GT1034
Precision Reference Board

User's Guide

GEOTEST – Marvin Test Systems, Inc.

Part No. GT94108 - Revision 1 (1/4/99)

Safety and Handling

Each product shipped by Geotest is carefully inspected and tested prior to shipping. The shipping carton provides protection for shipment and can be used to store both hardware and software.

The circuit board is extremely delicate and requires care in handling and installation. Do not remove the board from its protective shipping carton or plastic covering until you are ready to install it.

If the board is removed from the computer for any reason, be sure to store it in its original shipping carton. Do not store the board on a workbench or anywhere it can be dropped or subjected to strong electromagnetic or electrostatic fields. It is good practice to store the board in a protective, anti-electrostatic wrapper away from electromagnetic fields.

Make a single backup copy of the software diskette and store the original in a place safe from heat or electromagnetic or electrostatic fields.

Warranty

Geotest's products are warranted against defects in materials and workmanship for a period of 12 months (6 months for software products). Geotest shall repair or replace (at Geotest's discretion) any defective product during the stated warranty period. The software warranty includes any upgrades 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

If you need help at any time with the installation or use of this product, call Geotest technical support at 949-263-2222. In addition, you may get updated product information and drivers from our Internet Web Site at www.geotestinc.com.

Disclaimer

In no event shall Geotest or any of its representatives be liable for any consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other loss) arising out of the use of or inability to use this product, even if Geotest has been advised of the possibility for such damages.

Copyright

Trademarks

386, 486, Pentium	Intel Corporation
ATEasy	Geotest
Borland C++	Borland Corporation
Delphi	Borland Corporation
IBM or IBM-PC/AT	International Business Machines Corp. (IBM).
Microsoft Developer Studio	Microsoft Corporation
Microsoft Visual Basic	Microsoft Corporation
Microsoft Visual C++	Microsoft Corporation
MS-DOS	Microsoft Corporation
Windows 3.1, Windows 95, Windows 98 and Windows NT	Microsoft Corporation

Table of Contents

Chapter 1 - Introduction

About This Manual

This GT1034 User's Guide provides all the information needed to install, program, and use Geotest's GT1034 board. This manual assumes that the user has a general knowledge of PC-based computers, Windows 3.1/95/98/NT operating systems and an electronics background. Some knowledge of programming and development tools will permit computer program control of the board.

Scope and Organization

The manual is organized as follows:

Chapter	Content
Chapter 1	Introduces this GT1034 User's Guide.
Chapter 2	Summarizes the board's features, architecture, hardware and software.
Chapter 3	Furnishes step-by-step directions on how to install and setup the hardware and GTCAL software.
Chapter 4	Provides instructions for using the Virtual Panel.
Chapter 5	Presents details on how to program the board by using the supplied driver and example files.
Chapter 6	Contains an alphabetical list of driver functions including syntax, variables, programming comments and samples.
Appendix A	A reference for GTCAL driver error codes.
Appendix B	Summarizes GT1034 specifications.
Index	A roadmap to important topics and concepts in this manual.

Conventions Used in This Manual

There are several naming conventions used throughout this manual. The conventions used are:

Example	Description
Copy or **Paste**	Commands are indicated in bold type.
Shift+F1	Keys are often used in combinations. The example to the left instructs the user to hold down the shift key while pressing the FI key at the same time. When key combination instructions are separated by commas, such as ALT+D, A, hold the ALT key while pressing D, the press A.
Direction Keys	Refer to the up arrow (↑), down arrow (↓), right arrow (→), and left arrow (←) keys.
cd bold	Bolded text must be entered from the keyboard exactly as shown.
cd *directory name*	Italicized text is a placeholder for variables or other items that the user must define and enter from the keyboard.
`examples`	Examples and source code are indicated in Courier, a fixed pitch font.
0x*hexnumber*	An integer in hexadecimal notation, E.g., 0x10A equals 266 in decimal.

Technical Support

General Information

Geotest provides both pre-sales and post-sales technical support for all products. Our Technical Support engineers can help you select hardware and software for your application, understand the specifications, and assist you in designing a complete system. Call our Technical Support department from 8:30AM to 5:30PM Pacific Standard Time (PST). Call 949-263-2222 or send email to support@geotestinc.com.

Internet Home Page Download Area

Use Geotest's Internet download area to obtain new instrument drivers, application notes, and other valuable resource information for the board. Our Internet address is: www.geotestinc.com

Calibration

Geotest offers calibration services for some measurement instrumentation. For more information, please contact Geotest Technical Support at 949-263-2222.

Geotest Address and Telephone Numbers

Geotest – Marvin Test Systems, Inc.
2851 S. Pullman Street
Santa Ana, CA 92705
Phone: 949-263-2222
Fax: 949-263-1203
Email (sales): sales@geotestinc.com
Email (support): support@geotestinc.com

Chapter 2 - Overview

Introduction

The GT1034 board verifies the function of a wide range of measurement instruments. The board provides verification by outputting a defined signal via register banks, relays and finally through a 9-pin connector to which a measurement instrument can be connected. Designed for use with Automatic Testing Equipment (ATE), the GT1034 plugs directly into any 8- or 16-bit ISA bus slot. The board can be implemented to verify functional integrity on power-up or at regular intervals, thus providing additional safety and quality assurance.

Features

The GT1034 board provides a variety of reference sources and loads such as:

- Precision resistance
- 4-Wire from 1 to 10Ω, with a 0.005% tolerance.
- 2-Wire from 100Ω to 1 MΩ, with a 0.005% tolerance.
- Optional 2-Wire from 10 MΩ to 100 MΩ, with a 0.01% tolerance. (This option requires the industrial version of GT1034. Contact Geotest Sales for details.)
- Seven precision resistor banks, with 0.005% tolerance (0.01% above 10 MΩ).
- DC Voltage from 9 mV to 10 V, with an accuracy of 5µV±0.015%.
- AC Voltages from 4.5 mV to 5 V RMS, with an accuracy of 5µV±0.05%. (Includes a 5 V sine wave).
- Frequency references from 1 kHz to 10 MHz, with an accuracy of 0.01%.

Board Diagram

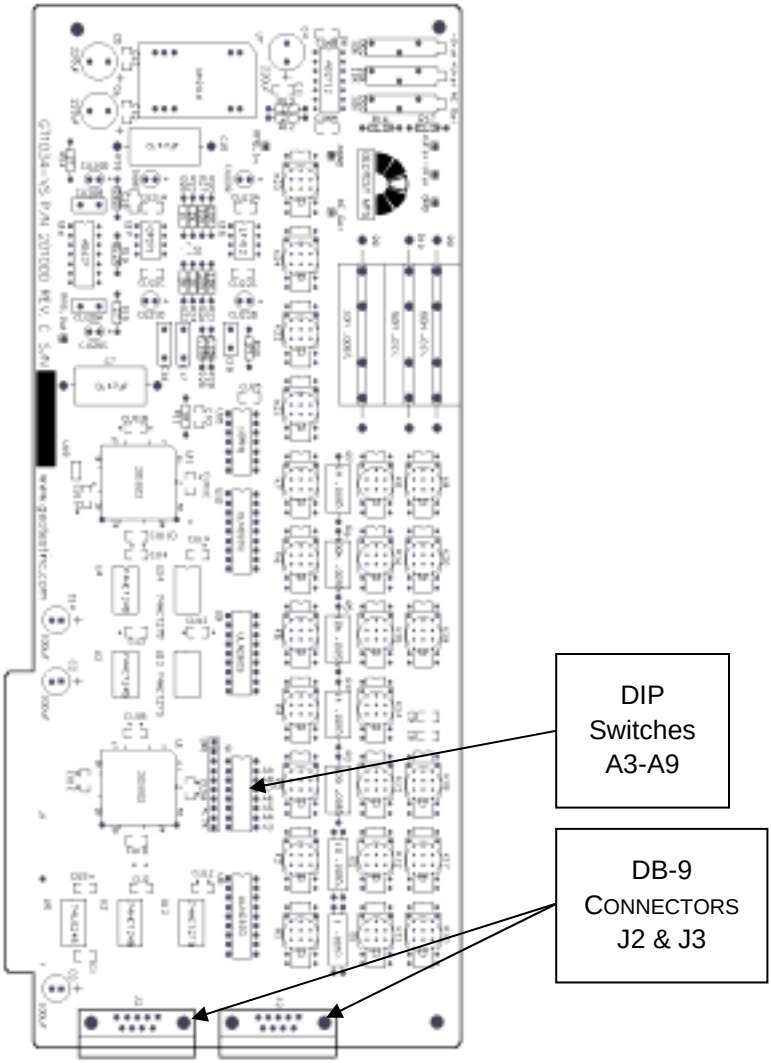

Figure 2-1: GT1034 Circuit Board

Architecture

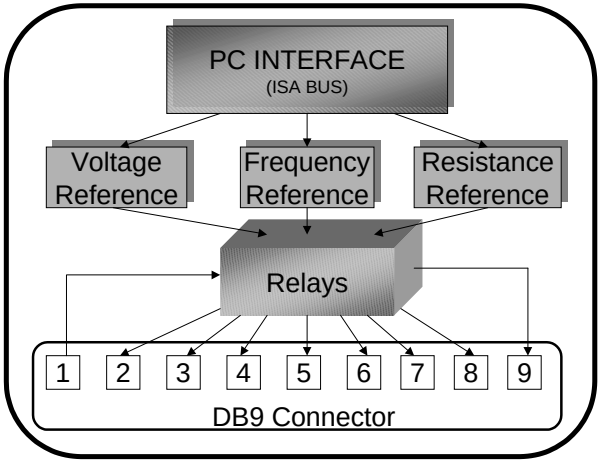

Figure 2-2: GT1034 Architecture

Caveat

This manual quotes certain nominal values and tolerances for resistance and voltage. Although the GT1034 is built to high quality standards, it is *not* intended to replace either a primary or secondary calibration standard.

