GTDIO ™

GT-DIO Product Family

GTDIO/DIOEasy Software

User’s Guide

Last Updated March 23, 2016

MARVIN TEST SOLUTIONS

Safety and Handling

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

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

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

Be sure to make a single copy of the software CD for installation. Store the original CD in a safe place away from electromagnetic or electrostatic fields. Return compact disks (CD) to their protective case or sleeve and store in the original shipping box or other suitable location.

Warranty

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

If You Need Help

Visit our web site at http://www.marvintest.com more information about Marvin Test Solutions products, services and support options. Our web site contains sections describing support options and application notes, as well as a download area for downloading patches, example, patches and new or revised instrument drivers. To submit a support issue including suggestion, bug report or questions please use the following link: http://www.marvintest.com/magic/

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

Disclaimer

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

Copyright

Trademarks

ATEasy®, CalEasy, DIOEasy®, DtifEasy, WaveEasy	Marvin Test Solutions, Inc., Geotest – Marvin Test Systems, Inc (prior company name)
C++ Builder, Delphi	Embarcadero Technologies Inc.
LabView, LabWindowstm/CVI	National Instruments
Microsoft Developer Studio, Microsoft Visual C++, Microsoft Visual Basic, .NET, Windows 95, 98, NT, ME, 2000, XP, VISTA, Windows 7, 8 and 10	Microsoft Corporation

All other trademarks are the property of their respective owners.

Table of Contents

Safety and Handling i

Warranty i

If You Need Help i

Disclaimer i

Copyright i

Trademarks ii

Table of Contents iii

Chapter 1 - Introduction 1

About This User Guide 1

Required User Knowledge and Skills 1

User Guide Organization 1

Conventions Used in this User Guide 2

Style Conventions 2

Definitions 2

Chapter 2 - DIOEasy 3

Overview 3

Concepts 3

Starting DIOEasy 3

Main Window Components 4

Creating a New Vector File 5

Navigation Shortcuts 6

Selecting Items 7

Selecting Cells in Vector View 7

Selecting Items in a Windows List Box 7

File Menu 8

New… 8

Open 10

Compare… 12

Close 12

Save 13

Save As... 14

CheckIt! 15

Properties... 16

Properties: General Window 17

Properties: Setup Window 18

Properties: Channels settings Window 20

Import File … 22

Exit....23

Edit Menu 23

Cut 23

Copy....23

Paste....23

Delete....23

Insert Steps....24

Select....24

GoTo... 25

Find Direction Forward 25

Find Next Command....25

Find Next Label 25

Find Next Command/Label....26

Find Next Signal Change 26

Find Next Failed Step 26

Find Next Failed Cell....26

View Menu 27

Toolbar....27

Status Bar....28

Command Bar....28

Comparison....29

Command Properties....29

Zoom In / Zoom Out / Zoom Fit 30

Split....30

Unsplit....31

Options… 32

Channel Menu....33

Name... 33

Import Names… 34

Order... 35

Show... 36

Hide 36

Remap ... .37

Fill Menu 37

Value... 38

Other Fill Function Patterns 39
Pseudo-Random... 39
Fill Ramp... 40
Toggle... 41
Shift/Rotate... 42
Clock... 43
Fill Direction 44
Window Menu 45
New Window 45
Cascade 45
Tile Vertically 45
Tile Horizontally 45
Open Windows Menu Area 45
Help Menu 45
Contents 45
Search 45
About DIOEasy… 45
Chapter 3 - Importing External Vector Files 47
The Import File Dialog 47
Signals Tab 48
Profile Tab 49
WGL File 51
STIL File 51
ATP File 51
VCD/eVCD File 52
Chapter 4 - Virtual Panel Description 53
Virtual Main Page Panel 54
Clock Group 54
Trigger Group 55
Ext. Events… 55
Delay Group 55
Vector Group 56
Run Group 56
Virtual Panel Configuration Page 57
Using Marvin Test Solutions' HW Device Driver 57
Using VISA Device Driver 57
Virtual Panel Gx5290/Gx5280: I/O Channels Settings Page 59

Virtual Panel Gx5295: I/O Channels Page 60

Output voltage group: 60

Imput Active Load group: 61

PMU Forced Current group: 61

PMU Forced Voltage group: 61

Virtual Panel Gx5295 Measure Page 62

Temperarures group: 62

I/O Voltages group: 62

PMU Curretns group: 63

Virtual Panel Advanced Page 64

Clock group: 64

Panel Options group: 65

Signals Polarity group: 65

Virtual Panel Gx529x Real Time Compare Page 66

Virtual Panel Gx5055 High Power Chassis Page 68

Virtual Panel About Page 70

Chapter 5 - Using DIOEasy with the GC5050/GX5050 71

Introduction 71

Planning and Writing Example Testware 71

Choosing the DIO Board 71

An Example UUT 72

Initializing the UUT 72

Testing the RAM 74

Writing to the RAM 74

Testing the D Flip-flop 77

Adding Data and Commands to Vector Files 77

Defining Channels 77

Hiding Unused Channels 78

Adding Data and Controlling GC5050/GX5050 I/O Direction 78

Commands and Labels 79

Using the Fill Clock Command 81

Using the Fill Ramp Command 82

Using Copy and Paste to Duplicate Data 82

Checking Your Work 82

Using the Panel 83

Saving Your Work 84

Chapter 6 - Using *DIOEasy* with the GX515X 85

Introduction..85
UUT Description...85
UUT Environment...85
DIO/UUT Interface..85
UUT Flow of Events..86
Implementing the example...86
Creating the Example Vector File..87
Assigning Vector Channels..87
Editing the vector file 88
Set All Channels Values to Zero...88
Initialize the system 88
Generating Waveform Data 90
Generating the DAC clock..91
Bus Request Line (BusReq) 91
Controlling Waveform Generation in the RUN State 91
Set Vector Properties 93
Enable/Disable Output Data Lines..94
Adding Labels...95
UUT / DIO Interface Signals 95
Chapter 7 - Programming the Board 97
The GTDIO Driver 97
Programming Using C/C++ Tools...97
Programming Using Visual Basic..97
Programming Using Pascal/Delphi..98
Programming DIO Boards Using ATEasy®..............................98
Programming Using LabView and LabView/Real Time 98
Using and Programming under Linux.......................................98
Using the GTDIO driver functions 99
Initialization, HW Slot Numbers and VISA Resource...............99
Board Handle 100
Reset...100
Error Handling 100
Driver Version...100
Panel...100
Distributing the Driver...101
Sample Programs 101
Sample Program Listing...101

Index 108

Chapter 1 - Introduction

About This User Guide

This User Guide provides information needed to install, configure, program, and use Marvin Test Solutions' family of digital input/output (DIO) Drivers and the ***DIOEasy*** programming environment. The software is used to control DIO circuit boards. DIO boards, supporting cards and accessories are discussed in their respective User Guides.

Required User Knowledge and Skills

This User Guide assumes a general knowledge of PC-based computers, proficiency with a 32-bit Microsoft Windows operating system and some knowledge of electronics. In order to program the board, skill in using program development tools such as ***ATEasy***™ or commercial C, C++ or Pascal integrated development environment (IDE) products is required.

User Guide Organization

This user guide is organized as follows:

Chapter	**Content**
Chapter 1	Introduction. Introduces this Software User's Guide.
Chapter 2	Overview and Using DIOEasy. Provides reference information on menu commands and dialog boxes. Summarizes DIO software and family board features and architecture.
Chapter 3	Virtual Panel Description. Provides information on using and operating the Panel windows to configure, initialize, and control a DIO domain.
Chapter 4	Using DIOEasy with GC5050/GX5050. A functional software and system description. Includes a tutorial example.
Chapter 5	Using DIOEasy with GX5150. A functional software and system description. Includes a tutorial example.
Chapter 6	Programming with the GTDIO Driver. A functional software and system description. Includes a tutorial example.
Index	A road map to important topics and concepts in this user guide.

Conventions Used in this User Guide

Style Conventions

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, then 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. For example, 0x10A equals 266 in decimal.

Definitions

The following table defines terms commonly used in this document:

Term	Definition
DIO	Digital Input/Output (I/O).
DIO board	Generically, any of Marvin Test Solutions' DIO circuit boards.
DIOEasy	***DIOEasy*** version *2.n* unless otherwise stated. "*n*" is a release number. All release numbers are backward compatible with version 2.0.
Testware	Test software. It is the software used to test a Unit Under Test (UUT) or Device Under Test (DUT).
UUT	Unit Under Test, this includes the broader sense, it refers to any user device that communicates with a DIO board, whether or not actually used in a test system.

Chapter 2 - DIOEasy

Overview

Marvin Test Solutions' DIO is a family of high-speed, PC-based, programmable, dynamic digital input and output (I/O) boards for high speed automated functional testing, device testing, simulation and data acquisition. The DIO family provides real-time digital stimulus, capture and synchronization with 32 bits per card or 256 bits per system.

The DIO family uses common software development tools to develop test procedures and test vectors. The test vectors contain digital data that needs to be transmitted to the UUT (stimulus) or data that is expected from the UUT (response/capture).

DIO ***DIOEasy***, Marvin Test Solutions vector development software that also allows manual control of the DIO hardware using the built-in DIO Virtual Instrument Panel. The DIO products are also provided with the DIO driver that allows control of the DIO family from common software development tools such as ***ATEasy***, Visual Basic, C, C++ and Pascal.

DIOEasy offers a quick and simple way to generate and preview vectors and to analyze returned data in a Graphical User Interface (GUI) environment. However, using the full power of DIO requires the driver and test programs. Supported Development Environments

Concepts

To test a unit, we create a vector file using the Driver or ***DIOEasy***. This vector file contains programmed data presented to all UUT pins. Test vectors are loaded to the DIO board using the Panel window (*See Chapter 3 - Virtual Panel Description*) or calls to Driver functions. The board can be ARMED and TRIGGERED from the Panel. Result data is read from input channels and saved on the DIO board.

When the program stops, the result data can be saved to a file. The result file can be loaded and compared to a file of expected results or results from a "gold standard" to determine if the test passes or fails.

Starting DIOEasy

The easiest way to start ***DIOEasy*** is to double-click the ***DIOEasy*** icon or double-click *dioeasy.exe* in Explorer.

To start ***DIOEasy*** from a command line prompt in any of the supported 32-bit operating systems, type *C:\dioeasy\dioeasy* and press **Enter**. Use the appropriate *drive letter:\directory* where ***DIOEasy*** is installed elsewhere.

When ***DIOEasy*** is opened, the Main window displays. When a vector file is opened, the Vector View window displays. These windows are explained in the following sections.

Main Window Components

After ***DIOEasy*** is started, the Main window (Figure 2-1) displays. File, View, Window and Help menus display in the Menu bar. View has two menu items that alternately enable (checked) and disable (unchecked) display of the Toolbar and Status bar. The Window menu has only the GTDIO Panel ("the Panel") enabled (shortcut key F10). Additional menus display after a file is loaded.

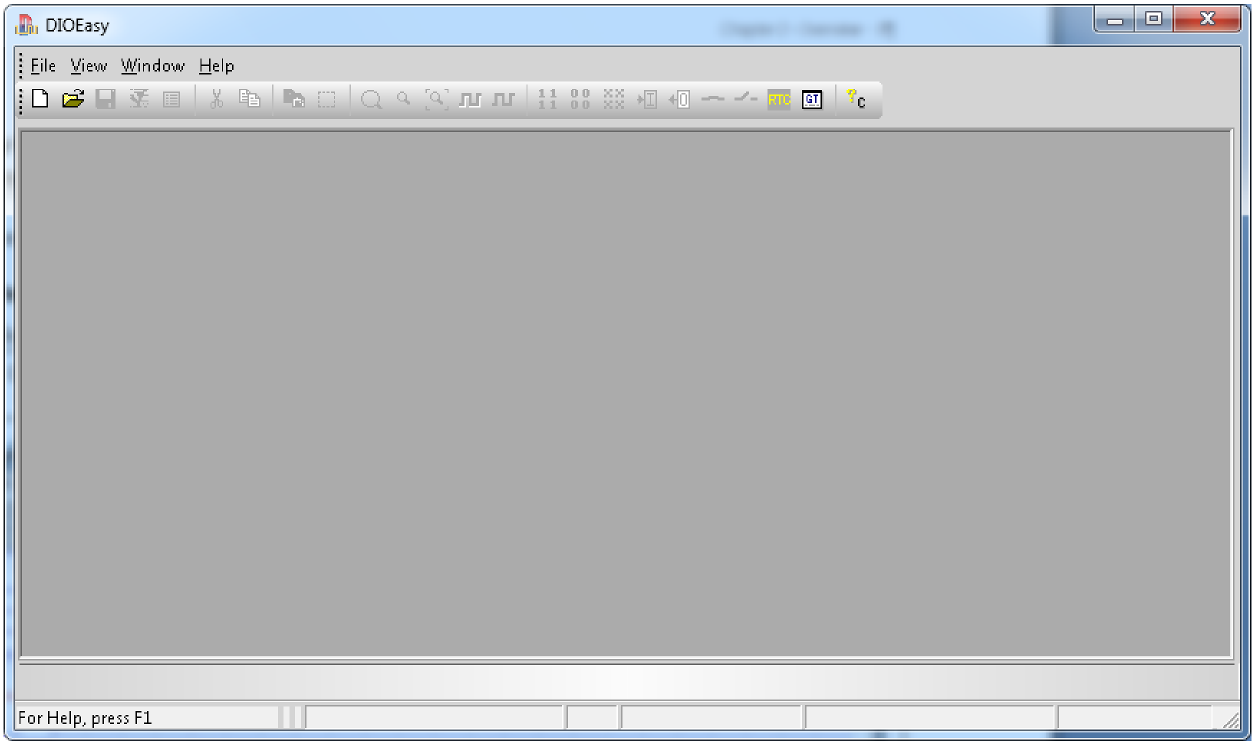


Figure 2-1: Main Window. No file is open.

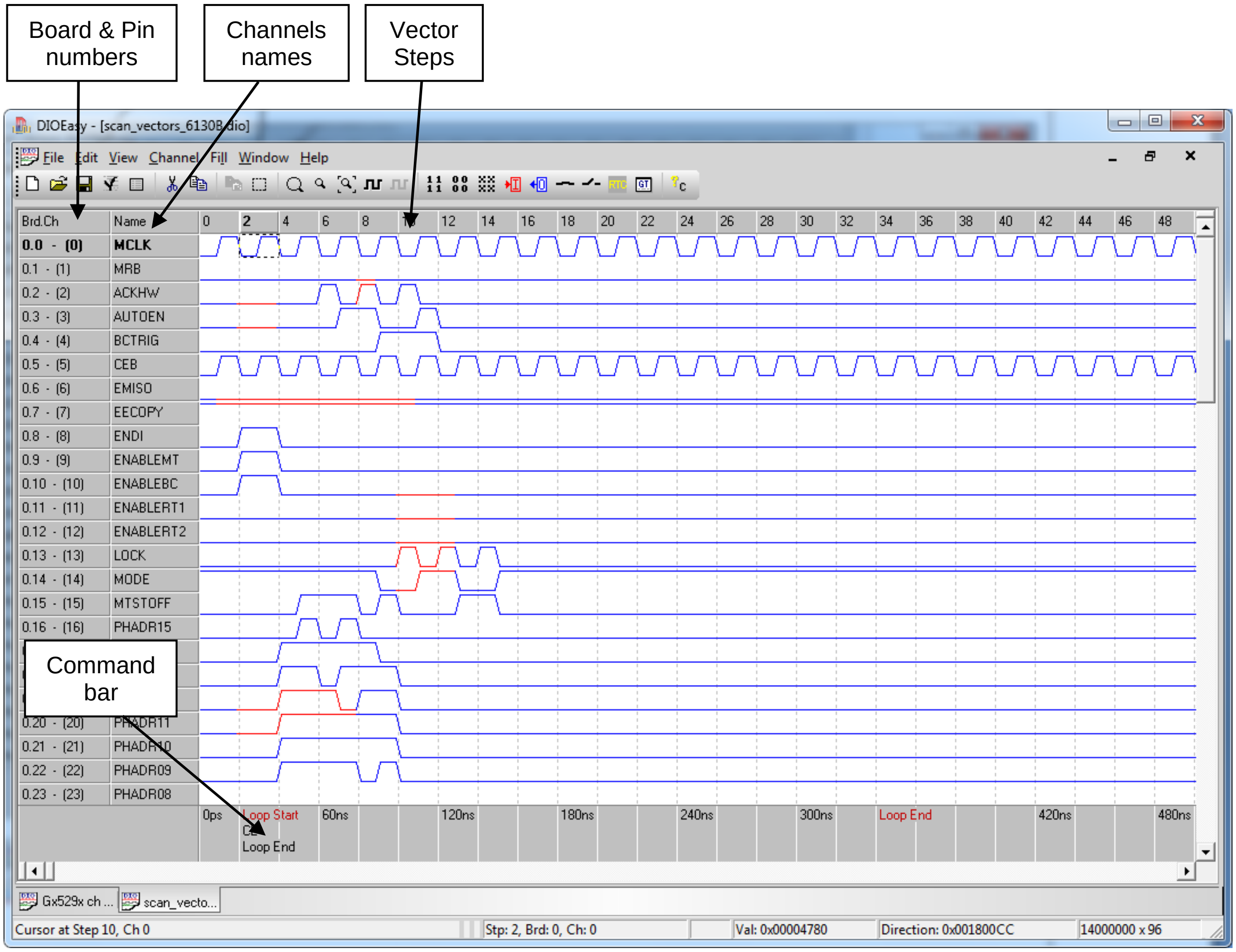
The following are ***DIOEasy*** Main window components (see operating system documentation for more information about standard Windows components):

Control Menu Box	The upper-left *DIO* Icon opens the Control menu. It contains commands to control the ***DIOEasy*** Main Window. Commands include Restore, Move, Size, Minimize, Maximize and Close.
Title Bar	Displays ***DIOEasy*** and the name of the active window Vector View file.
Menu Bar	A horizontal bar containing drop-down menus.
Toolbar	A set of buttons that launches frequently used commands with a click of the mouse.
Status Bar	Displays information about the active vector file.

Creating a New Vector File

To create a new vector file, select **New** in the File menu. A blank Vector View window opens.

Key window items display in Figure 2-2. They are explained below:



Brd.Ch	Name
0.0 - (0)	MCLK
0.1 - (1)	MRB
0.2 - (2)	ACKHW
0.3 - (3)	AUTOEN
0.4 - (4)	BCTRIG
0.5 - (5)	CEB
0.6 - (6)	EMISO
0.7 - (7)	EECOPY
0.8 - (8)	ENDI
0.9 - (9)	ENABLEMT
0.10 - (10)	ENABLEBC
0.11 - (11)	ENABLERT1
0.12 - (12)	ENABLERT2
0.13 - (13)	LOCK
0.14 - (14)	MODE
0.15 - (15)	MTSTOFF
0.16 - (16)	PHADR15
0.20 - (20)	PHADR11
0.21 - (21)	PHADR10
0.22 - (22)	PHADR09
0.23 - (23)	PHADR08

Cursor at Step 10, Ch 0 | Stp: 2, Brd: 0, Ch: 0 | Val: 0x00004780 | Direction: 0x001800CC | 14000000 x 96

Figure 2-2: Vector View Window

Title Bar	Displays the file name for the current window at the left.
Vertical Scroll Bar	Two scroll arrows and a scroll bar at the right. The arrows scroll window contents up or down one channel.
Horizontal Scroll Bar	Two scroll arrows and a scroll bar near the bottom. The arrows scroll the window contents one step right or left.
Command Bar	Displayed on the bottom of the Vector View, the command bar extends horizontally and displays commands and the step's label entered in the **Step Properties** dialog. Each step also displays the steps relative time form the start of the vector based on the Vector's frequency. Commands display at the step where they execute.

Board Number, Board Channel Number, Domain Channel Number	Channels display in the left-most column in the format *Board Number***.***Board Chan Number - (Domain Channel Number)*. Board numbers range from 0 - 15 and *board* channels range from 0 up to 31. (*board* channels can also range from 0-7 or 0-16, depending on how width is set.) *Domain* channel numbers display in parentheses and range from 0 up to 512, depending on the number of DIO boards (up to 16) and channels per board. *Domain* channel numbers are handy for identifying user pin assignments. When a board is added, it is assigned to the next higher number. Board 0 is the Master.
Channel or Channel Names	Displays in the second column. A column listing channel numbers (default) or user-assigned names. The mouse or arrow keys can select (all steps) in groups of channels.
Steps	A horizontal bar containing step numbers that display over each Vector View columns.
Vector Cells	The active portion of Vector View displays a grid pattern of channels and step data. Each vector cell contains a line representing 1, 0 or X (don't care) levels. Line colors specify whether a channel is input (default red) or output (default green). The **Options** command (View menu) permits user color choices (discussed later.)

Navigation Shortcuts

The following keyboard entries ease Vector View navigation:

Arrow keys	Move the selected cell by one cell in the indicated direction.
Shift + arrow keys	Select or deselect another cell in the indicated direction.
Page Up/Page Down	Advance the cursor one screen up/down.
Ctrl + Page Up/Ctrl + Page Down	Advance the cursor one screen left/right.
Home	Go to step 0.
End	Go to last step.
Ctrl + Home	Move 1000 steps to the left or step 0.
Ctrl + End	Move 1000 steps to the right or the last step.
Ctrl + Tab	Change the Vector View window focus.

Selecting Items

Selecting Cells in Vector View

To select any cell, click **Select** in the Edit menu. In the Select dialog box, enter First and Last values in Channel Range and Step Range. Click **OK**. The dialog box closes. Selected cells in Vector View now display highlighted.

Note: Channels are numbered in rows starting with 0. Channel numbers are fixed. This should not be confused with Brd.Chan numbers, which the user can reorder.

For cells that currently display in vector view, use the mouse.

Steps Select one or more steps. All channels are selected. Click a step number in Vector View to select it. To select a group of steps, click one and drag the mouse to the last step in the group.

Channels Selected one or more channels. All steps are selected. Click a channel in the Name or Board.Chan column. To select a group of channels, click a channel in the Name column and drag the mouse to the last channel.

Block A group of channels and steps are selected. Select a corner box and drag the mouse diagonally to the opposite corner. Rows and columns between the first and last box are block selected.

Unselect Click again to undo the selection of selected cells.

Selecting Items in a Windows List Box

Many ***DIOEasy*** list boxes support multiple item selection. The following standard Windows mouse procedure is used wherever a multiple selection is mentioned in this User's Guide. Users familiar with these methods may skip ahead.

To select a block of items in a list, click the first item, then hold down the **SHIFT** key and click the last item. The two clicked items and all items between are selected and display as highlighted.

