, pnWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnWaveform	PSHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PM modulation waveform:

```
SHORT   nWaveform;
GtWaveFuncGetPmWaveform (nHandle, GTWAVE_CHANNEL_A, &nWaveform, &nStatus);
```

See Also

GtWaveFuncSetPmWaveform, **GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncGetPmState, GtWaveGetErrorString**

GtWaveFuncGetPwmDeviation

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the pulse-width modulation deviation.

Syntax

GtWaveFuncGetPwmDeviation(*nHandle, nChannel, pdDeviation, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdDeviation	PDOUBLE	GX1120: Pulse-width modulation deviation (Gain) is 0% to 100%
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the pulse-width modulation deviation:

```
DOUBLE dDeviation;
GtWaveFuncGetPwmDeviation (nHandle, GTWAVE_CHANNEL_A, &dDeviation, &nStatus);
```

See Also

GtWaveFuncSetPwmDeviation, GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncGetPwmFrequency

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the pulse-width modulation frequency.

Syntax

GtWaveFuncGetPwmFrequency(*nHandle, nChannel, pdFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdFrequency	PDOUBLE	GX1120: Frequency range is 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the pulse-width modulation frequency:

```
DOUBLE  dFrequency;
GtWaveFuncGetPwmFrequency (nHandle, GTWAVE_CHANNEL_A, &dFrequency, &nStatus);
```

See Also

GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncGetPwmSource

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the pulse-width modulation source.

Syntax

GtWaveFuncGetPwmSource(*nHandle, nChannel, pnSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSource	PSHORT	PWM modulation source can be: 0. GTWAVE_FUNC_PWM_SOURCE_INTERNAL: pulse-width modulation source is internal waveform. 1. GTWAVE_FUNC_PWM_SOURCE_EXTERNAL: pulse-width modulation source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PWM source:

```
SHORT nSource;
GtWaveFuncGetPwmSource (nHandle, GTWAVE_CHANNEL_A, &nSource, &nStatus);
```

See Also

GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncGetPwmState

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the pulse-width modulation state.

Syntax

GtWaveFuncGetPwmState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	Pulse-width modulation state: 0. Disabled. 1. Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PWM state:

```
BOOL    bState;
GtWaveFuncGetPwmState (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncGetPwmWaveform

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the pulse-width modulation waveform.

Syntax

GtWaveFuncSetPwmWaveform(*nHandle, nChannel, pnWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnWaveform	PSHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PWM modulation waveform:

```
SHORT   nWaveform;
GtWaveFuncGetPwmWaveform (nHandle, GTWAVE_CHANNEL_A, &nWaveform, &nStatus);
```

See Also

GtWaveFuncSetPwmWaveform, **GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncGetPwmState, GtWaveGetErrorString**

GtWaveFuncGetPwmWidth

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the pulse-width modulation width.

Syntax

GtWaveFuncGetPwmWidth(*nHandle, nChannel, pdWidth, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdWidth	PDOUBLE	Pulse-width range is 0% to 100%
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PWM width:

```
DOUBLE  dWidth;
GtWaveFuncGetPwmWidth (nHandle, GTWAVE_CHANNEL_A, &dWidth, &nStatus);
```

See Also

GtWaveFuncSetPwmWidth, **GtWaveFuncGetPwmWaveform**, **GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncGetPwmState, GtWaveGetErrorString**

GtWaveFuncGetSquareWaveDutyCycle

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the square wave duty cycle.

Syntax

GtWaveFuncGetSquareWaveDutyCycle(*nHandle, nChannel, pdDutyCycle, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdDutyCycle	PDOUBLE	Square wave duty cycle is in percentage and can be 10% to 90%.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The Function Generator needs to be loaded with a Square wave prior calling this function.

Example

The following example returns the square wave duty cycle:

```
DOUBLE dDutyCycle;
GtWaveFuncGetSquareWaveDutyCycle (nHandle, GTWAVE_CHANNEL_A, &dDutyCycle, &nStatus);
```

See Also

GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncGetSweep

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the sweep parameters.

Syntax

GtWaveFuncGetSweep(*nHandle, nChannel, pdStartFreq, pdStopFreq , pdTime , pnMode , pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdStartFreq	PDOUBLE	Sweep stop frequency range is 1 uHz to 100MHz
pdStopFreq	PDOUBLE	Sweep stop frequency range is 1 uHz to 100MHz
pdTime	PDOUBLE	Sweep time range is from 1mSec to 500Sec
pnMode	PSHORT	Sweep Mode: 0. GTWAVE_GX1120_SWEEP_MODE_LINEAR: linear sweep 1. GTWAVE_GX1120_SWEEP_MODE_LOGARITHMIC: logarithmic sweep
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the sweep parameters:

```
DOUBLE dStartFreq, dStopFreq, dTime;
SHORT nMode;
GtWaveFuncGetSweep (nHandle, GTWAVE_CHANNEL_A, 100, 100000, 0.1, GTWAVE_GX1120_SWEEP_MODE_LINEAR,
                    &nStatus);
```

See Also

GtWaveFuncGetSweepState, GtWaveFuncGetSweep, GtWaveFuncSetSweep, GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncGetSweepState

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Returns the sweep state.

Syntax

GtWaveFuncGetSweepState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	Sweep state: 0. Disabled. 1. Enabled
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the sweep state:

```
GtWaveFuncGetSweepState (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveFuncSetSweepState, GtWaveFuncSetSweep, GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncGetWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Returns the loaded waveform type.

Syntax

GtWaveFuncGetWaveform(*nHandle, nChannel, pnWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnWaveform	PSHORT	Standard waveforms can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform 5. GTWAVE_WAVEFORM_RAMP_DC: Constant voltage 6. GTWAVE_WAVEFORM_NOISE: White noise
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example loads a sinusoid waveform:

```
SHORT   nWaveform;
GtWaveFuncGetWaveform (nHandle, GTWAVE_CHANNEL_A, &nWaveform, &nStatus);
```

See Also

GtWaveFuncSetWaveform, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveFuncReadWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Reads an array of data from the board's Function Generator waveform memory.

Syntax

GtWaveFuncReadWaveform (*nHandle, nChannel, dwStartAddress, dwSize, pnData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dwStartAddress	DWORD	GX1110: Waveform start address, any even value between 0 and 2097150. GX1120: Waveform start address, any even value between 0 and 33554430.
dwSize	DWORD	GX1110: Waveform start address, any even value between 0 and 2097151. GX1120: Waveform start address, any even value between 0 and 33554431.
pnData	PSHORT	Pointer to an array containing waveform data
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110:

When calling this function, the function will cast all the array values to 12-bit wide value.

Data point value of 2047 (0x7FF) represents t he highest output amplitude and -2047 (0x800) represents the lowest output amplitude.

GX1120:

Data point value of 0x7FFF represents t he highest output amplitude and -0x8000 represents the lowest output amplitude.

Example

The following example reads 4096 data points from the board's waveform memory stating at address 0:

```
GtWaveFuncReadWaveform (nHandle, GTWAVE_CHANNEL_A, 0, 4096, pnData, &nStatus);
```

See Also

GtWaveFuncWriteWaveform, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveFuncSetFmDeviation

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the FM deviation frequency.

Syntax

GtWaveFuncSetFmDeviation(*nHandle, nChannel, dDeviation, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dDeviation	DOUBLE	GX1120: FM deviation frequency range is 0Hz to 2MHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

When the FM source is set to internal then the FM deviation frequency range is 0Hz to 2MHz. When the FM source is set to external then the deviation frequency will be 2x the carrier frequency over the range of -5V to +5V. The deviation will increase linearly as a percentage of the max value. E.g. an external signal with peak voltage of 2.5V will result in a deviation that is +/- 50% of the carrier frequency.

Example

The following example sets the FM deviation frequency to 10KHz:

```
GtWaveFuncSetFmDeviation (nHandle, GTWAVE_CHANNEL_A, 10e3, &nStatus);
```

See Also

GtWaveFuncGetFmDeviation, GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncSetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncSetFmFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the FM frequency.

Syntax

GtWaveFuncGetFmFrequency(*nHandle*, *nChannel*, *dFrequency*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dFrequency	DOUBLE	GX1110: Frequency range is 0.01Hz to 20KHz GX1120: Frequency range is 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the FM modulation frequency to 10KHz:

```
GtWaveFuncSetFmFrequency (nHandle, GTWAVE_CHANNEL_A, 10e3, &nStatus);
```

See Also

GtWaveFuncGetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncSetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncSetFmSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the FM source.

Syntax

GtWaveFuncSetFmSource(*nHandle, nChannel, nSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSource	SHORT	FM source can be: 0. GTWAVE_FUNC_FM_SOURCE_INTERNAL: FM source is internal waveform. 1. GTWAVE_FUNC_FM_SOURCE_EXTERNAL: FM source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Note: External AM needs to be disabled before calling this function.

Example

The following example sets the FM modulation source to internal:

```
GtWaveFuncSetFmSource (nHandle, GTWAVE_CHANNEL_A, GTWAVE_FUNC_FM_SOURCE_INTERNAL, &nStatus);
```

See Also

GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncSetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncSetFmState

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the FM modulation state.

Syntax

GtWaveFuncSetFmState (*nHandle, nChannel, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	FM state: 0=Disabled. 1=Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Note: External AM needs to be disabled before calling this function.