Tolerances are guaranteed only at the time of manufacture and are covered by our standard warranty.

All tolerances are quoted at standard temperature (20^{o} C), pressure (1 atmosphere) and 50% humidity. Deterioration of components leading to performance degradation can occur due to normal aging, handling, pollution, extreme temperatures, radiation and certain other factors.

Note: If highly accurate measurements are required, calibration of the GT1034 against a traceable standard at the time and under the conditions of test is essential.

Connector

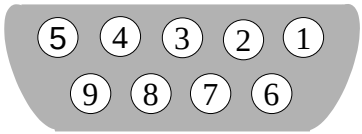

Figure 2-3: DB-9 Connector Pin Number Layout

Connections to the measurement system are made through a standard 9-pin, D-type male (DB-9) connector. As shown in the Board Diagram on page 6, identical DB-9 connectors, J2 and J3, enable two instruments to connect to the board. Figure 2-3 shows the pin number layout of each connector.

All connections to the GT1034 board are made through P2 and P3, which are standard D-type female connectors. One male connector is provided with each board. Details on installing the GT1034 Board and software installation details are discussed in Chapter 3.

Both board connectors are connected to the same reference circuit. Care should be taken to avoid a conflict between two simultaneously connected measurement functions, as in the instance of resistance measurements.

The GT1034 provides only one signal at a time from the output terminals, with the exception of the 5V AC signal which is always available. See “Appendix B” in this manual for output specifications.

Table 2-1 below shows external connections to either J2 or J3.

Pin #	Signal	Purpose
1	GND	Ground
2	Measure Lo	AC and DC voltages and resistance reference
3	Sense Lo	4-wire connections
4	Ext Ref Lo	External reference sources

Pin #	Signal	Purpose
5	AC Ref Lo	AC voltage (5V)
6	Measure Hi	AC and DC voltages and resistance reference
7	Sense Hi	4-wire connections
8	Ext Ref Hi	External reference sources
9	AC Ref Hi	AC voltage (5V)

Table 2.1: Connector Pinout

Note: A 5-volt sine wave is always available at pins 5 and 9.

The function and corresponding signal can be designated in one of two ways:

- Use the Virtual Panel that is included in the GTCAL software module (described in detail in the next section).
- Program the GT1034 board by using the DLL drivers that are supplied in the GTCAL software module.

Details for programming the board and functions within DLL drivers are in Chapters Five and Six of this manual.

A measurement function is verified by connecting the measurement channel to J2 or J3. Table 2-1 shows connector pin outs used for 2- and 4-wire connection points, the external reference input and permanent AC source.

Software

The GT1034 software module, also referred to as the GTCAL module, includes the following:

- Virtual Panel
- Function Library
- Programming examples
- Windows Help file

- Installation and Setup utility

Virtual Panel

The GTCAL driver includes a virtual panel program that enables the user to fully utilize various configuration and control modes.

The Virtual Panel interactively sets up and programs the GT1034 board. This functionality is provided both as part of the DLL driver and as a stand-alone executable file.

Figure 2-4: GT1034 Virtual Panel

Function Library

This part of the module contains functions that perform the following:

- Initialization of drivers.
- Resetting of the board, or parts of it.
- Setting, controlling or retrieving board parameters.

The functions are in various formats to support different operating systems and development tools. The drivers are supplied in two formats:

- Windows 16 and 32 bit DLLs. Used for Windows(3.x, 95/NT)-based applications programming.
- DOS Static Libraries. Used for DOS-based applications programming.

Exported functions are declared in C (using Pascal calling conventions) to provide access from various development tools. Interface files and examples are provided for various development tools and programming languages, such as C/C++, VB, Pascal, and Geotest's ***ATEasy***.

Programming Example

The example in Chapter 5 demonstrates how to program the board with the supplied drivers. The example is provided in various formats to demonstrate use of different development tools and operating systems.

A complete discussion of the Virtual Panel, files, drivers, examples, and functions library are located in chapters Four, Five and Six, respectively.

Chapter 3 - Setup and Installation

Introduction

This product consists of both the board and the software module. This chapter describes how to install the GT1034 board and software module referred to as GTCAL.

Packing List

This package includes one each:

- GT1034 User's Guide
- GTCAL/GT1034 Driver Disk
- GT1034 Calibration Verification board.

System Requirements

The GT1034 board is designed to run on an IBM compatible computer running DOS, Windows 3.X, Windows 95/98, or Windows NT, with at least one free 8/16-bit ISA bus slot.

Unpacking and Inspection

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. 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 this manual.

Discharge Static Electricity

To reduce the risk of damage, observe the following precautions:

- Leave the board in the anti-static bag until you install it. 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 the board or handle it roughly.
- Handle the board by the edges. Do not touch any components on the circuit board.

Caution: Do not attempt to insert or remove any board with the power on. Turn off the power before attempting installation.

Setting the I/O Base Address

The I/O Base address (default 0x310 can be changed by resetting the I/O address according to the values set forth in Table 3-1.

The GT1034 uses two 16-bit ports in the I/O space following the base address with an additional port used at the base address + 0x400. The factory default base address is set to 0x310. The board base address can be changed by setting a DIP Switch to any address between 0x200 and 0x3FF.

I/O Address Settings

Address (hex)	SW1-10 A4	SW1-9 A5	SW1-8 A6	SW1-7 A7	SW1-6 A8	SW1-5 A9
200	ON	ON	ON	ON	OFF	ON
210	OFF	ON	ON	ON	OFF	ON
220	ON	OFF	ON	ON	OFF	ON
230	OFF	OFF	ON	ON	OFF	ON
240	ON	ON	OFF	ON	OFF	ON
250	OFF	ON	OFF	ON	OFF	ON
260	ON	OFF	OFF	ON	OFF	ON
270	OFF	OFF	OFF	ON	OFF	ON
280	ON	ON	ON	OFF	OFF	ON
290	OFF	ON	ON	OFF	OFF	ON
2A0	ON	OFF	ON	OFF	OFF	ON
2B0	OFF	OFF	ON	OFF	OFF	ON
2C0	ON	ON	OFF	OFF	OFF	ON
2D0	OFF	ON	OFF	OFF	OFF	ON
2E0	ON	OFF	OFF	OFF	OFF	ON
2F0	OFF	OFF	OFF	OFF	OFF	ON
300	ON	ON	ON	ON	OFF	OFF
310	OFF	ON	ON	ON	OFF	OFF
320	ON	OFF	ON	ON	OFF	OFF
330	OFF	OFF	ON	ON	OFF	OFF
340	ON	ON	OFF	ON	OFF	OFF
350	OFF	ON	OFF	ON	OFF	OFF
360	ON	OFF	OFF	ON	OFF	OFF
370	OFF	OFF	OFF	ON	OFF	OFF
380	ON	ON	ON	OFF	OFF	OFF
390	OFF	ON	ON	OFF	OFF	OFF

Address (hex)	SW1-10 A4	SW1-9 A5	SW1-8 A6	SW1-7 A7	SW1-6 A8	SW1-5 A9
3A0	ON	OFF	ON	OFF	OFF	OFF
3B0	OFF	OFF	ON	OFF	OFF	OFF
3C0	ON	ON	OFF	OFF	OFF	OFF
3D0	OFF	ON	OFF	OFF	OFF	OFF
3E0	ON	OFF	OFF	OFF	OFF	OFF
3F0	OFF	OFF	OFF	OFF	OFF	OFF

Table 3-1: I/O Base Address Chart

Note: The default I/O address is 310h.

