GT1648

Analog Outputs Board

User's Guide

Version A (10/01/2002)

GEOTEST

MARVIN TEST SYSTEMS, INC.

Safety and Handling

Each product shipped by Geotest is carefully inspected and tested prior to shipping. The shipping box provides protection during shipment, and can be used for storage of both the hardware and the software when they are not in use.

The circuit boards are extremely delicate and require care in handling and installation. Do not remove the boards from their protective plastic coverings or from the shipping box until you are ready to install the boards into your computer.

If a board is removed from the computer for any reason, be sure to store it again in its original shipping box. Do not store boards on top of workbenches or other areas where they might be dropped or be exposed to strong electromagnetic or electrostatic fields. Boards should be stored in protective anti-electrostatic wrapping and away from electromagnetic fields.

Be sure to make a single copy of the software diskette and store the original in a safe place away from electromagnetic or electrostatic fields.

Warranty

Geotest products are warranted against defects in materials and workmanship for a period of 12 months (3 months for software products). Geotest shall repair or replace (at its discretion) any defective product during the stated warranty period. The software warranty includes any revisions or new versions released during the warranty period. If you need to return a board, please call 949-263-2222 for an RMA number.

If You Need Help

Visit our web site at http://www.geotestinc.com for more information about Geotest's products, services and support options. Our web site contains sections describing: Support options, Application notes, Download area for downloading patches, example, patches and new or revised instrument drivers. To submit a support issue including suggestion, bug report or question please use the following link:

http://www.geotestinc.com/magic

You can also use Geotest technical support phone line (949) 263-2222. This service is available between 9:00 AM and 5:00 PM Pacific Standard Time.

Disclaimer

Copyright

Trademarks

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

Table of Contents

Safety and Handling i
Warranty i
If You Need Help i
Disclaimer ii
Copyright ii
Trademarks ii
Chapter 1 - Introduction 1
Manual Scope and Organization 1
Manual Scope 1
Manual Organization 1
Conventions Used in this Manual 2
Chapter 2 - Overview 3
Introduction 3
Functional Overview 3
Board Description 4
Architecture 6
Specifications 7
Virtual Panel Description 8
Chapter 3 - Installation and Connections 13
Getting Started 13
Packing List 13
Unpacking and Inspection 13
Installation of the GT1648 Board 13
Introduction 13
Switch and Jumper Settings 14
I/O Base Address Chart 15
Connectors and Accessories 17

Connectors 17
Output Channels Connector (J3) 18
Installation of the GTAO Driver 19
Installation Directories 21
Driver Files Description 22
Driver File and Virtual Panel 22
Interface Files 22
On-line Help and Manual 22
ReadMe File 23
Example Programs 23
Setup Maintenance Program 25
Chapter 4 - Programming the Board 27
The GT1648 Driver 27
Programming Using C/C++ Tools 28
Programming Using Visual Basic 28
Programming Using Pascal/Delphi 28
Programming GTAO Boards Using ATEasy® 29
Using the GTAO driver functions 30
Board Handle 30
Reset 30
Error Handling 30
Driver Version 31
Panel 31
Distributing the Driver 31
Sample Programs 32
Sample Program Listing 33
Chapter 5 - Functions Reference 39
Introduction 39
GTAO Functions 41

GtAoGetBoardAccuracy 43
GtAoGetBoardSummary 44
GtAoGetChannelVoltage 46
GtAoGetDriverSummary 47
GtAoGetErrorString 48
GtAoGetGroupExternalUpdate 50
GtAoGetGroupUpdateState 51
GtAoGetGroupVoltageRange 52
GtAoGetPort 54
GtAoGetPortBit 55
GtAoGetPortDirection 56
GtAoInitialize 57
GtAoLoadChannelVoltage 58
GtAoPanel 59
GtAoReset 61
GtAoResetGroup 62
GtAoSetChannelVoltage 63
GtAoSetGroupExternalUpdate 64
GtAoSetGroupVoltageRange 65
GtAoSetPort 67
GtAoSetPortBit 68
GtAoSetPortDirection 69
GtAoUpdateAllGroupsVoltage 70
GtAoUpdateGroupVoltage 71

Index 73

Chapter 1 - Introduction

Manual Scope and Organization

Manual Scope

The purpose of this manual is to provide all the necessary information to install, use, and maintain the GT1648 instrument. This manual assumes the reader has a general knowledge of PC based computers, Windows operating systems, and some understanding of digital to analog conversion.

This manual also provides programming information using the GT1648 driver. Therefore, good understanding of programming development tools and languages may be necessary.

Manual Organization

The GT1648 manual is organized in the following manner:

Chapter	Content
Chapter 1 - Introduction	Introduces the GT1648 manual. Lists all the supported board and shows warning conventions used in the manual.
Chapter 2 – Overview	Describes the GT1648 features, board description, its architecture, specifications and the panel description and operation.
Chapter 3 – Installation and Connections	Provides instructions on how to install the GT1648 Board and its accompanying GTAO software.
Chapter 4 – Programming the Board	Provides a listing of GT1648 driver files, general purpose/generic driver functions, and programming methods. Discusses various supported operating systems and development tools.
Chapter 5 – Functions Reference	Contains a listing of the general GT1648 functions. Each function is described along with its syntax, parameters, and special programming comments. Samples are given for each function.

Conventions Used in this Manual

Symbol Convention	**Meaning**
⚡	Static Sensitive Electronic Devices. Handle Carefully.
🛑 STOP	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.
▼ TIP	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, programming function names and window control names.
Italic type	Specialized terms. Titles of other reference books. Placeholders for items you must supply, such as function parameters

Chapter 2 - Overview

Introduction

The GT1648 board is a full size ISA bus board that provides precision high-speed analog outputs. The board has *three groups, each with 16 analog outputs channels with 12-bit resolution each. The voltage range for each group can be programmed to 0-10V, ±10V, 0-5V or ±5V. Each group's channels voltages can be set simultaneously through software or by an internal or external command signal.

* The board is also available with an additional group (group D) of 16 analog output channels.

Functional Overview

Principal capabilities of the GT1648 board are summarized in the following list of features.

- 48 Precision High-Speed Analog Output Channels
- 12-Bit D/A Converter per Channel.
- Accuracy of 3mV ±1LSB.
- Each channel can source/sink up to 10 mA for any given voltage setting.
- Data update rate of 10 KSPS (Kilo samples per second).
- Outputs updated simultaneously or per channel (software-selectable).
- External outputs signal update for each group allows for simultaneous continuous update.
- Output voltage range for each group can be programmed to 0-10V, ±10V, 0-5V or ±5V.
- Fast settling time of 50 uSec to 0.1% of programmed value.
- All channels are automatically at zero volts DC after power reset.
- TTL I/O port with four pins.
- Upgradeable Firmware using software driver.

Board Description

The GT1648 board contains three groups with 16 channels each. All groups' channels have a dedicated high-speed 12-bit D/A converter (DAC). Each group can be independently set to one of the four available voltage ranges: 0-10V, ±10V, 0-5V or ±5V. Changing the voltage range will change the group's channels voltage resolution. For example changing the range ±10V to 0-5V will change the resolution from 4.88mV to 1.22mV. All channels will source or sink a minimum of 10mA of supply current for all voltage ranges.

The board has a local controller that performs all internal configuration and data manipulation between the ISA bus and the board. Values are first loaded to the specified channel output buffer before it is programmed to the corresponding DAC. Each DAC has an output buffer enabling users to load all (or part) of the board channels with new values and then update all outputs at once.

The local controller reads the 16-bit channel data value for each channel from the analog output buffer, and sends the value serially to the associated output DAC. The output DAC holds that word in an internal buffer until commanded to transfer the data to the output register that drives the DAC output. All output registers are updated simultaneously.

Each group has a dedicated external Update Line available at the 78-pin output connector (J3). A ground or TTL low on the External Update Line will cause the group's channels to continuously update the group outputs to update.

The board draws power from the ISA bus +5 VDC in accordance with the ISA specifications.

The board has a Digital I/O port with four bits. The port direction can be set either to output or input (default). A 78-pin D-type connector is used, for the 48 Output Channels, three external signals inputs, 4-bit TTL I/O port and ground signals.

The board requires +5VDC from the host's ISA expansion bus. The host PC, +5VDC supplies power to the board. The board requires a maximum of 2.5 ADC from the host PC.

Figure 2-1 shows the GT1648 with the J3 connector and SW1 that is used to set the board IO base address.