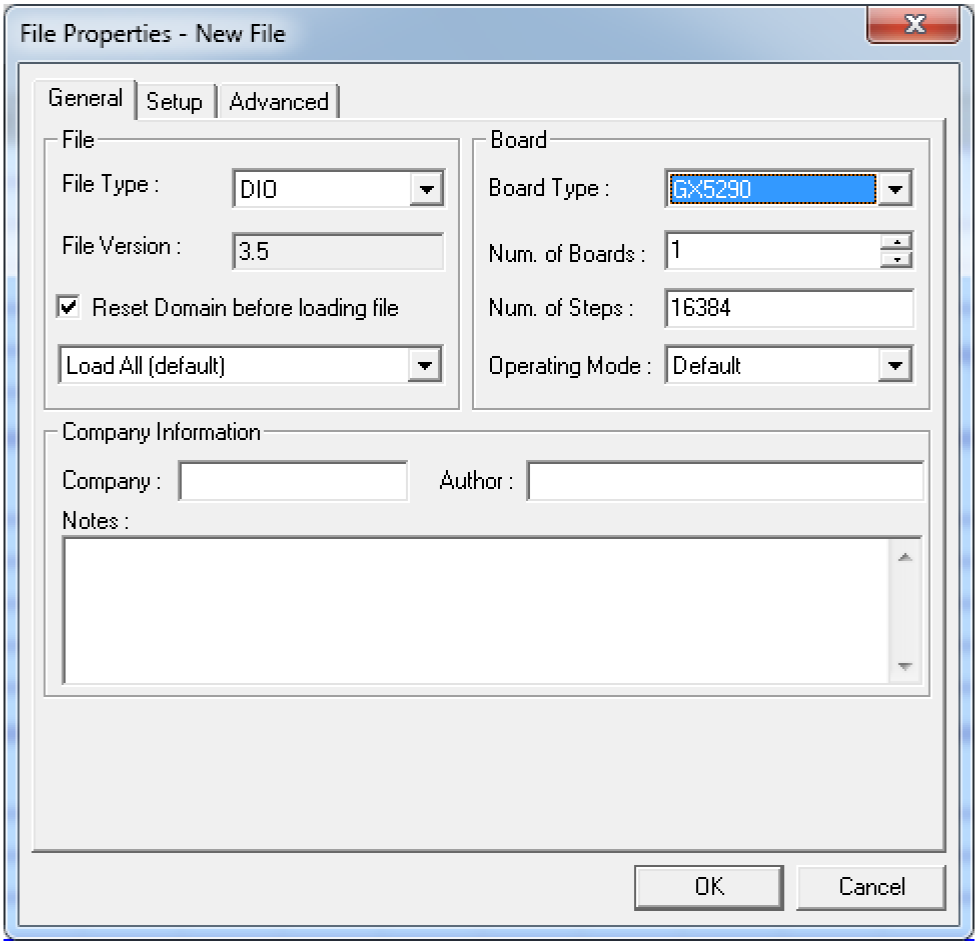
To select items in the list that do display in sequence, click the first. Now hold the **CTRL** key and click each additional item. All clicked items are now selected, but not items between.

File Menu

The commands discussed below are available from this menu.

New…

The dialog box of Figure 2-3 displays whenever **New…** is selected. Select the board type and enter the Number of Steps or accept the default value of 16384. If the number entered implies a larger memory than the selected board type supports, an error message displays.



File Type : DIO

File Version : 3.5

☑ Reset Domain before loading file

Load All (default)

Board Type : GX5290

Num. of Boards : 1

Num. of Steps : 16384

Operating Mode : Default

Company Information

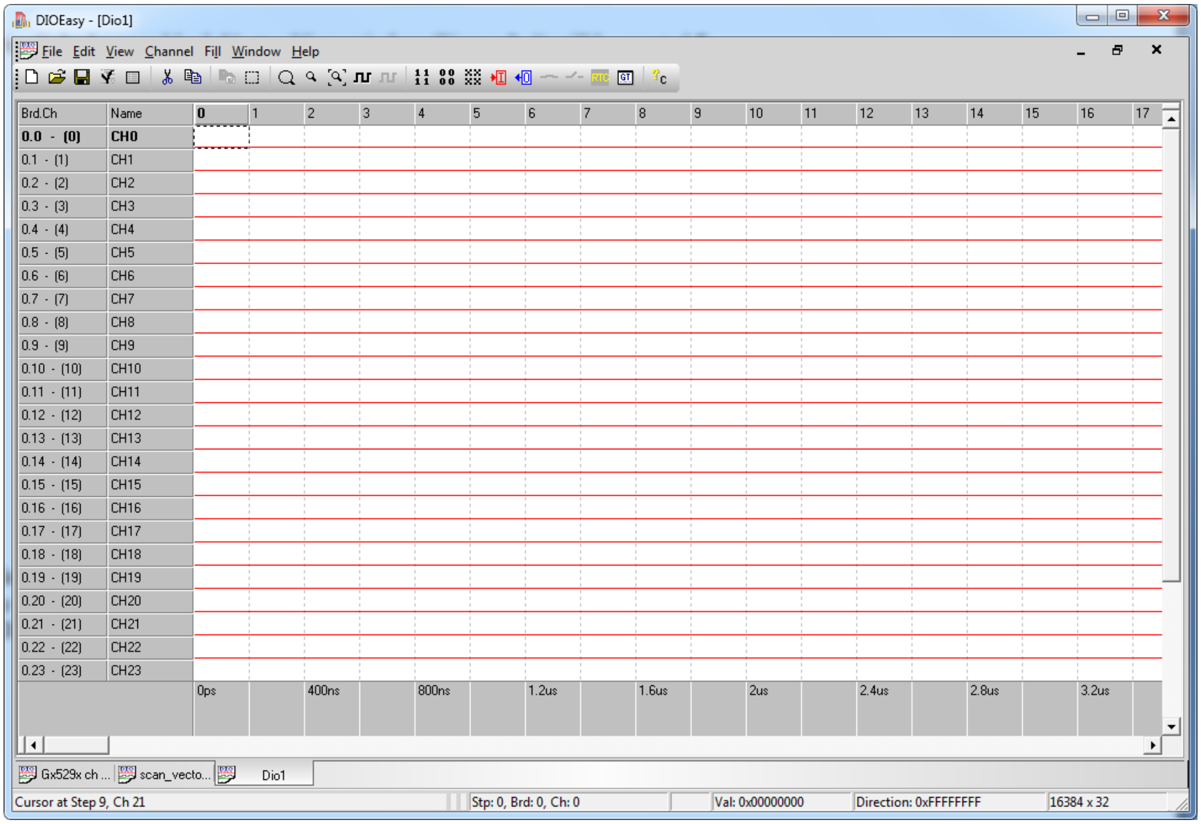
Company :

Author :

Notes :

Figure 2-3: New File Dialog for GX5290

When **OK** is clicked a new blank Vector View window (Figure 2-4) will be created.



Brd.Ch	Name
0.0 - (0)	**CH0**
0.1 - (1)	CH1
0.2 - (2)	CH2
0.3 - (3)	CH3
0.4 - (4)	CH4
0.5 - (5)	CH5
0.6 - (6)	CH6
0.7 - (7)	CH7
0.8 - (8)	CH8
0.9 - (9)	CH9
0.10 - (10)	CH10
0.11 - (11)	CH11
0.12 - (12)	CH12
0.13 - (13)	CH13
0.14 - (14)	CH14
0.15 - (15)	CH15
0.16 - (16)	CH16
0.17 - (17)	CH17
0.18 - (18)	CH18
0.19 - (19)	CH19
0.20 - (20)	CH20
0.21 - (21)	CH21
0.22 - (22)	CH22
0.23 - (23)	CH23

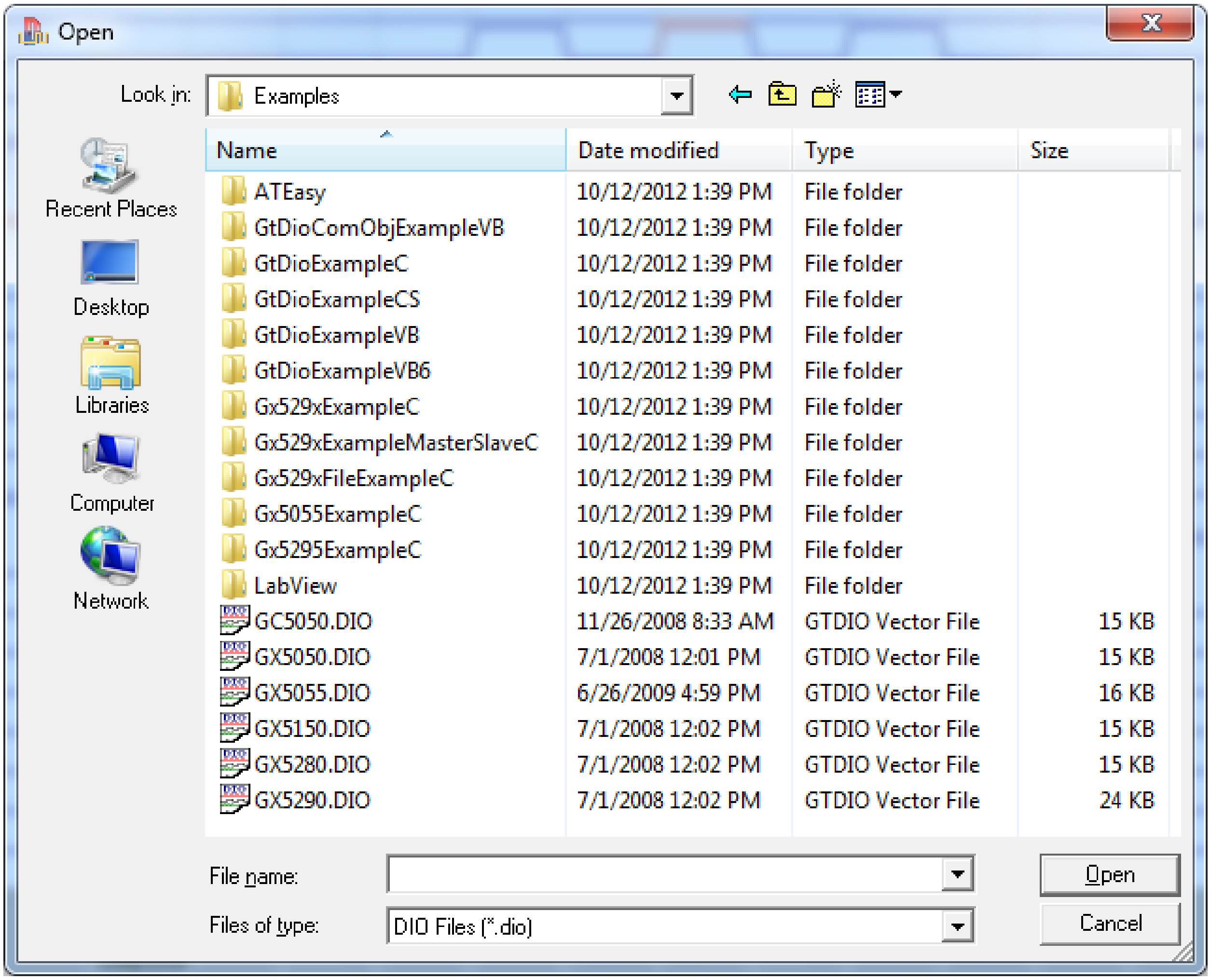
Cursor at Step 9, Ch 21 | Stp: 0, Brd: 0, Ch: 0 | Val: 0x00000000 | Direction: 0xFFFFFFFF | 16384 x 32

Figure 2-4: New Vector File Opens on Creation

This is replaced with the user-assigned file name when the file is saved. Several Vector View windows can display in the Main window (Figure 2-6). The active window's file name displays in brackets on the Main window's title bar after "DIOEasy".

Open...

The **Open…** command opens one or more saved vector files. The dialog window (Figure 2-5) display DIO files and subfolders in the default folder. The default folder is *DIOEasy*, which also contains the *DIOEasy.exe* program file. If the DIO file is in another folder, navigate to the other folder using standard Windows tools. The selected file opens and displays in a Vector View window. The new folder is now the default. The default folder reverts to *DIOEasy* when the program is reopened.



Look in: Examples

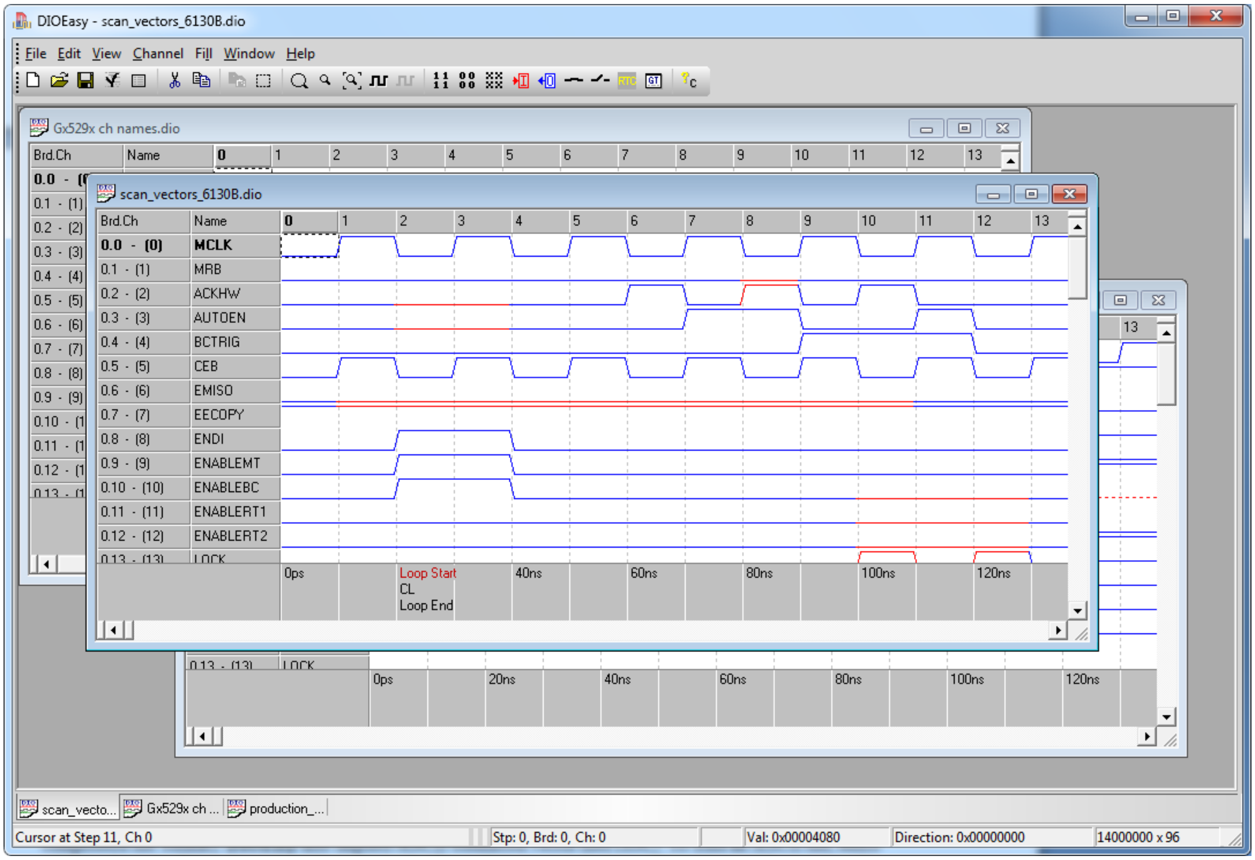
Name	Date modified	Type	Size
ATEasy	10/12/2012 1:39 PM	File folder	
GtDioComObjExampleVB	10/12/2012 1:39 PM	File folder	
GtDioExampleC	10/12/2012 1:39 PM	File folder	
GtDioExampleCS	10/12/2012 1:39 PM	File folder	
GtDioExampleVB	10/12/2012 1:39 PM	File folder	
GtDioExampleVB6	10/12/2012 1:39 PM	File folder	
Gx529xExampleC	10/12/2012 1:39 PM	File folder	
Gx529xExampleMasterSlaveC	10/12/2012 1:39 PM	File folder	
Gx529xFileExampleC	10/12/2012 1:39 PM	File folder	
Gx5055ExampleC	10/12/2012 1:39 PM	File folder	
Gx5295ExampleC	10/12/2012 1:39 PM	File folder	
LabView	10/12/2012 1:39 PM	File folder	
GC5050.DIO	11/26/2008 8:33 AM	GTDIO Vector File	15 KB
GX5050.DIO	7/1/2008 12:01 PM	GTDIO Vector File	15 KB
GX5055.DIO	6/26/2009 4:59 PM	GTDIO Vector File	16 KB
GX5150.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB
GX5280.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB
GX5290.DIO	7/1/2008 12:02 PM	GTDIO Vector File	24 KB

File name:

Files of type: DIO Files (*.dio)

Figure 2-5: File Open Command

Multiple Vector View files can be opened (Figure 2-6) in tandem.



Cursor at Step 11, Ch 0 | Stp: 0, Brd: 0, Ch: 0 | Val: 0x00004080 | Direction: 0x00000000 | 14000000 x 96

Figure 2-6: Multiple Opened Files in Vector View

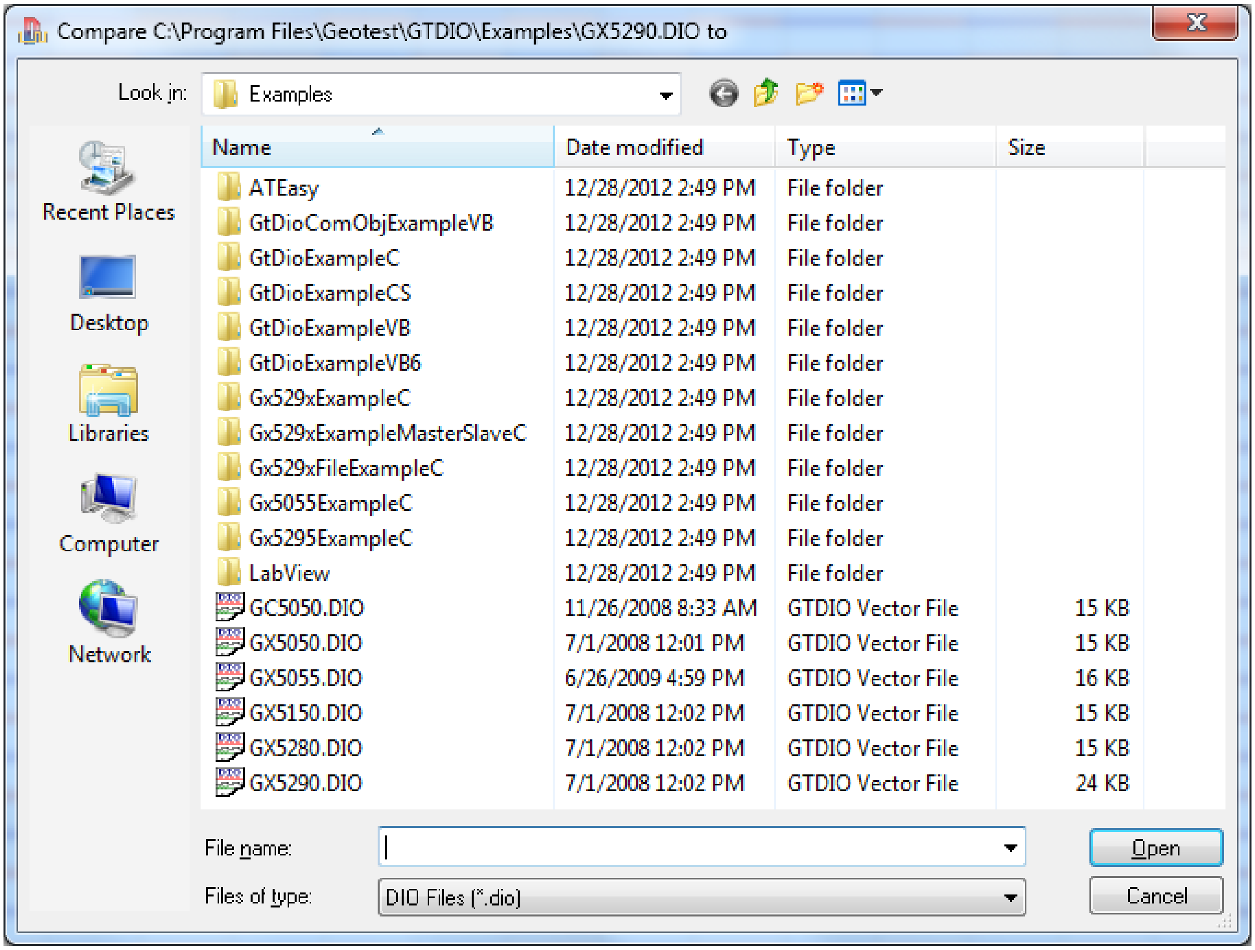
Each displays in its own window. By default, the last window opened occupies the entire Vector View area. Standard Windows controls allow the user to manipulate the window size and work in a selected window). Click anywhere on a window to select it and display it on the top.

A list of the four most recently opened files displays in the **File Menu Area** just above **Exit**. The quickest way to open a recently used file is to select it from the list in the **File Menu Area**. See the **File Menu Area topic** below.

DIOEasy uses the *DIO* default extension for vector files. The *DI* extension is used for ***DIOEasy*** result files (same compressed file format). ***DIOEasy*** also imports ASCII-formatted vector files *(ASC)*. To load an ASCII file, select *All Files* from the *Files of type* drop-down list in the **Open** dialog box and select a file with an *ASC* extension. See the Appendices for more file format information.

Compare…

Compare… selects a file to be compared against the active Vector View window. First click to activate the Vector View window to be compared, then click Compare… The Compare To dialog box as shown in Figure 2-7 is displayed. Navigate to the folder containing the desired DIO file. Click the file name and click **Open,** or else click **Cancel** to quit the dialog**. Open** closes the dialog box. The Vector View window now displays with differences highlighted. See **Comparison** in the View Menu for further information about using Compare.



Compare C:\Program Files\Geotest\GTDIO\Examples\GX5290.DIO to

Look in: Examples

Name	Date modified	Type	Size
ATEasy	12/28/2012 2:49 PM	File folder	
GtDioComObjExampleVB	12/28/2012 2:49 PM	File folder	
GtDioExampleC	12/28/2012 2:49 PM	File folder	
GtDioExampleCS	12/28/2012 2:49 PM	File folder	
GtDioExampleVB	12/28/2012 2:49 PM	File folder	
GtDioExampleVB6	12/28/2012 2:49 PM	File folder	
Gx529xExampleC	12/28/2012 2:49 PM	File folder	
Gx529xExampleMasterSlaveC	12/28/2012 2:49 PM	File folder	
Gx529xFileExampleC	12/28/2012 2:49 PM	File folder	
Gx5055ExampleC	12/28/2012 2:49 PM	File folder	
Gx5295ExampleC	12/28/2012 2:49 PM	File folder	
LabView	12/28/2012 2:49 PM	File folder	
GC5050.DIO	11/26/2008 8:33 AM	GTDIO Vector File	15 KB
GX5050.DIO	7/1/2008 12:01 PM	GTDIO Vector File	15 KB
GX5055.DIO	6/26/2009 4:59 PM	GTDIO Vector File	16 KB
GX5150.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB
GX5280.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB
GX5290.DIO	7/1/2008 12:02 PM	GTDIO Vector File	24 KB

File name:

Files of type: DIO Files (*.dio)

Figure 2-7: Compare To... dialog

Close

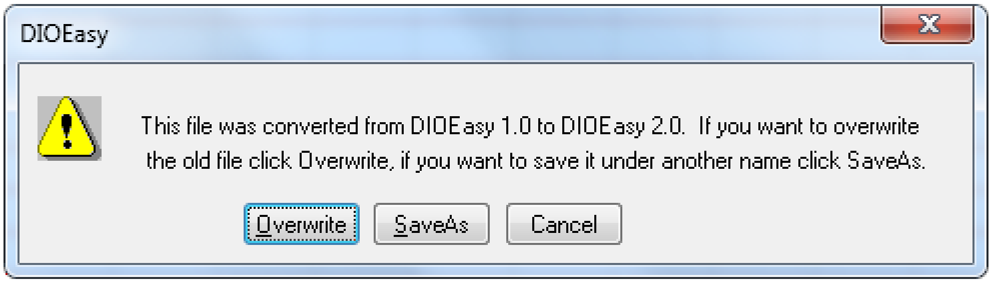
Closes the active (in-focus) Vector View window.

Save

This command saves the current active vector file under the current file name. If the file is new, **Save** defaults to **Save As….** To copy a file, use **Save As…** to avoid overwriting the original file. See **Save As…** for more information.

DIOEasy 2.0 can read files in the prior (version 1.x) format. When a vector or result file created under the older driver is loaded, it is automatically converted to the new file format when saved.