Example

The following example enables FM modulation:

```
GtWaveFuncSetFmState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncGetFmState, GtWaveFuncSetFmWaveform, GtWaveGetErrorString

GtWaveFuncSetFmWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the FM modulation waveform.

Syntax

GtWaveFuncSetFmWaveform(*nHandle, nChannel, nWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nWaveform	SHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform 5. GTWAVE_WAVEFORM_RAMP_DC: Constant voltage 6. GTWAVE_WAVEFORM_NOISE: White noise
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the FM modulation waveform to sine:

```
GtWaveFuncSetFmWaveform (nHandle, GTWAVE_CHANNEL_A, GTWAVE_WAVEFORM_SINE, &nStatus);
```

See Also

GtWaveFuncSetFmFrequency, GtWaveFuncSetFmSource, GtWaveFuncGetFmState, GtWaveGetErrorString

GtWaveFuncSetFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the function generator frequency.

Syntax

GtWaveFuncSetFrequency(*nHandle, nChannel, dFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dFrequency	DOUBLE	GX1110: Function generator frequency, frequency range is (GTWAVE_GX1110_FUNC_MIN_FREQUENCY) 10uHz to (GTWAVE_GX1110_FUNC_MAX_FREQUENCY) 30MHz GX1120: Function generator frequency, frequency range is (GTWAVE_GX1120_FUNC_MIN_FREQUENCY) 10uHz to (GTWAVE_GX1120_FUNC_MAX_FREQUENCY) 100MHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the function generator frequency to 10KHz:

```
DOUBLE dFrequency;
GtWaveFuncSetFrequency (nHandle, dFrequency, 10E3, &nStatus);
```

See Also

GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncGetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncSetFskFrequencies

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the Frequency shift keying modulation frequencies.

Syntax

GtWaveFuncSetFskFrequencies(*nHandle*, *nChannel*, *dHighFrequency*, *dLowFrequency*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dHighFrequency	DOUBLE	High Frequency shift keying modulation frequency. GX1110: Range from GTWAVE_GX1110_FSK_MIN_FREQUENCY (10.0E-6) to GTWAVE_GX1110_FSK_MAX_FREQUENCY (50.0E+6). GX1120: Range from GTWAVE_GX1120_FSK_MIN_FREQUENCY (1.0E-6) to GTWAVE_GX1120_FSK_MAX_FREQUENCY (100.0E+6)
dLowFrequency	DOUBLE	Low Frequency shift keying modulation frequency GX1110: Range from GTWAVE_GX1110_FSK_MIN_FREQUENCY (10.0E-6) to GTWAVE_GX1110_FSK_MAX_FREQUENCY (50.0E+6). GX1120: Range from GTWAVE_GX1120_FSK_MIN_FREQUENCY (1.0E-6) to GTWAVE_GX1120_FSK_MAX_FREQUENCY (100.0E+6)
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the frequency for shift keying modulation frequencies:

```
GtWaveFuncSetFskFrequencies (nHandle, GTWAVE_CHANNEL_A, 100, 1000, &nStatus);
```

See Also

GtWaveFuncGetFskFrequencies, GtWaveFuncSetFskRate, GtWaveFuncSetFskSource, GtWaveFuncSetFskState, GtWaveGetErrorString

GtWaveFuncSetFskRate

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the FSK switching rate for Frequency shift keying modulation.

Syntax

GtWaveFuncSetFskRate(*nHandle, nChannel, dRate, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dRate	DOUBLE	Rate of switching between the two frequencies for frequency shift keying modulation. GX1110: The rate can be set between 0.01Hz to 100KHz. GX1120: The rate can be set between 0.01Hz to 1MHz.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the FSK switching for frequency shift keying modulation to 100Hz:

```
GtWaveFuncSetFskRate (nHandle, GTWAVE_CHANNEL_A, 100, &nStatus);
```

See Also

GtWaveFuncSetFskFrequencies, GtWaveFuncGetFskRate, GtWaveFuncSetFskSource, GtWaveFuncSetFskState, GtWaveGetErrorString

GtWaveFuncSetFskSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the Frequency shift keying source.

Syntax

GtWaveFuncSetFskSource(*nHandle, nChannel, nSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSource	SHORT	Frequency shift keying source: 0. GTWAVE_FSK_SOURCE_INTERNAL: Frequency shift keying source is internal. 1. GTWAVE_FSK_SOURCE_EXTERNAL Frequency shift keying source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the Frequency shift keying source to internal:

```
SHORT   nSource;
GtWaveFuncSetFskSource (nHandle, GTWAVE_CHANNEL_A, GTWAVE_FSK_SOURCE_INTERNAL, &nStatus);
```

See Also

GtWaveFuncSetFskFrequencies, GtWaveFuncSetFskRate, GtWaveFuncGetFskSource, GtWaveFuncSetFskState, GtWaveGetErrorString

GtWaveFuncSetFskState

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the Frequency shift keying state.

Syntax

GtWaveFuncGetFskState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	Frequency shift keying state: 0. Disabled. 1. Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the Frequency shift keying state to enable:

```
BOOL    bState;
GtWaveFuncSetFskState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveFuncSetFskFrequencies, GtWaveFuncSetFskRate, GtWaveFuncSetFskSource, GtWaveFuncGetFskState, **GtWaveGetErrorString**

GtWaveFuncSetOutToSquareDutyCycle

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the output to square duty cycle value.

Syntax

GtWaveFuncSetOutToSquareDutyCycle (*nHandle, nChannel, dDutyCycle, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dDutyCycle	DOUBLE	Output to square duty cycle, range is from 20% to 80%.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

This function is only active when the Output to square state is enabled.

The function is only applicable when generating a sine waveform. When enabled, a comparator is connected to the sine waveforms and convert the output to a square waveform. The duty cycle of that square waveform can be programmed by calling the GtWaveFuncSetOutToSquareDutyCycle API. The output square wave is much cleaner waveform since it was derived directly from a sine wave.

Example

The following example sets the output to square duty cycle to 35%:

```
GtWaveFuncSetOutToSquareDutyCycle (nHandle, GTWAVE_CHANNEL_A, 35, &nStatus);
```

See Also

GtWaveFuncGetOutToSquareDutyCycle, **GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase**, **GtWaveGetErrorString**

GtWaveFuncSetOutToSquareState

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the output to square state.

Syntax

GtWaveFuncSetOutToSquareState (*nHandle, nChannel, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	Output to square state: 0=FALSE: Out to square state not active. 1=TRUE: Out to square state active.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function is only applicable when generating a sine waveform. When enabled, a comparator is connected to the sine waveforms and convert the output to a square waveform. The duty cycle of that square waveform can be programmed by calling the GtWaveFuncSetOutToSquareDutyCycle API. The output square wave is much cleaner waveform since it was derived directly from a sine wave.

Example

The following example sets the output to square state:

```
GtWaveFuncSetOutToSquareState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveFuncGetOutToSquareDutyCycle, **GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase**, **GtWaveGetErrorString**

GtWaveFuncSetPhase

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the function generator's output start phase in units of degrees.

Syntax

GtWaveFuncGetPhase(*nHandle, nChannel, dPhase, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dPhase	DOUBLE	Output start phase in units of degrees, -360 to 360.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The parameter defines the starting phase angle for the sine waveform.

Example

The following example sets the function generator phase to 180 degrees:

```
GtWaveFuncSetPhase (nHandle, GTWAVE_CHANNEL_A, 180, &nStatus);
```

See Also

GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncGetPhase, **GtWaveGetErrorString**

GtWaveFuncSetPmDeviation

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the Phase Modulation deviation

Syntax

GtWaveFuncSetPmDeviation(*nHandle, nChannel, dDeviation, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dDeviation	DOUBLE	GX1120: Phase Modulation deviation range is 0 to 360 degrees.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the Phase Modulation deviation:

```
GtWaveFuncSetPmDeviation (nHandle, GTWAVE_CHANNEL_A, 180, &nStatus);
```

See Also

GtWaveFuncGetPmDeviation, GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncSetPmFrequency

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the Phase Modulation frequency.

Syntax

GtWaveFuncSetPmFrequency(*nHandle, nChannel, pdFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdFrequency	DOUBLE	GX1120: Frequency range is 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the Phase modulation frequency:

```
GtWaveFuncSetPmFrequency (nHandle, GTWAVE_CHANNEL_A, 1000, &nStatus);
```

See Also

GtWaveFuncGetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncSetPmSource

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the Phase Modulation source.

Syntax

GtWaveFuncSetPmSource(*nHandle, nChannel, nSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSource	SHORT	PM modulation source can be: 0. GTWAVE_FUNC_PM_SOURCE_INTERNAL: Phase Modulation source is internal waveform. 1. GTWAVE_FUNC_PM_SOURCE_EXTERNAL: Phase Modulation source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the PM source:

```
GtWaveFuncSetPmSource (nHandle, GTWAVE_CHANNEL_A, GTWAVE_FUNC_PM_SOURCE_INTERNAL, &nStatus);
```

See Also

GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncSetPmState

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the Phase Modulation state.

Syntax

GtWaveFuncSetPmState(*nHandle, nChannel, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	Phase Modulation state: 0. Disabled. 1. Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the PM state:

```
GtWaveFuncSetPmState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncSetPmState, GtWaveFuncSetPmWaveform, GtWaveGetErrorString

GtWaveFuncSetPmWaveform

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the Phase Modulation waveform.

Syntax

GtWaveFuncSetPmWaveform(*nHandle, nChannel, nWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nWaveform	SHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the PM modulation waveform:

```
GtWaveFuncSetPmWaveform (nHandle, GTWAVE_CHANNEL_A, GTWAVE_WAVEFORM_SINE, &nStatus);
```

See Also

GtWaveFuncGetPmWaveform, **GtWaveFuncSetPmFrequency, GtWaveFuncSetPmSource, GtWaveFuncGetPmState, GtWaveGetErrorString**

GtWaveFuncSetPwmDeviation

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the pulse-width modulation deviation.

Syntax

GtWaveFuncSetPwmDeviation(*nHandle, nChannel, dDeviation, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dDeviation	DOUBLE	GX1120: Pulse-width modulation deviation (gain) is 0% to 100%
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the pulse-width modulation deviation:

```
GtWaveFuncSetPwmDeviation (nHandle, GTWAVE_CHANNEL_A, 50, &nStatus);
```

See Also

GtWaveFuncGetPwmDeviation, GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncSetPwmFrequency

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the pulse-width modulation frequency.

Syntax

GtWaveFuncSetPwmFrequency(*nHandle, nChannel, dFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dFrequency	DOUBLE	GX1120: Frequency range is 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the pulse-width modulation frequency:

```
GtWaveFuncSetPwmFrequency (nHandle, GTWAVE_CHANNEL_A, 1000, &nStatus);
```

See Also

GtWaveFuncGetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncSetPwmSource

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the pulse-width modulation source.

Syntax

GtWaveFuncSetPwmSource(*nHandle, nChannel, nSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSource	SHORT	PWM modulation source can be: 0. GTWAVE_FUNC_PWM_SOURCE_INTERNAL: pulse-width modulation source is internal waveform. 1. GTWAVE_FUNC_PWM_SOURCE_EXTERNAL: pulse-width modulation source is external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the PWM source:

```
GtWaveFuncSetPwmSource (nHandle, GTWAVE_CHANNEL_A, GTWAVE_FUNC_PWM_SOURCE_INTERNAL, &nStatus);
```

See Also

GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncSetPwmState

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the pulse-width modulation state.