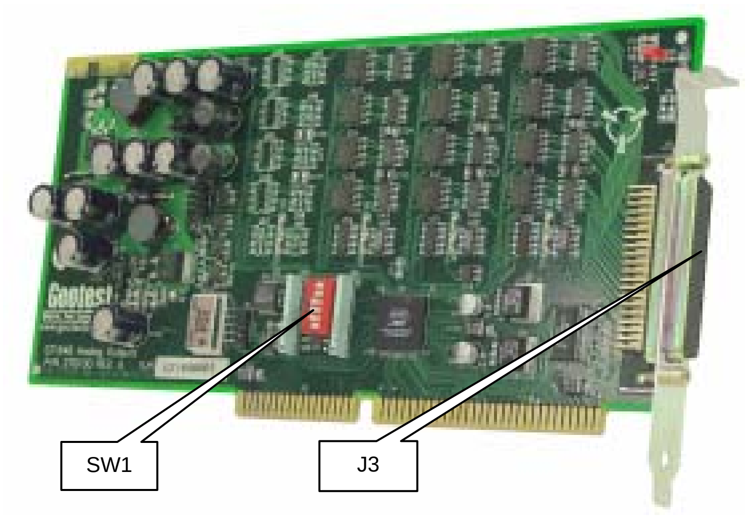


Figure 2-1: GT1648 Board (side view)

Architecture

Figure 2-2 illustrates the GT1648 architecture. The board is programmed by software using the ISA bus interface to set group range, D/A output and update, and the on-board 4 pins digital I/O port. The 78 pin connector on the right (J3) has pins that output the D/A and digital output and input signals, and the external update signals.

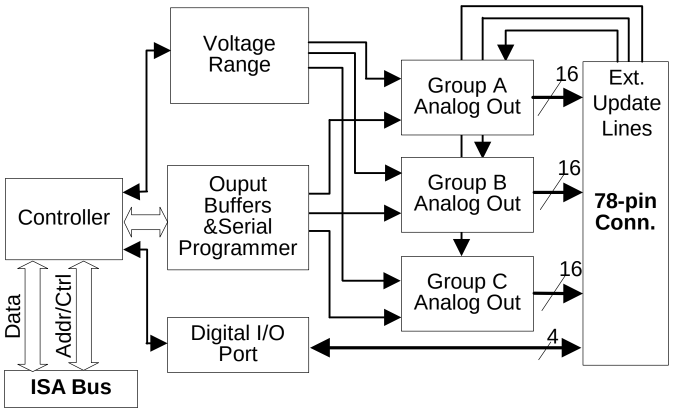


Figure 2-2 GT1648 Block Diagram

Specifications

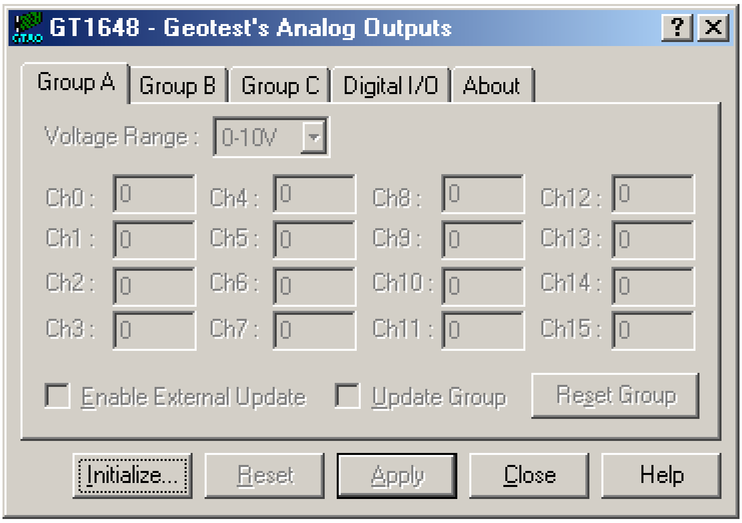
The following table outlines the specifications of the GT1648.

Number of Output channels	48 (64 - optional)
Output Voltage Ranges:	
Number of Ranges	4 (each group)
Selectable Output Voltage Ranges	0-10V, ±10V, 0-5V or ±5V
Resolution	12-bit
Accuracy	3mV ±1 LSB
Slew Rate	0.5V/µs
Settling time	50 uSec to 0.1%
Max. Current per channel	10 mA
Power supply 5 VDC Current	2.5A
Operating Temperature	0 to+55°C
Storage Temperature	-20 to+70°C
Size	10" long ISA card
Weight	12 oz.

Virtual Panel Description

The GT1648 includes a virtual panel program, which enables full utilization of the various configurations and controlling 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 **GT1648 Panel** from the **Geotest**, **GTAO** menu under the **Start** menu. The GT1648 virtual panel opens as shown here:


GT1648 - Geotest's Analog Outputs

Voltage Range : 0-10V

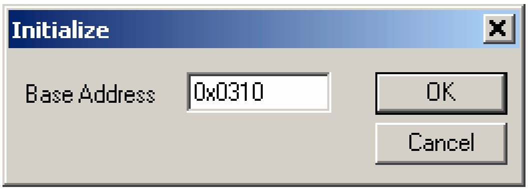
Ch0 :	0	Ch4 :	0	Ch8 :	0	Ch12 :	0
Ch1 :	0	Ch5 :	0	Ch9 :	0	Ch13 :	0
Ch2 :	0	Ch6 :	0	Ch10 :	0	Ch14 :	0
Ch3 :	0	Ch7 :	0	Ch11 :	0	Ch15 :	0

☐ Enable External Update ☐ Update Group

Figure 2-3: GT1648 Virtual Panel

The following controls are shown:

Initialize: Clicking on this button will cause the Initialize dialog box as shown in Figure 2-4 to be displayed. The user is prompted to type the base address where the board is installed.



0x0310

Figure 2-4: Initialize Dialog Box

Type the board base address and click **OK** to initialize the driver for the specified board. Successful initialization of the board will enable the panel controls. The panel will than display the current setting of the board (Initialization does not change the board settings).

Reset: Resets the board to its default state as follow:

- All groups voltage ranges set to 0-10V
- All groups channels set to 0V
- External updates disabled
- Digital I/O port value is set to zero and port direction set to input

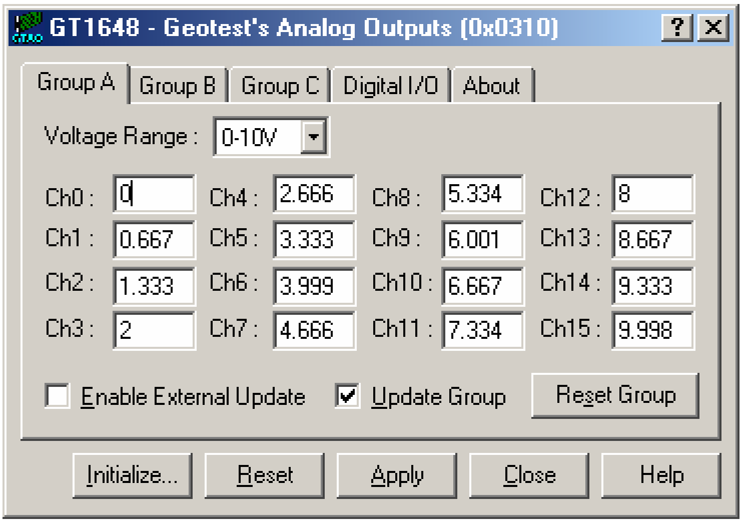
Close: Closes (exits) the GT1648 panel.

Help: Opens the GT1648 on-line help window.

Four tabs are displayed at the top of the current active page **Group A**, **Group B, Group C, Digital I/O** and **About**. These tabs are described below.

Virtual Panel Group A/B/C Pages

The following picture shows the **Group A page** followed by explanation of each one of the controls displayed on the page:


GT1648 - Geotest's Analog Outputs (0x0310)

Group A | Group B | Group C | Digital I/O | About

Voltage Range : 0-10V

Ch0 :	0	Ch4 :	2.666	Ch8 :	5.334	Ch12 :	8
Ch1 :	0.667	Ch5 :	3.333	Ch9 :	6.001	Ch13 :	8.667
Ch2 :	1.333	Ch6 :	3.999	Ch10 :	6.667	Ch14 :	9.333
Ch3 :	2	Ch7 :	4.666	Ch11 :	7.334	Ch15 :	9.998

☐ Enable External Update ☑ Update Group

Figure 2-5 GT1648 Virtual Panel – Group A page

Voltage Range Dropdown list: Select/display the group voltage range.

Ch 0-15 Edit Boxes: Each channel is displayed with an edit box for displaying the loaded voltage value. The value is updated continuously until the user types a new value to set. Clicking on the **Apply** button (or the **Enter** key) will load the board with the modified values and if the **Update Group** is checked will also output the new DAC values. After the **Apply** was pressed the panel will resume updating and display the board loaded values continuously.

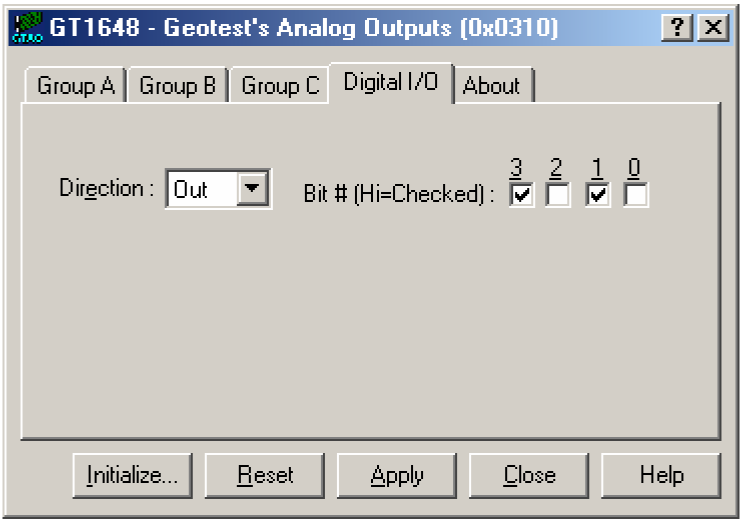
Update Group Check Box: When checked (default), clicking on the **Apply** button the group's channels will load and update the DACs to output the new values. When unchecked, clicking on the **Apply** button the group's channels will only be loaded and the DAC output will not change.