Under ***DIOEasy,*** the **Save** command defaults to **Save As…** if the file was originally opened in the old format. This gives the user an opportunity to rename the new file so it will not overwrite the old one. File edits made by ***DIOEasy*** cannot be saved in the old format.



This file was converted from DIOEasy 1.0 to DIOEasy 2.0. If you want to overwrite the old file click Overwrite, if you want to save it under another name click SaveAs.

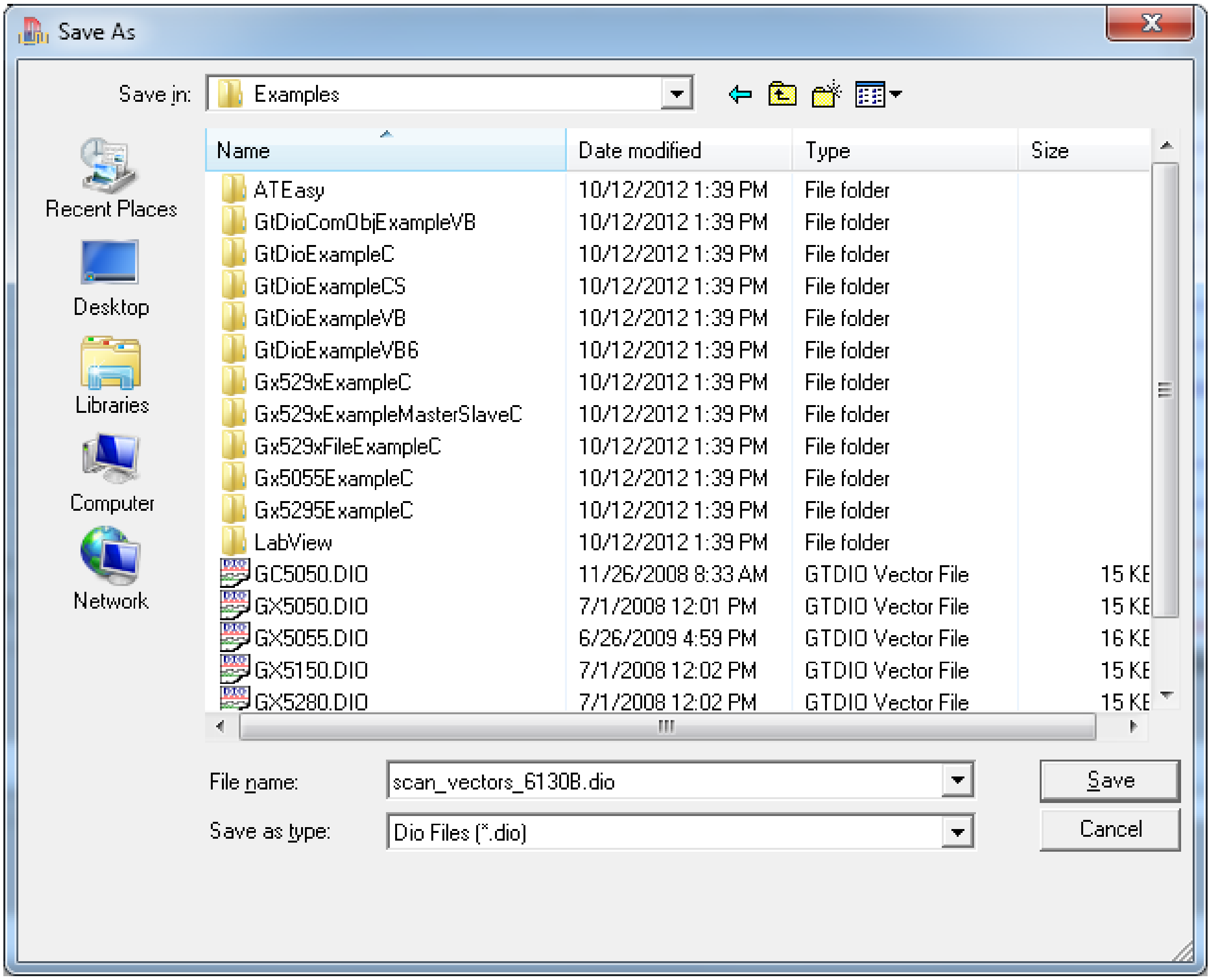
Figure 2-8: Saving in Prior File Format

Note: The user should upgrade all test stations to the new driver version as soon as possible.

Save As...

Select Save As… to save or copy the active Vector View window data to a different file, format or folder, or (optionally) to save a newly created vector file. The **Save As** dialog box (Figure 2-9) opens. Specify the file name and select or create a folder using standard Windows tools. Accept the default DIO extension or select an ASC extension in the “Save as types” combo box.

An ASC extension is used for ASCII files. Two types are available, with or without embedded commands. ASCII files are uncompressed and can be quite large, but they can be edited using any text editor that handles the file size. The **Save As** command can be used to convert ASC files to DIO files (or vice versa). See “Appendix C – ASCII File Formats at Programmer's Reference User's Guide” for more information about file formats.



Name	Date modified	Type	Size
ATEasy	10/12/2012 1:39 PM	File folder	
GtDioComObjExampleVB	10/12/2012 1:39 PM	File folder	
GtDioExampleC	10/12/2012 1:39 PM	File folder	
GtDioExampleCS	10/12/2012 1:39 PM	File folder	
GtDioExampleVB	10/12/2012 1:39 PM	File folder	
GtDioExampleVB6	10/12/2012 1:39 PM	File folder	
Gx529xExampleC	10/12/2012 1:39 PM	File folder	
Gx529xExampleMasterSlaveC	10/12/2012 1:39 PM	File folder	
Gx529xFileExampleC	10/12/2012 1:39 PM	File folder	
Gx5055ExampleC	10/12/2012 1:39 PM	File folder	
Gx5295ExampleC	10/12/2012 1:39 PM	File folder	
LabView	10/12/2012 1:39 PM	File folder	
GC5050.DIO	11/26/2008 8:33 AM	GTDIO Vector File	15 KB
GX5050.DIO	7/1/2008 12:01 PM	GTDIO Vector File	15 KB
GX5055.DIO	6/26/2009 4:59 PM	GTDIO Vector File	16 KB
GX5150.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB
GX5280.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB

File name: scan_vectors_6130B.dio

Save as type: Dio Files (*.dio)

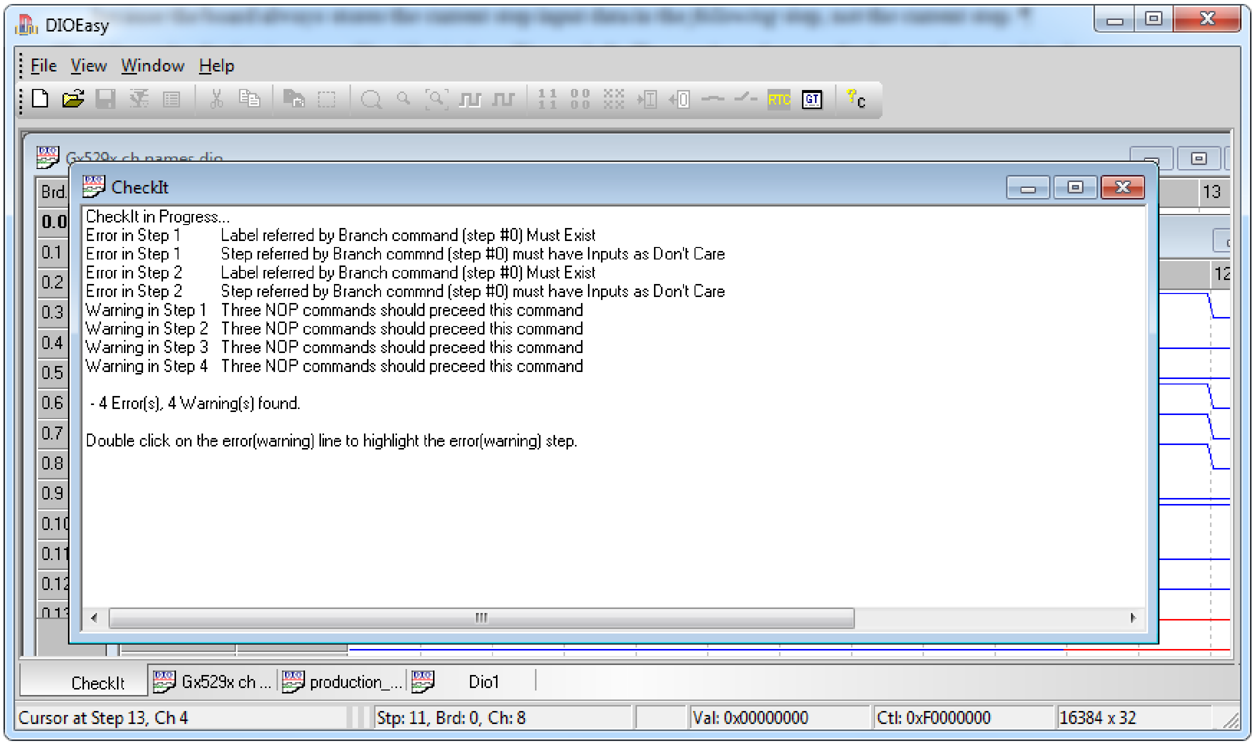
Figure 2-9: Using the ‘Save As’ Command.

CheckIt!

CheckIt! checks the current vector file format. The following rules are checked:

1. Branch command labels must exist; that is, the target address must have the specified label (**JUMP**, **LOOP**, **CALL**).
2. Near branch command labels (**JUMP NEAR**, **LOOP**) must exist in the current 128K page.
3. **CALL** label step numbers must be aligned to an 8 step boundary (for example, 8, 16…).
4. Program steps with labels referred by a branch command should contain a **NOP** command and should be followed by two more **NOP** commands.
5. Three **NOPs** must exist between every two commands of GT50.
6. Program steps with a label referenced by branch commands must have all inputs defined as "don't care". This is because the board always stores the current step input data in the *following* step, not the current step.

CheckIt results display in a new *CheckIt* window (Figure 2-10). The number of errors displays at the top of the list followed by a detailed breakdown. One Error can mask others or generate several error codes. Error code details are in “Appendix B – Error Codes at Programmer’s Reference User’s Guide”.



```
CheckIt in Progress...
Error in Step 1      Label referred by Branch command (step #0) Must Exist
Error in Step 1      Step referred by Branch commnd (step #0) must have Inputs as Don't Care
Error in Step 2      Label referred by Branch command (step #0) Must Exist
Error in Step 2      Step referred by Branch commnd (step #0) must have Inputs as Don't Care
Warning in Step 1    Three NOP commands should preceed this command
Warning in Step 2    Three NOP commands should preceed this command
Warning in Step 3    Three NOP commands should preceed this command
Warning in Step 4    Three NOP commands should preceed this command

 - 4 Error(s), 4 Warning(s) found.

Double click on the error(warning) line to highlight the error(warning) step.
```

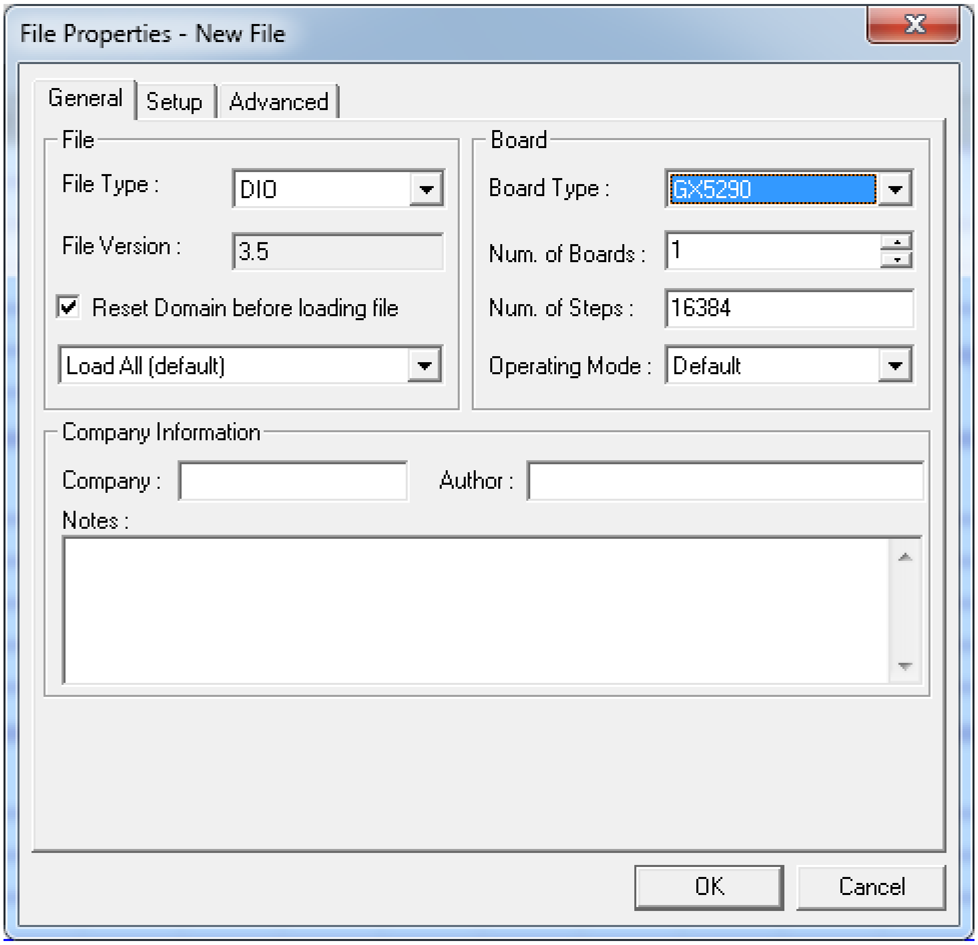
Cursor at Step 13, Ch 4 | Stp: 11, Brd: 0, Ch: 8 | Val: 0x00000000 | Ctl: 0xF0000000 | 16384 x 32

Figure 2-10: CheckIt! Window

Properties...

Properties… opens the File Properties dialog (Figure 2-11) containing the General tab (default) and Setup tab. These windows display information about the current Vector View file. Some fields are editable depending on configuration.

Click **OK** to save properties and close. Click **Apply** to save changes and continue editing. Click **Cancel** (or the Esc key) to close without applying changes.



File Properties - New File

General | Setup | Advanced

File

File Type : DIO

File Version : 3.5

☑ Reset Domain before loading file

Load All (default)

Board

Board Type : GX5290

Num. of Boards : 1

Num. of Steps : 16384

Operating Mode : Default

Company Information

Company :

Author :

Notes :

Figure 2-11: File Properties General Dialog Box

Properties: General Window

The General tabbed window (Figure 2-11) contains three groups: File, Board and Company Information.

The following fields are available in the File group (* means not editable):

File Type*	File header data.
File Version*	The version of ***DIOEasy*** that generated the current file.
Reset Domain before loading file (Checkbox)	Check will not reset all the boards in the domain before loading the new settings from the DIO file (default is to always reset the domain).
File Load Options (Liast Box): • Load All (default) • Load Vector Data only • Load Settings only	Check will only load the file data to the boards' in the domain.

The following fields are available in the Company Information group:

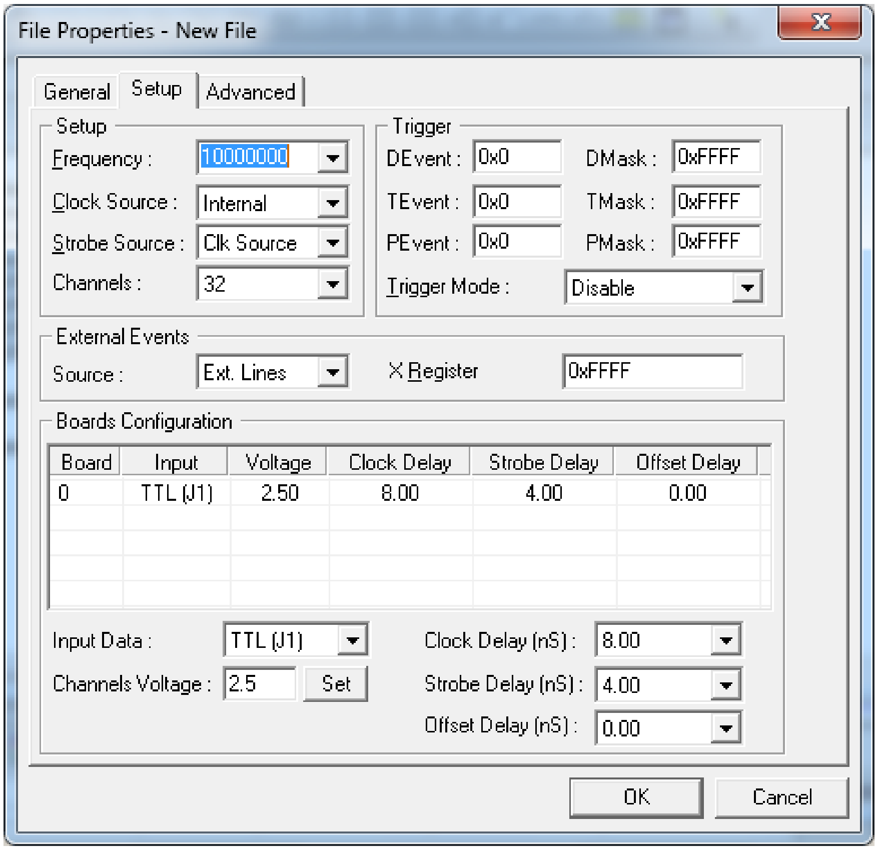
Company	Optional company name or other information.
Author	Optional name of the author or person responsible.
Notes	Optional text.

The following fields are available in the Board group:

Number of Boards	The number of boards used by this vector file, from one to eight, that this vector file uses. The default is the last value or one.
Board Type*	Not editable. The vector's board type.
Number of Steps	Number of steps that the vector file will have if the number of steps exceedsthe board type available memory an error message displays. If there are fewer steps entered than in the current vector file, a message warns that the file will be truncated.

Properties: Setup Window

The Setup tabbed window (Figure 2-12) contains six groups: Setup, Trigger, External Events, Board Channels, Registers and Direction.



Setup

Frequency : 10000000

Clock Source : Internal

Strobe Source : Clk Source

Channels : 32

Trigger

DEvent : 0x0 DMask : 0xFFFF

TEvent : 0x0 TMask : 0xFFFF

PEvent : 0x0 PMask : 0xFFFF

Trigger Mode : Disable

External Events

Source : Ext. Lines X Register 0xFFFF

Boards Configuration

Board	Input	Voltage	Clock Delay	Strobe Delay	Offset Delay
0	TTL (J1)	2.50	8.00	4.00	0.00

Input Data : TTL (J1) Clock Delay (nS) : 8.00

Channels Voltage : 2.5 Strobe Delay (nS) : 4.00

Offset Delay (nS) : 0.00

Figure 2-12: File Properties Setup Dialog Box

The following fields are available in the Setup group:

Frequency	Enter or view the internal clock frequency in Hz (5 MHz default). This is *not* editable if Clock Source is set to *EXTERNAL*
Ext Frequency	Editable if Clock Source is set to *PRG BY EXT*. It displays the selected external clock frequency. If the combined external and internal frequencies are outside the allowed range, the nearest available clock frequency displays.
Clock Source	Select *INTERNAL* (default), *PRG BY EXT* or *EXTERNAL*. This sets the clock to an internally generated frequency, a frequency synchronized to an external reference clock, or an external clock source, respectively.
Strobe Delay	Strobe Delay in nanoseconds the clock *lags* the Strobe by the selected amount.
Strobe Source	This field can be selected only if Clock Source is set to *EXTERNAL*. Select either CLK SOURCE or EXT STROBE to strobe on the external clock or use another external source.

The following fields are available in the Trigger group:

DEvent D Event Register displays as hex value, represents the state of the External Event Lines for a D Trigger Event to occur.

TEvent T Event Register displays as hex value, represents the state of the External Event Lines for a T Trigger Event to occur.

PEvent P Event Register displays as hex value, represents the state of the External Event Lines for a P Pause Event to occur.

DMask Four hex characters in the D Mask Register. 0xFFFF enables all bits. A zero value will set the bit condition as true. A 0x0000 will cause the D event to always to be true.

TMask Four hex characters in the T Mask Register. 0xFFFF enables all bits. A zero value will set the bit condition as true. A 0x0000 will cause the T event to always to be true.

PMask Four hex characters in the P Mask Register. 0xFFFF enables all bits. A zero value will set the bit condition as true. A 0x0000 will cause the T event to always to be true.

Trigger Mode Select DISABLE (default), or select D LEVEL, T LEVEL, DT LEVEL or TD LEVEL to enable Trigger events based upon External Event Inputs or X Bus words. In addition, the Pause or P Event is always active whenever a Trigger Mode selection is other than DISABLE.

The following fields are available in the External Events group:

Source Use radio buttons to select external lines or internal (X register) event simulation.

X Register Enter a hex number into the X Register. This is used for simulation of an external event.

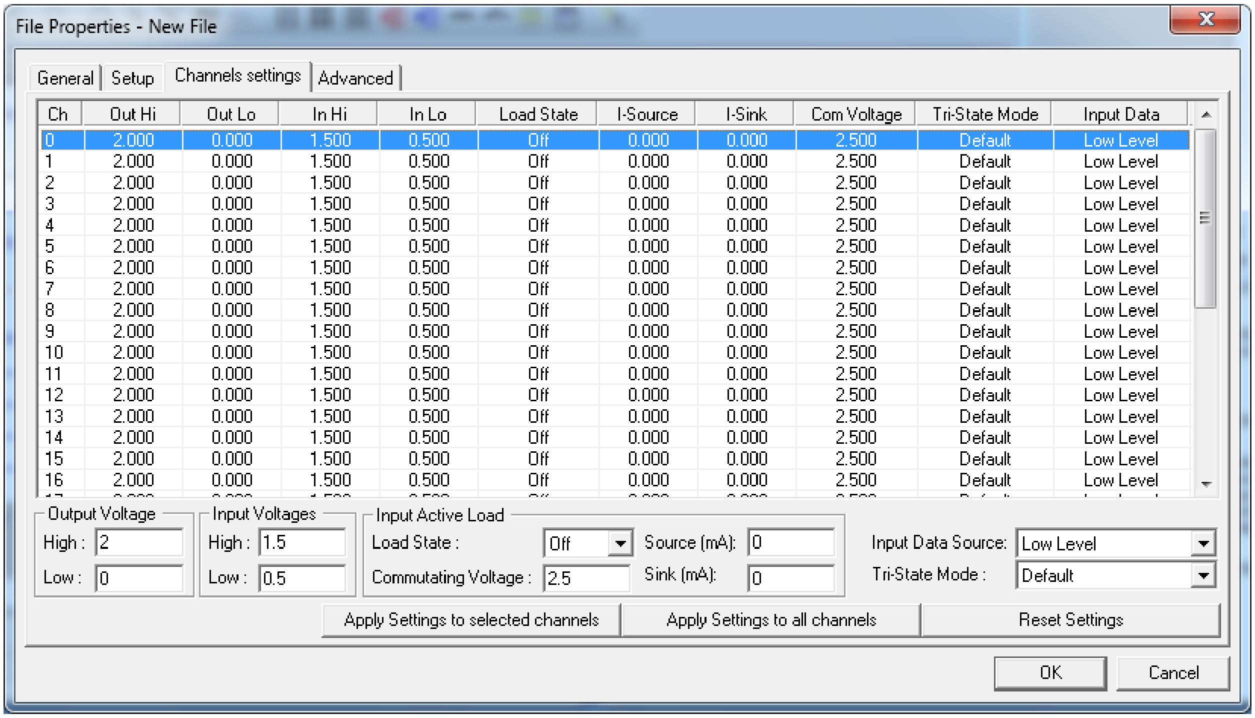
The following fields are available in the Board Channels group for a GX5150/GX5280/ GX5290 file, but are not available for a GX5050 file:

Bord Number Select the number of the board you want to specify active channels for.