Board Installation

Warning: Turn off the PC before attempting installation. Do not attempt to insert or remove any board while the computer is turned on.

Until it is ready for installation, keep the board in the anti-static bag with which it is shipped, and do not break the seal. This greatly reduces the risk of damage from static electricity.

Install the GT1034 board as follows:

1. Power off the system and remove the power cord.

2. Unfasten appropriate screws and remove the chassis.

3. Examine the rear panel of the system. Notice that each expansion slot has an opening at its end for mounting I/O connectors. Unused slots may have a metal plate covering this opening.

4. Locate an unused slot. If necessary, use a screwdriver to gently remove a metal plate that covers an unused expansion slot. Pry the lower end away and work the catilevered strip back and forth several times until it breaks off.

5. Carefully remove the GT1034 board from the anti-static bag. Save the anti-static bag for future storage if the board is later removed from the system.

Caution: Handle the GT1034 board by its edges. Do not touch any components on the circuit board.

---

6. Check the DIP switch. If a base address other than the default base address set by the manufacturer (310H) is required, refer to Table 3-1 for the proper DIP switch settings.

7. Carefully position the GT1034 board over the expansion slot and fit the connector through the rear panel opening.

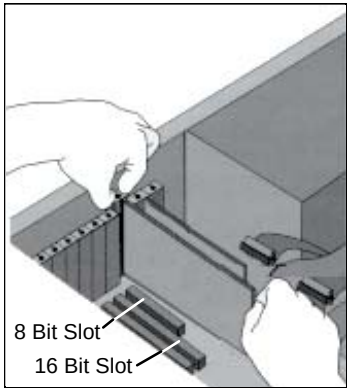

Figure 3-1. Typical Board Installation

8. Insert the board edge into the motherboard receptacle, making certain it is seated properly.

9. Use the screw from the cover plate to attach the mounting bracket to the rear panel, holding the board firmly in place.

10. Connect external cable(s) to the rear 9-pin connector(s).

11. Slide the chassis back on and re-fasten the screws.

12. Plug in the power cord and power up the system.

Software Installation

Installing the GTCAL module is straightforward and easy. The software can be installed under Windows (3.x, ’95, ’98 or NT), or under DOS. Both environments are described in the following sections.

Installation Under Windows

1. Boot the computer and log into windows as Administrator (or with administrator rights) where applicable.

2. Insert the 3.5-inch diskette into the floppy drive.

3. In Windows 95/98 or NT, click **Start** and select **Run**.

   Under Windows NT, the Setup program installs the kernel mode driver, HW.SYS. To successfully install the kernel mode driver, you must be logged on as a user with administrator privileges. Note that the kernel mode driver can also be installed manually after the Setup. Please refer to the section “Windows NT Kernel Mode Driver Installation” on page 19.

4. In Windows 3.x, select **File**, **Run** from Program Manager.

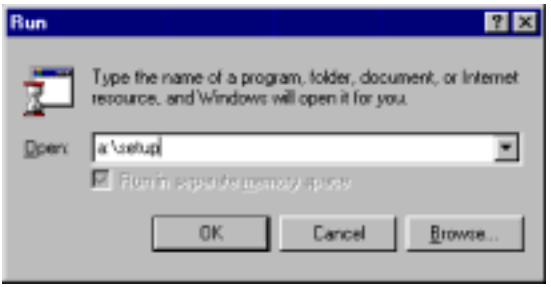

Figure 3-2. Run Window

5. In the command line field, enter A:\setup.exe.

   The Installation program copies the necessary files to the local hard disk. The default directory (folder) is C:\GTCAL.

Windows NT Kernel Mode Driver Installation

Under Windows NT, the kernel mode device driver, HW.SYS, must be installed and executed before the GTCAL driver can be used. Before installing the kernel mode driver, you must be logged on as a user with an administrator privilege.

The GTCAL Setup program normally installs the kernel mode driver and starts it automatically. If the current user is not logged-in as an administrator, the kernel mode driver installation will fail. This section explains how to install the kernel mode driver manually when the Setup program fails to do so.

To manually install the kernel mode driver, perform the following:

1. Log in as an administrator.

2. Open a Command Prompt window.

3. Change to the installation destination directory by using the CD command (e.g. **CD \GTCAL**).

4. From the command prompt, execute the following command:

   HWSETUP –vdd install start

   If the current working directory is different from the directory where the HW.SYS resides, you may specify your own custom path. For Example:

   HWSETUP –vdd install=a:\start

The Setup program installs the driver as a service. The service can be started or stopped from the Windows NT control panel Devices applet. The **–vdd** switch can be removed from the command if support for 16-bit drivers is not required (only the 32-bit DLL will be used in this case).

The Setup program HWSETUP.EXE, the device driver HW.SYS, the driver library, HWVDD.DLL, and the GTCALSW driver files may be distributed with an application for GT1034 board.

Additional HWSETUP.EXE command line options are available. To display these options, type **HWSETUP** with no command line options.

Installing Under DOS

The Setup program that is shipped with the GT1034 software module disk can only be used from Windows. To install the GTCAL module under DOS, manually copy the files in Table 3-2 to the local hard drive.

Filename	Purpose
README.1ST	Text file that includes updated information.
GTCAL16.BAS	Used with Visual Basic for DOS (with minor changes).
GTCAL16 .PAS	Used with Turbo Pascal (with minor changes).
GTCAL.H	C header file for GT1034 drivers.
GTCALBC.LIB	Borland static library for DOS applications.
GTCALMS.LIB	Microsoft static library for DOS applications.

Table 3-2. Files Installed Under DOS

Overview of GTCAL Files

The Setup program installs the GTCAL software module which includes:

- Driver Files
- Example Programs.

A complete discussion of how to use the files can be found in Chapter 5, “Programming the Board.”

Driver Files

Tables 3-3 through 3-6 describe the files included in the software.

Filename	Purpose
GTCALBC.LIB	Borland static library for DOS applications
GTCALMS.LIB	Microsoft static library for DOS applications
GTCAL16.DLL	16-bit DLL for Windows 3.x applications
CTL3DV2.DLL	Supports 3D Controls/Dialogs for GTCAL16.DLL
GTCAL16.LIB	16-bit import library for GTCAL16.DLL
GTCAL32.DLL	32-bit DLL for Windows 95/NT applications
GTCAL32.LIB	32-bit import library for GTCAL32.DLL
GTCAL32B.LIB	32-bit import library for GTCAL32B.DLL for Borland applications

Table 3-3. GTCAL Library Files

Filename	Purpose
GTCALP16.EXE	16-bit virtual panel for GTCAL boards
GTCALP32.EXE	32-bit virtual panel for GTCAL boards

Table 3-4. GTCAL Virtual Panel Executables

Filename	Purpose
HW.SYS	Windows NT kernel mode driver for the GTCAL32.DLL
HWVDD.DLL	Windows NT virtual device driver for use with the 16-bit DLL under Windows NT
HWSETUP.EXE	Setup program to install/uninstall the Windows NT kernel driver HW.SYS
HWTEST.EXE	Verification program to test the HW.SYS driver installation

Table 3-5. GTCAL NT Kernel Mode Device Drivers

Filename	Purpose
GTCAL.H	C header file for driver functions and constants
GTCAL16.BAS	Visual Basic (16 bit) file that contains a 16-bit DLL function and constant declarations
GTCAL32.BAS	Visual Basic (32 bit) file that contains a 16-bit DLL function and constant declarations
GTCAL.INS	***ATEasy*** 2.x driver for GT1034 board
GTCAL.PAS	Driver for use with Borland Pascal or Delphi

Table 3-6. GTCAL Interface Files

Example Programs

The GTCAL software module includes a C source file used for all samples and a MAK file (IDE file for Borland C++) for each of the following development tools:

- DOS Static Library (16-bit and 32-bit DLLs)
- Borland Static Library (16-bit and 32-bit DLLs)
- ***ATEasy***

The samples demonstrate how the resulting output module (executable file) uses the appropriate library and MAK file. Tables 3-7 to 3-11 describe sample files included in the software module.