Enable External Update: When checked the specified group can be updated by the groups' external input line, when unchecked the groups' external input line is disabled.

Reset Group Button: Sets all the groups' channels to zero volts, set voltage range to 0-10V and disables the External Update.

Virtual Panel Digital I/O Page

Clicking on the **Digital I/O** tab will show the **Digital I/O page** as shown in Figure 2-6:


GT1648 - Geotest's Analog Outputs (0x0310)

Group A | Group B | Group C | Digital I/O | About

Direction : Out

Bit # (Hi=Checked) : 3 ✓ 2 ☐ 1 ✓ 0 ☐

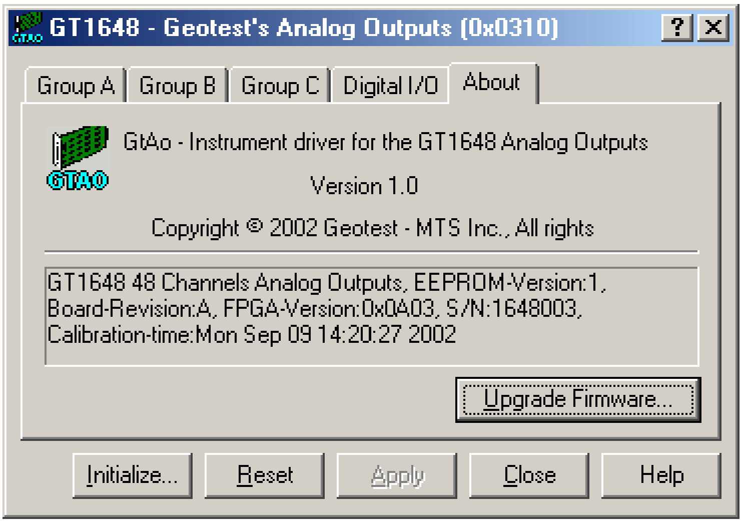
Figure 2-6: GT1648 Virtual Panel – Digital I/O Page

Direction: The direction dropdown sets the I/O port direction as Input or Output.

Bit # 0-3: When the port direction is Output, bits 0-3 are enabled for editing. When checked the specified bit number is set to high, when unchecked it's set to low.

Virtual Panel About Page

Clicking on the **About** tab will show the **About page** as shown in Figure 2-7:



GtAo - Instrument driver for the GT1648 Analog Outputs

Version 1.0

Copyright © 2002 Geotest - MTS Inc., All rights

GT1648 48 Channels Analog Outputs, EEPROM-Version:1, Board-Revision:A, FPGA-Version:0x0A03, S/N:1648003, Calibration-time:Mon Sep 09 14:20:27 2002

Figure 2-7: GT1648 Virtual Panel – About Page

The top part of the **About** page displays version and copyright of the GTAO driver. The bottom part displays the board summary, including the EEPROM version, the board Revision, the FPGA version, the board serial number and the calibration time. The **About** page also contains a button **Upgrade Firmware…** used to upgrade the board FPGA. This button maybe used only when the board requires upgrade as directed by Geotest 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 in order the new firmware to be used.

Chapter 3 - Installation and Connections

Getting Started

This product consists of both the board and the software module. This chapter describes how to install the GT1648 board and software module referred to as GTAO.

Packing List

All GT1648 boards have the same basic packing list, which includes:

1. GT1648 Board
2. GTAO Driver Disk/CD

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.

Installation of the GT1648 Board

Introduction

The GT1648 board I/O switch settings must be set before installation. If the board will be used in a GTXI chassis, the offset address switch on the GTXI carrier board must be set (for more information on how to install a board to the GTXI see the GTXI Instrumentation Chassis manual).

Switch and Jumper Settings

Switch Location:

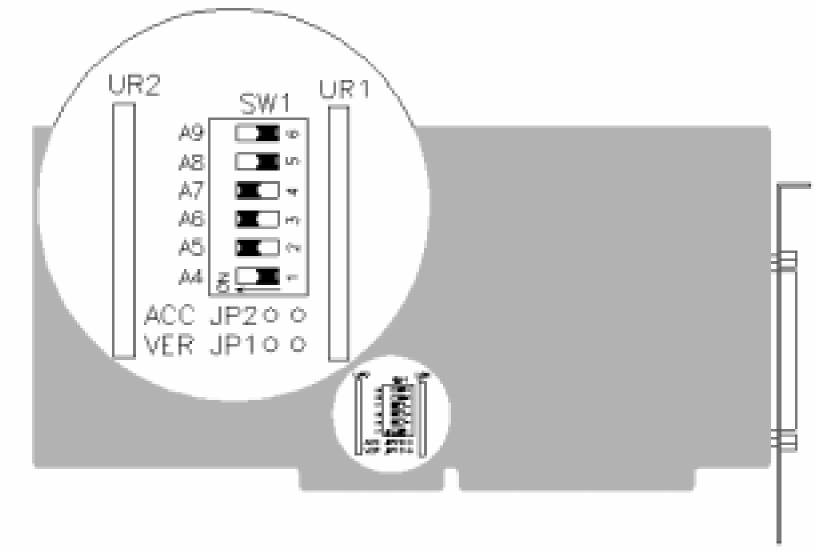


Figure 2-1: GT1648 Switch Location

SW1 Base Address Switch Settings: The GT1648 uses 16 consecutive I/O addresses. The base address switch (SW1) is used to set the base address of the board. The GT1648 has a default base address setting of 0x310 (shown in Figure 2-1). This means that addresses 0x310-0x31F must be available and not used by other system device.

Base addresses may be assigned within the range 0x000 to 0x3F0. Check your system configuration for available addresses.

Warning - It is strongly recommended that you only use the address range between 0x200 and 0x3F0.

The base address switch, SW1, is a six-position DIP switch located in the left bottom part of the GT1648 and is marked SW1. To set the appropriate base address uses a pen tip to move the individual switches into the ON and OFF positions. A switch in the ON position sets the related address bit to ‘0’ and a switch in the OFF position sets it to ‘1’. Figure 2-1 illustrates SW1 with the switches set to base address 0x310 (hex). Refer to Table 2-1 for possible switch combinations.

I/O Base Address Chart

SW1 / Address	**A9**	**A8**	**A7**	**A6**	**A5**	**A4**
200H	OFF	ON	ON	ON	ON	ON
210H	OFF	ON	ON	ON	ON	OFF
220H	OFF	ON	ON	ON	OFF	ON
230H	OFF	ON	ON	ON	OFF	OFF
240H	OFF	ON	ON	OFF	ON	ON
250H	OFF	ON	ON	OFF	ON	OFF
260H	OFF	ON	ON	OFF	OFF	ON
270H	OFF	ON	ON	OFF	OFF	OFF
280H	OFF	ON	OFF	ON	ON	ON
290H	OFF	ON	OFF	ON	ON	OFF
2A0H	OFF	ON	OFF	ON	OFF	ON
2B0H	OFF	ON	OFF	ON	OFF	OFF
2C0H	OFF	ON	OFF	OFF	ON	ON
2D0H	OFF	ON	OFF	OFF	ON	OFF
2E0H	OFF	ON	OFF	OFF	OFF	ON
2F0H	OFF	ON	OFF	OFF	OFF	OFF
300H	OFF	OFF	ON	ON	ON	ON
***310H**	OFF	OFF	ON	ON	ON	OFF
320H	OFF	OFF	ON	ON	OFF	ON
330H	OFF	OFF	ON	ON	OFF	OFF
340H	OFF	OFF	ON	OFF	ON	ON
350H	OFF	OFF	ON	OFF	ON	OFF
360H	OFF	OFF	ON	OFF	OFF	ON
370H	OFF	OFF	ON	OFF	OFF	OFF
380H	OFF	OFF	OFF	ON	ON	ON
390H	OFF	OFF	OFF	ON	ON	OFF

SW1 / Address	**A9**	**A8**	**A7**	**A6**	**A5**	**A4**
3A0H	OFF	OFF	OFF	ON	OFF	ON
3B0H	OFF	OFF	OFF	ON	OFF	OFF
3C0H	OFF	OFF	OFF	OFF	ON	ON
3D0H	OFF	OFF	OFF	OFF	ON	OFF
3E0H	OFF	OFF	OFF	OFF	OFF	ON
3F0H	OFF	OFF	OFF	OFF	OFF	OFF

Table 3-1: I/O Base Address

* Default address for GT1648 board.

Connectors and Accessories

The following accessories are available from Geotest for your switching board.