Number of Channels The number of channels (channels) to be available (width, the Vector step capacity (Depth) scales inversely.

Properties: Channels settings Window

The Channels settings tabbed window (Figure 2-12) contains six groups: Setup, Trigger, External Events, Board Channels, Registers and Direction.



Ch	Out Hi	Out Lo	In Hi	In Lo	Load State	I-Source	I-Sink	Com Voltage	Tri-State Mode	Input Data
0	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
1	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
2	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
3	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
4	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
5	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
6	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
7	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
8	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
9	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
10	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
11	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
12	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
13	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
14	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
15	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level
16	2.000	0.000	1.500	0.500	Off	0.000	0.000	2.500	Default	Low Level

Output Voltage
High : 2
Low : 0

Input Voltages
High : 1.5
Low : 0.5

Input Active Load
Load State : Off
Source (mA): 0
Commutating Voltage : 2.5
Sink (mA): 0

Input Data Source: Low Level
Tri-State Mode : Default

Apply Settings to selected channels
Apply Settings to all channels
Reset Settings
OK
Cancel

Figure 2-13: File Properties Gx5295 Channles Settings

Output Voltage High Group:

High   Displays/set the channel output high voltage, voltage can be set from -2V to +7.0V and need to be higher than the output low voltage settings.

Low   Displays/set the channel output low voltage, voltage can be set from -2V to +7.0V and need to be less than the output high voltage settings.

Input Voltage High Group:

High   Displays/set the channel input high threshold voltage, voltage can be set from -2V to +7.0V and need to be higher than the input low threshold voltage settings.

Low   Displays/set the channel input low threshold voltage, voltage can be set from -2V to +7.0V and need to be less than the input high threshold voltage settings.

Input Active Load Group:

Load State   Displays/set the channel input active load state; the load state can be turned On or Off

Commutating voltage   Displays/set the channel input active load sink and source currents commutating voltages value, voltage can be set from --2V to +7V.

Source (mA)   Displays/set the channel input active load source current settings; source current can be set form 0 to 24mA.

Sink (mA)   Displays/set the channel input active load sink current settings; sink current can be set form 0 to 24mA.

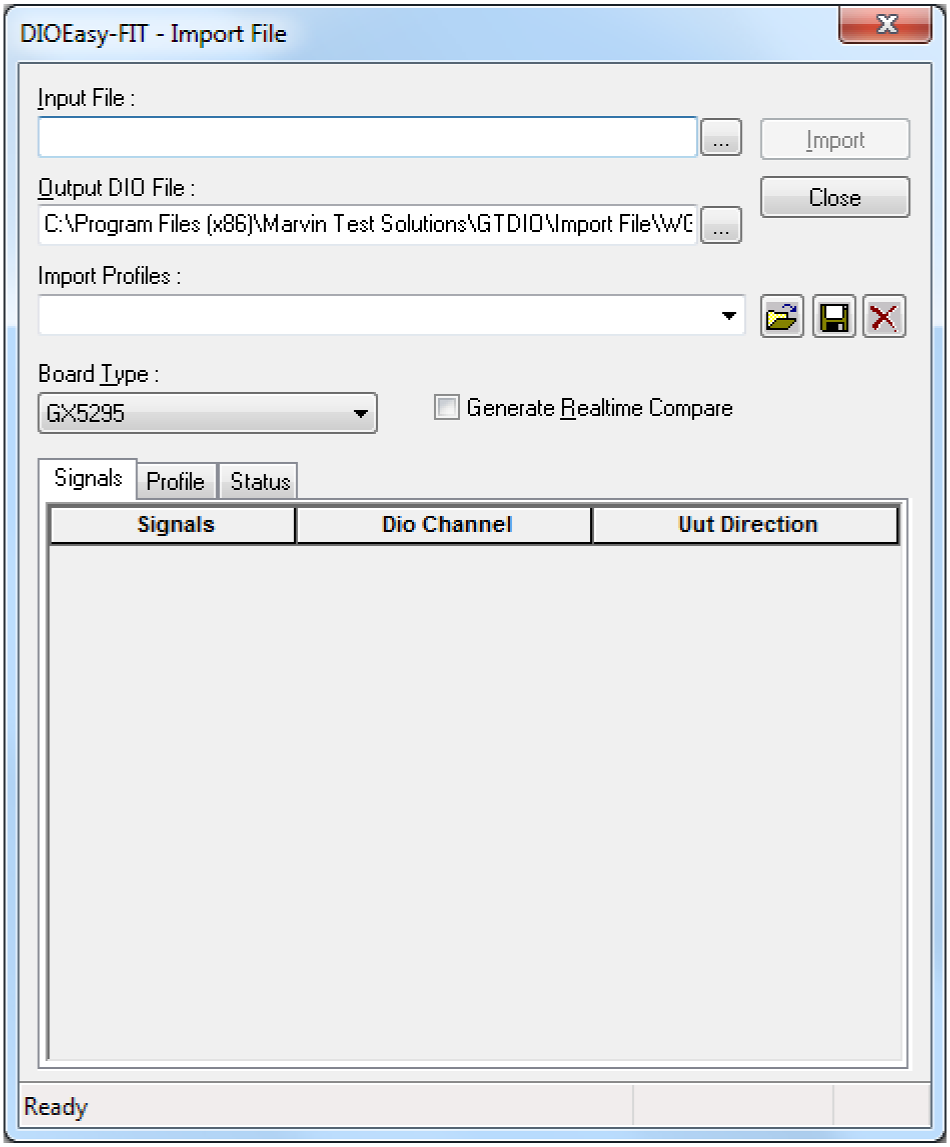
Input Data Source

Displays/set the channel input data comparator source, Input data comparator can be as follows:\n\rLow Threshold Comparator: data will be the result of the comparison done on the input data by the low threshold comparator, logic levels are as follows: \n\r\tLogic Low: whenever the input signal is below the low comparator threshold\n\r\tLogic High: whenever the input signal is above the low comparator threshold\n\r\n\rHigh Threshold Comparator: data will be the result of the comparison done on the input data by the high threshold comparator, logic levels are as follows: \n\r\tLogic Low: whenever the input signal is below the high comparator threshold. \n\r\tLogic High: whenever the input signal is above the high comparator threshold.

Import File …

The command will open the File Import dialog that is used to import WGL/STIL/VCD/eVCD and ATP file types and convert them to a GTDIO file. Using the dialog requires a GTDIO-FIT license to be purchased and installed on your machine.

You open the **Import File Dialog** by selecting **Import…** from the **GTDIO Panel File** menu.



Output DIO File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\WG

Board Type : GX5295

□ Generate Realtime Compare

Signals	Dio Channel	Uut Direction

Ready

Figure 2-14: Import File Dialog

For more information about see Chapter 3 - Importing External Vector Files

Exit

Exit closes ***DIOEasy***. The user is prompted to save new or changed files.

Edit Menu

The commands discussed below are available from this menu.

Cut

Deletes a selected area from the Vector View window and copies it to the clipboard. See the **Select** and **Delete** commands for more information.

Copy

Copies the selected vector data and commands to the clipboard. See the **Select** command for more information.

Paste

Takes a copy of the data from the clipboard and overwrites existing data at the selected area or point. If no area is selected, data is overwritten starting from the cursor location and proceeding to the right and down. If an area is selected, that selection will be overwritten. In this case, the selected area shape must match the clipboard copy. See the **Select** command for more information.

Delete

Deletes data, commands or both from a selected area, step or channel. The **Delete** commands function according to the selected area type:

Data — Deletes selected vector data. Remaining data is moved down the number of steps deleted. Data at the high end is replaced by zeros. Commands and labels remain at the same step. Only data is deleted.

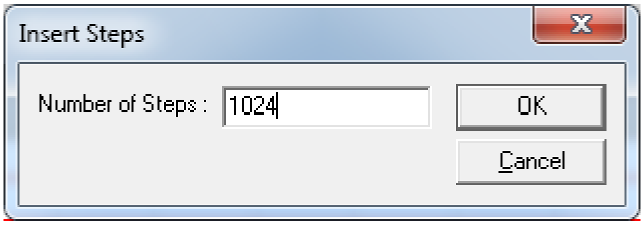
Commands — Deletes selected command steps. Remaining commands are moved down the number of commands deleted. Steps at the end are replaced by NOPs. Jump references to and from deleted steps are voided.

Both — Deletes both data and commands at the selected step(s) and channel(s).

Insert Steps...

Fill values can be zeros, ones, random or don't care. The value of data to be filled in new cells is set in the New Cells Fill group, in the Options dialog under the View menu. The fill value should be set before using Insert Steps… to avoid manual corrections.

This command is active when one or more step columns are selected in Vector View. The insertion point is the lowest numbered step selected. A submenu lists Data, Command or Both. Choosing any one opens the same Insert Steps dialog (Figure 2-15). Enter the number of steps to be inserted (default 0 will not insert any steps).



Number of Steps : 1024

Figure 2-15: Insert Steps Dialog Box

Click OK to insert steps at the insertion point. All steps after the insertion point will be moved to the right by the number of steps inserted. Inserted steps are filled in accordance with New Cells Fill value selected as described above.

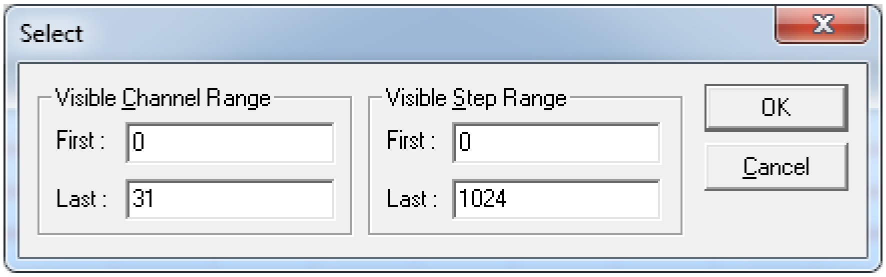
Data Only data steps are inserted at the insertion point.

Command Only command steps are inserted at the insertion point. Jump references are adjusted to point to the same labels. For a GX5150 domain, the number of steps inserted must be a multiple of 2, 4 or 8 depending on the smallest Board width set for the domain, or an error message displays.

Both Both Data and Command steps are inserted in accordance with the above. **Restrictions for the GX5150 domain apply**.

Select...

To specify a range of steps and channels to be selected, click this command. The **Select** dialog box displays in Figure 2-16:



Visible Channel Range
First : 0
Last : 31

Visible Step Range
First : 0
Last : 1024

Figure 2-16: Select Dialog Box

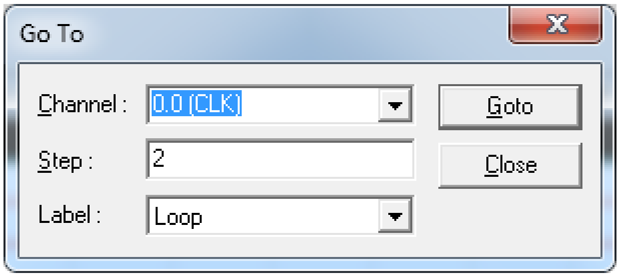
Enter first and last channels and step numbers, and then click **OK**. All fields must have valid data.

This dialog box method is useful for selecting a step range that is off the screen. The mouse method is convenient when the desired range displays or can be accessed easily by scrolling.

Note: It is easier to select with the mouse. To select a block using the mouse in Vector View, click a step-channel, drag the mouse to the opposite corner of the desired area and release the button. The desired area is selected and highlighted.

GoTo...

The **GoTo** dialog box in Figure 2-17 changes the active Vector View position. This is convenient for channels and especially steps that extend beyond the displayed part of the Vector View window.



Channel : 0.0 (CLK)

Step : 2

Label : Loop

Figure 2-17: GoTo Dialog Box

Enter a channel number or select a channel label from the **Channel** combo box. Enter a step number in the **Step** text box, or select a step label. Click **OK**. The Vector View window immediately displays the new highlighted channel-step position as if the user had scrolled there using the minimum number of steps.

The arrow keys can also be used to navigate to a cell position. If the position is past the window boundary, the Vector View display scrolls one cell position for each press of the arrow key. The GoTo dialog also displays by double-clicking the far left area of the Command bar.

Find Direction Forward

Sets the find direction for all the following Find Next menu items:

- Find Next Command
- Find Next Label
- Find Next Command/Label
- Find Next Signal Change

When clicked a check mark appears next to the Find Direction Forward menu item designating that the above Find Next item will be done searching forward from current selected step in the vector view, if unchecked it will search form this step backwards.

Find Next Command

Searches for the next Command in the vector view.

Find Next Label

Searches for the next label in the vector view.

Find Next Command/Label

Searches for the next label and/or command in the vector view.

Find Next Signal Change

Searches for the signal change in the vector view.

Find Next Failed Step

This command searches two files for Vector View steps that have a difference on at least one channel. Differences are highlighted in Vector View after executing a **Compare** operation (in the **File** menu). The cursor advances to the next step in Vector View where differences between files exist, scrolling as necessary. The search proceeds horizontally across steps.

This function is enabled only after a **Compare** operation (see the File menu) is set up. Use the **F3** shortcut for repetitive searches.

Find Next Failed Cell

This command searches for cells that have differences (comparison errors) between two files. Errors are detected and highlighted in Vector View after executing the **Compare** function (**File** menu).

The cursor advances to the next error, scrolling the Vector View display as necessary. The search proceeds vertically down channels, then to the next step in a serpentine pattern.

This function is enabled after a **Compare** is performed (see the File menu). The **Ctrl+F3** shortcut is easier to use for repetitive searches.

View Menu

The commands discussed below are available from this menu.

Toolbar

When this item is selected the Toolbar in Figure 2-18 alternately displays (checked) and hides (unchecked).

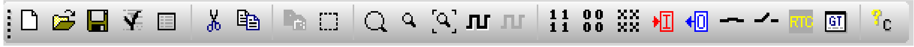


Figure 2-18: DIOEasy Toolbar

Pass the mouse over a button to view a short description of its function. Toolbar buttons are organized into five groups with related functions. The last group is for help only. Some buttons are available for one type board.

Each button corresponds to a menu function.

New DIO file

Open file

Save

Check the current active vector

Display the file properties dialog

Deletes a selected area from the Vector View window and copies it to the clipboard. See the **Select** and **Delete** commands for more information.

Copies the selected vector data and commands to the clipboard. See the **Select** command for more information.

Paste the data from the clipboard and overwrites existing data at the selected area or point.

Select vector data

Zoon in

Zoon out

Zoom out and fit the whole vector in the vector view.

Show the selected channels.

Hide the selected channels.

Fill the selected vector area with all 1's, 0's or don't cares.

Sets the selected vector area to output or input.

The Output Enable and Output Disable buttons enable and disable the output of a GX515X in groups of eight channels. To use this function, select an output channel in the desired group in Vector View and click a button to enable or disable the group.

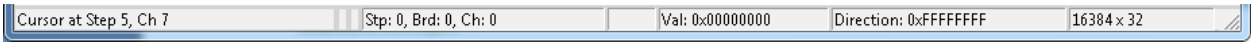
RTC Open the Real Time Compare data dialog

GT Opens the DIO virtual panel.

?c Display the DIOEasy About splash screen.

Status Bar

The Status Bar in Figure 2-19 displays under the Command Bar.



Cursor at Step 5, Ch 7 | Stp: 0, Brd: 0, Ch: 0 | Val: 0x00000000 | Direction: 0xFFFFFFFF | 16384 x 32

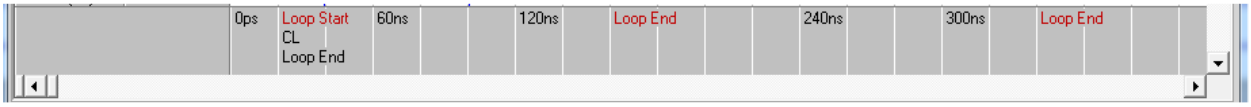
Figure 2-19: DIOEasy Status Bar

Step and channel number of the cell under the mouse cursor display at the left. Array size in steps and channels display at the right.

The second box from the left displays Channel and Step number of a selected Vector View cell. The step value displays at the second box from the right as binary number with Channel 0 the least significant bit. The Value can display as hex or binary, according to the Options setup in the View menu. The Value depends on channel number only. The third box from the right displays the direction of the step.

Command Bar

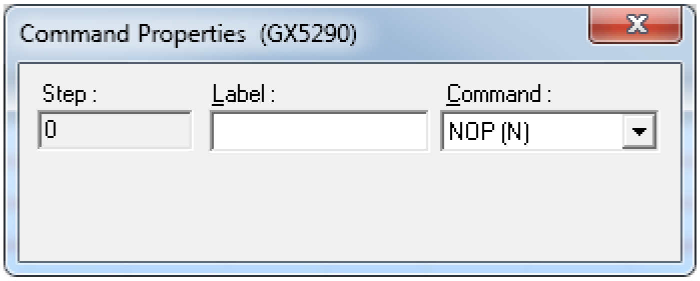
The *Command Bar* in Figure 2-20 displays between the Vector View windows and the Status Bar. Commands display in step columns.



0ps | Loop Start
CL
Loop End | 60ns | 120ns | Loop End | 240ns | 300ns | Loop End

Figure 2-20: DIOEasy Command Bar

Command Bar steps display commands and labels that are associated with a step. Each step also displays the steps relative time form the start of the vector based on the Vector's frequency. An instruction can be inserted or edited by double-clicking on **Command Bar** at any step. This action brings up the *Command Property* dialog box. The default step number that displays is the step that is double-clicked. The board type is identified in the *Command Properties* Window Title bar (Figure 2-21). The opening dialog box is the same for all DIO boards, but commands and options differ.



Command Properties (GX5290)

Step : 0

Label :

Command : NOP (N)

Figure 2-21: New Command Properties Dialog

Select a command. Enter a label and, if necessary, a condition from the appropriate lists. Depending on the command selected, additional selections may appear. Press ENTER to store the command. The command now displays on the Command Bar.

Select another step on the Command Bar and repeat until all commands have been set up. Close the window. All labels and commands now display on the command on the Command Bar.

To edit a command, double-click it on the Command Bar.The Command Properties dialog box opens for editing.

Comparison

Comparison is used to alternately show (checked) and hide (unchecked) differences between the compared files. When a DIO or DI file is selected for comparing against the active Vector View file (see **Compare** in the File menu). It is easier to use the **F8** shortcut to alternate views. Differences are crosshatched as shown in Figure 2-22:

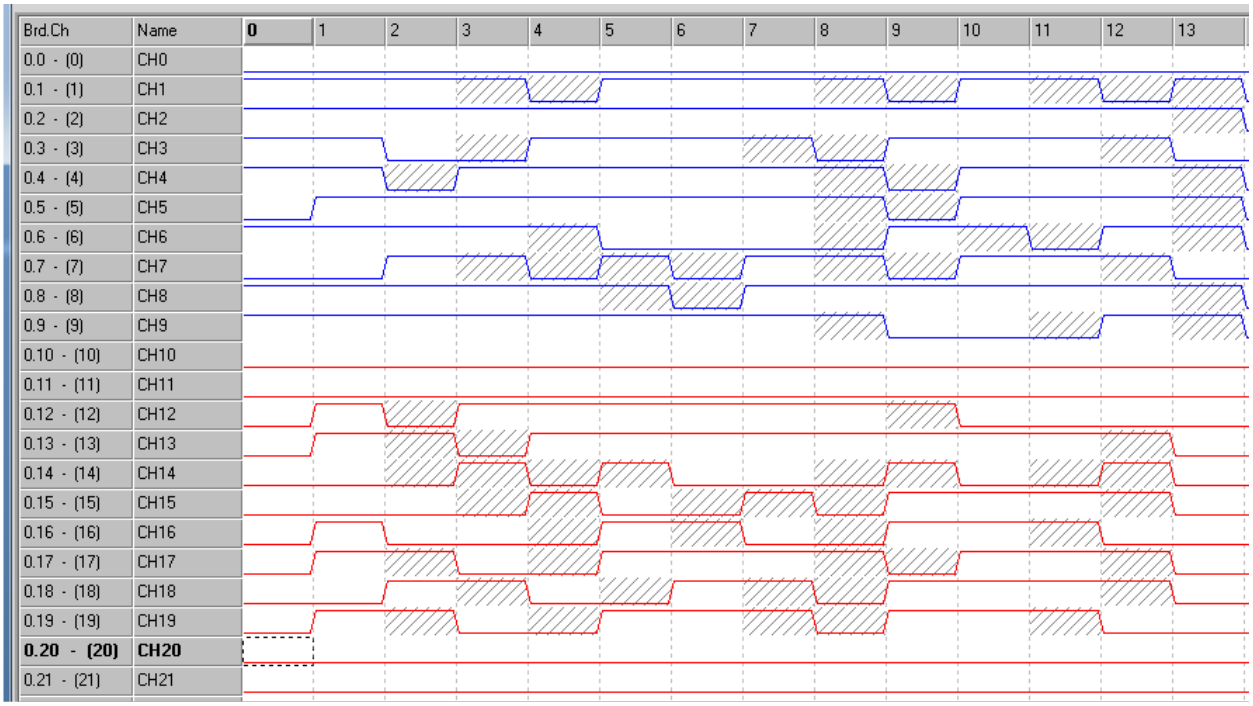
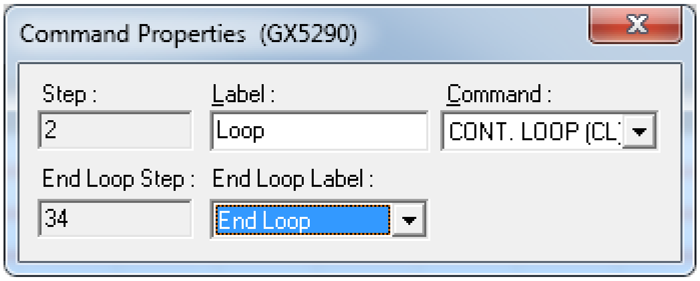


Figure 2-22: Comparing Two Files

The path to the selected comparison file displays [in brackets] on the Vector View title bar, after the Vector View file name. Comparison is available only if **Compare** was previously selected in the File menu.

Command Properties

Command Properties displays the Command Properties dialog box of Figure 2-23:



Step : 2

Label : Loop

Command : CONT. LOOP (CL

End Loop Step : 34

End Loop Label : End Loop

Figure 2-23: Command Properties Dialog Box

The displayed dialog for existing properties depends on the command. Refer to the “Theory of Operation” sections of the specified board type User Guide for a list of commands and conditions that are available.

Zoom In / Zoom Out / Zoom Fit

It is usually easiest to use the shortcut keys: **F9** to zoom in, **Ctrl + F9** to zoom out, and **Ctrl + F11** to zoom fit.

Zoom In increases the Vector View cell width but displays fewer steps. **Zoom Out** decreases Vector View cell width but displays more steps. Every click on zoom changes the cell the width by ten pixels. Eight zoom levels are available. Cell height is not affected.

Zoom Fit fit the whole vector in the vector view.

Split

Split partitions Vector View in half horizontally or vertically. When **Split** is selected, a submenu (Figure 2-24) pops up so that Horizontal or Vertical split modes can be selected. Use **Split** twice to partition the Vector View window into quadrants.