Syntax

GtWaveFuncSetPwmState(*nHandle, nChannel, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	Pulse-width modulation state: 0. Disabled. 1. Enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the PWM state:

```
GtWaveFuncSetPwmState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncSetPwmState, GtWaveFuncSetPwmWaveform, GtWaveGetErrorString

GtWaveFuncSetPwmWaveform

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the pulse-width modulation waveform.

Syntax

GtWaveFuncSetPwmWaveform(*nHandle, nChannel, nWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nWaveform	SHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the PWM modulation waveform:

```
GtWaveFuncSetPwmWaveform (nHandle, GTWAVE_CHANNEL_A, GTWAVE_WAVEFORM_SINE, &nStatus);
```

See Also

GtWaveFuncGetPwmWaveform, **GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncGetPwmState, GtWaveGetErrorString**

GtWaveFuncSetPwmWidth

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the pulse-width modulation width.

Syntax

GtWaveFuncSetPwmWidth(*nHandle, nChannel, dWidth, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dWidth	DOUBLE	Pulse-width range is 0% to 100%
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the PWM width:

```
DOUBLE  dWidth;
GtWaveFuncGetPwmWidth (nHandle, GTWAVE_CHANNEL_A, 50, &nStatus);
```

See Also

GtWaveFuncSetPwmWidth, **GtWaveFuncGetPwmWaveform**, **GtWaveFuncSetPwmFrequency, GtWaveFuncSetPwmSource, GtWaveFuncGetPwmState, GtWaveGetErrorString**

GtWaveFuncSetSquareWaveDutyCycle

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Sets the square wave duty cycle.

Syntax

GtWaveFuncSetSquareWaveDutyCycle(*nHandle, nChannel, dDutyCycle, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dDutyCycle	DOUBLE	Square wave duty cycle is in percentage and can be 10% to 90%.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The Function Generator needs to be loaded with a Square wave prior calling this function.

Example

The following example sets square wave duty cycle to 15%:

```
GtWaveFuncSetSquareWaveDutyCycle (nHandle, GTWAVE_CHANNEL_A, 15, &nStatus);
```

See Also

GtWaveFuncSetWaveform, GtWaveFuncGetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncSetSweep

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the sweep parameters.

Syntax

GtWaveFuncSetSweep(*nHandle, nChannel, dStartFreq, dStopFreq , dTime , nMode , pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dStartFreq	DOUBLE	Sweep start frequency range is 1 uHz to 100MHz
dStopFreq	DOUBLE	Sweep stop frequency range is 1 uHz to 100MHz
dTime	DOUBLE	Sweep time range is from 1mSec to 500Sec
nMode	SHORT	Sweep Mode: 0. GTWAVE_GX1120_SWEEP_MODE_LINEAR: linear sweep 1. GTWAVE_GX1120_SWEEP_MODE_LOGARITHMIC: logarithmic sweep
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the sweep parameters:

```
GtWaveFuncSetSweep (nHandle, GTWAVE_CHANNEL_A, 100, 100000, 0.1, GTWAVE_GX1120_SWEEP_MODE_LINEAR,
                    &nStatus);
```

See Also

GtWaveFuncGetSweepState, GtWaveFuncGetSweep, GtWaveFuncSetSweep, GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncSetSweepState

Supported By

Gx1120

Applies To

Function Generator mode only.

Purpose

Sets the sweep state.

Syntax

GtWaveFuncSetSweepState(*nHandle, nChannel, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	Sweep state: 0. Disabled. 1. Enabled
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the sweep state:

```
GtWaveFuncSetSweepState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveFuncGetSweepState, GtWaveFuncGetSweep, GtWaveFuncSetSweep, GtWaveFuncSetWaveform, GtWaveFuncSetSquareWaveDutyCycle, GtWaveFuncSetFrequency, GtWaveFuncSetOutToSquareState, GtWaveFuncSetPhase, **GtWaveGetErrorString**

GtWaveFuncSetWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Loads the specified standard waveform.

Syntax

GtWaveFuncSetWaveform(*nHandle, nChannel, nWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nWaveform	SHORT	Standard waveforms can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform 5. GTWAVE_WAVEFORM_RAMP_DC: Constant voltage 6. GTWAVE_WAVEFORM_NOISE: White noise
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example loads a sinusoid waveform:

```
GtWaveFuncSetWaveform (nHandle, GTWAVE_CHANNEL_A, GTWAVE_WAVEFORM_SINE, &nStatus);
```

See Also

GtWaveFuncGetWaveform, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveFuncWriteWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator mode only.

Purpose

Write an array of data to the board's Function Generator waveform memory.

Syntax

GtWaveFuncWriteWaveform (*nHandle, nChannel, dwStartAddress, dwSize, pnData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dwStartAddress	DWORD	GX1110: Waveform start address, any even value between 0 and 2097150. GX1120: Waveform start address, any even value between 0 and 33554430.
dwSize	DWORD	GX1110: Waveform start address, any even value between 0 and 2097151. GX1120: Waveform start address, any even value between 0 and 33554431.
pnData	PSHORT	Pointer to an array containing waveform data
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110:

When calling this function, the function will cast all the array values to 12-bit wide value.

Data point value of 2047 (0x7FF) represents the highest output amplitude and -2047 (0x800) represents the lowest output amplitude.

GX1120:

Data point value of 0x7FFF represents the highest output amplitude and -0x8000 represents the lowest output amplitude.

Example

The following example writes an array of 4096 data points to the board's waveform memory stating at address 0:

```
GtWaveArbWriteWaveformData (nHandle, GTWAVE_CHANNEL_A, 0, 4096, pnData, &nStatus);
```

See Also

GtWaveFuncGetWaveform, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveGetAmDepth

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the AM modulation depth in percentage.

Syntax

GtWaveGetAmDepth(nHandle, *nChannel, pdPercentDepth, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdPercentDepth	PDOUBLE	AM modulation depth in percentage 0% (no modulation) to 100% (full modulation).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the AM modulation depth in percentage:

```
DOUBLE dPercentDepth;
GtWaveGetAmDepth (nHandle, GTWAVE_CHANNEL_A, &dPercentDepth, &nStatus);
```

See Also

GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveSetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveGetAmFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the AM modulation frequency.

Syntax

GtWaveGetAmFrequency(*nHandle, nChannel, pdFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdFrequency	PDOUBLE	GX1110: AM modulation frequency range is from 0.01Hz to 20KHz GX1120: AM modulation frequency range is from 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the AM modulation frequency:

```
DOUBLE dFrequency;
GtWaveGetAmFrequency (nHandle, GTWAVE_CHANNEL_A, &dFrequency, &nStatus);
```

See Also

GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveSetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveGetAmplitude

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the waveform's programmed peak-to-peak amplitude voltage.

Syntax

GtWaveGetAmplitude(*nHandle, nChannel, pdAmplitude, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdAmplitude	PDOUBLE	Waveform peak-to-peak amplitude voltage GX1110: 0.1V to 8V (p-p) into 50 Ohm 0.2V to 16V (p-p) into open circuit. GX1120: 0.01V to 10V (p-p) into 50 Ohm 0.02V to 20V (p-p) into open circuit.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Offset may be applied to the output to shift the signal either positive or negative. Offset and amplitude are inter-related, i.e. the maximum positive or negative peak voltage at any time cannot be more then +/-4V (GX1110) and +/-5V (GX1120).

See the specification section for more details.

Example

The following example returns the waveform peak-to-peak amplitude voltage:

```
DOUBLE dAmplitude;
GtWaveGetAmplitude (nHandle, GTWAVE_CHANNEL_A, &dAmplitude, &nStatus);
```

See Also

GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveGetAmSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the AM modulation source.

Syntax

GtWaveGetAmSource(*nHandle, nChannel, pnSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSource	PSHORT	AM modulation source are: 0. GTWAVE_AM_SOURCE_INTERNAL: AM modulation source internal. 1. GTWAVE_AM_SOURCE_EXTERNAL: AM modulation source external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the AM modulation source:

```
SHORT nSource;
GtWaveGetAmSource (nHandle, GTWAVE_CHANNEL_A, &nSource, &nStatus);
```

See Also

GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveSetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveGetAmState

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the AM modulation state.

Syntax

GtWaveGetAmState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	AM modulation state: 0. Disabled. 1. Enabled
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the AM modulation state:

```
BOOL bState;
GtWaveGetAmState (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveSetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveGetAmWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the AM modulation waveform.

Syntax

GtWaveGetAmWaveform(*nHandle, nChannel, pnWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnWaveform	PSHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform 5. GTWAVE_WAVEFORM_RAMP_DC: Constant voltage 6. GTWAVE_WAVEFORM_NOISE: White noise
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example Returns the AM modulation waveform:

```
SHORT nWaveform;
GtWaveGetAmWaveform (nHandle, GTWAVE_CHANNEL_A, &nWaveform, &nStatus);
```

See Also

GtWaveSetAmWaveform, GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveGetAmState, GtWaveGetErrorString

GtWaveGetBoardSummary

Supported By

Gx1120, Gx1110

Purpose

Returns the board information.

Syntax

GtWaveGetBoardSummary (*nHandle, szSummary, nSumMaxLen, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
szSummary	PSTR	Buffer to contain the returned board info (null terminated) string.
nSumMaxLen	SHORT	Size of the buffer to contain the error string.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The board summary string provides the following data from the board in the order shown:

- Instrument Name (e.g., GX1110)
- Firmware (FPGA) Version (e.g., 0xB000)
- Serial Number (e.g. GX11100005-BB-BA-0000)
- Calibration date and time (e.g. Calibrated on: Sat Feb 23 23:36:14 2008)

For example, the returned string could look like the following:

"GtWave, FPGA-Version: 0xB000, Serial Number: GtWave0005-BB-BA-0000, Calibrated on: Sat Feb 23 23:36:14 2008"

Example

The following example returns the board summary:

```
SHORT nHandle, nStatus;
CHAR szSummary [256];

GtWaveGetBoardSummary(nHandle, szSummary, sizeof(szSummary), &nStatus);
```

See Also

GtWaveGetDriverSummary, **GtWaveInitialize**, **GtWaveGetErrorString**

GtWaveGetBoardType

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the board type.

Syntax

GtWaveGetBoardType(*nHandle, pnBoardType, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
pnBoardType	PSHORT	Board type are: 1. GTWAVE_BOARD_TYPE_1110: board type is GX1110. 2. GTWAVE_BOARD_TYPE_1120: board type is Gx1120.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the board type:

```
SHORT nBoardType;
GtWaveGetBoardType (nHandle, GTWAVE_CHANNEL_A, &nBoardType, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveGetDriverSummary

Supported By

Gx1120, Gx1110

Purpose

Returns the driver description string and version number.

Syntax

GtWaveGetDriverSummary (*szSummary*, *nSummaryMaxLen*, *pdwVersion*, *pnStatus*)

Parameters

Name	Type	Comments
pszSummary	LPSTR	Buffer to receive the summary string.
nSummaryMaxLen	SHORT	Buffer size passed by pszSummary.
pdwVersion	LPDWORD	Driver version
pnStatus	LPSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the driver summary.

```
SHORT nHandle, nStatus;
DWORD dwVersion;
CHAR szSummary[128];
GtWaveGetDriverSummary(szSummary, 128, &dwVersion, &nStatus);
After the function call the parameter SzSummary will be set to:
"GtWave Driver for GX1110, Version 1.00, Copyright(c) 2008 Marvin Test Solutions - MTS inc.".
```

See Also

GtWaveGetErrorString

GtWaveGetErrorString

Supported By

Gx1120, Gx1110

Purpose

Returns the error string associated with the specified error number.