Part / Model Number	**Description**
GT78-CON	Connector, D-type 78-pin Male with solder pins.
GT78-MCON	Military Grade Connector, D-type 78-pin Male with solder pins.
GTXI-765	1' harness, 78-pin male connector on one side.
GTXI-766	1' harness, 78-pin male connector on both sides.

Table 3-2: Spare Connectors and Accessories

Connectors

Figure 3-1 shows the available GT1648 board connector with description:

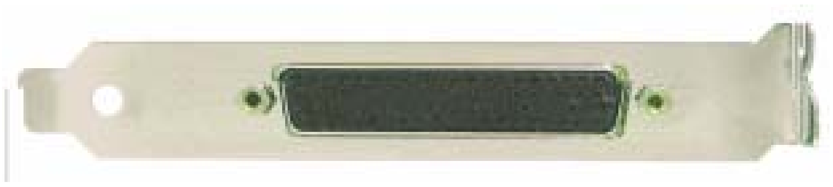


Figure 3-1: GT1648 J3 Output Channels Connector

Output Channels Connector (J3)

The following are the connector signal pin assignments for GT1648 module:

Pin#	Signal	Pin#	Signal	Pin#	Signal	Pin#	Signal
1	ChA0	21	ChB0	40	ChC0	60	ChD0
2	ChA1	22	ChB1	41	ChC1	61	ChD1
3	ChA2	23	ChB2	42	ChC2	62	ChD2
4	ChA3	24	ChB3	43	ChC3	63	ChD3
5	ChA4	25	ChB4	44	ChC4	64	ChD4
6	ChA5	26	ChB5	45	ChC5	65	ChD5
7	ChA6	27	ChB6	46	ChC6	66	ChD6
8	ChA7	28	ChB7	47	ChC7	67	ChD7
9	ChA8	29	ChB8	48	ChC8	68	ChD8
10	ChA9	30	ChB9	49	ChC9	69	ChD9
11	ChA10	31	ChB10	50	ChC10	70	ChD10
12	ChA11	32	ChB11	51	ChC11	71	ChD11
13	ChA12	33	ChB12	52	ChC12	72	ChD12
14	ChA13	34	ChB13	53	ChC13	73	ChD13
15	ChA14	35	ChB14	54	ChC14	74	ChD14
16	ChA15	36	ChB15	55	ChC15	75	ChD15
17	ExUpdateA	37	ExUpdateB	56	ExUpdateC	76	ExUpdateD
18	TTL I/O 0	38	TTL I/O 1	57	TTL I/O 2	77	TTL I/O 3
19	GND	39	GND	58	GND	78	GND
20	GND			59	GND		

Table 3-3: J3 – Output Channels Connector

Legend

- **ChX0 - ChX15**: Group X (Group A-C, D optional) channel output.
- **ExUpdateA - ExUpdateD**: External update line to the corespending group (A/B/C/D).
- **TTL I/O 0-3**: Digital I/O port bits 0 to 3.
- **GND**: Ground.

Installation of the GTAO Driver

The Setup installation program requires Windows, 95, 98, Me, NT, 2000 or above. In addition, Microsoft Windows Explorer (IE) version 4.0 or above is required to view the online help.

To run the GTAO Setup program:

Insert the GTAO CD-ROM disk in the CD-ROM drive (for floppy disk installation, insert Disk 1 to the floppy drive). The Setup program runs automatically if your drive is set up to auto play.

If Setup does *not* run automatically, select **Run** from the Start menu and when prompted, use the **Browse…** button to locate the Setup.exe located on the disk or type the following:

CD-ROM drive Only:

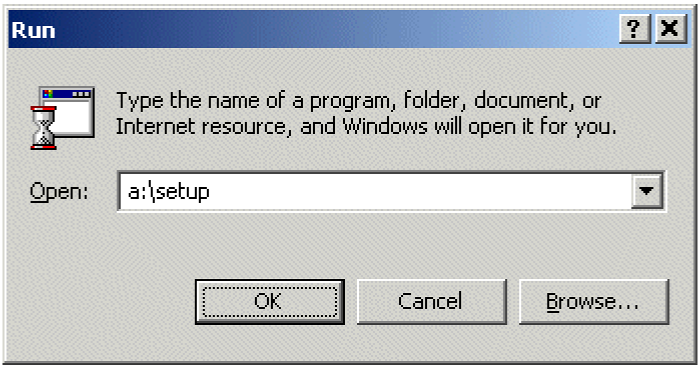
[drive letter]:\GTAO\Setup

Where [*drive letter*] is the drive letter assigned to your CD-ROM drive. As an example, type "**d:\gtao\setup**" when your CD-ROM is assigned to drive letter "D."

Floppy drive Only:

[drive letter]:\Setup

Where [*drive letter*] is the drive letter assigned to your Floppy drive (normally A:) as shown here:


Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: a:\setup

Figure 3-2: Run Window

Click **OK** to continue.

Note: When installing under Windows NT/2000/XP, you may be required to restart the setup after logging-in as a user with Administrator privileges. This is required to upgrade your system with newer Windows components and to install kernel-mode device drivers (HW.SYS and HWDEVICE.SYS) required by the GTAO driver to access resources on your board.

The first screen to appear is the Welcome screen. Click **Next** to continue.

Enter the folder where GTSW 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\Geotest\GTAO`.

Select the type of Setup you wish and click **Next.** You can choose between **Typical**, **Run-Time** and **Custom** setups. **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. **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 it finished if some of the components it replaced where used by another application during the installation. Please do so before attempting to use the software.

You can now test your installation by starting a panel program that let you control the board interactively. Launch the panel using the Start, Programs, Geotest, and GTAO menu located in the Windows Taskbar.

Installation Directories

The GT1648 driver files are installed in the default directory `C:\Program Files\Geotest\GTAO`. You can change the default `GTAO` directory to one of your choosing at the time of installation.

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

Name	**Purpose / Contents**
…\Geotest\GTAO	The GT1648 directory. Contains panel programs, programming libraries, interface files and examples, on-line help files and other documentation.
…\Geotest\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. GTAO 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 GTAO DLL driver and some upgraded system components, such as the HTML help viewer, etc.

Driver Files Description

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

Driver File and Virtual Panel

- GTAO.DLL - 32-Bit MS-Windows DLL for applications running under Windows, 95, 98, Me, NT, 2000, XP or above.
- GTAOPANEL.EXE – An instrument front panel program for all GTAO supported boards.

Interface Files

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

- GTAO.H - header file for accessing the DLL functions using the C/C++ programming language. The header file compatible with the following 32 bit development tools:
    - Microsoft Visual C++, Microsoft Visual C++ .NET
    - Borland C++
- GTAO.LIB - Import library for GTAO.DLL (used when linking C/C++ application that uses GTAO.DLL).
- GTAOBC.LIB - Import library for GTAO.DLL (used when linking Borland C/C++ application that uses GTAO.DLL).
- GTAO.PAS - interface file to support Borland Pascal or Borland Delphi.
- GTAO.BAS - Supports Microsoft Visual Basic 4.0, 5.0 and 6.0.
- GTAO.VB - Supports Microsoft Visual Basic .NET.
- Gt1648.drv - ATEasy driver File for Gt1648 and GTAO Virtual Panel Program

On-line Help and Manual

GTAO.CHM – On-line version of the GT1648 User’s Guide. The help file is provided in a Windows Compiled HTML help file (.CHM). The file contains information about the GT1648 board, programming reference and panel operation.

GT1648.PDF – On line, printable version of the GT1648 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 File

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 GTAO manuals. You can view and/or print this file using the Windows NOTEPAD.EXE or any other text file editors.

Example Programs

The sample program includes a C/C++ sample compiled with various development tools, Visual Basic example and an ATEasy sample. Other examples may be available for other programming tools.

Microsoft Visual C++ .NET example files:

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

Microsoft Visual C++ 6.0 example files:

- GtAoExampleC.cpp - Source file
- GtAoExampleC.ico - Icon file
- GtAoExampleC.rc - Resource file
- GtAoExampleC.dsp - VC++ project file
- GtAoExampleC.exe - Example executable

Borland C++ example files:

- GtAoExampleC.cpp - Source file
- GtAoExample.ico - Icon file
- GtAoExampleC.rc - Resource file
- GtAoExampleC.bpr - Borland project file

- GtAoExampleC.exe - Example executable

Microsoft Visual Basic .NET example files:

- GtAoExampleVB.vb - Example form.
- GtAoExampleVB.resx - Example form resource.
- GtAoExampleVBapp.config - Example application configuration file.
- GtAoExampleVBAssembleyInfo.vb - Example application assembly file
- GtAoExampleVB.vbproj - Project file
- GtAoExampleVB.exe - Example executable

Microsoft Visual Basic 6.0 example files:

- GtAoExampleVB6.frm - Example form
- GtAoExampleVB6.frx - Example form binary file
- GtAoExampleVB6.vbp - Project file
- GtAoExampleVB6.exe - Example executable.