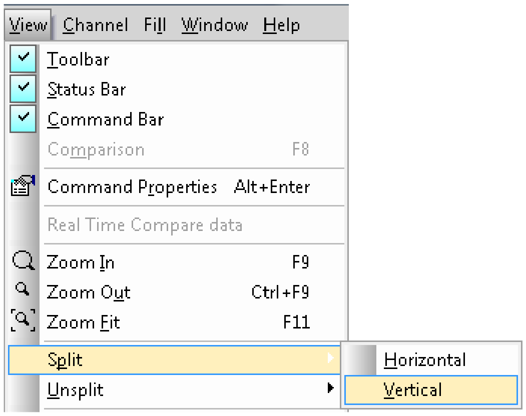
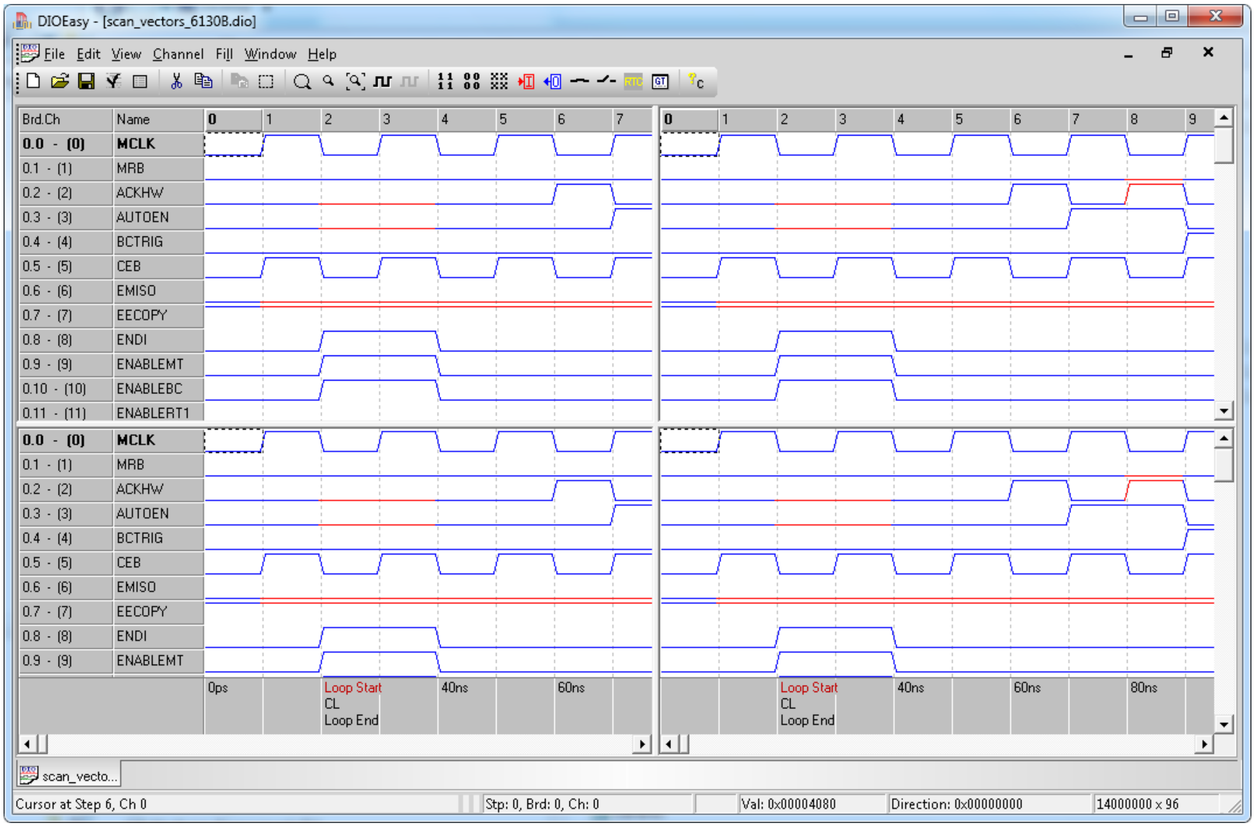


Figure 2-24: Split Menu

A split bar appears at the split midpoint (Figure 2-25). Each split has independent scroll bars.



Cursor at Step 6, Ch 0 | Stp: 0, Brd: 0, Ch: 0 | Val: 0x00004080 | Direction: 0x00000000 | 14000000 x 96

Figure 2-25: Splitting the Vector View Vertically

When the window is split two ways, the scroll bars are shared among the quadrants. Split windows can be zoomed, but the same magnification factor applies to all partitions. Data is not affected by splitting.

Unsplit

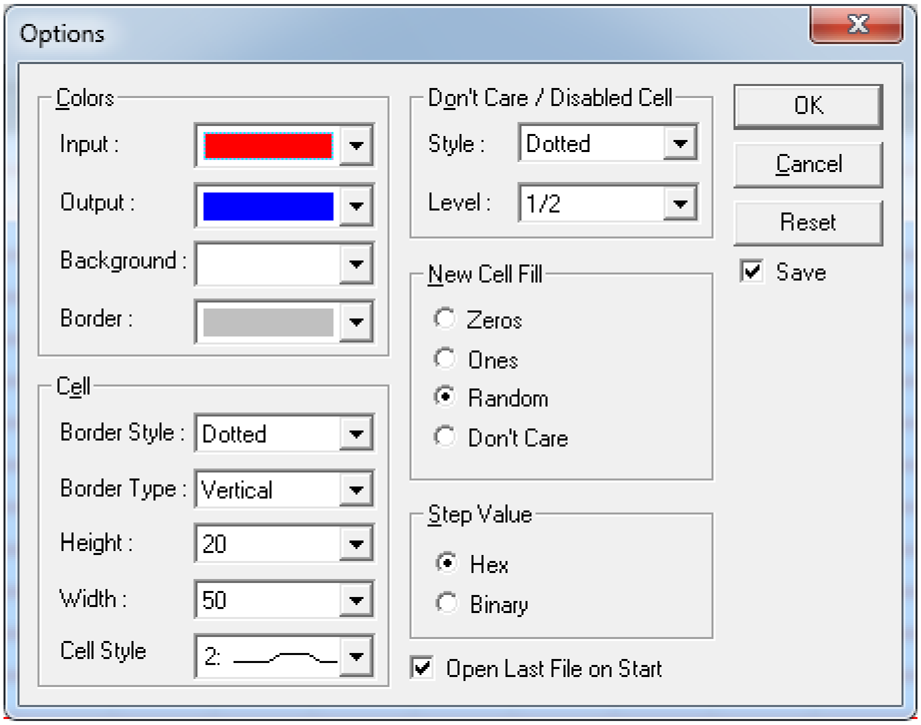
Unsplit restores a split Vector View window partition to its normal (unsplit) state. Choose the Horizontal or Vertical split to be restored from the dialog pop up window. If the split is two ways, then select either one.

A window can also be unsplit by moving the split bar to an edge. This method permits better control over which half displays.

Options…

The Options command allows the user to customize the Vector View display and specify some data replacement options.

The Options window is divided into five groups. Various group controls allow the user to specify a new configuration (Figure 2-26).



Options

Colors

Input : 🟥

Output : 🟦

Background : ⬜

Border : ⬜

Cell

Border Style : Dotted

Border Type : Vertical

Height : 20

Width : 50

Cell Style 2:

Don't Care / Disabled Cell

Style : Dotted

Level : 1/2

New Cell Fill

○ Zeros
○ Ones
● Random
○ Don't Care

Step Value

● Hex
○ Binary

☑ Open Last File on Start

Figure 2-26: Option Window (Default)

Click **Reset** to return all options to default.

Check the **Save** checkbox to save the current configuration whn clicking OK.

Click **OK** to implement the new configuration and exit the dialog window. To retain the new configuration, check the Save checkbox *before* **OK**.

Click **Cancel** to close without changing the current configuration.

Colors

All selections use combo boxes. The drop-down list displays color options. The user can specify one of 16 different colors for Input, Output, Background and Border. Defaults are *Red*, *Blue*, *White* and *Light Gray*, respectively.

Cell

All selections use combo boxes. Click the arrow to see choices.

Select Border Style: None, Solid, Dashed or *Dotted*.

Select Border Type: Both, *Vertical* or Horizontal.

Select cell Height: 10 to 35 (default *20*) in increments of five pixels.

Select cell Width: 5 to 80 (default *50)* in increments of five pixels.

Note: The current zoomed value is also retained when Save is clicked.

Don't Care/Disabled Cell

Use the combo boxes to specify which value display. Click the arrow to select style and level.

Select Style: Solid, Dashed or Dotted.

Select Level: 0, 1 or ½.

Cells containing unspecified data are now displayed with these settings.

New Cells Fill

Use the radio buttons to choose how inserted cells are filled.

Select Zeros, Ones, Random or Don't Care.

If Don't Care is selected, it displays as set up in the Don't Care Cell panel.

Step Value

Use the radio buttons to choose Hex or Binary number format to display the state value for a step. The *Value* field displays the state of the selected step at the lower left corner of the Command bar.

Hexadecimal is more compact. Binary shows the actual bit pattern with Channel 31 at the left and Channel 0 at the right.

Open Last File on Start

Use this check box to load the last file accessed the next time ***DIOEasy*** is opened.

Channel Menu

The commands discussed below are available from this menu.

Name...

Users can assign mnemonic names to channels instead of using default names such as CH0, CH19, etc.

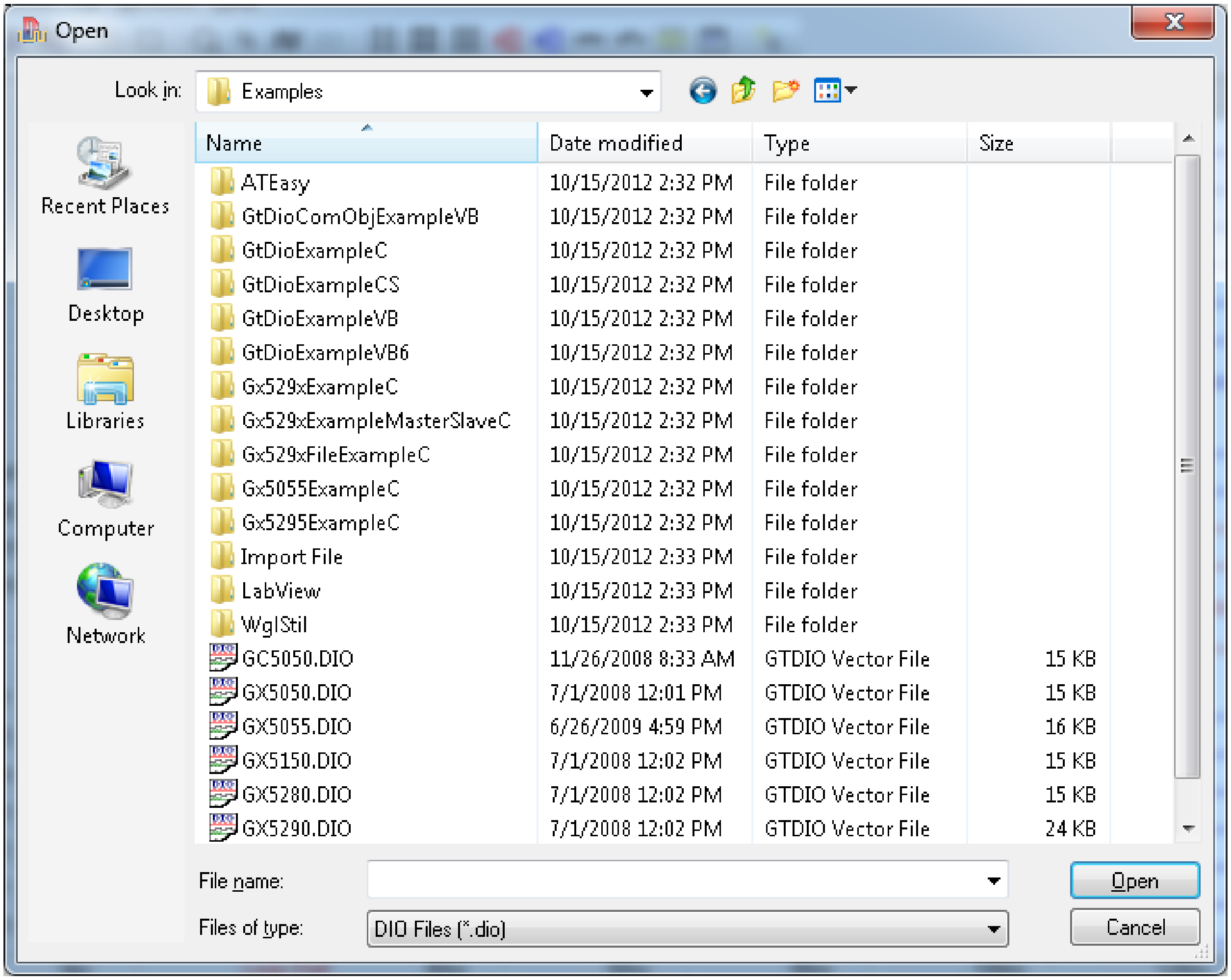
The Name command (shortcut **F2**) allows a name of up to 20 characters to be assigned to a channel. Select a channel's name column to enable editing or double-clicks the channel name. This opens an edit session in the *Name* column of the selected channel.

Type a new name or edit an existing name up to 20 characters. Longer names will be truncated. Press **Enter** or the down arrow key (↓) to retain the new name, or press Esc to revert. The down arrow key opens editing on the next channel. This is useful when assigning or editing a block of names.

When setting up a new DIO file, rather than type in 32 channel names, it may be easier to use names that were previously assigned in another vector, especially if many names are the same. This can be done using the Import Names… command described below.

Import Names…

Click Import Names… to copy channel names from another vector file to the current working file. A dialog box displays to allow source file selection (Figure 2-27). When the file is selected, all channel names are copied from the selected DIO file and assigned to channels in the current Vector View. To retain those names, save the current file.



Look in: Examples

Name	Date modified	Type	Size
ATEasy	10/15/2012 2:32 PM	File folder	
GtDioComObjExampleVB	10/15/2012 2:32 PM	File folder	
GtDioExampleC	10/15/2012 2:32 PM	File folder	
GtDioExampleCS	10/15/2012 2:32 PM	File folder	
GtDioExampleVB	10/15/2012 2:32 PM	File folder	
GtDioExampleVB6	10/15/2012 2:32 PM	File folder	
Gx529xExampleC	10/15/2012 2:32 PM	File folder	
Gx529xExampleMasterSlaveC	10/15/2012 2:32 PM	File folder	
Gx529xFileExampleC	10/15/2012 2:32 PM	File folder	
Gx5055ExampleC	10/15/2012 2:32 PM	File folder	
Gx5295ExampleC	10/15/2012 2:32 PM	File folder	
Import File	10/15/2012 2:33 PM	File folder	
LabView	10/15/2012 2:33 PM	File folder	
WglStil	10/15/2012 2:33 PM	File folder	
GC5050.DIO	11/26/2008 8:33 AM	GTDIO Vector File	15 KB
GX5050.DIO	7/1/2008 12:01 PM	GTDIO Vector File	15 KB
GX5055.DIO	6/26/2009 4:59 PM	GTDIO Vector File	16 KB
GX5150.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB
GX5280.DIO	7/1/2008 12:02 PM	GTDIO Vector File	15 KB
GX5290.DIO	7/1/2008 12:02 PM	GTDIO Vector File	24 KB

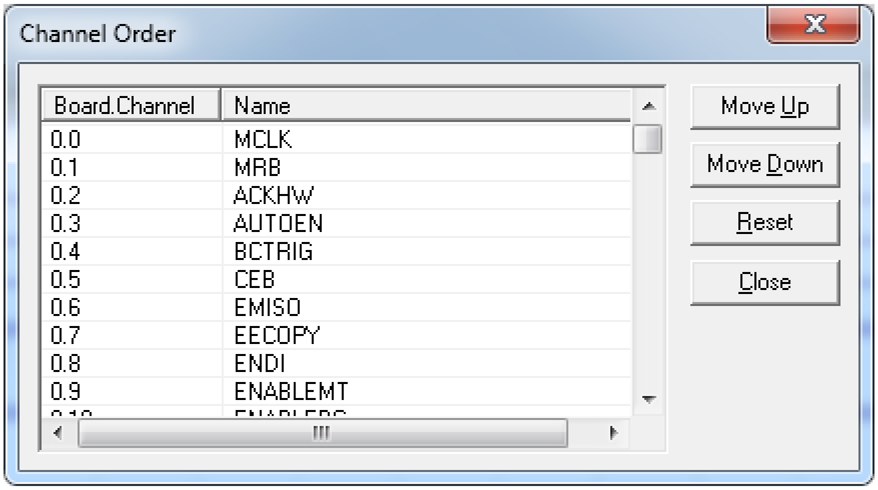
File name:

Files of type: DIO Files (*.dio)

Figure 2-27: Import Names Dialog

Order...

Click **Order…** to reorder channel name assignments. The **Channel Order** dialog box (Figure 2-28) displays.



Board.Channel	Name
0.0	MCLK
0.1	MRB
0.2	ACKHW
0.3	AUTOEN
0.4	BCTRIG
0.5	CEB
0.6	EMISO
0.7	EECOPY
0.8	ENDI
0.9	ENABLEMT

Figure 2-28: Channel Order Dialog Box

A scrollable list of domain channels (Brd.Channel) and names displays in the list box. To change a channel order, select the channel or name and click **Up** or **Down** to move the name. Names that are passed in the move are shifted. This allows moving a block of names one up or down per move, which is often faster than changing names in Vector View using drag and drop.

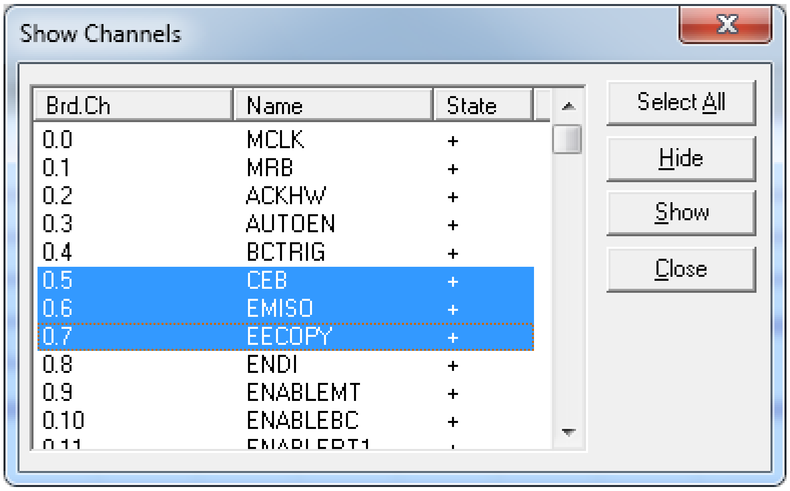
When names are moved in Vector View only the displayed order of channels and names are altered, but names are not reassigned.

To go back to the original order click the Reset button.

Click **OK** to implement the new name assignments. The dialog box closes and Vector View now displays the new names.

Show...

The Show Channels dialog box (Figure 2-29) is used primarily to restore visibility of previously hidden channels. It displays and controls Vector View channel visibility.



Brd.Ch	Name	State
0.0	MCLK	+
0.1	MRB	+
0.2	ACKHW	+
0.3	AUTOEN	+
0.4	BCTRIG	+
0.5	CEB	+
0.6	EMISO	+
0.7	EECOPY	+
0.8	ENDI	+
0.9	ENABLEMT	+
0.10	ENABLEBC	+
0.11	ENABLEBT1	+

Figure 2-29: Show Channels Dialog Box

Channel visibility displays in the Status column as + (visible) and – (hidden). The list box supports multiple item selection. Visibility of selected channels is controlled by the **Show** and **Hide** buttons.

To show one or more hidden channels, select them and click **Show**. All selected channels are now visible in Vector View. Previously visible channels are unaffected.

To hide visible channels, select them and click **Hide**. Selected channels no longer display in Vector View. Previously hidden channels are unaffected.

Hint: To just hide channels, it is easier to use the **Hide** command (see **Hide** below).

To show only a few channels in Vector View, click **Select All**. All channels are selected (highlighted). Hold the CTRL key and click to deselect channels to be displayed. Release CTRL and Click **Hide.** Only the selected channels now display.

To unconditionally restore all channels in Vector View, click **Select All** then **Show**.

Hide

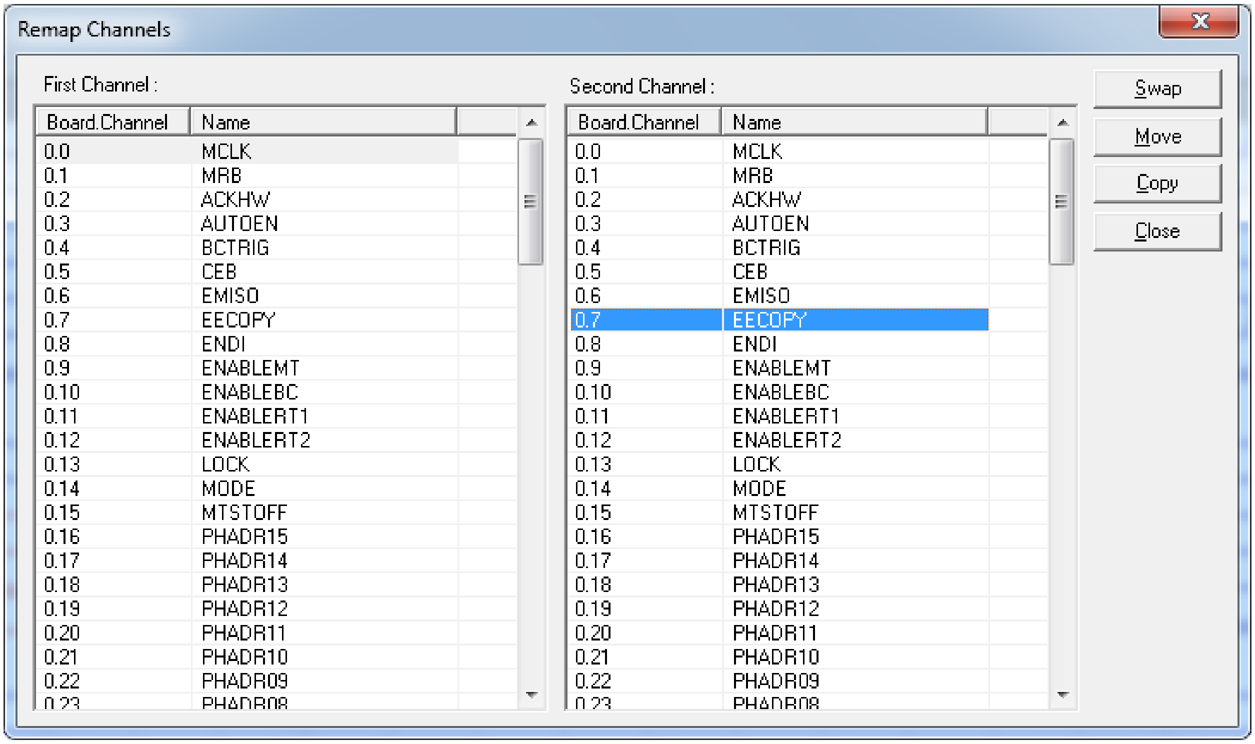
Channel hiding helps remove clutter on the Vector View display. To hide channels in Vector View, select them and click **Hide**. The selected channel(s) no longer display. Use the **Show** command to redisplay hidden channels.

If no channels are selected, **Hide** is grayed out (not selectable).

The **Show…** command can both show and hide channels and may be easier for changing the Vector View display (see **Show…**).

Remap ...