Syntax

GtWaveGetErrorString (*nError*, *pszMsg*, *nErrorMaxLen*, *pnStatus*)

Parameters

Name	Type	Comments
nError	SHORT	Error number.
pszMsg	PSTR	Buffer to the returned error string.
nErrorMaxLen	SHORT	The size of the error string buffer.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function returns the error string associated with the *nError* as returned from other driver functions.

The following table displays the possible error values; not all errors apply to this board type:

Resource Errors

0	No error has occurred
-1	Unable to open the HW driver. Check if HW is properly installed
-2	Board does not exist in this slot/base address
-3	Different board exist in the specified PCI slot/base address
-4	PCI slot not configured properly. You may configure using the PciExplorer from the Windows Control Panel
-5	Unable to register the PCI device
-6	Unable to allocate system resource for the device
-7	Unable to allocate memory
-8	Unable to create panel
-9	Unable to create Windows timer
-10	Bad or Wrong board EEPROM
-11	Not in calibration mode
-12	Board is not calibrated
-13	Function is not supported by the specified board

General Parameter Errors

-20 Invalid or unknown error number

-21 Invalid parameter

-22 Illegal slot number

-23 Illegal board handle

-24 Illegal string length

-25 Illegal operation mode

Execution errors

-40 Failed to program the board Firmware

-41 Unable to set the specified clock frequency value

-42 Error occurred when trying to program the specified clock frequency

-43 EEPROM gain value is out of range

-44 EEPROM offset value is out of range

-45 Unable to initialize the COM library

-46 Unable to create COM object

-47 Error reading the specified file data

General Execution errors

-50 Offset value is out of range, allowed range is -5.0V to +5.0V

-51 Amplitude value is out of range, allowed output voltage range is 0V to +8V peak-to-peak (including offset voltage)

-52 Clock source is out of range, clock source can be either internal or external

-53 Attenuation value is out of range, see user's guide for allowed values

-54 Illegal timeout or not enough time to finish the operation

-55 Board operating mode is out of range, operation mode can either be Function Generator mode or Arbitrary Wave Generator

Trigger

-60 Burst counter value is out of range, allowed range if from 1 to 1048576

-61 Trigger internal rate value is out of range, see user's guide for allowed values

-62 Trigger mode value is out of range, see user's guide for allowed values

-63 Trigger source value is out of range, see user's guide for allowed values

-64 Trigger edge value is out of range, slope value can be positive or negative

-65 PXI trigger line value is out of range, see user's guide for allowed values

-66 Sync source value is out of range, see user's guide for allowed values

Amplitude Modulation

-70 AM depth value is out of range, see user's guide for allowed values

-71 AM frequency value is out of range, see user's guide for allowed values

-72 AM waveform value is out of range, see user's guide for allowed values

-73 AM source value is out of range, AM source can be either internal or external

Function Generator

-80 Function Generator frequency value is out of range, see user's guide for allowed values

-81 Function Generator FM deviation value is out of range, see user's guide for allowed values

-82 Function Generator FM gain value is out of range, see user's guide for allowed values

-83 Function Generator FM frequency value is out of range, see user's guide for allowed values

-84 Function Generator FM waveform value is out of range, see user's guide for allowed values

-85 Function Generator FM source value is out of range, FM source can be either internal or external

-86 Function Generator FSK frequency value is out of range, see user's guide for allowed values

-87 Function Generator FSK rate value is out of range, see user's guide for allowed values

-88 Function Generator FSK source value is out of range, FSK source can be either internal or external

-89 Function Generator waveform value is out of range, see user's guide for allowed values

-90 Function Generator phase value is out of range, phase value can be between 0.0 to 360.0 degrees

-91 Function Generator waveform start address value is out of range, see user's guide for allowed values

-92 Function Generator waveform length value is out of range, see user's guide for allowed values

-93 Function Generator sync mode value is out of range, see user's guide for allowed values

Arbitrary Wave Generator

-100 Arbitrary Waveform Generator sample rate value is out of range, see user's guide for allowed values

-101 Arbitrary Waveform Generator waveform value is out of range, see user's guide for allowed values

-102 Arbitrary Waveform Generator waveform address range value is out of range, see user's guide for allowed values

-103 Arbitrary Waveform Generator wave min num steps value is out of range, see user's guide for allowed values

-103 Arbitrary Waveform Generator memory test failed

Example

The following example initializes the board. If the initialization failed, the following error string is printed:

```
CHAR    sz[256];
SHORT   nStatus, nHandle;
..
GtWaveInitialize (3, &Handle, &nStatus);
if (nStatus<0)
{   GtWaveGetErrorString(nStatus, sz, sizeof sz, &nStatus);
    printf(sz); // prints the error string returns
}
```

GtWaveGetFilterMode

Supported By

Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the filter mode.

Syntax

GtWaveGetFilterMode (*nHandle, nChannel, pnMode, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1100 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A.
pnMode	PSHORT	Filter mode: 0. GTWAVE_FILTER_MODE_AUTO – the driver applies the best filter for the active waveform automatically. 1. GTWAVE_FILTER_MODE_BAND_PASS – output filter is set to the band pass mode. This filter is most suitable when the board is in function generator mode and the waveform is not a square wave. 2. GTWAVE_FILTER_MODE_LOW_PASS - output filter is set to the low band pass filter mode. This filter is most suitable when the board is in Arbitrary Waveform Generator mode or when the board is in Function Generator mode and the waveform is a square wave.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The output filter setting is different for Sinusoidal waveform and a Square waveform versus the Arbitrary mode of operation. The user can determine the filter settings or let the driver to apply the best filter for the active waveform.

Example

The following example returns the filter mode:

```
SHORT nMode;
GtWaveGetFilterMode (nHandle, GTWAVE_CHANNEL_A, &nMode, &nStatus);
```

See Also

GtWaveSetFilterMode, **GtWaveGetErrorString**

GtWaveGetMarkerToPxiTriggerBusLine

Supported By

Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the Marker to PXI trigger bus line assignment and the state.

Syntax

GtWaveGetMarkerToPxiTriggerBusLine(*nHandle, nChannel, pnLine, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A.
pnLine	PSHORT	A PXI trigger bus line can be assigned to the marker. Only a single line may be connected at any given time. PXI trigger bus line values are: 0. GTWAVE_PXI_TRIGGER_BUS_LINE0 1. GTWAVE_PXI_TRIGGER_BUS_LINE1 2. GTWAVE_PXI_TRIGGER_BUS_LINE2 3. GTWAVE_PXI_TRIGGER_BUS_LINE3 4. GTWAVE_PXI_TRIGGER_BUS_LINE4 5. GTWAVE_PXI_TRIGGER_BUS_LINE5 6. GTWAVE_PXI_TRIGGER_BUS_LINE6 7. GTWAVE_PXI_TRIGGER_BUS_LINE7
pbState	PBOOL	Marker to PXI trigger bus line state: 0. Disconnected. 1. Connected.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the marker to PXI trigger bus line assignment and the state:

```
SHORT nLine
BOOL bState;
GtWaveGetMarkerToPxiTriggerBusLine (nHandle, GTWAVE_CHANNEL_A, &nLine, &bState, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, **GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveGetErrorString**

GtWaveGetOffset

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the programmed output offset voltage.

Syntax

GtWaveGetOffset(*nHandle, nChannel, pdOffset, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdOffset	PDOUBLE	GX1110: Output offset voltage ranges from -4.0V to +4.0V. The output level may be programmed from 0.1V to 8V p-p into 50 Ohm (0.2V to 16V p-p into an open circuit). GX1120: Output offset voltage ranges from -5.0V to +5.0V. The output level may be programmed from 0.01V to 10V p-p into 50 Ohm (0.2V to 20V p-p into an open circuit).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Offset may be applied to the output to shift the signal either positive or negative. Offset and amplitude are inter-related, i.e. the maximum positive or negative peak voltage at any time cannot be more then +/- 4V (GX1110) and +/- 5V (GX1120).

See the specification section for more details.

Example

The following example Returns the output offset voltage:

```
DOUBLE dOffset;
GtWaveGetOffset (nHandle, GTWAVE_CHANNEL_A, &dOffset, &nStatus);
```

See Also

GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveGetOperationMode

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the board's operation mode.

Syntax

GtWaveGetOperationMode(*nHandle, nChannel, pnMode, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnMode	PSHORT	Each channel can be programmed to operate as Function Generator or as an Arbitrary Waveform Generator, modes are as follows: 0. GTWAVE_OPERATING_MODE_FUNC: Function Generator mode 1. GTWAVE_OPERATING_MODE_ARB: Arbitrary Waveform Generator mode 2. GTWAVE_OPERATING_MODE_ARB_HIGH_SPEED: High speed Arbitrary Waveform Generator mode. In this mode both channels A and B are combined into one channel (Channel A) to produce a high speed Arbitrary Waveform Generator that can run as high as 400 MS/s.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Each channel can be programmed at any point in time to operate as Function Generator or as an Arbitrary Waveform Generator by calling the **GtWaveSetOperationMode** function. After calling **GtWaveSetOperationMode** function the board will be set to its default settings for the specifed operation mode.

Example

The following example returns the board's operation mode:

```
SHORT nMode;
GtWaveGetOperationMode (nHandle, GTWAVE_CHANNEL_A, &nMode, &nStatus);
```

See Also

GtWaveSetOperationMode, GtWaveReset, GtWaveGetErrorString

GtWaveGetOutputState

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the specified channel's output state.

Syntax

GtWaveGetOutputState(*nHandle, nChannel, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pbState	PBOOL	Output state: 0. Output is disabled. 1. Output is enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the specified channel's output state:

```
BOOL bState;
GtWaveGetOutputState (nHandle, GTWAVE_CHANNEL_A, &bState, &nStatus);
```

See Also

GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveGetPxiTriggerBusLine

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the PXI trigger bus line assignment and state.

Syntax

GtWaveGetPxiTriggerBusLine(*nHandle, nChannel, pnLine, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnLine	PSHORT	A PXI trigger bus line is assigned as a trigger. Only a single line may be connected at any given time. PXI trigger bus line value are: 0. GTWAVE_PXI_TRIGGER_BUS_LINE0 1. GTWAVE_PXI_TRIGGER_BUS_LINE1 2. GTWAVE_PXI_TRIGGER_BUS_LINE2 3. GTWAVE_PXI_TRIGGER_BUS_LINE3 4. GTWAVE_PXI_TRIGGER_BUS_LINE4 5. GTWAVE_PXI_TRIGGER_BUS_LINE5 6. GTWAVE_PXI_TRIGGER_BUS_LINE6 7. GTWAVE_PXI_TRIGGER_BUS_LINE7
pbState	PBOOL	PXI trigger bus line state: 0. Disconnected. 1. Connected.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the PXI trigger bus line assignment and state:

```
SHORT nLine
BOOL bState;
GtWaveGetPxiTriggerBusLine (nHandle, GTWAVE_CHANNEL_A, &nLine, &bState, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, **GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveGetErrorString**

GtWaveGetReferenceClockSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the reference clock source.