ATEasy driver and examples files (ATEasy Drivers directory):

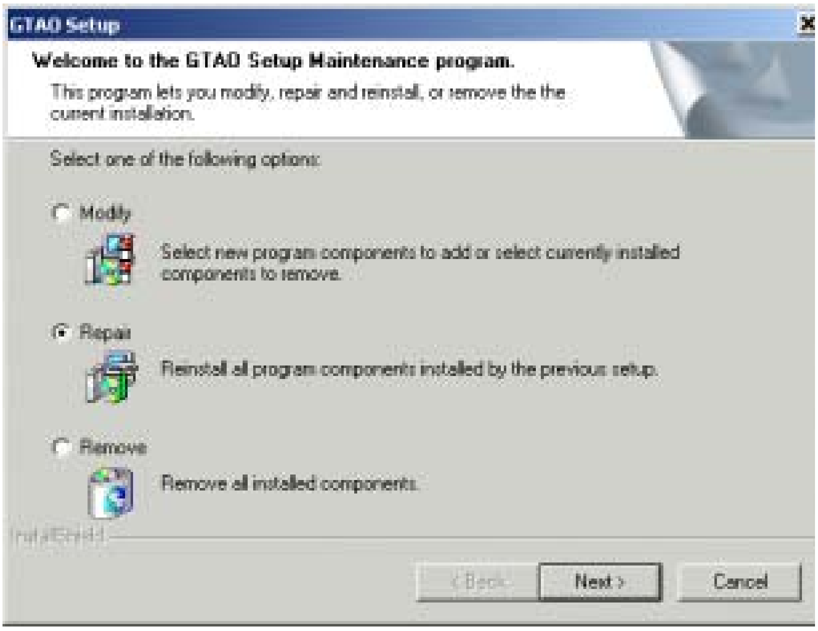
- Gt1648.drv - driver
- Gt1648.prj - example project
- Gt1648.sys - example system
- Gt1648.prg - example program

Setup Maintenance Program

You can run Setup again after GTAO 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 GT1648 installation. The following options are available in Maintenance mode:

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

The Maintenance mode screen is shown below:



GTAO Setup

Welcome to the GTAO Setup Maintenance program.

This program lets you modify, repair and reinstall, or remove the the current installation.

Select one of the following options:

Modify [✓ Repair] Remove

Modify: Select new program components to add or select currently installed components to remove.

Repair: Reinstall all program components installed by the previous setup.

Remove: Remove all installed components.

Figure 3-3: Setup Maintenance Window

Select one of the options and click **Next**.

Follow the instruction on the screen until Setup is complete.

Chapter 4 - Programming the Board

This chapter contains information about how to program the switching instruments using the GT1648 driver. The GTAO driver contains functions to initialize, reset, and control the switching instruments. 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 GTAO driver supports many development tools. Using these tools with the driver is described in this chapter. In addition, the GTAO 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 an outline for other programming languages, programming tools, and other GTAO driver types.

The GT1648 Driver

The GT1648 driver is a 32-bit Windows DLL file: GTAO.DLL. The DLL is used with 32 bit applications running under Windows 95/98/ME and Windows NT/2000/XP. The DLL uses a device driver to access the board resources. The device driver HW.SYS (on Windows NT/2000/XP) or HW.VXD (on Windows 9x/Me) is installed by the GTAO setup program and is shared by other Geotest products (ATEasy, GTAO, etc).

The 32-bit DLL 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 GT1648 driver with C/C++ development tools:

- Include the GTAO.H header file in the C/C++ source file that uses the GT1648 function. This header file is used for all driver types. The file contains function prototypes and constant declarations to be used by the compiler for the application.
- Add the required .LIB file to the projects. This can be import library GTAO.LIB for Microsoft Visual C++ and GTAOBC.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 GTAO 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 GTAO.BAS to the project. For Visual Basic .NET use the GTAO.VB.

The file can be loaded using *Add File* from the Visual Basic *File menu*. The GTAO.BAS/.VB 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 GTAO.PAS to the project. The GTAO.PAS file contains a **unit** with function prototypes for the DLL functions. Include the GTAO unit in the **uses** statement before making calls to the GTAO functions.

Programming GTAO Boards Using ATEasy®

The GTAO package is supplied with an ATEasy driver for each of the board types supported by this package. The ATEasy driver uses the GTAO.DLL to program the board. Each ATEasy driver includes an example that contains a program and a system file for use with the ATEasy driver. Use the driver shortcut property page from the System Drivers sub-module to correct the board slot number (PCI) or base address (ISA) before attempting to run the example.

Plain language 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 commands 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. GTAO1, GTAO2 for GT1648.
- 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 use of plain language commands is self-documenting. The 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.

Using the GTAO driver functions

The GTAO driver contains a set of functions for each of the supported instrument boards. The function name starts with the board type.

Every Geotest driver has similar functions that initialize the board driver, reset the board, and display the instrument virtual panel. In addition, all the board types use handles (see below) to access the boards and use the same error handling method. The following paragraphs describe the steps required to program the boards.

The **GtAoInitialize** function returns a handle that must be used with other driver functions to program the board. This handle is usually saved in the program in a global variable for later use when calling other functions. The initialize function does not change the state of the board or its setting.

Board Handle

The board handle argument *nHandle* passed (by reference) to the parameter *pnHandle* of the **GtAoInitialize** is a short integer (16 bits) number. It is used by the GTAO 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 identify which board is being programmed.

The *nHandle* is created when the application calls the **GtAoInitialize** function. There is no need to destroy the handle. Calling **GtAoInitialize** with the same slot number (PCI) or base address (ISA) will return the same handle.

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

Reset

The Reset function **GtAoReset** opens all switches and sets the board to a known initial state. A reset is usually performed after the board is initialized.

Error Handling

All the GTAO function returns status named *pnStatus* in the last parameter. This parameter can be later used for error handling. The status is zero for success, less than zero for failure or error. When the status is error, the program can call the **GtAoGetErrorString** function to return a string representing the error. The **GtAoGetErrorString** reference contains possible error numbers and their associated error strings.

Driver Version

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

Panel

Calling the **GtAoPanel** will display the instrument front panel window. The panel can be used to display its current setting and to control the board interactively. The panel function may be used by the application to allow the user to directly interact with the board.

The **GtAoPanel** function is also used by the GTAOPANEL.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 (GTAO.DLL and the HW device driver files located in the HW folder) can be shipped with the application. Typically, the GTAO.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 GTAO driver files. Alternatively, you can provide the GTAO disk to be installed along with the board.

Sample Programs

The following example demonstrates how to program the board using the C programming language under Windows. The example shows how to get or set a group or channel voltage.

To run, Enter the following command line:

GtAoExample <Baseaddress> <operation> <group> <channel> <voltage>

Where:

<Baseaddress>	Board base address where the board resides.
<Operation>	Operation code: GCV=Get channel voltage SCV=Set channel voltage SVR=Set group voltage range GVR=Get group voltage range
<Group >	Group number: 0 to 2
<Channel|Range>	Channel number: 0 to 15 Voltage Range: 0 to 3
<Value>	Set or get value.