File Name	Description
GTCALX32.EXE	Output module (executable)
GTCALX32.MAK	VC++/MS-C Make file
GTCALX.C	C source file
GTCALX.RC	Resource file
GTCALX.ICO	Icon file
GTCAL32.DLL	32-Bit DLL

Table 3-7. Files Used for Windows 95/NT Sample

File Name	Description
GTCALX16.EXE	Output module (executable)
GTCALX16.MAK	VC++/MS-C Make file
GTCALX.C	C source file
GTCALX.DEF	Module definition file
GTCALX.RC	Resource file
GTCALX.ICO	Icon file
GTCAL16.DLL	16-Bit DLL

Table 3-8. Files Used for Windows 3.x/95 Sample

File Name	Description
GTCALXMS.EXE	Output module (executable)
GTCALXMS.MAK	VC++/MS-C Make file
GTCALX.C	C source file (same source for all samples)
GTCALMS.LIB	DOS Static Library

Table 3-9. Files Used for DOS Sample (with Microsoft C Static Library)

File Name	Description
GTCALBC.IDE	Borland C++ 5.0 contains 3 projects, Windows 32, Windows 16 and DOS
GTCALX.C	C source file
GTCALX.DEF	Module definition file
GTCALX.RC	Resource file
GTCALX.ICO	Icon file
GTCALBC.LIB	Borland Static Library

Table 3-10. Files Used for Borland Sample

File Name	Description
GTCAL.PRG	GTCAL example program
GTCAL.CFG	GTCAL.PRG example CFG file
GTCAL.INS	Driver file for ***ATEasy*** version 2.x

Table 3-11. Files Used for *ATEasy* Sample

Figure 3-3 below shows the resulting window when the output module (with the exception of GTCALXMS.EXE) is executed.

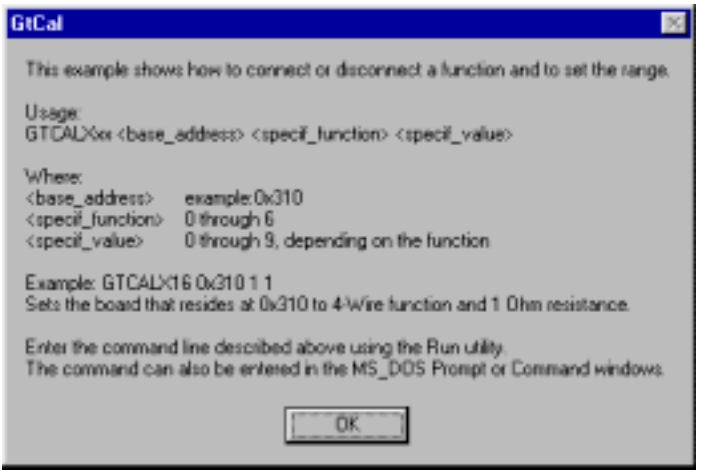

Figure 3-3. GTCAL Sample

Distributing the Software

The setup program HWSETUP.EXE, the device drivers HW.SYS and HWVDD.DLL, and the GTCAL driver files may be distributed with any application that uses the GT1034 board.

Chapter 4 - Using the Virtual Panel

Introduction

This chapter provides the basics for using the GT1034 Board and the Virtual Instrument Panel. We assume the user has carefully read "Chapter 2 – Overview," which provides important information on the theory of operation and function for both the GT1034 Board and the GTCAL software module.

Other procedures, such as programming the board and using supplied drivers and functions, are discussed in Chapters Five and Six of this manual.

Opening the Panel

Two Virtual Panel versions are available – for 16 and 32-bit operating systems.

In Windows 95/98/NT, you can run the 32-bit Virtual Panel by clicking the **Start** button on the taskbar, then highlighting **Programs**, **GTCAL**, and clicking **GTCAL Panel – 32 bit**.

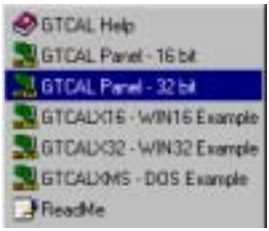

Figure 4-1: Windows 95/NT Menu via Start Button

To run the 16-bit version in Windows 3.1, click the icon below from the GTCAL group window in Program Manager.

Figure 4-2: GT1034 Icon for Windows 3.1

The 16-bit Virtual Panel will also run under Windows 95/98/NT and may be accessed using the **Start** button, selecting **Programs and GTCAL,** and clicking **GTCAL Panel – 16 bit**.

Initializing

After GTCALP16.EXE or GTCALP32.EXE is executed, the

Virtual Panel appears with all options disabled, as shown below.

Figure 4-3: GT1034 Virtual Panel with Options Disabled

Before any of the options can be selected, the GT1034 board must be initialized. To do so, click on the **Initialize…** button.

Figure 4-4: Initialize Button

When the Initialize button is clicked, the following dialog box appears:

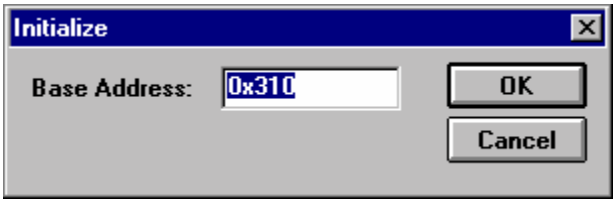

Figure 4-5: Changing the Base Address (See Caution)

Enter the base address set for this board and click **OK**. Note that initializing the board has no effect on the board setting.

Caution: The Base Address (default 310h) is set prior to installing the board. If the Base Address conflicts with other system hardware, change it. Refer to the section "I/O Address Settings" in Chapter 3 for appropriate switch settings for other I/O addresses..

Functions and Values

When the GT1034 board has been successfully initialized, the Virtual Panel appears with all functions and values enabled as indicated below:

Figure 4-6: GT1034 Virtual Panel with Options Enabled

The Virtual Panel consists of the following six radio buttons that designate the function to be used:

- 4-Wire
- 2-Wire
- VDC
- VAC
- Frequency
- External Source

Tip: A 4-wire connection is recommended for resistor values below 10 KΩ. If no 4-wire connection is available on the instrument side, then the 0Ω setting can be used to determine the load resistance for load compensation.

For each function, there is a corresponding range that is displayed in a listbox. Table 4-1 displays the corresponding range of values that can be selected for each function.

Function	Selectable Range Values
4-Wire	0 – 100 KΩ in 10x steps
2-Wire	0 – 1 MΩ in 10x steps. Optional: 1 MΩ, and either 50 or 100 MΩ
VDC	±9 mV – ±9 V in 10x steps
VAC	4.5 mV – 4.5 V in 10x steps
Frequency Ranges	1 KHz – 10 MHz in 10x steps
External Source	(Not applicable)

Table 4-1. Selectable Functions and Value Ranges

Note: When the External Source radio button is selected, the listbox is disabled because function values are determined by the external device connected to the DB 9 connector on the GT1034 board. Click the External Source option when using a device as an external reference source.

Reselecting Values

To perform a new or different test, select the appropriate function and corresponding value. There is no need to re-initialize the board.

Resetting the Board

Click on the Reset button to revert back to the previously selected function and value.

Figure 4-7: Reset Button

Closing the Virtual Panel

Click on the Close button to exit from the Virtual Panel. Note that when the Virtual Panel is closed, but later re-opened to begin a new test, the GT1034 board must be reinitialized.

Figure 4-8: Close Button

Chapter 5 - Programming the Board

Introduction

This chapter contains information on how to program the GT1034 board using the GTCAL software package. The GTCAL drivers contain functions to initialize, reset, and control the GT1034 board. A brief description of the functions, as well as how and when to use them, is included in this chapter. Chapter Six beginning on page 49, describes these functions in detail.

This chapter also describes how the GTCAL driver supports different operating systems and development tools. It also includes an example showing how to program the GT1034 board with the C programming language. 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 GTCAL driver formats. The following development tools are supported:

- Visual C++
- Visual Basic
- DOS (Microsoft or Borland C)
- Borland C++
- ***ATEasy***

Refer to the README.1ST file of the installation disk for the latest list of programming languages and development tools that are supported, as well as other example files that are available on the disk.

Windows 95/98/NT, Windows 3.1 and DOS Drivers

The GTCAL driver file comes in three main formats

- 32-Bit MS-Windows DLL (Dynamic Link Library) – GTCAL32.DLL - Used by WIN32 application running under Windows 95, 98 and Windows NT.
- 16-Bit MS-Windows DLL - GTCAL16.DLL - Used by 16-bit Windows application running under Windows 3.x/95/98/NT.
- DOS static libraries – GTCALMS.LIB (Microsoft format) and GTCALBC.LIB (Borland format) - Used by DOS applications that are compiled with Microsoft or Borland development tools. The libraries are compiled in large memory model libraries with floating point emulation.

Import Libraries

The following files should also be used during development with specific applications, as follows:

- GTCAL16.LIB is the import library for GTCAL16.DLL.
- GTCAL32.LIB is the import library for GTCAL32.DLL.
- The GTCAL32B.LIB is the import library for Borland applications using GTCAL32.DLL.

Note: All three import libraries mentioned above must be linked into the Windows applications just like any other static library.