Syntax

GtWaveGetReferenceClockSource(*nHandle, nChannel, pnSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSource	PSHORT	Reference clock source are: 0. GTWAVE_REF_CLOCK_SOURCE_INTERNAL: Reference clock source is the internal 10MHz clock 1. GTWAVE_REF_CLOCK_SOURCE_EXTERNAL: Reference clock source is the external clock 2. GTWAVE_REF_CLOCK_SOURCE_PXI10MHZ Reference clock source is the PXI backplane 10MHz clock
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the reference clock source:

```
SHORT nSource;
GtWaveGetReferenceClockSource (nHandle, GTWAVE_CHANNEL_A, &nSource, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveGetErrorString

GtWaveGetStatusRegister

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the status register.

Syntax

GtWaveGetStatusRegister(*nHandle, nChannel, pdwStatus, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdwStatus	PDWORD	Returned status register, see comments for bits description..
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

GX1110: Status register bits description:

Bit #	Description
0	Operation mode: 0= Function Generator, 1= Arbitrary Waveform Generator
1	Run mode: 0=Run disabled, 1= Run enabled
2	Over voltage: 0=No Over voltage, 1= Over voltage
3-31	Not used

GX1120: Status register bits description:

Bits #	Description
0-1	Operation mode: 0= Function Generator, 1= Arbitrary Waveform Generator, 2= High speed Arbitrary Waveform Generator.
2	Run mode: 0=Run disabled, 1= Run enabled
3	Over voltage: 0=No Over voltage, 1= Over voltage
4-31	Not used

Example

The following example rturns the status register:

```
DWORD dwStatus;
GtWaveGetStatusRegister (nHandle, GTWAVE_CHANNEL_A, &dwStatus, &nStatus);
```

See Also

GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveGetErrorString

GtWaveGetTriggerBurstCount

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the trigger burst count.

Syntax

GtWaveGetTriggerBurstCount(*nHandle, nChannel, pdwCount, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdwCount	PDWORD	GX1110: Trigger burst count can range from 1 (GTWAVE_GX1110_MIN_BURST_COUNT) to 9999999 (GTWAVE_GX1110_MAX_BURST_COUNT). GX1120: Trigger burst count can range from 1 (GTWAVE_GX1120_MIN_BURST_COUNT) to 9999999 (GTWAVE_GX1120_MAX_BURST_COUNT).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

When the trigger mode (by calling **GtWaveGetTriggerMode** function) is set to burst mode, which is an extension of the triggered mode, the generator can be programmed to output a pre-determined number of waveforms. The sources to trigger a burst are the same as for the trigger mode.

Example

The following example returns the trigger burst count:

```
DWORD dwCount;
GtWaveGetTriggerBurstCount (nHandle, GTWAVE_CHANNEL_A, &dwCount, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveGetTriggerDelay

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the external trigger delay.

Syntax

GtWaveGetTriggerDelay (*nHandle, nChannel, pdDelay, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdDelay	PDOUBLE	External trigger delay range is 0 - 15 sec, with 4 ns resolution. Units are in uSec, e.g. if pdDelay is equal 10 then the delay is 10uSec.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the external trigger edge:

```
DOUBLE dDelay;
GtWaveGetTriggerDelay (nHandle, GTWAVE_CHANNEL_A, &dDelay, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveGetTriggerEdge

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the external trigger edge.

Syntax

GtWaveGetTriggerEdge(*nHandle, nChannel, pnSlope, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSlope	PSHORT	External trigger edge slope can be: 0. GTWAVE_EXTERNAL_TRIGGER_RISING_EDGE 1. GTWAVE_EXTERNAL_TRIGGER_FALLING_EDGE
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the external trigger edge:

```
SHORT nSlope;
GtWaveGetTriggerEdge (nHandle, GTWAVE_CHANNEL_A, &nSlope, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveGetTriggerHoldoff

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the external trigger hold off delay.

Syntax

GtWaveGetTriggerHoldoff (*nHandle, nChannel, pdHoldoff, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdHoldoff	PDOUBLE	External trigger holdoff range is 0 - 15 sec, with 4 ns resolution. Units are in uSec, e.g. if pdDelay is equal 10 then the hold off is 10uSec.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the external trigger hold off:

```
DOUBLE dHoldoff;
GtWaveGetTriggerHoldoff (nHandle, GTWAVE_CHANNEL_A, &dHoldoff, &nStatus);
```

See Also

GtWaveSetTriggerHoldoff, **GtWaveSetTriggerDelay, GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString**

GtWaveGetTriggerInternalFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the trigger internal frequency.

Syntax

GtWaveGetTriggerInternalFrequency(*nHandle, nChannel, pdFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdFrequency	PDOUBLE	GX1110: Trigger internal frequency range can be form GTWAVE_GX1110_MIN_TRIGGER_INTERNAL_FREQUENCY (0.01Hz) to GTWAVE_GX1110_MAX_TRIGGER_INTERNAL_FREQUENCY (50MHz). GX1120: Trigger internal frequency range can be form GTWAVE_GX1120_MIN_TRIGGER_INTERNAL_FREQUENCY (0.01Hz) GTWAVE_GX1120_MAX_TRIGGER_INTERNAL_FREQUENCY (1 MHz)
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

A built-in programmable trigger generator can be used as a replacement of an external trigger source. The number of waveform cycles generated for each internal trigger is determined by the **GtWaveGetTriggerBurstCount** function.

Example

The following example returns the trigger internal frequency:

```
DOUBLE dFrequency;
GtWaveGetTriggerInternalFrequency (nHandle, GTWAVE_CHANNEL_A, &dFrequency, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveGetTriggerLevel

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the external trigger level.

Syntax

GtWaveGetTriggerLevel (*nHandle, nChannel, pdLevel, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdLevel	PDOUBLE	External trigger level range is -5V to +5V with 10mV of resolution.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the external trigger hold off:

```
DOUBLE dLevel;
GtWaveGetTriggerLevel (nHandle, GTWAVE_CHANNEL_A, & dLevel, &nStatus);
```

See Also

GtWaveSetTriggerLevel, **GtWaveSetTriggerHoldoff**, **GtWaveSetTriggerDelay, GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString**

GtWaveGetTriggerMode

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the trigger mode.

Syntax

GtWaveGetTriggerMode(*nHandle, nChannel, pnMode, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnMode	PSHORT	Trigger mode are: 0. GTWAVE_TRIGGER_MODE_CONTINUOUS 1. GTWAVE_TRIGGER_MODE_TRIGGERED 2. GTWAVE_TRIGGER_MODE_GATED 3. GTWAVE_TRIGGER_MODE_BURST
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Function Generator:

GTWAVE_TRIGGER_MODE_CONTINUOUS	The waveform is generated continuously by repeatedly cycling through the waveform table with the programmed waveform parameters.
GTWAVE_TRIGGER_MODE_TRIGGERED	Output is quiescent until triggered by an internal or external trigger, then, one waveform cycle is generated with the programmed waveform parameters.
GTWAVE_TRIGGER_MODE_GATED	The waveform generates continuously by repeatedly cycling through the waveform table as long as the gate is active.
GTWAVE_TRIGGER_MODE_BURST	After a trigger is received, waveform generation will be executed for the number of cycles that were defined by the **GtWaveGetTriggerBurstCount** function.

Arbitrary Waveform Generator mode:

GTWAVE_TRIGGER_MODE_CONTINUOUS	The waveform is generated continuously by repeatedly cycling through the defined waveform length at programmed waveform parameters. The waveform generation starts from the specified first address and continues through to the specified waveform length. After the last step is completed, the waveform generation loops back to the specified first address and continues until it is stopped.
GTWAVE_TRIGGER_MODE_TRIGGERED	Output quiescent until triggered by an internal or external trigger, then one waveform cycle is generated to programmed waveform parameters.
GTWAVE_TRIGGER_MODE_GATED	The waveform generates continuously by repeatedly cycling through the waveform table as long as the gate is active.
GTWAVE_TRIGGER_MODE_BURST	After a trigger is received, waveform generation starts from the specified first address and continues through to the specified waveform length. The number of cycles that was defined in the **GtWaveGetTriggerBurstCount** function.

Example

The following example returns the trigger mode:

```
SHORT nMode;
GtWaveGetTriggerMode (nHandle, GTWAVE_CHANNEL_A, &nMode, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveGetTriggerSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the trigger source.

Syntax

GtWaveGetTriggerSource(*nHandle, nChannel, pnSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnSource	PSHORT	Trigger sources are as follows: 0. GTWAVE_TRIGGER_SOURCE_SOFTWARE: Trigger source is software (immediate). 1. GTWAVE_TRIGGER_SOURCE_INTERNAL: Trigger source is the internal programmable trigger generator. 2. GTWAVE_TRIGGER_SOURCE_EXTERNAL: Trigger source is external input. 3. GTWAVE_TRIGGER_SOURCE_PXI_STAR_TRIG: Trigger source is the PXI star trigger.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The trigger source can be set to one of the four sources and also to one of the PXI trigger lines. See the **GtWaveSetPxiTriggerBusLine** function for details.

Example

The following example returns the trigger source:

```
SHORT nSource;
GtWaveGetTriggerSource (nHandle, GTWAVE_CHANNEL_A, &nSource, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveGetTriggerToPxiTriggerBusLine

Supported By

Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the trigger to PXI trigger bus line assignment and state.

Syntax

GtWaveGetTriggerToPxiTriggerBusLine(*nHandle, nChannel, pnLine, pbState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A.
pnLine	PSHORT	Assigned trigger to PXI trigger bus line value can be one of the following: 0. GTWAVE_PXI_TRIGGER_BUS_LINE0 1. GTWAVE_PXI_TRIGGER_BUS_LINE1 2. GTWAVE_PXI_TRIGGER_BUS_LINE2 3. GTWAVE_PXI_TRIGGER_BUS_LINE3 4. GTWAVE_PXI_TRIGGER_BUS_LINE4 5. GTWAVE_PXI_TRIGGER_BUS_LINE5 6. GTWAVE_PXI_TRIGGER_BUS_LINE6 7. GTWAVE_PXI_TRIGGER_BUS_LINE7
pbState	PBOOL	Trigger to PXI trigger bus line state: 0. Disconnected. 1. Connected.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Note: Only single line can be connected at any given time.

Example

The following example returns the trigger to PXI trigger bus line assignment and state:

```
SHORT nLine
BOOL bState;
GtWaveGetTriggerToPxiTriggerBusLine (nHandle, GTWAVE_CHANNEL_A, &nLine, &bState, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, **GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveGetErrorString**

GtWaveGetVoltageRangeMode

Supported By

Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Returns the amplitude and offset voltage settings, mode and voltage range.