Sample Program Listing

```
/*********************************************************
    FILE        : GtAoExampleC.cpp

    PURPOSE     : WIN32 sample program for GT1648 board
                  using the GTAO driver.

    CREATED     : Aug. 2002

    COPYRIGHT   : Copyright 2002 GEOTEST - MTS Inc.

    COMMENTS    :

    To compile the WIN32 example:

    1. Microsoft VC++
        Load GtAoExampleC.dsp, .vcproj or .mak, depends on
        the VC++ version from the Project\File/Open... menu
        Select Project/Rebuild all from the menu

    2. Borland C++ Builder
        Load GtAoExampleC.bpr from the Project/Open
            Project... menu
        Select Project/Build all from the menu

*********************************************************/
#include "windows.h"
#include "GtAo.h"
#include "stdio.h"

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

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

//*********************************************************
//      DisplayUsage
//*********************************************************
void DisplayUsage(void)
{
    DisplayMsg(
        "\r\nThis example shows how to set the GT1648 
voltage:\r\n\r\n"

        "Usage:\r\n"
        "GtAoExample <baseaddress> <operation> <group> 
                     <channel|range> <voltage>"
        "\r\n\r\nWhere : \r\n"
        "<baseaddress> - board base address as set by the on-
                            borad DIP switch (SW1)\r\n"
        "<operation> - one of the followings :\r\n"
        "\tSCV=Set channel voltage\r\n"
        "\tGCV=Get channel value\r\n"
        "\tSVR=Set group voltage range\r\n"
        "\tGVR=Get group voltage range\r\n"
        "<group> - group number 0 to 3\r\n"
        "<channel|range> - depends on the operation:\r\n"
        "\tchannel number :\t0-15 for SCV/GCV operations\r\n"
        "\trange number :\t0-3 for SVR/GVR operations\r\n"
        "<voltage> - channel's voltage\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\r\n"
        );
    exit(1);
}
```

```
//*********************************************************
//      CheckStatus
//*********************************************************
void CheckStatus(SHORT nStatus)
{
    CHAR   sz[128];

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

//*********************************************************
//      MAIN
//
// This main function receives five parameters
//
// GT1648 board base address (e.g. 0x310)
// GT1648 operation (e.g. GCV=get channel voltage)
//      SCV=set channel voltage,
//      GCV=get channel voltage
//      SVR=set group voltage range
//      GVR=get group voltage range
// GT1648 channel/range depends on operation
//          0 - 15 for channel
//          0 - 3 for voltage range
// GT1648 voltage value double
//

//*********************************************************
int main(int argc, char **argv)
{
    short   nBaseaddress;      // Board Base Address
    char*   sOperation;        // Board Operation
    short   nGroup;                    // group number
    short   nChannel_Range;    // channel or voltage range
    short   nHandle;           // Board handle
    short   nStatus;           // Returned status
    double  dVoltage;          // channel's voltage
```

```
// Check number of arguments recived
if (argc<4) DisplayUsage();

// Parse command line parameters
nBaseaddress=(SHORT)strtol(*(++argv), NULL, 0);
sOperation = strupr(*(++argv));
nGroup=(SHORT)strtol(*(++argv), NULL, 0);
nChannel_Range =(SHORT)strtol(*(++argv), NULL, 0);

if (nBaseaddress<0x100|| nGroup<0 || nGroup>3)
    DisplayUsage();

AoInitialize(nBaseaddress, &nHandle, &nStatus);
CheckStatus(nStatus);

if(!strcmp(sOperation, "SCV"))
{   // set channel voltage
    if (argc<5) DisplayUsage();
    dVoltage=strtod(*(++argv), NULL);
    AoSetChannelVoltage(nHandle, nGroup, nChannel_Range,
        dVoltage, &nStatus);
    CheckStatus(nStatus);
    printf("Set Group %i Channel %i voltage to %f.\n",
        nGroup, nChannel_Range, dVoltage);
}
else if(!strcmp(sOperation, "GCV"))
{   // get channel voltage
    AoGetChannelVoltage(nHandle, nGroup, nChannel_Range,
        &dVoltage, &nStatus);
    CheckStatus(nStatus);
    printf("Group %i Channel %i voltage is %f.\n", nGroup,
        nChannel_Range, dVoltage);
}
else if(!strcmp(sOperation, "SVR"))
{   // set group voltage range
    AoSetGroupVoltageRange(nHandle, nGroup, nChannel_Range,
        &nStatus);
    CheckStatus(nStatus);
    printf("Group %i voltage range to %i.\n", nGroup,
        nChannel_Range);
}
else if(!strcmp(sOperation, "GVR"))
{   // set group voltage range
```

```
        AoGetGroupVoltageRange(nHandle, nGroup, &nChannel_Range,
            &nStatus);
        CheckStatus(nStatus);
        printf("Group %i voltage range is %i.\n", nGroup,
            nChannel_Range);
    }

    return 0;
}

//***********************************************************
//      End Of File
//***********************************************************
```

Chapter 5 - Functions Reference

Introduction

The GTAO driver functions reference chapter is organized in 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 sections.

All function parameters syntax follows 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 **GtAoInitialize** 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 less than a zero on error. The description of the error is available using the **GtAoGetErrorString** function or by using a predefined constant, defined in the driver interface files: GTAO.H, GTAO.BAS, GTAO.VB, GTAO.PAS or GT1648.DRV.
- Group numbers are specified as *nGroup* with 0 to 3 for groups A to D. However, you may only have group A to C installed.

- 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 5-1: Parameter Name Prefixes

GTAO Functions

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

Driver Functions	**Description**
General	
GtAoInitialize	Initializes the GTAO driver for the specified base address.
GtAoPanel	Opens a virtual panel used to interactively control the GTAO.
GtAoReset	Resets the GTAO board to its default settings.
GtAoGetDriverSummary	Returns the driver name and version.
GtAoGetErrorString	Returns the error string associated with the specified error number.
Functions	
GtAoGetBoardAccuracy	Returns the board per channel accuracy.
GtAoGetBoardSummary	Returns the board summary from the on-board EEPROM.
GtAoGetChannelVoltage	Returns the specified channel's group voltage.
GtAoGetGroupExternalUpdate	Returns the specified group external update enable state.
GtAoGetGroupUpdateState	Returns the specified group update state.
GtGtAoGetGroupVoltageRange	Return the specified group voltage range.
GtAoGetPort	Return the Digital I/O lines value.
GtAoGetPortBit	Return the specified Digital I/O bit value.
GtAoGetPortDirection	Return the Digital I/O lines value.
GtAoLoadChannelVoltage	Load the specified groups' channel with new voltage value.

Driver Functions	**Description**
GtAoResetGroup	Reset the specified group to the default state.
GtAoSetGroupExternalUpdate	Sets the specified group external update enable state.
GtAoSetChannelVoltage	Sets the specified channel voltage.
GtAoSetGroupVoltageRange	Sets the specified group voltage range.
GtAoSetPort	Sets the Digital I/O lines value.
GtAoSetPortBit	Sets the specified Digital I/O bit value.
GtAoSetPortDirection	Sets the Digital I/O lines direction.
GtAoUpdateAllGroupsVoltage	Update all groups' channels outputs.
GtAoUpdateGroupVoltage	Update the specified groups with their loaded voltages.

GtAoGetBoardAccuracy

Purpose

Returns the board minimum per channel resolution.

Syntax

GtAoGetBoardAccuracy (*nHandle*, *pnAccuracy*, *pnStatus*)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle to a GT1648 board.
pnAccuracy	PSHORT	Returns the board Accuracy, values are: 0. 2mV 1. 5mV
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Board accuracy depends on the board factory installed output amplifier type.

Example

The following example returns the board accuracy:

```
SHORT nStatus, nAccuracy;

GtAoGetBoardAccuracy (nHandle, &nAccuracy, &nStatus);
```

See Also

GtAoGetBoardSummary, GtAoGetDriverSummary, GtAoGetErrorString

GtAoGetBoardSummary

Purpose

Returns the board name, description, S/N, firmware version and revision, and calibration time.

Syntax

GtAoGetBoardSummary(*nHandle, pszSummary, nSummaryMaxLen, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
pszSummary	PSTR	Buffer to contain the returned board info (null terminated) string.
nSummaryMaxLen	SHORT	Size of the buffer to contain the error string.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The GT1648 summary string provides the following data from the on-board nonvolatile electrically erasable programmable read-only memory (EEPROM) in the order shown:

- Instrument Name (e.g., GT1648)
- Instrument Description (e.g. 48 Channels Analog Output)
- EEPROM format version (e.g., 1.00)
- PCB revision (e.g., 'A')
- FPGA version (e.g. 0xA003)
- Serial Number (e.g., 1648008)
- Calibration time and date

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

```
“GT1648 48 Channels Analog Outputs, EEPROM-Version:1,
Board-Revision:A, FPGA-Version:0x0A03, S/N 1648002,
Calibration-time:Wed Aug 14 17:30:04 2002”
```

Example

The following example returns the board summary:

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

GtAoGetBoardSummary(nHandle, szSummary, sizeof(szSummary),
                    &nStatus)
```

See Also

GtAoGetDriverSummary, GtAoInitialize, GtAoGetErrorString

GtAoGetChannelVoltage

Purpose

Returns the specified channel's group voltage.

Syntax

GtAoGetChannelVoltage (*nHandle, nGroup, nChannel, pdVoltage,*

pnStatus)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Group number 0-3.
nChannel	SHORT	Channel number 0-15.
pdVoltage	PDOUBLE	Returned channel voltage
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the voltage for group 0 channel two:

```
SHORT nStatus;
DOUBLE dVoltage;

GtAoGetChannelVoltage (nHandle, 0, 2, &dVoltage, &nStatus);
```

See Also

GtAoSetChannelVoltage, GtAoSetGroupVoltageRange, GtAoGetGroupVoltageRange, GtAoGetErrorString

GtAoGetDriverSummary

Purpose

Returns the driver name and version.

Syntax

GtAoGetDriverSummary (*pszSummary ,nSummaryMaxLen, pdwVersion, pnStatus*)

Parameters

Name	**Type**	**Comments**
pszSummary	PSTR	Buffer to the returned driver summary string.
nSummaryMaxLen	SHORT	The size of the summary string buffer.
pdwVersion	PDWORD	Returned version number. The high order word specifies the major version number where the low order word specifies the minor version number.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The returned string is:

```
"GTAO Driver for GT1648. Version 1.0, Copyright © Geotest
2002"
```

Example

The following example prints the driver version:

```
CHAR    sz[128];
DWORD   dwVersion;
SHORT   nStatus;
GtAoGetDriverSummary (sz, sizeof sz, &dwVersion, &nStatus);
printf("Driver Version %d.%d", (INT)(dwVersion>>16), (INT)
       dwVersion &0xFFFF);
```

See Also

GtAoGetErrorString

GtAoGetErrorString

Purpose

Returns the error string associated with the specified error number.

Syntax

GtAoGetErrorString (*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

Parameter Errors

-40	Invalid group number, allowed range is 0-3
-41	Invalid channel number, allowed range is 0-15
-42	Invalid range voltage range, allowed range is 0-3
-43	Unable to set I/O port since port direction is set to input
-44	Invalid port bit number, allowed range is 0-3
-45	Voltage is out of the allowed range
-46	Channel is busy, return on timeout
-47	Unable to lock channel or group for voltage settings

Example

The following example initializes the board. If the initialization failed, the following error string is printed:

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

GtAoInitialize (3, &Handle, &Status);
if (nStatus<0)
{   GtAoGetErrorString(nStatus, sz, sizeof sz, &nStatus);
    printf(sz); // prints the error string returns
}
```

GtAoGetGroupExternalUpdate

Purpose

Returns the specified group external update enable state.