The GTCAL.H header file contains prototypes of the GTCAL driver functions. It must be included at the beginning of any C/C++ file that makes GTCAL function calls.

Using the DLL Driver with Windows NT

Under Windows NT direct access to hardware (e.g. I/O ports) is limited to code that runs in kernel mode. Windows applications (16 or 32-bit) cannot access the hardware because they run in user mode. To access hardware, these applications rely on drivers that reside in kernel mode. The GTCAL package has built-in support that overcomes some of the limitations of Windows applications.

HW.SYS (kernel-mode device driver) and HWVDD.DLL (virtual device driver) are both installed with the GTCAL module. HW.SYS enables any 16- or 32-bit Windows application to access the I/O ports. The HWVDD.DLL driver traps any I/O-to-PC port from a Windows 16-bit application and routes these calls to HW.SYS, which then exxecutes the command.

Note: Under Windows NT the HW drivers will trap any I/O port in the range of 0x100-0xFFFE, excluding the range 0x3BB-0x3BF.

How Does the HW Driver Work?

Figure 5-1 describes how HW.SYS works.

The following steps explain Figure 5-1:

1. Assume a 16-bit Windows application attempts to write to an I/O port.
2. NT traps the I/O port write attempt and routes it to HWVDD.DLL.
3. HWVDD.DLL routes the call to HW.SYS.
4. HW.SYS executes the port I/O (write) instruction and returns.

A similar process is executed when reading from a port.

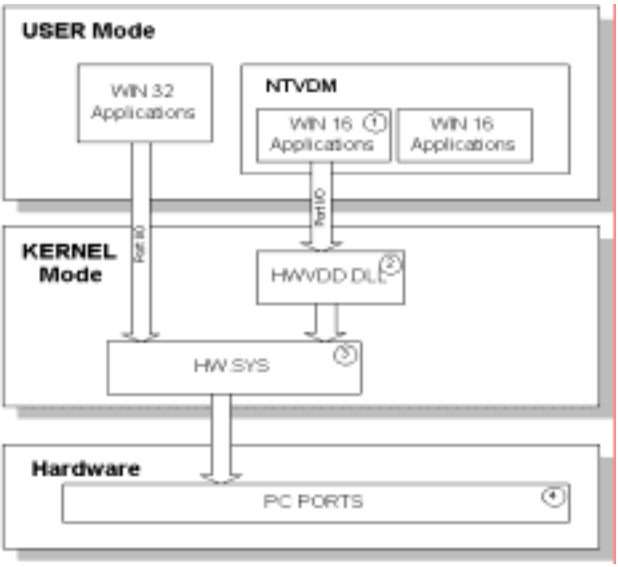

Figure 5-1. How the Hardware Driver works

Using the Windows 32-bit DLL

The Windows 32-bit GTCAL32.DLL can be used for applications that are developed for Windows 95 or Windows NT. Refer to Chapter 3, “Setup and Installation” for special installation instructions for Windows NT. The 32-bit DLL can be used with various development tools such as:

- Microsoft Visual C++
- Borland C
- ***ATEasy***
- Microsoft Visual Basic
- Borland Pascal
- Borland Delphi

The DLL may be shipped with the GTCAL application. Refer to one of the sections beginning on page 38 for specific information about the development tools listed above.

To use the DLL driver from an application, the DLL must reside in one of the following directories:

- Application
- Windows
- Windows System
- One of the directories specified in the PATH statement

The GTCAL32.DLL file can be distributed with the GT1034 board and any associated applications.

Note: Under Windows NT, HWVDD.DLL and HW.SYS must be installed before using the 32-bit DLL. Refer to Microsoft's Windows NT Kernel Mode Driver Installation procedure.

Using the Windows 16-bit DLL

The Windows 16-bit DLL, GTCAL16.DLL, is used for writing 16-bit applications that run under Windows 3.1/95/NT. The DLL can be used with various development tools such as:

- Microsoft Visual C++ (1.5 or less)
- Borland C (for 16-bit)
- ***ATEasy*** 2.0
- Microsoft Visual Basic (3.0 or less)
- Borland Pascal
- Borland Delphi (for 16-bit)

Once developed, the DLL can be shipped with the application. Refer to one of the sections beginning on page 38 for specific information about one of the development tools listed above.

To use the DLL driver from an application, the DLL must reside in one of the following directories:

- Application
- Windows
- Windows System

- One of the directories specified in the PATH statement

The GTCAL16.DLL file can be distributed with the GT1034 board and any associated applications.

Note: Under Windows NT, HW.SYS must be installed before the 16-bit DLL can be used. See "NT Kernel Mode Driver Installation procedure" on page 19.

Programming with the DOS Static Library

When programming the board under DOS, the application must be linked with either GTCALBC.LIB (Borland static library) or GTCALMS.LIB (Microsoft static library). These files are Static Libraries in which functions reside. The Setup disk contains static libraries for many development tools. See the README.1ST file for a complete list of supported development tools.

Supported Development Tools

The GTCAL driver supports C/C++, Visual Basic, Borland Pascal/Delphi, Geotest ***ATEasy***, and any other language that can use a Windows DLL, such as LabView.

Programming with C/C++ Tools

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

- Include the GTCAL.H header file in the C/C++ source file that uses the GTCAL function. This header is used for all driver formats. The file has function prototypes and constant declarations used by the compiler for the application.
- For Windows applications, make sure the DLL is installed in the proper directory (see previous sections beginning on page 34 that describe how to use the DLL).
- Add the required .LIB file to the projects. This can be the import library GTCAL32.LIB for 32-bit application, GTCAL32BC.LIB for 32-bit applications that uses Borland

C++, GTCAL16.LIB for 16-bit application, or any of the DOS static libraries for DOS application. Windows based applications that explicitly load the DLL by calling the Windows **LoadLibrary()** API need not include the .LIB file in the project.

- Add code to call the GTCAL as required by the application.
- Build the project.
- Run, test, and debug the application.

Programming with Visual Basic

GTCAL16.BAS and GTCAL32.BAS files contain function declarations for the GTCAL16.DLL and GTCAL32.DLL drivers, respectively. These BAS files must be loaded using **Load File** from the Visual Basic File menu before the functions can be used.

Programming with Borland Pascal/Delphi

To use the driver with Borland Pascal or Delphi, include the GTCAL.PAS in the project. The GTCAL.PAS file contains a unit with function prototypes for all the DLL (16 and 32-bit DLLs) functions. Include the GTCAL unit in the **USES** statement before making calls to the GTCAL functions. .

Programming with *ATEasy* 2.0

The ***ATEasy*** driver (GTCAL.INS) supplied with this product uses the GTCAL16.DLL to program the board. These files must reside in the ***ATEasy*** INS and DLL directories, respectively. To use the driver, the INS, CFG, PRG and 16-bit DLL files must be copied to the appropriate ***ATEasy*** directory. The GTCAL.INS driver must also be included in the current system file (.CFG) in order to program the board.

In addition, GTCAL.INS is supplied with an example that contains a program and a system file configured with the instrument driver set to the default base address (in the System Editor Driver Setup dialog - module number field).

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. This parameter is handled automatically by the driver. (***ATEasy*** uses driver logical names).
- The *nStatus* parameter is omitted. Use the Get Status commands instead to check status.

The ***ATEasy*** driver also contains additional commands to permit easier access to the GT1034 board features (instead of typing parameter values). The commands are self-documenting. Their syntax is similar to English sentences and can be generated from the program editor in the Instruments menu. GTCAL.INS is documented in the ***ATEasy*** Notes dialog, which can be viewed from the Driver Editor Summary contained within the Edit menu. In addition, more detailed documentation exists in the ***ATEasy*** knowledge base, ATEASYKB.HLP, that is shipped with ***ATEasy***.

The GTCAL Setup program updates the ***ATEasy*** directory with the new DLL-based driver and example files.

Programming with the Driver

The GTCAL driver contains a set of functions that initialize the board driver, reset the board, and display the instrument virtual panel. In addition, the GTCAL driver uses handles (see below) to access the GT1034 board and process errors. The following paragraphs describe the steps and topics that are required to control the board.

Initialization and the Board Handle

The **GtCalInitialize** function initializes the driver for the board at the specified base address. The function returns a handle that can be used later with other 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.

Board Handles

The Board handle parameter - *nHandle* - is a short integer number that is used by the GTCAL driver functions to identify the board being accessed by the application. The *nHandle* parameter is required to identify the board being programmed.

The *nHandle* is created when the application calls the **GtCalInitialize** function. There is no need, or command, to destroy the handle. Calling **GtCalInitialize** with the same base address will return the same handle.

Once the board is initialized, the handle can be used with other functions calls to control the board.