The RemapChannels dialog box (Figure 2-30) allows the user to remapthe channels



First Channel :

Board.Channel	Name
0.0	MCLK
0.1	MRB
0.2	ACKHW
0.3	AUTOEN
0.4	BCTRIG
0.5	CEB
0.6	EMISO
0.7	EECOPY
0.8	ENDI
0.9	ENABLEMT
0.10	ENABLEBC
0.11	ENABLERT1
0.12	ENABLERT2
0.13	LOCK
0.14	MODE
0.15	MTSTOFF
0.16	PHADR15
0.17	PHADR14
0.18	PHADR13
0.19	PHADR12
0.20	PHADR11
0.21	PHADR10
0.22	PHADR09
0.23	PHADR08

Second Channel :

Board.Channel	Name
0.0	MCLK
0.1	MRB
0.2	ACKHW
0.3	AUTOEN
0.4	BCTRIG
0.5	CEB
0.6	EMISO
0.7	EECOPY
0.8	ENDI
0.9	ENABLEMT
0.10	ENABLEBC
0.11	ENABLERT1
0.12	ENABLERT2
0.13	LOCK
0.14	MODE
0.15	MTSTOFF
0.16	PHADR15
0.17	PHADR14
0.18	PHADR13
0.19	PHADR12
0.20	PHADR11
0.21	PHADR10
0.22	PHADR09
0.23	PHADR08

Figure 2-30: Remap Channels Dialog Box

The Remap Channels Dialog Box has three operations:

Swap (Button): swaps the entire first selected channel's data with the second selected channels.

Move (Button): moves the first channel to the second channel location while pushing the channels in between up or down. E.g. if the first selected channel is channel 0, and the second selected channel is channel 5, then the channels contents will be: 5,0,1,2, 3, 4.

Copy (Button): copy the first selected channel's data to the second selected channel.

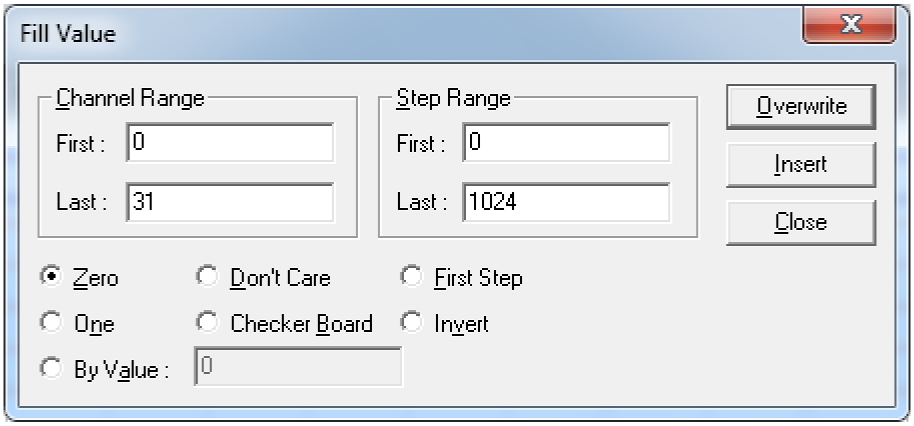
Fill Menu

Fill menu commands are used to insert or overwrite data in the current Vector View. Each of the Fill commands opens a dialog box that displays the fill range, fill parameters and **Insert!** or **Overwrite!** buttons. Icons of fills that are available on the button bar are displayed.

Commands discussed below are available from this menu.

Value...

This command opens the **Fill Value** dialog box (Figure 2-31).



Channel Range
First : 0
Last : 31

Step Range
First : 0
Last : 1024

● Zero ○ Don't Care ○ First Step
○ One ○ Checker Board ○ Invert
○ By Value : 0

Figure 2-31: Fill Value Dialog Box for GC5050/GX5050

The dialog can fill a selected range with any of six values: Zero, One, Don't Care, Checker Board, First Step and Invert.

The **Fill Value…** dialog box that opens contains the boundary channels and steps of the selected group of Vector View cells (default 0 if no cells are selected). The Vector View cells are selected, using the mouse or keyboard, before the Fill Value command is issued. The range that displays can be changed by editing the values in the **First** and **Last Channel Range** and **Step Range** text boxes.

Button functions that are discussed below apply to both DIO board families except where noted.

Insert!	Inserts selected value(s) into the selected range. Steps after the insertion point are renumbered. The last step(s) are truncated.
Overwrite!	Fills the selected range with new data, overwriting the existing data.
Cancel	No changes are made.

Radio buttons are used to select fill or insert values. Radio button selections are described below.

Zero	Fills steps with zeros.
One	Fills steps with ones.
Don’t Care/Disable	Fills with Don’t care or disable.
Invert	Invert the selected step values. Zero changes to one, one changes to zero.
Checker Board	Fills steps like a checkerboard. Adjacent channels are inverted. Vector View rows alternate.
By Value	Filled the selected steps with the specified value.

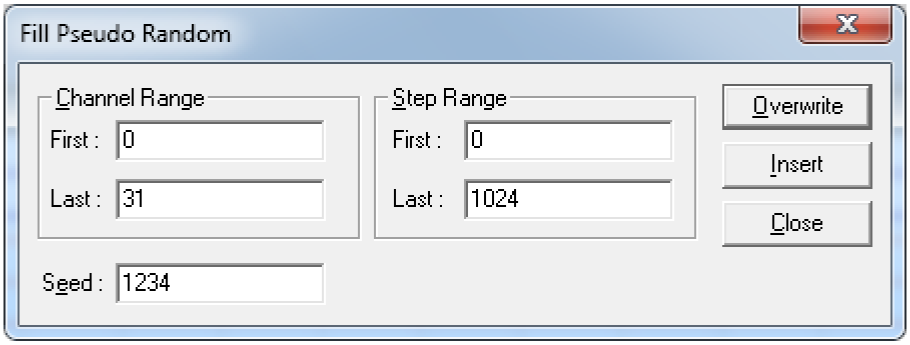
Other Fill Function Patterns

This topic discusses other fill functions that are accessed from the Fill menu. These are: Pseudo Random, Ramp, Toggle, Shift Rotate and Clock.

Each fill function displays a dialog box that accepts **First** and **Last Channel Range**, and **First** and **Last Step Range** inputs. If a range has been pre-selected with the mouse or keyboard, selected boundaries are automatically entered as default. Also, each dialog has **Overwrite!**, **Insert!** and **Cancel** buttons. These fill functions are described in subtopics below.

Pseudo-Random...

This command displays the **Fill Pseudo Random** dialog box (see Figure 2-32).



Fill Pseudo Random

Channel Range
First : 0
Last : 31

Step Range
First : 0
Last : 1024

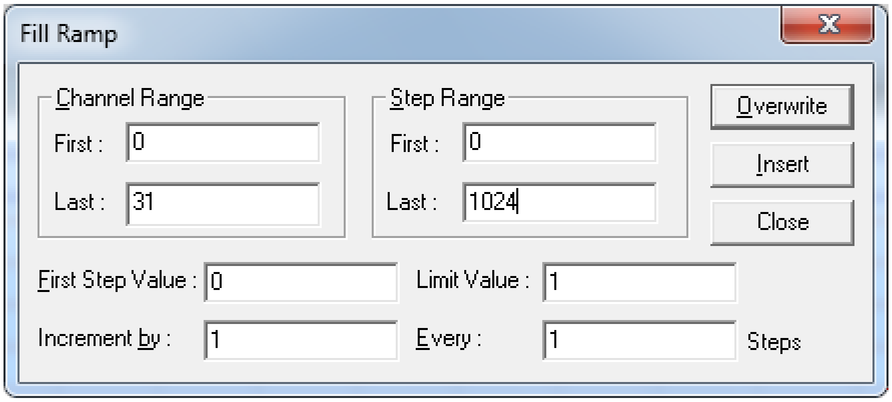
Seed : 1234

Figure 2-32: Fill Pseudo Random Dialog Box

Enter a seed number to generate a pseudo-random pattern or accept the default seed (1). The seed property determines the pseudo-random sequence that is generated. If a repetitive pattern is needed, specify the same seed each time.

Fill Ramp...

This command displays the **Fill Ramp** dialog box (Figure 2-33).



Fill Ramp

Channel Range

First : 0

Last : 31

Step Range

First : 0

Last : 1024

First Step Value : 0

Limit Value : 1

Increment by : 1

Every : 1 Steps

Figure 2-33: Fill Ramp Dialog Box

The "Ramp" is really the binary value of a software counter. It occupies the selected *Channel Range* and is ordered in ascending channel order, that is, the lowest order bit is assigned to the lowest channel number. Each channel in the Fill Ramp function represents one counter bit. To set up counter properties, enter:

- The *First Step Value.* An integer that defaults to 0. It is the starting ramp value.
- The *Limit Value.* An integer that defaults to the maximum binary count represented by the range of channels. It is the final ramp value.
- *Increment by.* The integer number of counts for each increment. This defaults to 1. A negative value will cause the function to decrement.
- *Every ___ Steps.* The integer number of steps between increment. This defaults to 1 and must be positive.

Here are some important points concerning the *Fill Ramp* function.

First Step Value specifies an initial count. It affects the ramp's starting value. The *Limit Value* should exceed *First Step Value* if the Increment is negative. It should be smaller if the Increment is positive.

The *Limit Value* specifies the maximum ramp value. It defaults to $2^n - 1$, where *n* is the number of selected channels. For example, if *First* and *Last* channels are 4 and 13, ten steps are spanned, the default value would be $2^{10} - 1 = 1023$.

The following points apply to the *Limit Value* parameter:

- If Limit is reached before the last step, the ramp will remain at the *Limit Value* until the last step is reached.
- If the ramp reaches the *Last Step* before the *Limit* is reached, the ramp ends.
- Channels are assigned to ramp bits starting with the least significant bit. Therefore, if sufficient channels are not allocated to fully represent a (user-modified) *Limit Value*, only the least significant ramp bits are assigned to channels. However, the ramp is still fully generated internally. Without more significant bits, the ramp will appear to "repeat" with a period that depends on all values entered. The ramp ends if the internal *Limit Value* or *Last Step* is exceeded.
- If there are more channels in the range than are necessary to represent the *Limit Value*, the ramp will stop when the Limit is reached, or when the *Last Step* is exceeded. Therefore, higher order channels will not switch.

- The limiting operation of *Limit Value* is such that if the next increment would exceed that limit, the ramp value remains unchanged. Thus, if *Increment by* is set to 5, *Limit Value* to 12 and *First Step Value* to zero, the ramp sequence 0, 5, 10, 10, 10… would be generated. 12 would never be attained because the next value, 15, would exceed the *Limit Value*.

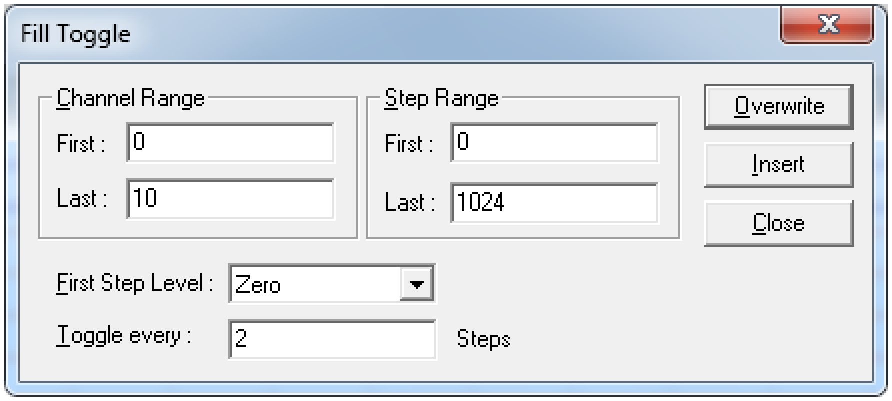
The ratio of *Increment by* value to the *Every (value) Steps* determines the effective slope of the ramp. For example, if set to increment by 3 every 2 steps, the effective slope is 1.5 units per step and increments can occur every second step. If set to increment 6 every 4 steps, the effective slope is still 1.5, but increments occur every fourth step and the step size is doubled.

Ramp values are written into the defined Channel and Step Ranges only. Other values that may be present are not changed.

A downward sloping ramp can be generated by first making an upward sloping ramp with the desired characteristics. Next, use the Invert function in the *Fill Value* dialog box (Fill menu, Value item) to invert all ramp values across selected channels. This shows that repeated application of the Fill function may be used to generate a complex waveform.

Toggle...

The **Fill Toggle** dialog box (Figure 2-34) fills the selected area with a square wave.



Fill Toggle

Channel Range

First : 0

Last : 10

Step Range

First : 0

Last : 1024

First Step Level : Zero

Toggle every : 2 Steps

Figure 2-34: Fill Toggle Dialog Box

The square wave's starting value and toggling interval are controlled in the dialog box.

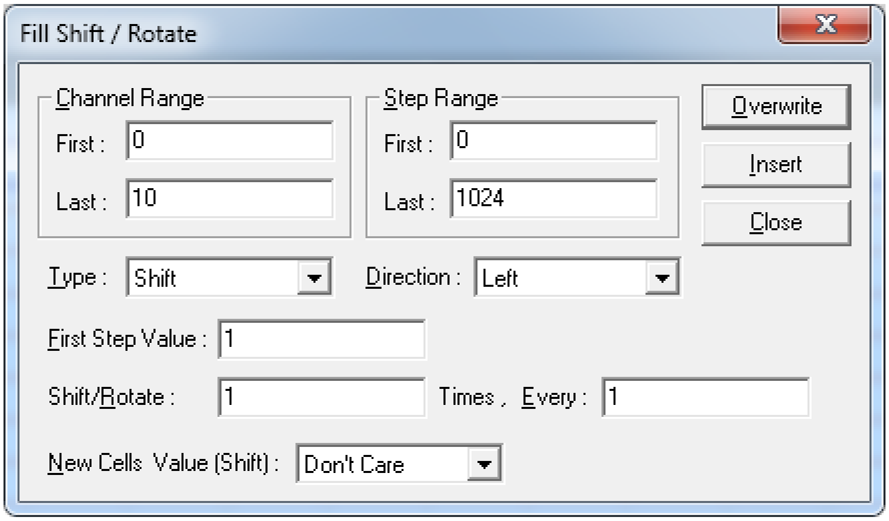
First Step Level selections are Zero (default) or One.

The *Toggle* interval can be any integer. The default is 2 steps.

Shift/Rotate...

If the Type selected is circular, then the value shifted out of the last channel fills the first channel. Then New Cells Value (Shift) is not available.

This command fills the first step of the selected channels with the binary value entered in *First Step Value* (See Figure 2-35).


Fill Shift / Rotate

Channel Range
First : 0
Last : 10

Step Range
First : 0
Last : 1024

Type : Shift
Direction : Left
First Step Value : 1
Shift/Rotate : 1 Times , Every : 1
New Cells Value (Shift) : Don't Care

Figure 2-35: Shift/Rotate Dialog Box

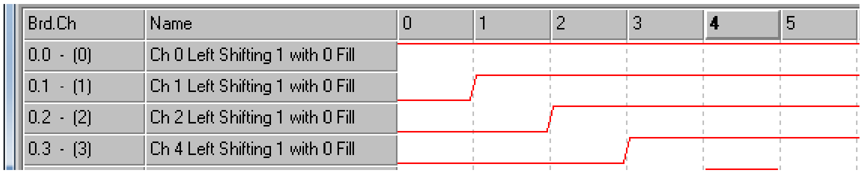
It then shifts the data in each cell left or right according to the *Direction* selected. The value of the vacated bit position depends on the *Type* of process selected and the New Cells Value (Shift) selection.

If the *Type* of process selected is Shift, then the value shifted out of the end channel is lost and the value selected in New Cells Value (Shift) fills the beginning channel.

Shift/Rotate (number) *Times* specifies the number of shifts per step. *Every* (number) determines the number of steps between shifts.

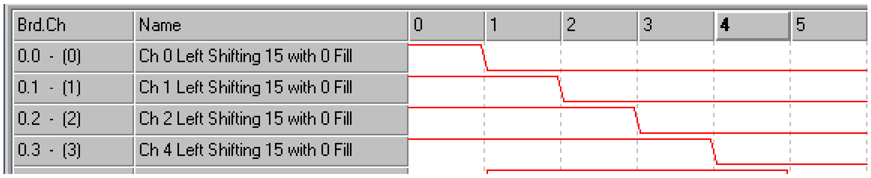
Figure 2-35 shows default values that appear when Shift/Rotate is selected.

Figure 2-36 through Figure 2-37 show some of the many variations for different initial values and both circular and shift processes.



Brd.Ch	Name	0	1	2	3	4	5
0.0 - (0)	Ch 0 Left Shifting 1 with 0 Fill						
0.1 - (1)	Ch 1 Left Shifting 1 with 0 Fill						
0.2 - (2)	Ch 2 Left Shifting 1 with 0 Fill						
0.3 - (3)	Ch 4 Left Shifting 1 with 0 Fill						

Figure 2-36: Left Shifting 1 with 0 Fill



Brd.Ch	Name	0	1	2	3	4	5
0.0 - (0)	Ch 0 Left Shifting 15 with 0 Fill						
0.1 - (1)	Ch 1 Left Shifting 15 with 0 Fill						
0.2 - (2)	Ch 2 Left Shifting 15 with 0 Fill						
0.3 - (3)	Ch 4 Left Shifting 15 with 0 Fill						

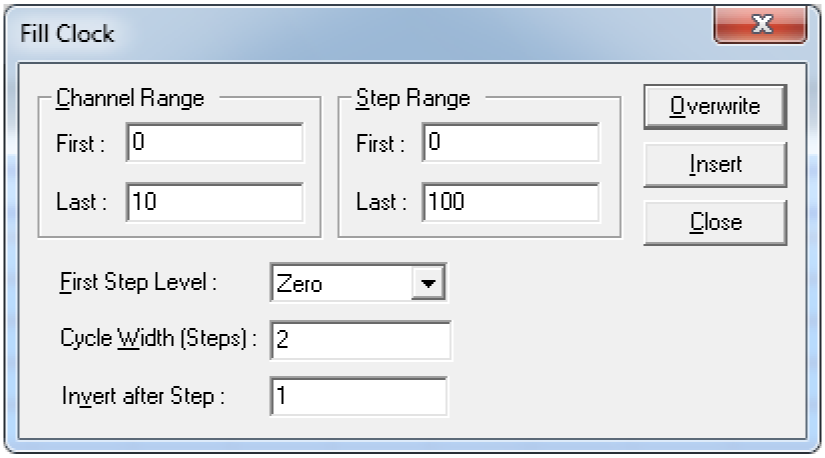
Figure 2-37: Left Shifting 15 with 0 Fill

Clock...

This selection displays the **Fill Clock** dialog box. This command fills the selected area starting with *First Step Level* (Zero or One). The default level is Zero.

Each clock cycle is inverted after a specified number of steps. The *Invert after Step* default is *1*. *Cycle Width (Steps*) is the number of steps to complete a clock cycle (must be greater than the *Invert after Step* entry).

Figure 2-38 displays the Fill Clock dialog for a 3 step clock on Channel 0. It starts with a *zero* and is followed by two *ones*.



Fill Clock

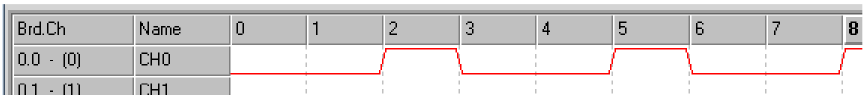
Channel Range
First : 0
Last : 10

Step Range
First : 0
Last : 100

First Step Level : Zero
Cycle Width (Steps) : 2
Invert after Step : 1

Figure 2-38: Fill Clock Dialog Box

Figure 2-39 displays the Vector View result on Channel 0.



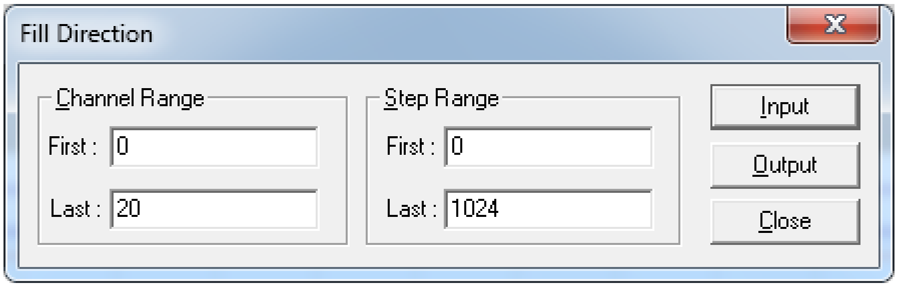
Brd.Ch	Name	0	1	2	3	4	5	6	7	8
0.0 - (0)	CH0									
0.1 - (1)	CH1									

Figure 2-39: Result of Fill Clock Dialog in Figure 4-39

Fill Direction

The **Fill Direction** dialog can be invoked for GX5050 DIO boards. This command is not available for GX5150 boards because the GX5150 direction cannot be changed after configuration.

The Fill Direction dialog (Figure 2-40) designates GC5050/GX5050 channel directions as either input (default) or output at selected steps and channels.



Channel Range
First : 0
Last : 20

Step Range
First : 0
Last : 1024

Figure 2-40: Fill Direction Dialog Box

Directions are always assigned in blocks of eight channels. Channels 0-7, 8-15, 14-31 and 24-31 are always in the same direction. If a direction for one channel in a block is changed, directions for all channels in that block are also changed.

Select input or output direction for any channel and step range in the target block. When the input is filled with values, the channel selected is defined as a receiver for these steps. These values serve as data to be compared to actual data received.

Note: The DIOs always strobes data into memory at the end of an input interval. Input data that is presented at pins during an interval is available in memory during the following interval or later, depending on strobe setup and clock phase. The timing delay is at least one interval, but is test configuration dependent. True delay must be accounted for when viewing or comparing input data.

Window Menu

New Window

Use this command to open a **New Window** and duplicate the current Vector View window. You can open multiple document windows to view different parts of a file at the same time. If you change the contents in one window, all other windows containing the same document reflect those changes. When you open a new window, it becomes the active window and is displayed on top of all other open windows. The copy number of each new window displays after a colon in the title bar.

Cascade

Use this command to arrange multiple opened view windows in an overlapped fashion.

Tile Vertically

Use this command to arrange multiple opened view windows in a non-overlapped fashion vertically.

Tile Horizontally

Use this command to arrange multiple opened windows in a non-overlapped fashion horizontally.

Open Windows Menu Area

This is a list of all opened ***DIOEasy*** windows (**Vector Window** and **CheckIt**). Click one to make it the currently active window.

Help Menu

Contents

Displays a Table of Contents for the on-line version of this User Guide. A standard Windows Help file format is used.

Search

Displays a standard Windows dialog that facilitates search by keyword or topic.