Syntax

GtWaveGetVoltageRangeMode (*nHandle, nChannel, pnMode,* p*dVoltageRange, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1100 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A.
pnMode	PSHORT	Amplitude and Offset voltages setting modes: 0. GTWAVE_VOLTAGE_RANGE_AUTO – each time the Amplitude or Offset voltages are set the driver runs an algorithm to produce the best possible resolution for the specified Amplitude and offset voltage settings by changing the output attenuators. The dVoltageRange variable is ignored in this mode. 1. GTWAVE_VOLTAGE_RANGE_BY_VALUE - the driver sets the best output attenuators for the specified (*dVoltageRange*) peak-to-peak voltage range. In this mode setting the Amplitude and/or Offset voltages will not change the output attenuators settings.
pdVoltageRange	PDOUBLE	Peak-to-peak voltage range. Value is only used when *nMode* is set to GTWAVE_VOLTAGE_RANGE_BY_VALUE.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

After calling this function the Offset is set to 0 and the amplitude to p*dVoltageRange*.

Note: calling reset command will reset the mode back to the Auto.

Example

The following example returns the Amplitude and Offset voltages setting mode and voltage range:

```
SHORT nMode;
DOUBLE dVoltageRange;
GtWaveGetVoltageRangeMode (nHandle, GTWAVE_CHANNEL_A, &nMode, &dVoltageRange, &nStatus);
```

See Also

GtWaveSetVoltageRangeMode, **GtWaveGetErrorString**

GtWaveInitialize

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Initializes the driver for the specified PXI slot using the HW device driver.

Syntax

GtWaveInitialize (*nSlot, pnHandle, pnStatus*)

Parameters

Name	Type	Comments
nSlot	SHORT	GX11X0 board slot number.
pnHandle	PSHORT	Returned Handle for a GX11X0 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The Marvin Test Solutions HW device driver is installed with the driver and is the default device driver. The function returns a handle that is used with other Counter functions to program the board. The function does not change any of the board settings.

The specified PXI slot number is displayed by the **PXI/PCI Explorer** applet that can be opened from the Windows **Control Panel**. You may also use the label on the chassis below the PXI slot where the board is installed. The function accepts two types of slot numbers:

- A combination of chassis number (chassis # x 256) with the chassis slot number. For example 0x105 (chassis 1 slot 5).
- A Legacy nSlot is used by earlier versions of HW/VISA. The slot number contains no chassis number and can be changed using the **PXI/PCI Explorer** applet (1-255).

Example

The following example initializes two GX1110 boards at slot 1 and 2.

```
SHORT nHandle1, nHandle2, nStatus;
Gx6616Initialize (1, &nHandle1, &nStatus);
Gx6616Initialize (2, &nHandle2, &nStatus);
if (nHandle1==0 || nHandle2==0)
{   printf("Unable to Initialize the boards")
        return;
}
```

See Also

GtWaveInitializeVisa, GtWaveGetErrorString, GtWaveReset

GtWaveInitializeVisa

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Initializes the driver for the specified PXI slot using the default VISA provider.

Syntax

GtWaveInitializeVisa (*szVisaResource, pnHandle, pnStatus*)

Parameters

Name	Type	Comments
szVisaResource	LPCTSTR	String identifying the location of the specified board in order to establish a session.
pnHandle	PSHORT	Returned Handle (session identifier) that can be used to call any other operations of that resource
pnStatus	PSHORT	Returned status: 0 on success, 1 on failure.

Comments

The **GtWaveInitializeVisa** opens a VISA session to the specified resource. The function uses the default VISA provider configured in your system to access the board. You must ensure that the default VISA provider support PXI/PCI devices and that the board is visible in the VISA resource manager before calling this function.

The first argument *szVisaResource* is a string that is displayed by the VISA resource manager such as NI Measurement and Automation (NI_MAX). It is also displayed by Marvin Test Solutions PXI/PCI Explorer as shown in the prior figure. The VISA resource string can be specified in several ways as follows:

- Using chassis, slot: “PXI0::CHASSIS1::SLOT5”
- Using the PCI Bus/Device combination: “PXI9::13::INSTR” (bus 9, device 9).
- Using alias: “ARB1”. Use the PXI/PCI Explorer to set the device alias.

The function returns a board handle (session identifier) that can be used to call any other operations of that resource. The session is opened with VI_TMO_IMMEDIATE and VI_NO_LOCK VISA attributes. On terminating the application the driver automatically invokes **viClose**() terminating the session.

Example

The following example initializes a GX1110 boards at PXI bus 5 and device 11.

```
SHORT nHandle, nStatus;
GtWaveInitializeVisa(“PXI5::11::INSTR”, &nHandle, &nStatus);
if (nHandle==0)
{
    printf("Unable to Initialize the board")
    return;
}
```

See Also

GtWaveInitialize, GtWaveGetErrorString, GtWaveReset

GtWavePanel

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Opens a virtual panel used to interactively control the board.

Syntax

GtWavePanel *(pnHandle, hwndParent, nMode, phwndPanel, pnStatus)*

Parameters

Name	Type	Comments
pnHandle	PSHORT	Handle to a GX11X0 board.
hwndParent	HWND	Panel parent window handle. A value of 0 sets the desktop as the parent window.
nMode	SHORT	The mode in which the panel main window is created. 0 for modeless window and 1 for modal window.
phwndPanel	HWND	Returned window handle for the panel.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function is used to create the panel window. The panel window may be opened as a modal or a modeless window depending on the *nMode* parameters.

If the mode is set to modal dialog (*nMode*=1), the panel will disable the parent window (*hwndParent*) and the function will return only after the window closes. In that case, the *pnHandle* may return the handle created by the user using the panel Initialize dialog. This handle may be used when calling other GX110 functions.

If a modeless dialog is created (*nMode*=0), the function returns immediately after creating the panel window returning the handle to the panel - *phwndPanel*. It is the responsibility of calling program to dispatch windows messages to this window so that the window can respond to messages.

Example

The following example opens the panel in modal mode:

```
DWORD dwPanel;
SHORT nHandle=0, nStatus;

GtWavePanel(&nHandle, 0, 1, &dwPanel, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveReset

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the board to its default settings

Syntax

GtWaveReset (*nHandle, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Calling this function does not change the operation mode.

The function calls **GtWaveSynchronizePhases** API before returning.

Example

The following example resets the board:

```
GtWaveReset (nHandle, GTWAVE_CHANNEL_A, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveResetChannel

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the board to its default settings

Syntax

GtWaveResetChannel (*nHandle, nChannel, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Calling this function does not change the operation mode, but resets the specified channel to its default state.

The function calls **GtWaveSynchronizePhases** API before returning.

Example

The following example resets the board:

```
GtWaveResetChannel (nHandle, GTWAVE_CHANNEL_A, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveReset**, **GtWaveGetErrorString**

GtWaveRun

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Enables the board for running.

Syntax

GtWaveRun(*nHandle*, *nChannel*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example enables the board for running:

```
GtWaveRun (nHandle, GTWAVE_CHANNEL_A, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveSetAmDepth

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the AM modulation depth in percentage.

Syntax

GtWaveSetAmDepth(nHandle, *nChannel*, *dPercentDepth*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dPercentDepth	DOUBLE	AM modulation depth in percentage 0% (no modulation) to 100% (full modulation).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the AM modulation depth to 50%:

```
GtWaveSetAmDepth (nHandle, GTWAVE_CHANNEL_A, 50, &nStatus);
```

See Also

GtWaveGetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveSetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveSetAmFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the AM modulation frequency.

Syntax

GtWaveGetAmFrequency(*nHandle, nChannel, dFrequency, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dFrequency	DOUBLE	GX1110: AM modulation frequency range is from 0.01Hz to 20KHz GX1120: AM modulation frequency range is from 0.01Hz to 20KHz
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the AM modulation frequency to 100Hz:

```
GtWaveSetAmFrequency (nHandle, GTWAVE_CHANNEL_A, 100, &nStatus);
```

See Also

GtWaveSetAmDepth, GtWaveGetAmFrequency, GtWaveSetAmSource, GtWaveSetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveSetAmplitude

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the waveform peak-to-peak amplitude voltage.

Syntax

GtWaveSetAmplitude(*nHandle, nChannel, pdAmplitude, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dAmplitude	DOUBLE	Waveform peak-to-peak amplitude voltage GX1110: 0.1V to 8V (p-p) into 50 Ohm 0.2V to 16V (p-p) into open circuit. GX1120: 0.01V to 10V (p-p) into 50 Ohm 0.02V to 20V (p-p) into open circuit.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Offset may be applied to the output to shift the signal either positive or negative. Offset and amplitude are inter-related, i.e. the maximum positive or negative peak voltage at any time cannot be more then +/-4V (GX1110) and +/-5V (GX1120).

See the specification section for more details.

Example

The following example sets the peak-to-peak amplitude voltage to 4V:

```
GtWaveSetAmplitude (nHandle, GTWAVE_CHANNEL_A, 4.0, &nStatus);
```

See Also

GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveGetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveSetAmSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the AM modulation source.

Syntax

GtWaveSetAmSource(*nHandle, nChannel, nSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSource	SHORT	AM modulation source are: 0. GTWAVE_AM_SOURCE_INTERNAL: AM modulation source internal. 1. GTWAVE_AM_SOURCE_EXTERNAL: AM modulation source external.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the AM modulation source to internal:

```
GtWaveSetAmSource (nHandle, GTWAVE_CHANNEL_A, GTWAVE_AM_SOURCE_INTERNAL, &nStatus);
```

See Also

GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveGetAmSource, GtWaveSetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveSetAmState

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the AM modulation state.