Reset

The Reset function, **GtCalReset**, opens all switches in the board and sets the board to a known state. A reset is usually performed after the board is initialized. See Chapter 6 in this manual for more information regarding the board reset.

Error Handling

All the GTCAL functions return the status, *pnStatus*, as the last parameter. This parameter can be used for error handling. *pnStatus* returns zero if the function completes successfully. It returns negative if an error occurred. The error description can be retrieved using the **GtCalGetErrorString** function.

Driver Version

The **GtCalGetDriverSummary** function can be used to return the current GTCAL driver description summary and version number. This can be used to check for driver changes. See Chapter 6 for more information.

Virtual Panel

The panel function, **GtCalPanel**, displays the instrument front panel in a window. The panel can be used to initialize the board and to control and read all of the board settings. The panel function may be used by the application to allow direct interaction with the board.

The **GtCalPanel** function is also used by GTCALP16.EXE and GTCALP32.EXE, which are supplied with the GTCAL package and provide a stand alone Windows application that displays the instrument panel. This function is available only under the Windows version of the driver and is not supported under the DOS driver.

Programming Example

The following example demonstrates how to program the board using the C programming language under Windows and DOS. To run, enter the following on the command line:

GTCALXXX <*base_address*> <*specif_command*><*specif_value*>

Where:

<*base_address*>	I/O Base Address in hexadecimal. For example: 0x310
<*specif_command*>	Can be 0 through 6 – Output function. See GtCalSetFunction( ) in Chapter 6, "Functions Reference."
<*specif_value*>	Can be 0 through 9 (depending on the function). See GtCalSetFunction( ) in Chapter 6, "Functions Reference."

Compiling the Example

To compile the Windows 32-bit DLL example in Microsoft Visual C++ 2.0 or above:

1. Load GTCALX32.MAK from the Project/File/Open menu.

2. Select Project/Rebuild all from the menu.

To compile the Windows 32-bit DLL example in Borland C++ 4.0/4.5/5.0:

1. Load GTCALXBC.IDE from the Project/Open Project menu.

2. Select Project/Build all from the menu.

To compile the Windows 16-bit DLL example in Microsoft VC++ 1.0/1.5 or above:

1. Load GTCALX16.MAK from the Project/Open menu.

2. Select Project/Rebuild all from the menu.

To compile the Windows 16-bit DLL example in Microsoft C 5.1 or above using the command line compiler:

Run NMAKE GTCALX16.MAK /A from the DOS command line.

To compile the Windows 16-Bit DLL Example in Borland C++ 4.0/4.5/5.0:

1. Load GTCALXBC.IDE from the Project/Open Project command.

2. Select Project/Build all.

To compile the DOS example in Microsoft VC++:

1. Load GTCALXMS.MAK from the Project/Open command.

2. Select Project/Rebuild all.

To compile the DOS example in Microsoft C 5.1 or above:

Run NMAKE GTCALXMS.MAK /A from the DOS command line.

To compile the DOS example in Borland C++ 4.0/4.5/5.0:

1. Load GTCALXBC.IDE from the Project/Open Project command.

2. Select Project/Build all.

MAK Files

The C++ sample includes a MAK file for each of the following tools:

DOS Static Library - Microsoft C (VC++/MS-C) Example File:

- GTCALXMS.EXE - Sample for DOS (with Microsoft C Static Library)
- GTCALXMS.MAK - VC++/MS-C Make file for the example
- GTCALX.C - C source file for the example (same source for all C examples)

16-bit DLL (Windows 3.x) - Microsoft C (VC++/MS-C) Example Files:

- GTCALX16.EXE - Sample for Windows 3.x/95
- GTCALX16.MAK - VC++/MS-C Make file for the example
- GTCALX.C - C source file for the example (same source for all C examples)
- GTCALX.DEF - Module definition file
- GTCALX.RC - Resource file
- GTCALX.ICO - Icon file

32-bit DLL (Windows 95/NT) example files:

- GTCALX32.EXE - Sample for Windows NT/95
- GTCALX32.MAK - VC++/MS-C Make file for the example
- GTCALX.C - C source file for the example (same source for all C examples)
- GTCALX.RC - Resource file
- GTCALX.ICO - Icon file

Borland Static/16-bit DLL/32-bit DLL Examples:

- GTCALBC.IDE - Borland C++ 5.0 contains 3 projects
- GTCALX.C - C source file for the example (same source for all C examples)
- GTCALX.DEF - Module definition file

- GTCALX.RC - Resource file
- GTCALX.ICO - Icon file

ATEasy Examples Files:

- GTCAL.PRG - example program
- GTCAL.CFG - example CFG file

Example Program Listing

The following example is included in the GTCAL software module. The example shows how to connect or disconnect a function and set a range.

```
/*********************************************************

     FILE     : GtCalX.C

     PURPOSE  : Windows16|32|DOS sample program for GT1034 boards
                using GTCAL drivers.

     CREATED  : Feb. 1999

     COPYRIGHT: Copyright 1999 Geotest, a Marvin Engineering Co.

     COMMENTS :


To compile the Windows 32 bit DLL example:


    1. Microsoft VC++ 2.0 or above
       - Load GTCALX32.MAK from the Project\File/Open... menu
       - Select Project/Rebuild all from the menu

    2. Borland C++ 4.0/4.5/5.0
       - Load GtCalXBC.IDE from the Project/Open
         Project... menu
       - Select Project/Build all from the menu

    To compile the Windows 16 bit DLL example:

    1. Microsoft VC++ 1.0/1.5 or above
       - Load GTCALX16.MAK from the Project/Open... menu
       - Select Project/Rebuild all from the menu

    2. Microsoft C 5.1 or above using command line compiler
       - Run 'NMAKE GtCalX16.MAK /A' from DOS command line

    3. Borland C++ 4.0/4.5/5.0

       - Load GTCALXBC.IDE from the Project/Open
         Project... menu
       -   Select Project/Build all from the menu
```

```
    To compile the DOS example with:

    1. Microsoft VC++
       - Load GTCALXMS.MAK from the Project/Open... menu
       - Select Project/Rebuild all from the menu

    2. Microsoft C 5.1 or above
       - Run 'NMAKE GTCALXMS.MAK /A' from DOS command line

    3. Borland C++ 4.0/4.5/5.0
       - Load GTCALXBC.IDE from the Project/Open
         project... menu
       - Select Project/Build all from the menu

*********************************************************/

#if defined(_WINDOWS) || defined(_WIN32) || defined (_Windows)
    #include <windows.h>
#elif !defined(_DOS)
    #define _DOS
#endif
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "gtcal.h"

//*********************************************************
//      DispMsg
//*********************************************************
void DispMsg(LPSTR szMsg)
{
    #ifndef _DOS
    //if windows use MessageBox
     MessageBeep(0);
     MessageBox(0, szMsg,"GtCal", MB_OK);
    #else
    // use printf for DOS
      printf("%s \r\n",szMsg);
    #endif
     return;
}

//*********************************************************
//      DispUsage
//*********************************************************
void DispUsage(void)
{
    DispMsg(
        "\r\nThis example shows how to "
        "Connect or disconnect a function and to set the
         range\r\n"
        "Usage:\r\n"
        "GtCalXxxx <base_address> <specif_function>
         <specif_value>"
        "\r\n\r\nWhere:\r\n"
        "<base_address>"
        "\t(example : 0x310)\r\n"
        "<specif_function> function - can be 0 through 6\r\n"
         "<specif_value> value - can be 0 through 9 (depending on
         the function)\r\n"
```

```
            "Example:\r\n"
            "GtCalX16 0x310 1 1\r\n"
            "Sets the board reside at 0x310 base address to 4-Wire
             function and 1 ohm resistance\r\n"

        #ifdef _WINDOWS
            "\r\nTo change command line under Windows \r\n"
            "Select File/Properties from Program Manager \r\n"
            "Menu and change the command line edit box as \r\n"
            "shown above."
        #else

        #endif
            );
        exit(1);
    }

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

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

    //*********************************************************
    //      Main
    //
    //  This main function receives three parameters:
    //  GTCAL board base address (e.g. 0x310)
    //  GTCAL setup operation (e.g. F=Function, R=Range)
    //
    //*********************************************************

    int main(int argc, char **argv)
    {
        short  nBaseAddr;              // Board base address
        char   cFunction;              // Board Function
        short  nValue;                 // Function value
        short  nHandle;                // Board handle
        short  nStatus;                // Returned status

        // ** Check number of arguments rcvd
        if (argc < 3 ) DispUsage();

        // ** Parse command line parameters
        nBaseAddr=(SHORT)strtol(*(++argv), NULL, 0);
        cFunction=**(++argv);

        // ** Process operation (close/open)
        cFunction=(char)toupper(cFunction);
        if (cFunction>0x36 || cFunction<0x30)
            DispUsage();

        // ** Get Function, Range, initialize, and perform operation
              on GTCAL card
```

```
            nValue = (SHORT)strtol(*(++argv), NULL, 0);
            GtCalInitialize (nBaseAddr, &nHandle, &nStatus);
            CheckStatus(nStatus);
            GtCalSetFunction (nHandle, (SHORT)(cFunction-0x30),
                  nValue, &nStatus);
            CheckStatus(nStatus);
            DispMsg ("Function & value were been set successfully");
            return 0;
      }

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

Chapter 6 - Functions Reference

Introduction

This chapter presents information on the GTCAL driver functions. in alphabetical order. Each function begins with a syntax example of the function followed by a short description of the function parameters and types, Comments, Example (written in C), and See Also sections.

All function parameters follow these same rules:

- All pointers are far pointers (applicable only on 16-bit drivers).
- Strings are ASCIIZ (null or zero character terminated).
- The first parameter for all functions is *nHandle* (16-bit integer), which is required and is returned by the board's **GtCalInitialize** function. The *nHandle* identifies the board when calling a function to program and control the board.
- All functions return a status with the last parameter named *pnStatus*. *pnStatus* is 0 if the function was successful, or <0 on error. The description of the error can be retrieved by using the **GtCalGetErrorString** function, or by using a predefined constant, defined in the driver interface file. See Appendix A for available error codes.

Parameter Prefix Names

Parameters naming prefixes are as follows:

Type	Prefix	Type Description	Example
ARRAY	a	Prefix precedes simple prefix type.	*anArray*
BOOL	b	Boolean. Signed 8-bit integer. TRUE if 0, FALSE otherwise.	bSelected
BYTE	uc	Unsigned 8-bit integer (unsigned CHAR type).	*ucSection*
CHAR	c	Character. Signed 8-bit integer.	*cKeyPress*
DOUBLE	d	Double precision 64-bit floating point.	*dReading*
DWORD	dw	Unsigned 32-bit integer (non-handle).	*dwTimeout*
DWORD	hwnd	Window handle - unsigned 32-bit integer used as a handle. Only the lower 16-bits are used under Windows 3.1.	*hwndPanel*
LONG	l	Signed 32-bit integer.	*lBits*
LP	p	32-bit long pointer. (Far pointer for Windows 3.1). Usually returns a value. Prefix precedes simple prefix type.	*pnStatus*
LPSTR	sz	String. Zero-terminated ASCIIZ text string.	*szMsg*
SHORT	n	Signed 16-bit integer.	*nMode*
WORD	w	Unsigned 16-bit integer.	*wCount*

Table 6-1: Parameter Name Prefixes

GtCalGetBoardSummary

Purpose

Return the board summary from the on-board EEPROM.

Syntax

GtCalGetBoardSummary(*nHandle*, *szBoardSum*, *nSumMaxLen*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1034 board.
szBoardSum	LPSTR	Buffer to contain the returned board info string (null terminated string).
nSumMaxLen	SHORT	Size of the buffer to contain the error string.
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

Comments

The board information string reads from the on-board EEPROM the following data in this order:

- Instrument Name (e.g., GT1034)
- EEPROM format version (e.g. 1.00)
- PCB revision (e.g. 'A')
- Serial Number
- Calibration time and date
- Optional resistors (10 MΩ and either 50 or 100 MΩ)
- Available voltage sources

For example the returned string looks as follows:

```
GtCal, S/N 1034009, Ver. 1.00, Rev. A, 100MOhms,
installed. Available voltage sources:  +10VDC, -10VDC,
5VAC, last calibrated time Fri Oct 16 18:30:25 1998
```

Example

The following example initializes GTCAL board at addresses 0x310 and returns the board summary:

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

GtCalInitialize(0x310, &nHandle1, &nStatus);
GtCalGetBoardSummary(nHandle, &szBoardSum,
    sizeof( szBoardSum), &nStatus);
```

See Also

GtCalGetDriverSummary, GtCalInitialize, GtCalGetErrorString

GtCalGetDriverSummary

Purpose

Returns the driver description string and version number.

Syntax

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

Parameters

Name	Type	Comments
pszSummary	LPSTR	Buffer to receive the summary string.
nSummaryMaxLen	SHORT	Buffer size passed by *szSummary*.
pdwVersion	LPDWORD	Returned version number. The high 16 bits contain the major version while the lower 16 bits contain the minor version number.
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

Comments

The GT1034 driver summary string provides the following information:

- Instrument driver name e.g. GtCal
- Version number. E.g. Version 1.0

For example the returned string could look like this:

```
GtCal Driver for GtCal, Version 1.0, Copyright (c)
Geotest Inc.
```

The value of *pdwVersion* is 10 (prints as 1.0).

Example

The following example initializes the board at addresses 0x310 and gets the driver summary:

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

GtCalInitialize(0x310, &nHandle, &nStatus);
GtCalGetDriverSummary(&szSummary, 256, &dwVersion,
    nStatus);
```

See Also

GtCalInitialize, GtCalGetErrorString

GtCalGetErrorString

Purpose

Returns the error string as specified by the error number.

Syntax

GtCalGetErrorString(*nError*, *pszError*, *nErrorMaxLen*, *pnStatus*)

Parameters

Name	Type	Comments
nError	SHORT	Error number as returned by the *pnStatus* of any of the GtCal functions. See “Appendix A – Error Codes” for possible values. For errors, the number should be less than zero. If the number is not zero, the function returns the "No error has occurred" string.
pszMsg	LPSTR	Buffer to contain the returned error string (null terminated string)
nStrMaxLen	SHORT	Size of the buffer to contain the error string.
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

Example

The following example initializes the board at addresses 0x310 and gets the error string:

```
SHORT nHandle, nStatus;
CHAR  szError[128];

GtCalInitialize (0x310, &nHandle, &nStatus);
GtCalGetErrorString (nStatus, &szError, sizeOf(szError),
    &nStatus);
```

See Also

GtCalInitialize, GtCalGetErrorString

GtCalGetFunction

Purpose

Returns the current board function source and the value for the specified function source.

Syntax

GtCalGetFunction(*nHandle*, *pnFunction*, *pnValue*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1034 board.
pnFunction	SHORT	Corresponding function and returned number: 4-Wire 0 2-Wire 1 Volt DC 2 Volt AC 3 Frequency 4 External Source 5 Not connected -1
pnValue	SHORT	Each function has its own corresponding values, as follows: 4-Wire: see Table 6-2. 2-Wire: see Table 6-3. Volt DC: see Table 6-4. Volt AC: see Table 6-5 Frequency: see Table 6-6**.**
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

pnValue	**Description**
0	0Ω
1	1Ω
2	10Ω
3	100Ω
4	1KΩ
5	10KΩ
6	100KΩ
-1	Not connected

Table 6-2: Four-Wire *pnValue* **Descriptions (***pnFunction***=0)**

pnValue	**Description**
0	0Ω
1	1Ω
2	10Ω
3	100Ω
4	1KΩ
5	10KΩ
6	100KΩ
7	1MΩ
8	10 MΩ
9	100 (or 50)* MΩ
-1	Not connected

Table 6-3: Two-Wire *pnValue* **Descriptions (***pnFunction***=1)**

*See GtCalGetBoardSummary for more information about this option.

pnValue	**Description**
0	9.000 Volts
1	0.900 Volts
2	0.090 Volts
3	0.009 Volts
10	-9.000 Volts
11	-0.900 Volts
12	-0.090 Volts
13	-0.009 Volts
-1	Not connected

Table 6-4: Volt DC *pnValue* **Descriptions (***pnFunction***=2)**

pnValue	Description
0	4.500 Volts
1	0.450 Volts
2	0.045 Volts
3	0.0045 Volts
-1	Not connected

Table 6-5: Volt AC *pnValue* **Descriptions (***pnFunction***=3)**

pnValue	Description
0	1KHz
1	10KHz
2	100KHz
3	1MHz
4	10MHz

Table 6-6: Frequency *pnValue* **Descriptions (***pnFunction***=4)**

Example

The following example gets the function and value from the board:

```
SHORT   nFunction, nValue, nStatus;