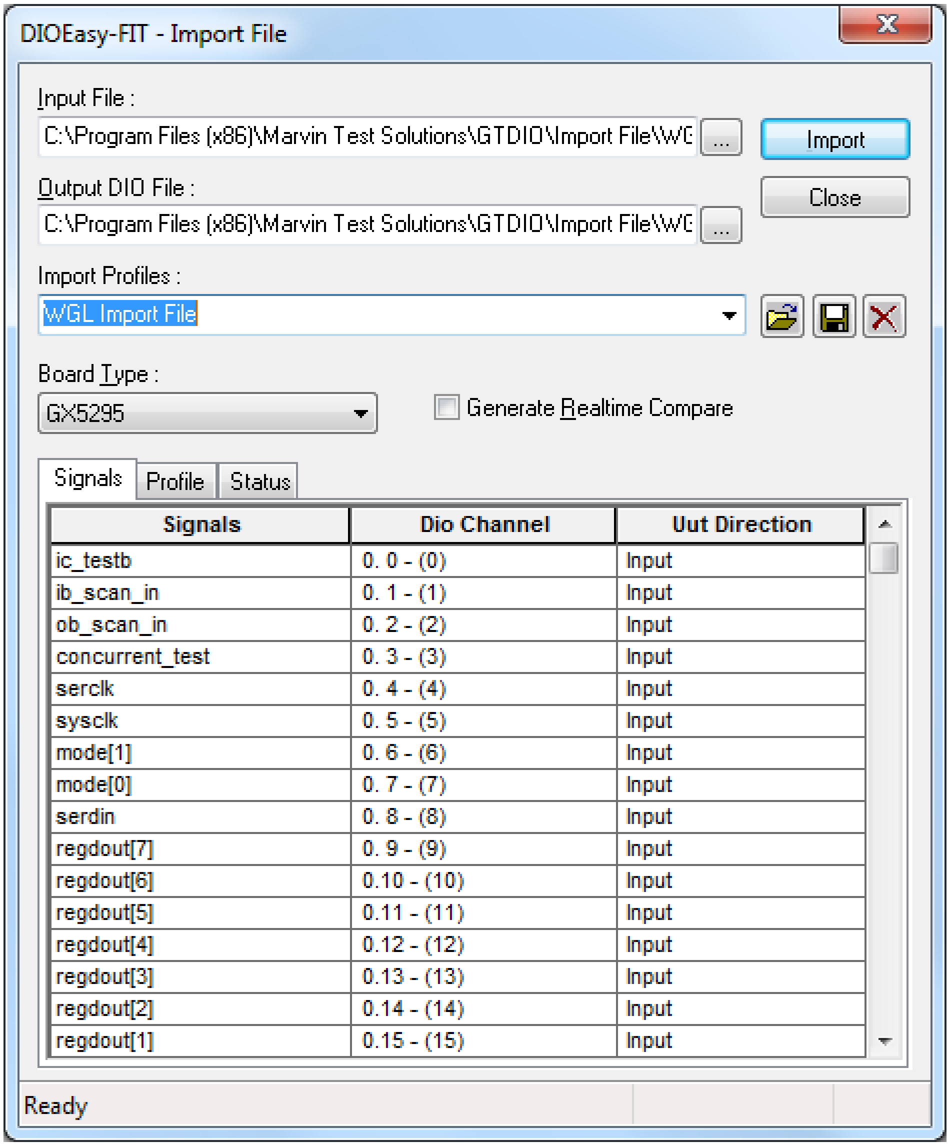
About DIOEasy…

Presents a window that displays the software version number and copyright notice.

Chapter 3 - Importing External Vector Files

The Import File Dialog

You can open the Import File Dialog from DIOEasy File, Import File… menu. The following image shows the Import dialog after selecting a WGL file as the Input File:



DIOEasy-FIT - Import File

Input File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\WG

Output DIO File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\WG

Import Profiles : WGL Import File

Board Type : GX5295

□ Generate Realtime Compare

Signals	Dio Channel	Uut Direction
ic_testb	0. 0 - (0)	Input
ib_scan_in	0. 1 - (1)	Input
ob_scan_in	0. 2 - (2)	Input
concurrent_test	0. 3 - (3)	Input
serclk	0. 4 - (4)	Input
sysclk	0. 5 - (5)	Input
mode[1]	0. 6 - (6)	Input
mode[0]	0. 7 - (7)	Input
serdin	0. 8 - (8)	Input
regdout[7]	0. 9 - (9)	Input
regdout[6]	0.10 - (10)	Input
regdout[5]	0.11 - (11)	Input
regdout[4]	0.12 - (12)	Input
regdout[3]	0.13 - (13)	Input
regdout[2]	0.14 - (14)	Input
regdout[1]	0.15 - (15)	Input

Ready

Figure 3-1: Import File Dialog - WGL File

The following controls are available:

Input file (Edit Box): The user can set the Input Waveform Generation Language source file by typing its full path and name or by clicking on the □ button that opens the file open dialog

Output DIO File Name: (Edit Box): The output DIO file name field will be automatically will be filled with the same file name and location but with the DIO file extension. The user can than rename or change the location of the output DIO file.

Import Profiles (drop list): The Import Profiles shows a list pf profile teach contains settings such as signal maps and import variables settings used by the import. The profiles are saved to your system registry and are used when you need to import several files with similar signal map and settings. The Profile can be viewed in the Profile tab which can be edited. The user can copy the profile text and pass it to the **GTDIOFileImport** API (see the function reference chapter for details). You can type a new profile name in the drop list and click the Save button to create a new profile.

Open Profile File (button): this button is used to open a text file containing the profile text as shown in the Profile tab. You can create a profile file by copying the text displayed in the profile tab and pasting it to a text editor and saving it to a text file.

Save Import Profiles (button): Typing a profile name and clicking on the save button saves the profile to the system registry under profile name. The next time the import dialog is opened the Import Profiles drop list will be populated with all saved profiles.

Delete Import Profiles (button): Clicking on the delete button deletes the selected import profile from the registry.

Board Type (List Box): Sets the target supported **DIO** board type.

Import (Button): Clicking on this button will start the file import process. When importing a file, a progress bar will appear at the bottom of the dialog and in the Status tab.

Close (Button): Closes the dialog.

Signals Tab

The Signals tab displays rows each showing the import file signal name, the associated DIO channel and the direction of the UUT/DUT signal.

Signals column (List View): Displays the imported file signals name list. The user can change any of the signals connections to a specified channel by selecting any signal and changing it to another signal.

Dio Channel column:(List View): Displays the import file signals to channels list. The user can remap the physical channel connection by clicking on the channels name and selecting a different DIO channel form the list or select **Unused** if the signal is not needed and will be ignored during the import. The DIO Channels are displayed in the format b.c –(#), where b is the board number in the DIO domain (1 based), c is the channel number and # is the domain channel number. Channels are always 0 based.

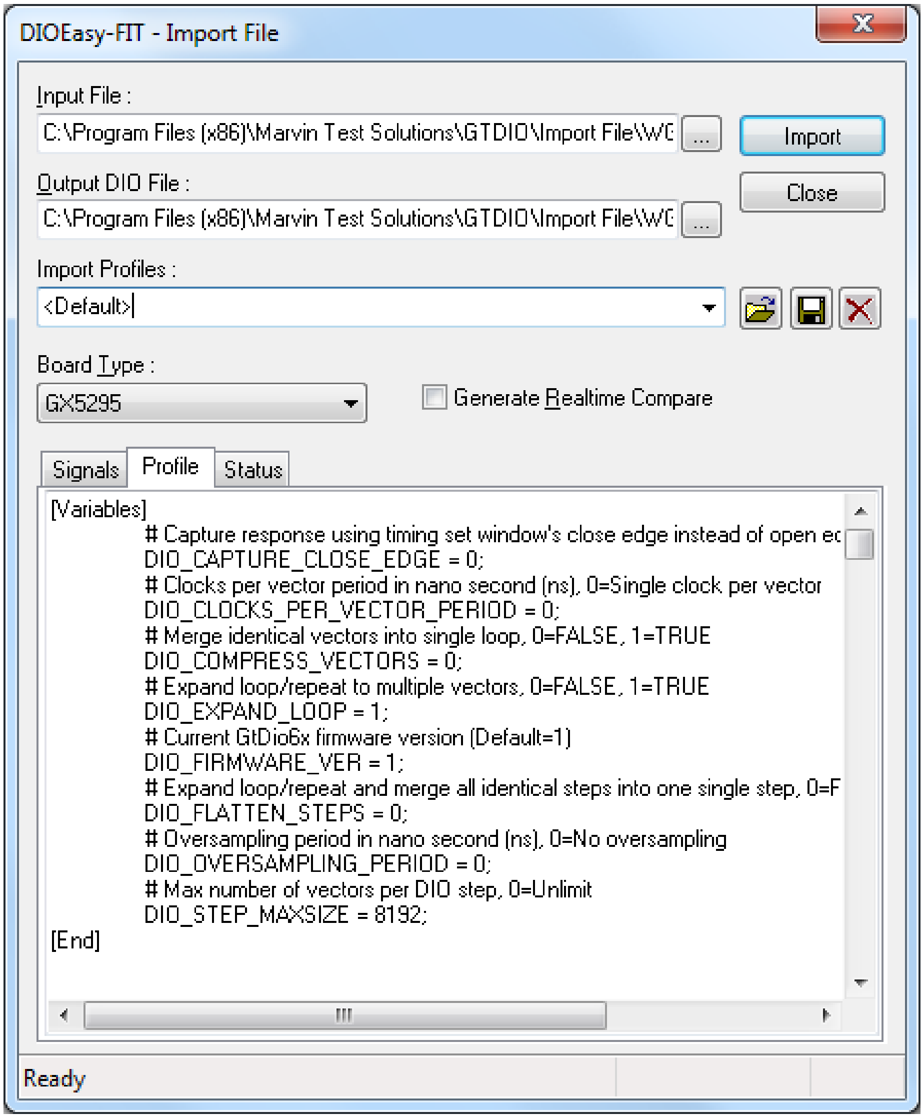
Direction column (List View): Displays the imported file signals direction. When importing VCD or EVCD file the user can set the direction since the files do not specify signal direction. Direction can be Input, Output or Bidir (bi-direction, input and output).

Profile Tab

The **Profile tab** displays the import settings associated with the current **Import Profile** name drop list. The profiles are saved to your system registry and can be copied for use on a different system or when using the **GTDIOFileImport** API so you can perform batch importing for several vector files with the same pin map or settings.

The profile is a text editor displaying two sections:

1. The **[Signals]** section that contains a list of DIO channels and the associated import file signal, followed by direction of the signal.
2. The **[Variables]** section contains pre-defined variables and their values. Each variables is preceded with a comment (start with #) explaining the usage of the variable. Variables are used to modify the behavior of the import process or outcome. This section also contains variables that were defined inside the imported file. These variables are listed under the **#user defined variables** comment. You must set these variables values by typing the value in the editor before importing the file.



Input File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\WC

Output DIO File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\WC

Import Profiles : <Default>

Board Type : GX5295

☐ Generate Realtime Compare

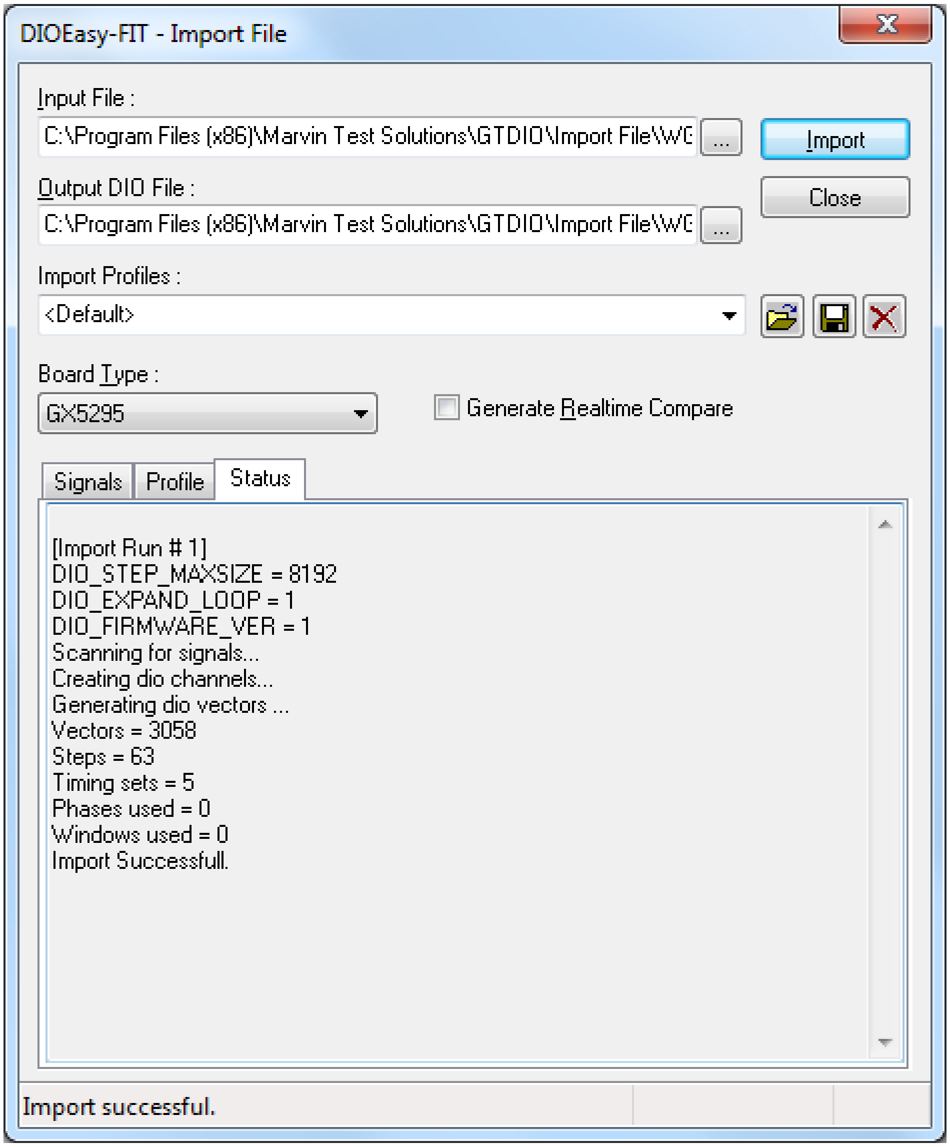
```
[Variables]
            # Capture response using timing set window's close edge instead of open ec
            DIO_CAPTURE_CLOSE_EDGE = 0;
            # Clocks per vector period in nano second (ns), 0=Single clock per vector
            DIO_CLOCKS_PER_VECTOR_PERIOD = 0;
            # Merge identical vectors into single loop, 0=FALSE, 1=TRUE
            DIO_COMPRESS_VECTORS = 0;
            # Expand loop/repeat to multiple vectors, 0=FALSE, 1=TRUE
            DIO_EXPAND_LOOP = 1;
            # Current GtDio6x firmware version (Default=1)
            DIO_FIRMWARE_VER = 1;
            # Expand loop/repeat and merge all identical steps into one single step, 0=F
            DIO_FLATTEN_STEPS = 0;
            # Oversampling period in nano second (ns), 0=No oversampling
            DIO_OVERSAMPLING_PERIOD = 0;
            # Max number of vectors per DIO step, 0=Unlimit
            DIO_STEP_MAXSIZE = 8192;
[End]
```

Ready

Figure 3-2: File Import – Profile Tab

Status Tab

The status tab displays the progress of the import process. The import will try to match the patterns and timing settings to the DIO Board limitation. As part of the process seal file iterations need to be performed to find the best import setting that will result in the best vector file that matches the imported vector file. This process may take a while and the status tab will show the progress and the setting used in addition to warnings that were found because of the DIO board limitation.



Input File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\WG

Output DIO File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\WG

Import Profiles : <Default>

Board Type : GX5295

☐ Generate Realtime Compare

```
[Import Run # 1]
DIO_STEP_MAXSIZE = 8192
DIO_EXPAND_LOOP = 1
DIO_FIRMWARE_VER = 1
Scanning for signals...
Creating dio channels...
Generating dio vectors ...
Vectors = 3058
Steps = 63
Timing sets = 5
Phases used = 0
Windows used = 0
Import Successfull.
```

Import successful.

Figure 3-3: File Import – Status Tab

Importing a File

To import a file to a DIO file format used by the DIO boards follow these steps:

1. Select the Input File (file to import). The dialog will scan the input file and display the available signals/pins in the Signals tab.
2. Type in the result Output DIO File (if needed)
3. Select target Board Type used to generate the DIO file for.
4. Map the signals to a DIO channel using the Signals tab, you can select the Unused to ignore a signal that is not needed. You can also set pin direction (apply only to VCD/EVCD).
5. The signal map and direction settings are displayed will display in the Profile tab. These settings can be saved by typing a profile name under the Import Profiles drop list, and clicking on the Save button. The profile will be save to your system registry. Additional settings for the profile can be typed in the Profile tab by assigning values to predefined variables that will be used by the import utility. These variables are explained in the Profile tab.
6. Click on Import to start the import process. The Status tab will show the import progress which may take a while since the import will try to accommodate the patterns and time settings in the imported file to the board type you have selected.

WGL File

The Waveform Generation Language (WGL) is a data description language supported by Test Systems Strategies Inc. A WGL file uses an ASCII representation of the digital waveform data, WGL is an intermediate file format used by the semiconductor industry for converting digital test patterns from a logic simulator to tester hardware, and back again.

STIL File

Standard Test Interface Language (STIL) file is an IEEE standard (1450) that describes an interface between digital test generation tools and test equipment. A test description language is defined that: (a) facilitates the transfer of digital test vector data from CAE to ATE environments; (b) specifies pattern, format, and timing information sufficient to define the application of digital test vectors to a DUT; and (c) supports the volume of test vector data generated from structured tests

The Importing STIL File dialog is similar to the importing WGL file dialog with one exception that the user can select which pattern in the STIL file to import (STILL file format may contain number of patterns).

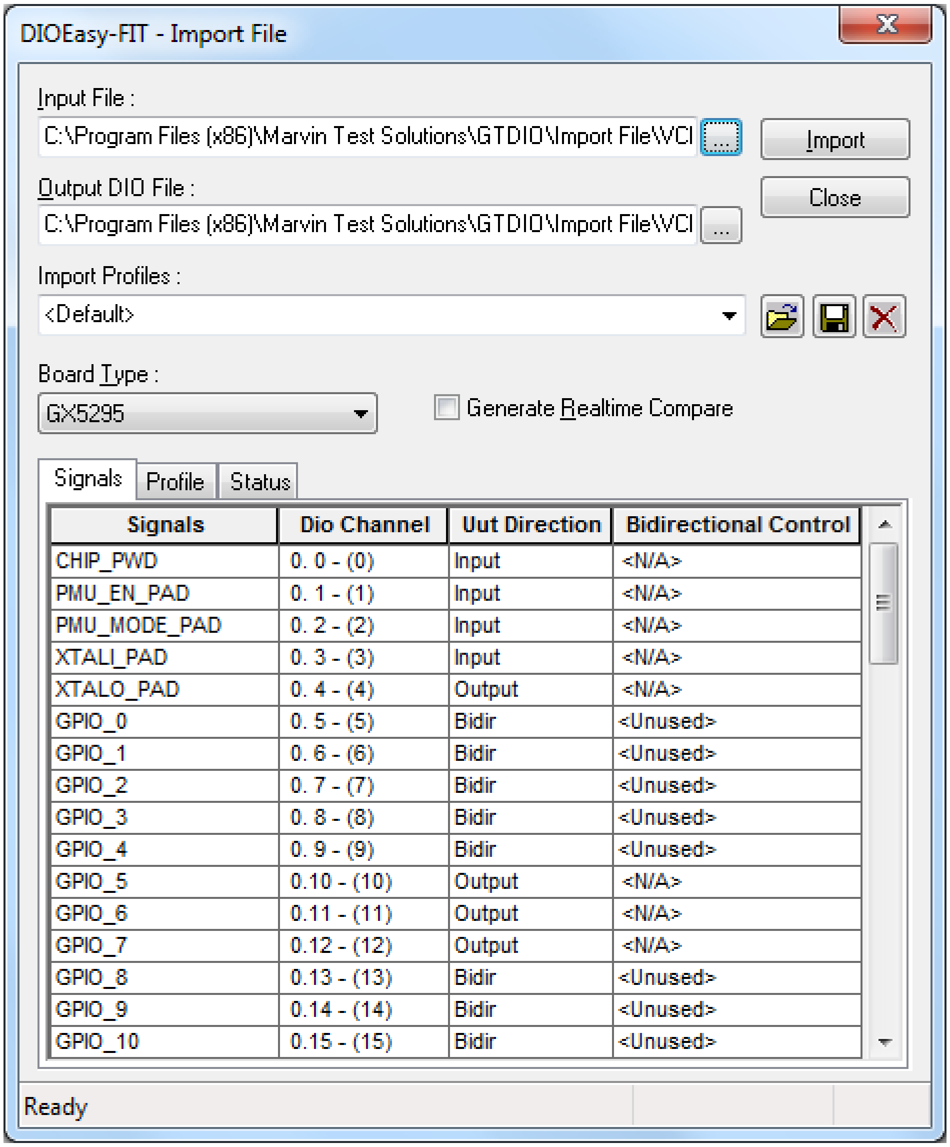
ATP File

Standard Test Interface Language (ATP) file is a Teradyne file format used by Teradyne testers. The **.atp file** is the main file that contains vector patterns, additional files can be supplied. 1) pin map file is an optional file and must have **.pinmap file** extension, 2) pin voltage or current levels is an optional file and must have the **.level file** extension), 3) timing set file, must be provided, this file should have **.tset file** extension. Since there is no standard way of specifying the additional files, The File import utility requires you to place all the files in the same folder, they must have the same file name as the main file (.atp). For example: if the main file is called **scantest1.atp** the timing set filename that has to be provided is called **scantest1.tset**, additional files **scantest1.pinmap** and **scantest1.level** can be also provided.

VCD/eVCD File

Value Change Dump (VCD) is an IEEE standard (1364) describing an ASCII-based format for dump files generated by EDA logic simulation tools. The standard, four-value VCD format was defined along with the Verilog hardware description language by the IEEE Standard 1364-1995 in 1995. An **Extended VCD** (eVCD) format defined six years later in the IEEE Standard 1364-2001 supports the logging of signal strength and directionality. The simple and yet compact structure of the VCD format has allowed its use to become ubiquitous and to spread into non-Verilog tools such as the VHDL simulator GHDL and various kernel tracers. A limitation of the format is that it is unable to record the values in memories.

The following image shows importing of VCD or eVCD file:



Input File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\VCI

Output DIO File : C:\Program Files (x86)\Marvin Test Solutions\GTDIO\Import File\VCI

Import Profiles : <Default>

Board Type : GX5295

□ Generate Realtime Compare

Signals	Dio Channel	Uut Direction	Bidirectional Control
CHIP_PWD	0. 0 - (0)	Input	<N/A>
PMU_EN_PAD	0. 1 - (1)	Input	<N/A>
PMU_MODE_PAD	0. 2 - (2)	Input	<N/A>
XTALI_PAD	0. 3 - (3)	Input	<N/A>
XTALO_PAD	0. 4 - (4)	Output	<N/A>
GPIO_0	0. 5 - (5)	Bidir	<Unused>
GPIO_1	0. 6 - (6)	Bidir	<Unused>
GPIO_2	0. 7 - (7)	Bidir	<Unused>
GPIO_3	0. 8 - (8)	Bidir	<Unused>
GPIO_4	0. 9 - (9)	Bidir	<Unused>
GPIO_5	0.10 - (10)	Output	<N/A>
GPIO_6	0.11 - (11)	Output	<N/A>
GPIO_7	0.12 - (12)	Output	<N/A>
GPIO_8	0.13 - (13)	Bidir	<Unused>
GPIO_9	0.14 - (14)	Bidir	<Unused>
GPIO_10	0.15 - (15)	Bidir	<Unused>

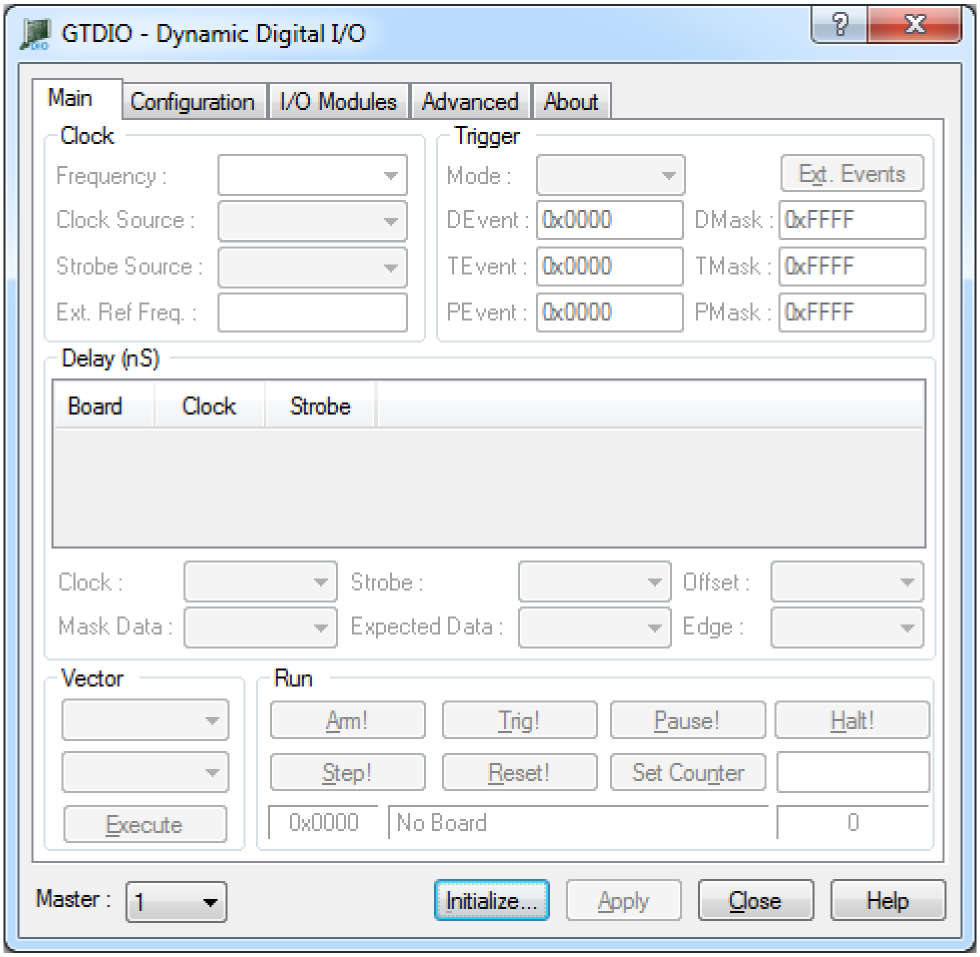
Figure 3-4: Importing VCD

Since the VCD file does not contain the direction information you must select the **UUT Direction** for each signal: **Input**, **Output** or **Bidir** (bidirectional). If you have selected the **Bidir** direction you must also select the signals that will be used to be used as direction signals from the **Bidirectional Control** column.