Syntax

GtWaveSetAmState(*nHandle, nChannel, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	AM modulation state: 0. Disabled. 1. Enabled
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example enables the AM modulation:

```
BOOL bState;
GtWaveSetAmState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveGetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveSetAmWaveform

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the AM modulation waveform.

Syntax

GtWaveSetAmWaveform(*nHandle, nChannel, nWaveform, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nWaveform	SHORT	Standard waveform can be one of the following: 0. GTWAVE_WAVEFORM_SINE: Sinusoidal waveform. 1. GTWAVE_WAVEFORM_SQUARE: Square waveform. 2. GTWAVE_WAVEFORM_TRIANGLE: Triangular waveform. 3. GTWAVE_WAVEFORM_RAMP_UP: Positive ramp waveform 4. GTWAVE_WAVEFORM_RAMP_DOWN: Negative ramp waveform 5. GTWAVE_WAVEFORM_RAMP_DC: Constant voltage 6. GTWAVE_WAVEFORM_NOISE: White noise
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the AM modulation waveform to Sine:

```
GtWaveSetAmWaveform (nHandle, GTWAVE_CHANNEL_A, GTWAVE_WAVEFORM_SINE, &nStatus);
```

See Also

GtWaveGetAmWaveform, GtWaveSetAmDepth, GtWaveSetAmFrequency, GtWaveSetAmSource, GtWaveGetAmState, GtWaveSetAmWaveform, GtWaveGetErrorString

GtWaveSetFilterMode

Supported By

Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the filter mode.

Syntax

GtWaveSetFilterMode (*nHandle*, *nChannel*, *nMode*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1110 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A.
nMode	SHORT	Filter mode: 0. GTWAVE_FILTER_MODE_AUTO – the driver applies the best filter for the active waveform automatically. 1. GTWAVE_FILTER_MODE_BAND_PASS – output filter is set to the elliptical band pass mode. This filter is most suitable when the board is in function generator mode and the waveform is not a square waveform. 2. GTWAVE_FILTER_MODE_LOW_PASS - output filter is set to the low pass filter mode. This filter is most suitable when the board is in Arbitrary Waveform Generator mode or when the board is in Function Generator mode and the waveform is a square waveform.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The output filter settings are different for Sinusoidal waveform and a Square waveform versus the Arbitrary mode. The user can determine the filter settings or let the driver to apply the best filter for the active waveform.

Example

The following example sets the filter mode to auto:

```
GtWaveSetFilterMode (nHandle, GTWAVE_CHANNEL_A, GTWAVE_FILTER_MODE_AUTO, &nStatus);
```

See Also

GtWaveGetFilterMode, **GtWaveGetErrorString**

GtWaveSetMarkerToPxiTriggerBusLine

Supported By

Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the Marker to PXI trigger bus line assignment and the state.

Syntax

GtWaveSetMarkerToPxiTriggerBusLine(*nHandle, nChannel, nLine, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A.
nLine	SHORT	A PXI trigger bus line assigned to the marker. Only single line may be connected at any given time. PXI trigger bus line value are: 0. GTWAVE_PXI_TRIGGER_BUS_LINE0 1. GTWAVE_PXI_TRIGGER_BUS_LINE1 2. GTWAVE_PXI_TRIGGER_BUS_LINE2 3. GTWAVE_PXI_TRIGGER_BUS_LINE3 4. GTWAVE_PXI_TRIGGER_BUS_LINE4 5. GTWAVE_PXI_TRIGGER_BUS_LINE5 6. GTWAVE_PXI_TRIGGER_BUS_LINE6 7. GTWAVE_PXI_TRIGGER_BUS_LINE7
bState	BOOL	Marker to PXI trigger bus line state: 0. Disconnected. 1. Connected.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example assigns the marker to PXI trigger bus line 1 and the state to enable:

```
GtWaveSetMarkerToPxiTriggerBusLine (nHandle, GTWAVE_CHANNEL_A, GTWAVE_PXI_TRIGGER_BUS_LINE1,
TRUE, &nStatus);
```

See Also

GtWaveGetMarkerToPxiTriggerBusLine, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveGetErrorString

GtWaveSetOffset

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the output DC offset voltage.

Syntax

GtWaveSetOffset(*nHandle, nChannel, dOffset, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dOffset	DOUBLE	GX1110: Output offset voltage ranges from -4.0V to +4.0V. The output level may be programmed from 0.1V to 8V p-p into 50 Ohm (0.2V to 16V p-p into an open circuit). GX1120: Output offset voltage ranges from -5.0V to +5.0V. The output level may be programmed from 0.01V to 10V p-p into 50 Ohm (0.2V to 20V p-p into an open circuit).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Offset may be applied to the output to shift the signal either positive or negative. Offset and amplitude are inter-related, i.e. the maximum positive or negative peak voltage at any time cannot be more then +/- 4V (GX1110) and +/- 5V (GX1120).

See the specification section for more details.

Example

The following example sets the output offset voltage to +2.0V:

```
DOUBLE dOffset;
GtWaveSetOffset (nHandle, GTWAVE_CHANNEL_A, 2.0, &nStatus);
```

See Also

GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveGetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveSetOperationMode

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the board's operation mode.

Syntax

GtWaveSetOperationMode(*nHandle, nMode, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nMode	SHORT	Each channel can be programmed to operate as Function Generator or as an Arbitrary Waveform Generator, modes are as follows: 0. GTWAVE_OPERATING_MODE_FUNC: Function Generator mode 1. GTWAVE_OPERATING_MODE_ARB: Arbitrary Waveform Generator mode 2. GTWAVE_OPERATING_MODE_ARB_HIGH_SPEED: High speed Arbitrary Waveform Generator mode. In this mode both channels A and B are combined into one channel (Channel A) to produce a high speed Arbitrary Waveform Generator that can run as high as 400 MS/s.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Each channel can be programmed at any point in time to operate as Function Generator or as an Arbitrary Waveform Generator by calling the **GtWaveSetOperationMode** function. After calling **GtWaveSetOperationMode** function the baord will be set to its default settings for the specifed operation mode.

Example

The following example sets the board's operation mode to GTWAVE_OPERATING_MODE_ARB:

```
GtWaveSetOperationMode (nHandle, GTWAVE_CHANNEL_A, GTWAVE_OPERATING_MODE_ARB, &nStatus);
```

See Also

GtWaveSetOperationMode, GtWaveReset, GtWaveGetErrorString

GtWaveSetOutputState

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the specified channel's output state.

Syntax

GtWaveSetOutputState(*nHandle, nChannel, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
bState	BOOL	Output state: 0. Output is disabled. 1. Output is enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example enables the specified channel's output state:

```
GtWaveSetOutputState (nHandle, GTWAVE_CHANNEL_A, TRUE, &nStatus);
```

See Also

GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetFilterMode, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveGetOutputState, GtWaveSetReferenceClockSource, GtWaveSetVoltageRangeMode, GtWaveGetErrorString

GtWaveSetPxiTriggerBusLine

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Assigns the PXI trigger bus line and state.

Syntax

GtWaveSetPxiTriggerBusLine(*nHandle, nChannel, nLine, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nLine	SHORT	A PXI trigger bus line is assigned as a trigger. Only a single line may be connected at any given time. PXI trigger bus line value are: 0. GTWAVE_PXI_TRIGGER_BUS_LINE0 1. GTWAVE_PXI_TRIGGER_BUS_LINE1 2. GTWAVE_PXI_TRIGGER_BUS_LINE2 3. GTWAVE_PXI_TRIGGER_BUS_LINE3 4. GTWAVE_PXI_TRIGGER_BUS_LINE4 5. GTWAVE_PXI_TRIGGER_BUS_LINE5 6. GTWAVE_PXI_TRIGGER_BUS_LINE6 7. GTWAVE_PXI_TRIGGER_BUS_LINE7
pbState	PBOOL	PXI trigger bus line state: 0. Disconnected. 1. Connected.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example connects PXI trigger bus line 1 and enables it

```
GtWaveSetPxiTriggerBusLine (nHandle, GTWAVE_CHANNEL_A, GTWAVE_PXI_TRIGGER_BUS_LINE1, TRUE,
&nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveGetPxiTriggerBusLine, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveGetErrorString

GtWaveSetReferenceClockSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the reference clock source.

Syntax

GtWaveSetReferenceClockSource(*nHandle, nChannel, nSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSource	SHORT	Reference clock source are: 0. GTWAVE_REF_CLOCK_SOURCE_INTERNAL: Reference clock source is the internal 10MHz clock 1. GTWAVE_REF_CLOCK_SOURCE_EXTERNAL: Reference clock source is the external clock 2. GTWAVE_REF_CLOCK_SOURCE_PXI10MHZ Reference clock source is the PXI backplane 10MHz clock
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the reference clock source to internal:

```
GtWaveSetReferenceClockSource (nHandle, GTWAVE_CHANNEL_A, GTWAVE_REF_CLOCK_SOURCE_INTERNAL,
&nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveGetReferenceClockSource, GtWaveGetErrorString

GtWaveSetTriggerBurstCount

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the trigger burst count.

Syntax

GtWaveSetTriggerBurstCount(*nHandle, nChannel, dwCount, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dwCount	DWORD	GX1110: Trigger burst count can range from 1 (GTWAVE_GX1110_MIN_BURST_COUNT) to 9999999 (GTWAVE_GX1110_MAX_BURST_COUNT). GX1120: Trigger burst count can range from 1 (GTWAVE_GX1120_MIN_BURST_COUNT) to 9999999 (GTWAVE_GX1120_MAX_BURST_COUNT).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

When the trigger mode (by calling **GtWaveGetTriggerMode** function) is set to burst mode, which is an extension of the triggered mode, the generator can be programmed to output a pre-determined number of waveforms. The sources to trigger a burst are the same as for the trigger mode.

Example

The following example sets the trigger burst count:

```
GtWaveSetTriggerBurstCount (nHandle, GTWAVE_CHANNEL_A, 1000, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveGetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveSetTriggerDelay

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the external trigger delay.

Syntax

GtWaveSetTriggerDelay (*nHandle, nChannel, dDelay, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1120 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dDelay	DOUBLE	External trigger delay range is 0 - 15 sec, with 4 ns resolution. Units are in uSec, e.g. if pdDelay is equal 10 then the delay is 10uSec.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the external trigger edge:

```
DOUBLE dDelay;
GtWaveSetTriggerDelay (nHandle, GTWAVE_CHANNEL_A, 100, &nStatus);
```

See Also

GtWaveGetTriggerDelay, GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveSetTriggerEdge

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the external trigger edge.