Syntax

GtAoGetGroupExternalUpdate (*nHandle, nGroup, pbEnabled, pnStatus*)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Specified group number 0-3.
pbEnabled	PBOOL	Return TRUE if the group external update is enabled and FALSE if disabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

When enabled a low on the group's external update pin will continuously update the group voltage causing its DACs to be loaded from the channel register as set by the **GtAoSetChannelVoltage**.

Example

The following example returns group zero external update enable state:

```
SHORT nStatus
BOOL bEnabled;

GtAoGetChannelVoltage (nHandle, 0, &bEnabled, &nStatus);
```

See Also

GtAoSetGroupExternalUpdate, GtAoSetChannelVoltage, GtAoSetGroupVoltageRange, GtAoGetGroupVoltageRange, GtAoGetErrorString

GtAoGetGroupUpdateState

Purpose

Returns whether the group channels loaded values were loaded to it DACs.

Syntax

GtAoGetGroupUpdateState (*nHandle, nGroup, pbUpdated, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Specified group number 0-3.
pbUpdated	PBOOL	Returns low if the specified group was not updated, high if it was updated
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Loading new value to any or all the specified group channels using **GtAoSetChannelVoltage** will set *pbUpdated* to FALSE. Updating the specified group channels will set load the value to the board DAC and will cause *pbUpdated* to return TRUE.

Example

The following example returns group zero external update enable state:

```
SHORT nStatus;
BOOL pbnUpdated;

GtAoGetGroupUpdateState (nHandle, 0, &bUpdated, &nStatus);
```

See Also

GtAoGetGroupExternalUpdate, **GtAoSetChannelVoltage, GtAoLoadChannelVoltage, GtAoUpdateGroupVoltage, GtAoGetErrorString**

GtAoGetGroupVoltageRange

Purpose

Return the specified group voltage range.

Syntax

GtAoGetGroupVoltageRange (*nHandle, nGroup, pnVoltageRange, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Specified group number 0-3.
pnVoltageRange	PSHORT	Return the specified group voltage range: 0. 0 to 10V (default) 1. -10 to 10V 2. 0 to 5V 3. -5 to -5V
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The voltage resolutions of all the channels in the specified group will increase/decrease as follow:

Range	**Resolution**
0 to 10V	2.44mV
-10 to 10V	4.88 mV
0 to 5V	1.22 mV
-5 to –5V	2.44mV

Any call to **GtAoSetGroupVoltageRange** resets all groups ’channels to zero volts.

Example

The following example returns group zero voltage range:

```
SHORT nStatus, nVoltageRange;

GtAoGetGroupVoltageRange (nHandle, 0, &nVoltageRange, &nStatus);
```

See Also

GtAoSetGroupVoltageRange, GtAoLoadChannelVoltage, GtAoUpdateGroupVoltage, GtAoGetErrorString

GtAoGetPort

Purpose

Return the Digital I/O lines value.

Syntax

GtAoGetPort (*nHandle, pucData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
pucData	PBYTE	Digital I/O lines value
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The four Digital I/O lines are TTL lines arranged as follow:

Bit #	**I/O port lines**
0	I/O 0
1	I/O 1
2	I/O 2
3	I/O 3

The Digital I/O lines can only be set when the port direction is output.

Example

The following example returns the I/O port lines value:

```
SHORT nStatus;
BYTE ucData;

GtAoGetPort (nHandle, &ucData, &nStatus);
```

See Also

GtAoSetPort, GtAoSetPortBit, GtAoGetPortBit, GtAoSetPortDirection, GtAoGetPortDirection, GtAoGetErrorString

GtAoGetPortBit

Purpose

Return the specified Digital I/O bit state.

Syntax

GtAoGetPortBit (*nHandle*, *nBit*, *pbHi*, *pnStatus*)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle to a GT1648 board.
nBit	SHORT	Digital I/O bit number 0-3.
pbHi	PBOOL	Digital I/O bit lines state: TRUE for high, FALSE for low.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The four Digital I/O lines are TTL lines arranged as follow:

Bit #	**I/O port lines**
0	I/O 0
1	I/O 1
2	I/O 2
3	I/O 3

The Digital I/O lines can only be set when the port direction is output.

Example

The following example returns the Digital I/O line number two state:

```
SHORT nStatus;
BOOL bHi;

GtAoGetPortBit (nHandle, 2, &bHi, &nStatus);
```

See Also

GtAoSetPortBit, GtAoSetPort, GtAoGetPort, GtAoSetPortDirection, GtAoGetPortDirection, GtAoGetErrorString

GtAoGetPortDirection

Purpose

Return the four I/O port lines value.

Syntax

GtAoGetPortDirection (*nHandle, nGroup, pnDirection, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Specified group number 0-3.
pnDirection	PSHORT	Return the I/O port lines direction 0. Input (default) 1. Output
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the I/O port lines direction:

```
SHORT nStatus, nDirection;

GtAoGetPortDirection (nHandle, & nDirection;, &nStatus);
```

See Also

GtAoSetPort, GtAoGetPort, GtAoSetPortDirection, GtAoGetErrorString

GtAoInitialize

Purpose

Initializes the driver for the specified board base address.

Syntax

GtAoInitialize (*nBaseaddress, pnHandle, pnStatus*)

Parameters

Name	Type	Comments
nBaseaddress	SHORT	Board base address.
pnHandle	PSHORT	Returned Handle for a GT1648 board.
pnStatus	PSHORT	Returned status: 0 on success, 1 on failure.

Comments

The function returns a handle that can be used with other GT1648 functions to program the board. The function does not change any of the board settings.

The function verifies that a GT1648 board exists in the given base address.

Example

The following example initializes a GT1648 boards at base address 0x310.

```
SHORT nHandle, nStatus;

GtAoInitilize(0x310, &nHandle, &nStatus);
if (nHandle==0)
{
   printf("Unable to Initialize the board")
   return;
}
```

See Also

GtAoGetErrorString, GtAoReset

GtAoLoadChannelVoltage

Purpose

Load the specified groups' channel with new voltage value.

Syntax

GtAoLoadChannelVoltage (*nHandle, nGroup, nChannel, dVoltage, pnStatus*)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Group number 0-3.
nChannel	SHORT	Channel number 0-15.
dVoltage	DOUBLE	Channel new voltage value.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Loading a channel with a new value does not affect the channels output. The channel's DAC will be programmed with the new value only after calling **GtAoUpdateGroupVoltage**. Calling **GtAoSetChannelVoltage** will only update all the channels in the specified group with their current loaded values. This function allows the user to load different values to different channels/groups and update all the channels at once.

Calling this function will set **GtAoGetGroupUpdateState** to high (i.e. new value is waiting for the group to be updated).

If the new voltage is out of range the function will return an error.

Example

The following example load channel five group 1 with 5.8 volts:

```
SHORT nStatus, nDirection;
GtAoLoadChannelVoltage (nHandle, 1, 5, 5.8, &nStatus);
```

See Also

GtAoSetChannelVoltage, GtAoGetChannelVoltage, GtAoGetGroupUpdateState, GtAoUpdateGroupVoltage, GtAoGetErrorString

GtAoPanel

Purpose

Opens a virtual panel used to interactively control the GT1648.

Syntax

GtAoPanel (p*nHandle, hwndParent, nMode, phwndPanel, pnStatus*)

Parameters

Name	**Type**	**Comments**
pnHandle	PSHORT	Handle to a GT1648 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	PHWND	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 was closed by the user. 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 GT1648 functions.

If a modeless dialog was created (*nMode*=0), the function returns immediately after creating the panel window returning the window handle to the panel - *phwndPanel*. It is the responsibility of the 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;
GtAoPanel(&nHandle, 0, 1, &dwPanel, &nStatus);
```

See Also

GtAoInitialize, GtAoGetErrorString

GtAoReset

Purpose

Resets the GT1648 board to its default settings.

Syntax

GtAoReset (*nHandle, pnStatus*)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle for a GT1648 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function does the following:

- All groups' voltage ranges are set to 0-10V.
- All channels of all groups are set to 0V.
- All external updates are disabled
- I/O port values are set to zero
- I/O port direction set to input

Example

The following example initializes and resets the GT1648 board:

```
GtAoInitialize (1, &nHandle, &nStatus);
GtAoReset (nHandle, &nStatus);
```

See Also

GtAoInitialize, GtAoGetErrorString

GtAoResetGroup

Purpose

Resets the specified group to the default state.

Syntax

GtAoResetGroup (*nHandle, nGroup, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Group number 0-3.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The specified groups will be reset to:

- Group’s channels set to 0V.
- Group’s voltage range is set to 0-10V.
- External update is disabled.

Example

The following example resets group 2:

```
SHORT nStatus;