GtCalGetFunction(nHandle, &nFunction, &nValue,
    &nStatus);
```

See Also

GtCalSetFunction, GtCalReset, GtCalInitialize, GtCalGetErrorString

GtCalInitialize

Purpose

Initializes the driver for the board at the specified base address. The function returns a handle that can be used with other GTCAL functions to program the board.

Syntax

GtCalInitialize(*nBaseAddress*, *pnHandle*, *pnStatus*)

Parameters

Name	Type	Comments
nBaseAddress	SHORT	Base address of the board.
pnHandle	LPSHORT	Returned handle for the board. The handle is set to zero on error and <> 0 on success.
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

Comments

The GT1034 uses two 16-bit ports in the I/O space following the base address with an additional port used at the base address + 0x400. The factory default base address is set to 0x310. The board base address can be changed by setting a DIP Switch to any address between 0x200 and 0x3FF.

The **GtCalInitialize** function verifies whether or not the board exists in the specified address. However, the function does not have any effect on the board settings.

The returned handle *pnHandle* identifies the specified board with other GTCAL functions.

Example

The following example initializes two GT1034 boards at addresses 0x310 and 0x320:

```
SHORT nHandle1, nHandle2, nStatus;

GtCalInitialize(0x310, &nHandle1, &nStatus);
GtCalInitialize(0x320, &nHandle2, &nStatus);
if (nHandle1==0 || nHandle2==0)
{
    printf(“Unable to Initialize the board”)
    return;
}
```

See Also

GtCalReset, GtCalInitialize, GtCalGetErrorString

GtCalPanel

Purpose

Opens a virtual panel used to interactively control the GT1034 board.

Syntax

GtCalPanel (*pnHandle*, *hwndParent*, *nMode*, *phwndPanel*, *pnStatus*)

Parameters

Name	Type	Comments
pnHandle	LPSHORT	Handle to a GT1034 board. This number may be zero if the panel window initializes the board.
hwndParent	DWORD	Sets the panel parent window handle. A value of 0 sets the desktop as the parent window.
nMode	SHORT	The mode in which to create the panel main window. 0 for modeless and 1 for modal window.
phwndPanel	LPDWORD	Returned window handle for the panel (for modeless panel only).
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

Comments

The function is used to create the panel window. The panel window may be open as a modal or a modeless window depending on the *nMode* parameters.

If the mode is set to modal dialog (*nMode*=1), the panel disables the parent window (*hwndParent*) and the function returns only after the user closes the window. In that case, the *pnHandle* may return the handle created by the panel Initialize dialog.

If a modeless dialog was created (*nMode*=0), the function returns immediately after creating the panel window and returns the window handle to the panel – *phwndPanel.* It is the responsibility of the calling program to dispatch window messages to this window, so that the window can respond to messages.

This function is supplied only with DLL versions of the driver.

Example

The following example opens the panel in modal mode:

```
DWORD   hwndPanel;
SHORT   nHandle=0, nStatus;

GtCalInitialize(0x310, &nHandle, &nStatus);
GtCalPanel(&nHandle, 0, 1, &hwndPanel, &nStatus);
```

See Also

GtCalInitialize, GtCalGetErrorString

GtCalReset

Purpose

Opens all the relays, disconnects all function sources.

Syntax

GtCalReset(*nHandle*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GT1034 board.
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

Example

The following example initializes the GT1034 board at address 0x310 and then issues a reset command.

```
SHORT nHandle, nStatus;

GtCalInitialize(0x310, &nHandle, &nStatus);
GtCalReset(nHandle, &nStatus);
```

See Also

GtCalInitialize, GtCalGetErrorString

GtCalSetFunction

Purpose

Sets the board function source and value.

Syntax

GtCalSetFunction(*nHandle, nFunction, nValue, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to the GT1034 board.
nFunction	SHORT	Function and corresponding number: 4-Wire 0 2-Wire 1 Volts DC 2 Volts AC 3 Frequency 4 External Source 5
nValue	SHORT	Each function has its own values settings as follows: 4-Wire: see Table 6-7 2-Wire: see Table 6-8. Volts DC: see Table 6-9. Volts AC: see Table 6-10. Frequency: see Table 6-11.
pnStatus	LPSHORT	Returned status: 0 on success, <0 on failure.

nValue	Description
0	0 Ω
1	1 Ω
2	10 Ω
3	100 Ω
4	1 KΩ
5	10 KΩ
6	100 KΩ

Table 6-7: Four-Wire *nValue* Descriptions (*nFunction*=0)

nValue	Description
0	0 Ω
1	1 Ω
2	10 Ω
3	100 Ω
4	1 KΩ
5	10 KΩ
6	100 KΩ
7	1 MΩ
8	10 MΩ
9	100 (50) MΩ

Table 6-8: Two-Wire *nValue* Descriptions (*nFunction*=1)

nValue	**Description**
0	9.000 Volts
1	0.900 Volts
2	0.090 Volts
3	0.009 Volts
10	-9.000 Volts
11	-0.900 Volts
12	-0.090 Volts
13	-0.009 Volts

Table 6-9: Volts DC ***nValue*** **Descriptions (***nFunction***=2)**

nValue	**Description**
0	4.500 Volts
1	0.450 Volts
2	0.045 Volts
3	0.0045 Volts

Table 6-10: Volts AC ***nValue*** **Descriptions (***nFunction***=3)**

nValue	**Description**
0	1 KHz
1	10 KHz
2	100 KHz
3	1 MHz
4	10 MHz

Table 6-11: Frequency ***nValue*** **Descriptions (***nFunction***=4)**

Example

The following example initializes the board at addresses 0x310 and sets the function to 2-wire and value of 10 ohms.

```
SHORT nHandle, nStatus;

GtCalInitialize (0x310, &nHandle, &nStatus);
GtCalSetFunction (nHandle, GtCal_FUNCTION_2WIRE,
    GtCal_2WIRE_10OHM, &nStatus);
```

See Also

GtCalGetFunction, **GtCalReset, GtCalInitialize, GtCalGetErrorString**

Appendix A – Error Codes

This appendix includes values for the *nError* parameter of the **GtCalGetErrorString** function. (See page 55 in Chapter 6.)

Note: Specifying an error code value of zero ("0") indicates no errors, and the function returns the "No error has occurred" string.

The error code is returned by all functions with *pnStatus*. Tables A-1 through A-4 describe the possible error code values for these Error Types:

- Resource Errors
- Parameter Error
- Board Errors (Fatal Errors)
- Miscellaneous Errors

Resource Errors

Error Code	Description
-1	Board does not exist in this base address
-2	Too many boards
-3	Unable to create panel
-4	Unable to create Windows timer
-5	Out of memory

Table A-1. Resource Error Code Values

Parameter Errors

Error Code	Description
-20	Invalid parameter
-21	Illegal base address
-22	Illegal boards handle
-23	Invalid function number
-24	Illegal source value number
-25	Illegal string length
-26	Illegal mode
-27	Invalid calibration mode
-28	Invalid calibration parameter

Table A-2. Parameter Error Code Values

Board (Fatal) Errors

These errors indicate that the board has a malfunction.

Error Code	Description
-50	Fatal error: Unable to set 4-wire function
-51	Fatal error: Unable to set 2-wire function
-52	Fatal error: Unable to set volt DC function
-53	Fatal error: Unable to set volt AC function
-54	Fatal error: Unable to set frequency function
-55	Fatal error: Unable to set to external source
-56	Fatal error: Unable to reset
-57	Fatal error: Unable to close a relay
-58	Fatal error: Unable to open a relay
-59	Fatal error: Unable to open/close a relay

Error Code	Description
-60	Fatal error: EEPROM offset address is out of range
-61	Fatal error: the board is not calibrated
-62	Fatal error: the board does not support the specified value
-63	Fatal error: the board does not support the specified function

Table A-3. Board Errors (Fatal Error) Code Values

Miscellaneous Errors

Error Code	Description
0	No error has occurred
-99	Invalid or unknown error number

Table A-4. Miscellaneous Error Code Values

Appendix B - Specifications

The following table outlines the specifications for the GT1034:

Functions	**Values**	**Tolerances**	**Notes**
Resistance:			
4-Wire, Ω	1, 10	.005%	
2-Wire, Ω	100, 1 K , 10 K, 100 K, 1 M	.005%	
2-Wire, Ω	10 M, 100 M	.01%	Optional
Voltage:			
DC Full Scale	+/-10 V	+/- 1 mV	
DC Ranges, Volts	10.000, 9.000, 0.900, 0.090, 0.009		Available across programmable resistor network, 10K, 1K, 100 and 10 Ω, .005% Resistors
AC Full Scale	5 V RMS	0.05%	Sine Wave
AC Ranges, Volts RMS	5.000, 4.500, 0.450, 0.045, 0.0045		Available across programmable resistor network, 10K, 1K, 100 and 10 Ω, .005% Resistors
Frequency:			
Ranges, Hz	1 K, 10 K, 100 K, 1 M, 10 M	+/- 100 PPM (0.01%)	

Index