Syntax

GtWaveSetTriggerEdge(*nHandle, nChannel, nSlope, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSlope	SHORT	External trigger edge slope can be: 0. GTWAVE_EXTERNAL_TRIGGER_RISING_EDGE 1. GTWAVE_EXTERNAL_TRIGGER_FALLING_EDGE
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets external trigger edge to rising edge:

```
GtWaveSetTriggerEdge (nHandle, GTWAVE_CHANNEL_A, GTWAVE_EXTERNAL_TRIGGER_RISING_EDGE,
&nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveGetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveSetTriggerHoldoff

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the external trigger hold off delay.

Syntax

GtWaveSetTriggerHoldoff (*nHandle, nChannel, dHoldoff, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dHoldoff	DOUBLE	External trigger holdoff range is 0 - 15 sec, with 4 ns resolution. Units are in uSec, e.g. if pdDelay is equal 10 then the hold off is 10uSec.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the external trigger hold off:

```
GtWaveSetTriggerHoldoff (nHandle, GTWAVE_CHANNEL_A, 100, &nStatus);
```

See Also

GtWaveGetTriggerHoldoff, **GtWaveGetTriggerDelay, GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString**

GtWaveSetTriggerInternalFrequency

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the trigger generator's internal frequency.

Syntax

GtWaveSetTriggerInternalFrequency(*nHandle, nChannel, dRate, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dFrequency	DOUBLE	GX1110: Trigger internal frequency range can be form GTWAVE_GX1110_MIN_TRIGGER_INTERNAL_FREQUENCY (0.01Hz) to GTWAVE_GX1110_MAX_TRIGGER_INTERNAL_FREQUENCY (50MHz). GX1120: Trigger internal frequency range can be form GTWAVE_GX1120_MIN_TRIGGER_INTERNAL_FREQUENCY (0.01Hz) GTWAVE_GX1120_MAX_TRIGGER_INTERNAL_FREQUENCY (1 MHz)
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

A built-in programmable trigger generator can be used as a replacement of an external trigger source. The number of waveforms output for each internal trigger is determined by the **GtWaveGetTriggerBurstCount** function.

Example

The following example sets the trigger internal frequency to 10KHz:

```
GtWaveGetTriggerInternalFrequency (nHandle, GTWAVE_CHANNEL_A, 10e3, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveGetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveSetTriggerLevel

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the external trigger level.

Syntax

GtWaveSetTriggerLevel (*nHandle, nChannel, dLevel, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
dLevel	DOUBLE	External trigger level range is -5V to +5V with 10mV of resolution.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the external trigger hold off:

```
GtWaveSetTriggerLevel (nHandle, GTWAVE_CHANNEL_A, 2.5, &nStatus);
```

See Also

GtWaveGetTriggerLevel, **GtWaveSetTriggerHoldoff**, **GtWaveSetTriggerDelay, GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString**

GtWaveSetTriggerMode

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the trigger mode.

Syntax

GtWaveSetTriggerMode(*nHandle, nChannel, nMode, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nMode	SHORT	Trigger mode are: 0. GTWAVE_TRIGGER_MODE_CONTINUOUS 1. GTWAVE_TRIGGER_MODE_TRIGGERED 2. GTWAVE_TRIGGER_MODE_GATED 3. GTWAVE_TRIGGER_MODE_BURST
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Function Generator:

GTWAVE_TRIGGER_MODE_CONTINUOUS	The waveform generates continuously by repeatedly cycling through the waveform table using the programmed waveform parameters.
GTWAVE_TRIGGER_MODE_TRIGGERED	Output is quiescent until triggered by an internal or external trigger, and then one waveform cycle is generated using the programmed waveform parameters.
GTWAVE_TRIGGER_MODE_GATED	The waveform generates continuously by repeatedly cycling through the waveform table as long as the gate is active
GTWAVE_TRIGGER_MODE_BURST	After a trigger is received, waveform generation will be executed for the number of cycles that were defined via the **GtWaveGetTriggerBurstCount** function

Arbitrary Waveform Generator mode:

GTWAVE_TRIGGER_MODE_CONTINUOUS	The waveform generates continuously by repeatedly cycling through the defined waveform length using the programmed waveform parameters. The waveform generation starts from the specified first address and continues through to the specified waveform length. After the last step is completed, the waveform generation loops back to the specified first address and continues until it is stopped.
GTWAVE_TRIGGER_MODE_TRIGGERED	Output is quiescent until triggered by an internal or external trigger, then one waveform cycle is generated using the programmed waveform parameters.
GTWAVE_TRIGGER_MODE_GATED	The waveform generates continuously by repeatedly cycling through the waveform table as long as the gate is active
GTWAVE_TRIGGER_MODE_BURST	After a trigger is received, waveform generation starts from the specified first address and continues through to the specified waveform length. Waveform generation will be executed for the number of cycles that were defined via the **GtWaveGetTriggerBurstCount** function.

The function calls **GtWaveSynchronizePhases** API before returning.

Example

The following example sets trigger mode to continuous:

```
GtWaveSetTriggerMode (nHandle, GTWAVE_CHANNEL_A, GTWAVE_TRIGGER_MODE_CONTINUOUS, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveSetTriggerSource

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the trigger source.

Syntax

GtWaveSetTriggerSource(*nHandle, nChannel, nSource, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nSource	SHORT	Trigger source are as follows: 0. GTWAVE_TRIGGER_SOURCE_SOFTWARE: Trigger source is software (immediate). 1. GTWAVE_TRIGGER_SOURCE_INTERNAL: Trigger source is the internal programmable trigger generator. 2. GTWAVE_TRIGGER_SOURCE_EXTERNAL: Trigger source is an external input. 3. GTWAVE_TRIGGER_SOURCE_PXI_STAR_TRIG: Trigger source is the star trigger. 4. `GTWAVE_TRIGGER_ALTERNATE_CHANNEL:` Trigger source is the alternate channel.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The trigger source can be set to one of the four sources and also to one of the PXI trigger lines. See the **GtWaveSetPxiTriggerBusLine** function for details.

The function calls **GtWaveSynchronizePhases** API before returning.

Example

The following example sets the trigger source to software:

```
GtWaveSetTriggerSource (nHandle, GTWAVE_CHANNEL_A, GTWAVE_TRIGGER_SOURCE_SOFTWARE, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveGetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString

GtWaveSetTriggerToPxiTriggerBusLine

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the trigger to a PXI trigger bus line and state.

Syntax

GtWaveSetTriggerToPxiTriggerBusLine(*nHandle, nChannel, nLine, bState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
nLine	SHORT	Assigned trigger to PXI trigger bus line value can be one of the following: 0. GTWAVE_PXI_TRIGGER_BUS_LINE0 1. GTWAVE_PXI_TRIGGER_BUS_LINE1 2. GTWAVE_PXI_TRIGGER_BUS_LINE2 3. GTWAVE_PXI_TRIGGER_BUS_LINE3 4. GTWAVE_PXI_TRIGGER_BUS_LINE4 5. GTWAVE_PXI_TRIGGER_BUS_LINE5 6. GTWAVE_PXI_TRIGGER_BUS_LINE6 7. GTWAVE_PXI_TRIGGER_BUS_LINE7
bState	BOOL	Trigger to PXI trigger bus line state: 0. Disconnected. 1. Connected.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Note: Only a single line can be connected at any given time.

Example

The following example sets the trigger to PXI trigger bus line 1 and state enabled:

```
GtWaveSetTriggerToPxiTriggerBusLine (nHandle, GTWAVE_CHANNEL_A, GTWAVE_PXI_TRIGGER_BUS_LINE1,
TRUE, &nStatus);
```

See Also

GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, **GtWaveRun, GtWaveStop, GtWaveTrig, GtWaveGetStatusRegister, GtWaveSetAmplitude, GtWaveSetOffset, GtWaveSetOperationMode, GtWaveSetOutputState, GtWaveSetReferenceClockSource, GtWaveGetErrorString**

GtWaveSetVoltageRangeMode

Supported By

Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Sets the Amplitude and Offset voltages setting mode and voltage range.

Syntax

GtWaveSetVoltageRangeMode (*nHandle, nChannel, nMode, dVoltageRange, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1110 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A.
nMode	SHORT	Sets the amplitude and Offset voltages setting mode: 0. GTWAVE_VOLTAGE_RANGE_AUTO – each time the Amplitude or Offset voltages are set the driver runs an algorithm to produce the best possible resolution for the specified Amplitude and offset voltage settings by changing the output attenuators. The dVoltageRange variable is ignored in this mode. 1. GTWAVE_VOLTAGE_RANGE_BY_VALUE - the driver sets the best output attenuators for the specified (*dVoltageRange*) peak-to-peak voltage range. In this mode setting the Amplitude and/or Offset voltages will not change the output attenuators settings.
dVoltageRange	DOUBLE	Specified the peak-to-peak voltage range. Value is only used when *nMode* is set to GTWAVE_VOLTAGE_RANGE_BY_VALUE.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

After calling this function the Offset is set to 0 and the amplitude to p*dVoltageRange*.

Note: Calling the reset command will reset the mode back to the Auto.

Example

The following example sets the amplitude and offset voltages to by value mode and voltage range to 4.0V:

```
GtWaveReset (nHandle, GTWAVE_CHANNEL_A, GTWAVE_VOLTAGE_RANGE_BY_VALUE, 4.0, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveStop

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Disables the board from running.

Syntax

GtWaveStop(*nHandle, nChannel, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example disables the board from running:

```
GtWaveStop (nHandle, GTWAVE_CHANNEL_A, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveSynchronizePhases

Supported By

Gx1120

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Synchronize both channels phases.

Syntax

GtWaveSynchronizePhases (*nHandle, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example Synchronize both channels phases:

```
GtWaveSynchronizePhases (nHandle, &nStatus);
```

See Also

GtWaveGetTriggerLevel, **GtWaveSetTriggerHoldoff**, **GtWaveSetTriggerDelay, GtWaveSetMarkerToPxiTriggerBusLine, GtWaveSetPxiTriggerBusLine, GtWaveSetTriggerBurstCount, GtWaveSetTriggerEdge, GtWaveSetTriggerInternalFrequency, GtWaveSetTriggerMode, GtWaveSetTriggerSource, GtWaveSetTriggerToPxiTriggerBusLine, GtWaveGetErrorString**

GtWaveTestMemory

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Runs a memory self-test.

Syntax

GtWaveTestMemory(*nHandle, nChannel, pdwFailedStep, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pdwFailedStep	PDWORD	Returned first failed address, if no failure were found returns -1.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function runs a memory self-test, after calling this function all the memory data will be set to 0.

Example

The following example runs a memory self-test:

```
DWORD dwFailedStep;
GtWaveTestMemory (nHandle, GTWAVE_CHANNEL_A, &dwFailedStep, &nStatus);
if (dwFailedStep==-1)
    printf("Memory pass self-test");
else
    printf("Memory failed self-test, first failed address is at %i", dwFailedStep);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

GtWaveTrig

Supported By

Gx1120, Gx1110

Applies To

Function Generator and Arbitrary Waveform Generator modes.

Purpose

Issue a software trigger.

Syntax

GtWaveTrig(*nHandle, nChannel, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX11X0 board.
nChannel	SHORT	Specified Channel number. GX1110: 0 = GTWAVE_CHANNEL_A: Channel A. GX1120: 0 = GTWAVE_CHANNEL_A: Channel A. 1 = GTWAVE_CHANNEL_B: Channel B.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example issues a software trigger:

```
GtWaveTrig (nHandle, GTWAVE_CHANNEL_A, &nStatus);
```

See Also

GtWaveInitialize, **GtWaveGetErrorString**

Index