GtAoResetGroup(nHandle, 2, &nStatus);
```

See Also

GtAoSetChannelVoltage, GtAoGetChannelVoltage, GtAoSetGroupVoltageRange, GtAoGetGroupVoltageRange, GtAoGetErrorString

GtAoSetChannelVoltage

Purpose

Sets the specified channel's group voltage.

Syntax

GtAoSetChannelVoltage (*nHandle, nGroup, nChannel, dVoltage, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Group number 0-3.
nChannel	SHORT	Channel number 0-15.
dVoltage	DOUBLE	Channel's voltage
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The voltage must be in the group's voltage range otherwise the function returns an error.

Calling this function will set **GtAoGetGroupUpdateState** to low (i.e. the group was updated).

Example

The following example sets the voltage for group 0 channel two to 2.34V:

```
SHORT nStatus;
DOUBLE dVoltage;

GtAoGetChannelVoltage (nHandle, 0, 2, &dVoltage, &nStatus);
```

See Also

GtAoGetChannelVoltage, GtAoSetGroupVoltageRange, GtAoGetGroupVoltageRange, GtAoGetErrorString

GtAoSetGroupExternalUpdate

Purpose

Sets the specified group external update enable state.

Syntax

GtAoSetGroupExternalUpdate (*nHandle, nGroup, bEnabled, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Group number 0-3.
bEnabled	BOOL	A high enable the group external update line, a low disable the external update line.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

When enabled a low on the group's external update pin will continuously update the group voltage.

Example

The following example enable group zero external update:

```
SHORT nStatus;

GtAoSetChannelVoltage (nHandle, 0, TRUE, &nStatus);
```

See Also

GtAoGetGroupExternalUpdate, GtAoSetChannelVoltage, GtAoSetGroupVoltageRange, GtAoGetGroupVoltageRange, GtAoGetErrorString

GtAoSetGroupVoltageRange

Purpose

Sets the specified group voltage range.

Syntax

GtAoSetGroupVoltageRange (*nHandle, nGroup, nVoltageRange, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Group number 0-3.
nVoltageRange	SHORT	Set the specified group voltage range: 0. 0 to 10V (default) 1. -10 to 10V 2. 0 to 5V 3. -5 to -5V
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The voltage resolutions of all the channels in the specified group will increase/decrease as follow:

Range	**Resolution**
0 to 10V	2.44mV
-10 to 10V	4.88 mV
0 to 5V	1.22 mV
-5 to –5V	2.44mV

Any call to "**GtAoSetGroupVoltageRange**" resets all groups 'channels to zero volts.

Example

The following example set group zero voltage range to –5 to +5V:

```
SHORT nStatus;

GtAoSetGroupVoltageRange (nHandle, 0, 3, &nStatus);
```

See Also

GtAoGetGroupVoltageRange, GtAoLoadChannelVoltage, GtAoUpdateGroupVoltage, GtAoGetErrorString

GtAoSetPort

Purpose

Sets the four Digital I/O lines value.

Syntax

GtAoGetPort (*nHandle, nGroup, ucData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
ucData	BYTE	I/O port lines data (0-15).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The four Digital I/O lines are TTL lines arranged as follow:

Bit #	I/O port lines
0	I/O 0
1	I/O 1
2	I/O 2
3	I/O 3

The Digital I/O lines can be set when the port direction is output.

Example

The following example sets the Digital I/O port lines to 0xA:

```
SHORT nStatus;

GtAoGetPort (nHandle, 0xA, &nStatus);
```

See Also

GtAoGetPort, GtAoSetPortDirection, GtAoGetPortDirection, GtAoGetErrorString

GtAoSetPortBit

Purpose

Sets the specified Digital I/O bit state.

Syntax

GtAoGetPortBit (*nHandle, nBit, bHi, pnStatus*)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle to a GT1648 board.
nBit	SHORT	Digital I/O bit number 0-3.
bHi	BOOL	Digital I/O bit lines state one for high, 0 for low.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The four Digital I/O lines are TTL lines arranged as follow:

Bit #	**I/O port lines**
0	I/O 0
1	I/O 1
2	I/O 2
3	I/O 3

The Digital I/O lines can only be set when the port direction is output.

Example

The following example sets Digital I/O line two to high:

```
SHORT nStatus;
BOOL bHi;

GtAoSetPortBit (nHandle, 2, TRUE, &nStatus);
```

See Also

GtAoGetPortBit, GtAoSetPort, GtAoGetPort, GtAoSetPortDirection, GtAoGetPortDirection, GtAoGetErrorString

GtAoSetPortDirection

Purpose

Sets the TTL I/O port lines direction.

Syntax

GtAoSetPortDirection (*nHandle, pnDirection, pnStatus*)

Parameters

Name	**Type**	**Comments**
nHandle	SHORT	Handle to a GT1648 board.
nDirection	SHORT	I/O port lines direction as follow: 0. Input (default) 1. Output
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example sets the I/O port lines direction to output:

```
SHORT nStatus;

GtAoSetPortDirection (nHandle, 1, &nStatus);
```

See Also

GtAoSetPort, GtAoGetPort, GtAoGetPortDirection, GtAoGetErrorString

GtAoUpdateAllGroupsVoltage

Purpose

Update all groups' channels outputs.

Syntax

GtAoUpdateAllGroupsVoltage (*nHandle*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example updates all the groups:

```
SHORT nStatus;

GtAoUpdateAllGroupsVoltage (nHandle, &nStatus);
```

See Also

GtAoUpdateAllGroupsVoltage, **GtAoSetGroupVoltageRange**, **GtAoGetGroupVoltageRange**, **GtAoLoadChannelVoltage**, **GtAoGetErrorString**

GtAoUpdateGroupVoltage

Purpose

Update all the specified group channels outputs.

Syntax

GtAoUpdateGroupVoltage (*nHandle, nGroup, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1648 board.
nGroup	SHORT	Group number 0-3.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function update all the specified group channels outputs with the currently loaded values.

Example

The following example updates group 1:

```
SHORT nStatus;

GtAoUpdateGroupVoltage (nHandle, 1, &nStatus);
```

See Also

GtAoSetGroupVoltageRange, GtAoGetGroupVoltageRange, GtAoLoadChannelVoltage, GtAoGetErrorString

Index

.

.NET ii

A

Accessories 17

Architecture 1, 6

ATEasy .... ii, 21, 22, 23, 27, 29

B

Board Description 1, 4

Board Handle 30

Borland .... ii, 22, 27, 28

Borland-Delphi 28

C

C/C++ 22, 23, 28

C++ 27

Connector 17

Connectors 4, 17

Connectors and Accessories 17

Corrupt files 25

D

Delphi .... ii, 22, 27, 28

Directories 21

Distributing 31

Driver

  Directory 21

  Files 21

E

Error-Handling 30

Example 23

F

Functions

  Boards-Supported 30

Functions Reference 39

G

GT1648 .... 1, 5, 13

GTAO 13, 19

  Driver-Description 27

  Header-file 28

  Help-File-Description 22

  Panel-File-Description 22

  Supported-Development-Tools 27

GTAO Functions 41

GTAO.BAS 22, 28

GTAO.DLL 22, 27, 29

GTAO.H 22, 28

GTAO.LIB 22, 28

GTAO.PAS 22, 28

GTAO.VB 22

GTAOBC.LIB 28

GTAOBC.LIB 22

GtAoGetBoardAccuracy 43

GtAoGetBoardSummary 44

GtAoGetChannelVoltage 46

GtAoGetDriverSummary 31, 47

GtAoGetErrorString.....30, 39, 48
GtAoGetGroupExternalUpdate 50
GtAoGetGroupUpdateState 51
GtAoGetGroupVoltageRange ..52
GtAoGetPort 54
GtAoGetPortBit 55
GtAoGetPortDirection 56
GtAoInitialize...............30, 39, 57
GtAoLoadChannelVoltage.......58
GtAoPanel..........................31, 59
GTAOPANEL.EXE...........22, 31
GtAoReset................................61
GtAoResetGroup......................62
GtAoSetChannelVoltage..........63
GtAoSetGroupExternalUpdate.64
GtAoSetGroupVoltageRange...65
GtAoSetPort.............................67
GtAoSetPortBit........................68
GtAoSetPortDirection..............69
GtAoUpdateAllGroupsVoltage 70
GtAoUpdateGroupVoltage.......71

H

Handle......................................30
HW.........................20, 21, 27, 31

I

I/O Base Address Chart............15
Installation Directories.............21

J

J3...4, 18

L

Listing...................................... 33

N

nHandle 29, 30

O

OnError.................................... 29
Output Channels Connector..... 18

P

Panel .8, 10, 11, 12, 22, 25, 31, 59
  About Page 12
  Digital I/O............................ 11
  Group A/B/C Page............... 10
Pascal........................... 22, 27, 28
PCI... 21
pnStatus 30, 39
Program-File-Descriptions 22
Programming
  Borland-Delphi..................... 28
  Error-Handling..................... 30
  Panel-Program 31

R

README.TXT 21, 22, 23
Readme-File 23
Reset.. 30

S

Sample 32, 33
Setup.......... 13, 19, 20, 21, 22, 25
Setup Maintenance 25
Specifications 1, 7

Switch Settings 14

System

Directory...............................21

V

Virtual Panel8, 10, 11, 12, 20, 22, 30, 59

Visual Basic .... ii, 28

Visual C++.............. ii, 22, 27, 28