Chapter 4 - Virtual Panel Description

The GTDIO Panel displays the **GTDIO Dynamic Digital I/O** window. This window is used to configure, initialize, load and save the vector file from/to the GTDIO circuit boards. It can be opened from a **GTDIO Panel** icon (links to 32-bit DIOPNL32.EXE or 64-bit DIOPANEL64.EXE), ***DIOEasy*** or by calling the **DioPanel** function.

The Panel window of Figure 4-1:displays before the boards have been initialized and when Board Panel is selected in the Window menu, or when **F10** is pressed. Fields are discussed below.



DEvent : 0x0000 DMask : 0xFFFF

TEvent : 0x0000 TMask : 0xFFFF

PEvent : 0x0000 PMask : 0xFFFF

0x0000 No Board 0

Master : 1

Figure 4-1: Panel not Initialized

Master Selection Combo Box: Select a Master board in the combo box. This identifies the domain. A domain board is configured using the **Configuration** Tab.

Initialize initialize the board driver. The current settings of the selected counter **will not change after calling initialize**. The panel will reflect the current settings of the counter after the Initialize dialog closes.

Reset - resets the board settings to their default state and clears the reading.

Apply – applies changed settings to the board

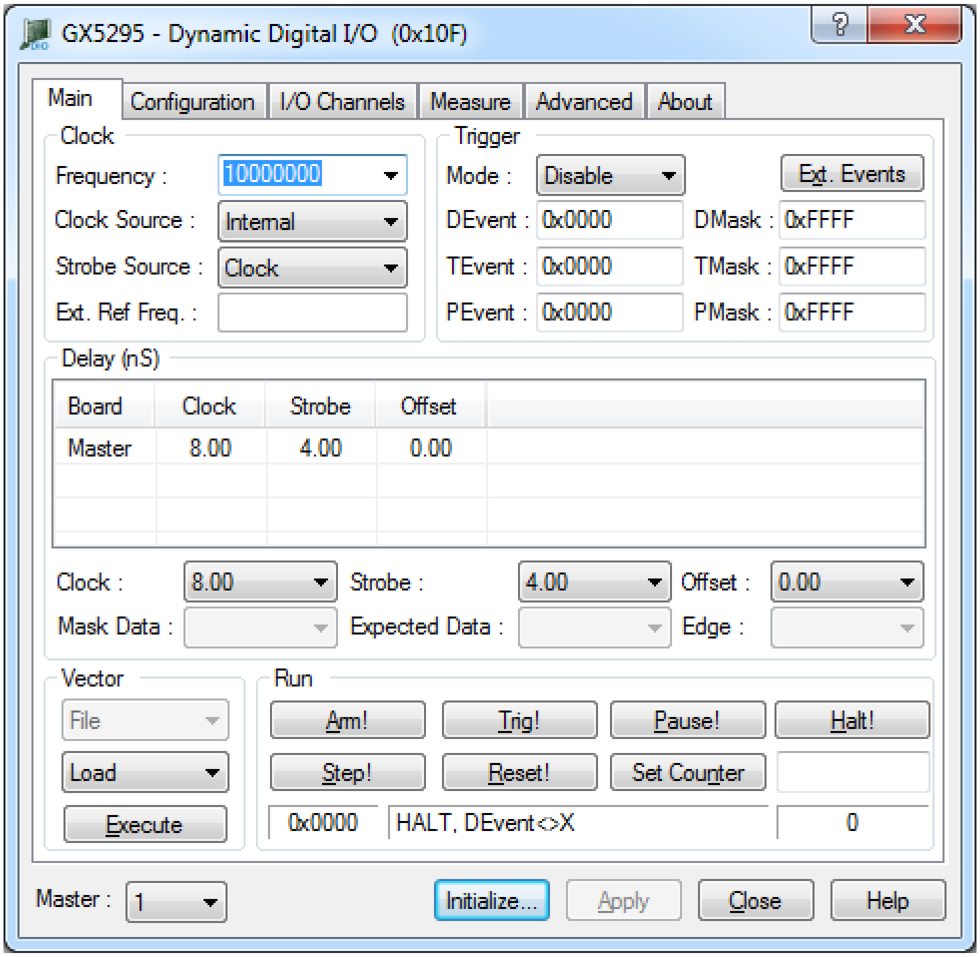
Close - closes the panel. Closing the panel **does not affect** the board settings.

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

The GTDIO virtual panel changes its appearance after initialization to fit the functionality of the initialized DIO family member.

Virtual Main Page Panel

After the board is initialized, the panel is enabled and will display the current setting of the board. The following picture shows the **Main** tab settings:



GX5295 - Dynamic Digital I/O (0x10F)

Clock

Frequency : 10000000

Clock Source : Internal

Strobe Source : Clock

Ext. Ref Freq. :

Trigger

Mode : Disable

DEvent : 0x0000 DMask : 0xFFFF

TEvent : 0x0000 TMask : 0xFFFF

PEvent : 0x0000 PMask : 0xFFFF

Delay (nS)

Board	Clock	Strobe	Offset
Master	8.00	4.00	0.00

Clock : 8.00 Strobe : 4.00 Offset : 0.00

Mask Data : Expected Data : Edge :

Vector

File

Load

Run

0x0000 HALT, DEvent<>X 0

Master : 1

Figure 4-2: Virtual Panel (Initialized)

Clock Group

Frequency: Set/displays a range of desired clock frequencies from 5 Hz to 50 MHz (default). This control is disabled if Clock Source is set to External.

Clock Source : Set/displays the clock source: Internal, External or Program by External.

- ***Internal* generates** a clock at the frequency specified by the **Frequency** selection control. The clock is synthesized from an internal 10 MHz reference.
- ***External*** uses a remote clock source as the clock directly. No frequency synthesizes is involved.

- ***Program by External*** generates a clock at the frequency specified by the **Frequency** selection control. The clock is synthesized from an external reference. The external reference frequency is entered into the **Ext. Frequency** control.

Strobe Source: Select either *Clock Source* or *External*. This option is available *only* if the **Clock Source** control is set to *External*. If *Clock Source* is selected, then the strobe is synchronous with the external clock, but advanced by the **Strobe Delay** setting. If *External Strobe* is selected, then the strobe is timed externally and assumed to be asynchronous with the internal clock (although it may be synchronized externally). Because the strobe frequency is unknown, **Strobe Delay** is not available.

Ext. Ref Freq: Set/displays the external source's *reference frequency*. The *internal* reference frequency is 10 MHz. The clock and strobes are synthesized from this reference. This control is available *only* if the **Clock Source** selection box is set to *Prg. by External (Program by External)*. Enter or select the external reference frequency in Hz from 1 to 60 MHz (50 MHz default).

Trigger Group

Trigger Mode: set/displays the trigger mode. Use the Mode control combo box to select Disable, D level, T level, DT level or TD level. Events are used to set conditions for a board trigger

DEvent: Sets/displays the D register event, values are entered in decimal or in hex (using the "0x" prefix).

DMask: Sets/displays the D register mask, values are entered in decimal or in hex (using the "0x" prefix).

TEvent: Sets/displays the T register event, values are entered in decimal or in hex (using the "0x" prefix).

TMask: Sets/displays the T register mask, values are entered in decimal or in hex (using the "0x" prefix).

PEvent: Sets/displays the conditions that pauses the board Puase register event, values are entered in decimal or in hex (using the "0x" prefix).

PMask: Sets/displays the Puase register mask, values are entered in decimal or in hex (using the "0x" prefix).

See the board's Theory of Operation in its user guide for an explanation of board registers and masks.

Ext. Events…

Click **the Ext. Events** to display the External Event Register dialog (Figure 4-3) Use the radio buttons to select either *External Lines* (default) or *X Register* as an external source. By writing a value into the X Register, the user can simulate events on the External Event Line. The Actual Value text box displays the current value in the X Register as a hex number. To change that value, enter a hex or decimal integer in the upper text box and click **Write**. When entering a hex value, the first two characters must be "0x". The Actual Value text box now displays the hex value of the number entered.

Delay Group

Clock Delay: Select the delay between the main clock and the clock. The DIO family determines what delay selections are available.

Strobe Delay: Select the delay between the strobe and clock. The DIO family determines what delay selections are available. This control is *not* available if the **Strobe Source** control is set to *External Strobe*.

Offset Delay (Gx528X /Gx529X only)**:** Additional offset that advance or delay the specifed DIO board's Clock and Strobe. The offset values vacan be set between -3.0 nSec to +3.0 nSec with resolution of 0.25nSec. The main purpose of this delaty is to ease timing aligment between DIO boards in a domain.

Sets the specified data source clock cycles delay

Mask Data (Gx529X only whne in Real Time Compare opearting mode): sets/display the number of clock cycles that the mask memory data will be delayed by from the clock.

Expected Data (Gx529X only whne in Real Time Compare opearting mode): sets/display the number of clock cycles that the expected memory data will be delayed by from the strobe.

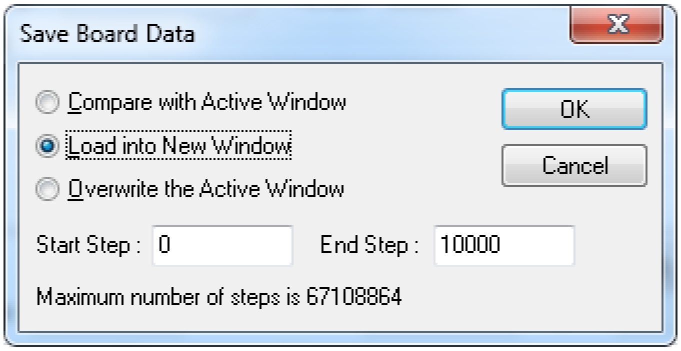
Edge (Gx529X only whne in Real Time Compare opearting mode): sets/display the clock edge of rising or falling.

Vector Group

Source dropdown list: Sets/display the source or target for the DIO vector file. Select *File* (default) to work with a DIO or ASC file. Select *Window* to work with the current active window in ***DIOEasy***. The window is made active by selecting it in Vector View. The choice affects operation of other controls in this group. The *Window* is available only if a Vector View window is active and the Panel is opened within ***DIOEasy***.

Load/Save dropdown list: When *File* is selected, the DIO file that is loaded initializes the board. When *Window* is selected, the file opened in ***DIOEasy*** initializes the board. In file mode, **Save…** reads the board's configuration and data and saves it to a DIO file. The default file extension for data read back from a board is DI, but the user can specify the extension, *.DIO, or any other.

When Window source is selected, the Save Board Data dialog opens (Figure 4-3).



Save Board Data

○ Compare with Active Window
● Load into New Window
○ Overwrite the Active Window

Start Step : 0
End Step : 10000

Maximum number of steps is 67108864

Figure 4-3: Save Board Data Dialog Window

"Compare with Active Window" (default) compares vector differences between files opened in ***DIOEasy*** and board data. Differences appear as shaded cells in ***DIOEasy*** current Vector View.

"Load into New Window" writes DIO data to a new Vector View window while maintaining the previous one.

"Overwrite the Active Window" replaces data currently in the active window with data from the board.

Run Group

This group has buttons for immediate function execution. It displays certain system parameters for all boards bound to the selected Master.

Arm: Changes board states from HALT to PAUSE.

Trig: Changes board states from PAUSE to RUN.

Pause: Changes board states from RUN to PAUSE.

Step: Advances the sequencer a single step.

Halt: Changes board states to HALT.

Rese!: Changes board states to HALT and restores the Master and Slaves to the default setup.

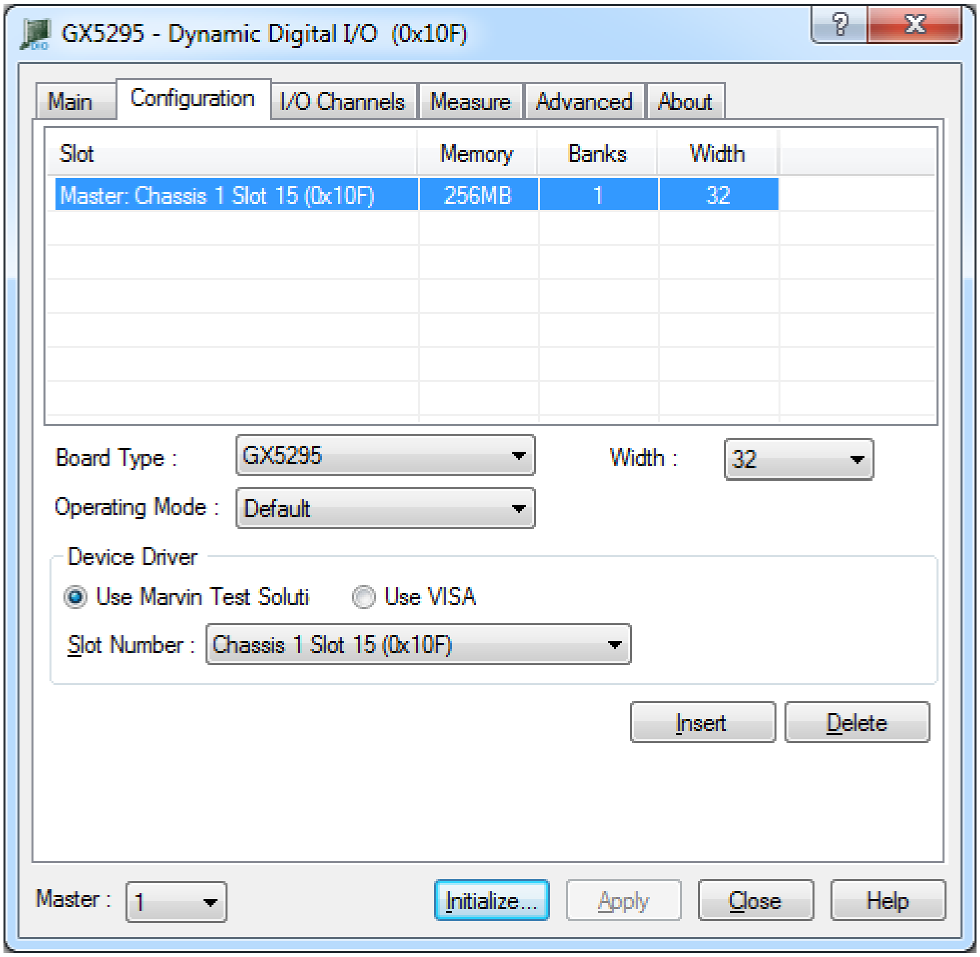
Set Counter: This button is enabled for GX5150 domains only. It sets the Program Counter to the hex or decimal step value entered in the text box at the right. Hex values are preceded with "0x". The resulting Program Counter value displays in hex in the right window of the status bar. (See the Status Display topic below).

Status Display: The Status Display at the bottom of the Run group, has three text boxes: left, middle and right. The left box displays the Status Register contents as a hex code. Status codes are different for the GC5050/GX5050 and GX515X boards. See the DioReadStatusRegister function in Chapter 7 – Function Reference for an explanation of codes. The middle box displays additional details concerning status and errors. See the same reference as above for details. The right box displays the Program Counter step number.

Virtual Panel Configuration Page

The **Configuration** tab dialog supports two different device drivers that can be used to access and control the board

Double click the **GT-DIO Panel** icon or open the Panel in ***DIOEasy*** (press **F10** or select it under the Windows menu). The DIO Panel window opens (see **Figure 4-4**), and then Click on the **Configuration** tab.



Slot	Memory	Banks	Width
Master: Chassis 1 Slot 15 (0x10F)	256MB	1	32

Board Type : GX5295

Width : 32

Operating Mode : Default

Device Driver

● Use Marvin Test Soluti ○ Use VISA

Slot Number : Chassis 1 Slot 15 (0x10F)

Master : 1

Figure 4-4: Configuration window Displaying GX5290 board

Using Marvin Test Solutions' HW Device Driver

The Marvin Test Solutions' HW Device Driver installed by the setup program and is the default driver. When selected, the **Slot Number** list displays the available boards installed and their slots. The chassis, slots, devices and their resources are also displayed by the HW resource manager, **PXI/PCI Explorer** applet that can be opened from the Windows Control Panel. The **PXI/PCI Explorer** can be used to configure the system chassis, controllers, slots and devices. The configuration is saved to PXISYS.INI and PXIeSYS.INI located in the Windows folder. These configuration files are also used by VISA.

Using VISA Device Driver

Use VISA – this is a third party device driver usually provided by National Instrument (NI-VISA). When selected, the **Resource** list displays the available boards installed in the system and their VISA resource address. The chassis, slots, devices and their resources are also displayed by the VISA resource manager, **Measurement & Automation** (NI-MAX) and in Marvin Test Solutions **PXI/PCI Explorer**. The following figure shows PXI9::13::INSTR as the VISA resource (PCI bus 9 and Device 13). This is VISA resource string argument (*szVisaResource*) passed by the panel when calling the driver **DioSetupInitializationVisa** function to initialize the driver with the specified board.

The following describes how to configure a DIO domain using Marvin Test Solutions' HW Device Driver (the process is the same when either using Marvin Test Solutions' HW Device Driver or VISA).

Note: when the **Configuration** tab dialog is selected it will fill the list with the current stored configuration for the specified Master number.

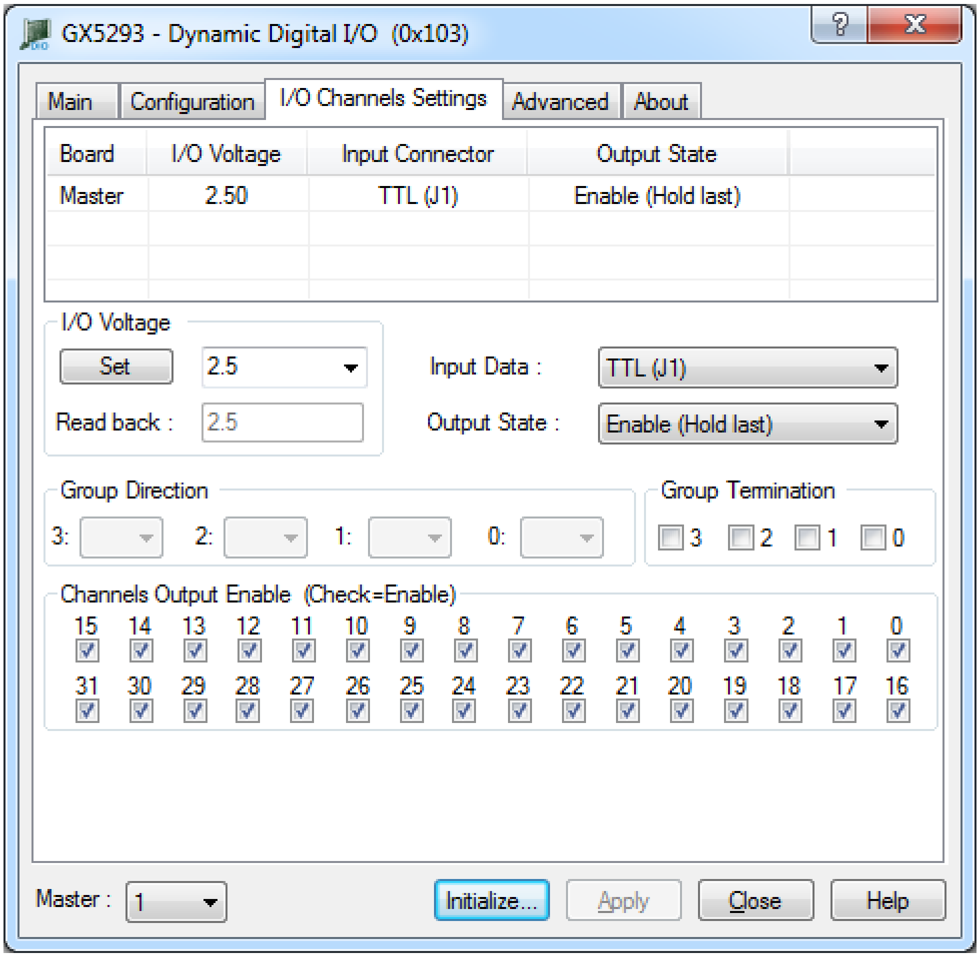
1. If you want to delete the current configuration select the first line (which always points to a Master) and click **Delete**.
2. Select the board family type From the **Board Type** dropdown list. The panel will scan your system and fill the **Slot Number** dropdown list with all the matching boards' with their slot resource numbers and designation as Master or Salve. If none were found the list box will display "No board was found"
3. From the list select the Master and click Insert (the Master board has to be first in the list). The new board appears with the entered parameters. The user can select/change the width and direction of the currently selected board in the list.
4. Add additional Slaves if needed.
5. To delete board configuration, select it and click **Delete**. The board is no longer display in the list.

Figure 4-4 shows a Gx5290 Master board at slot number 0x10A (chassis 1 Slot 10) with two Salves. This is the same slot number argument (*nSlot*) passed to the driver when calling **DioSetupInitialization** function used to initialize driver with the specified board.
To check that the configuration is OK simply click **Initialize** button to initialize all the board in the configuration list. Each board in the list should now has the memory size and number of banks, as well as the Panel's caption display the Master's slot number.

Figure 4-4 show the DIO domain initialized with each board displays its memory size and number of banks. The Panel's caption displays the Master's slot number.

Virtual Panel Gx5290/Gx5280: I/O Channels Settings Page

Clicking on the I/O Channels Settings Page tab (Gx5290/Gx5280 only) will show the I/O Channels Settings Page as shown in Figure 4-5:



Board	I/O Voltage	Input Connector	Output State
Master	2.50	TTL (J1)	Enable (Hold last)

I/O Voltage: 2.5

Read back : 2.5

Input Data : TTL (J1)

Output State : Enable (Hold last)

Group Termination: □ 3 □ 2 □ 1 □ 0

Channels Output Enable (Check=Enable)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑

Master : 1

Figure 4-5: Virtual Panel –I/O Channels Settings Page

Board List: Displays/sets the current active board settings.

Input Connector (dropdown list):

I/O Voltage group: Sets/Displays the I/O Voltage settings.

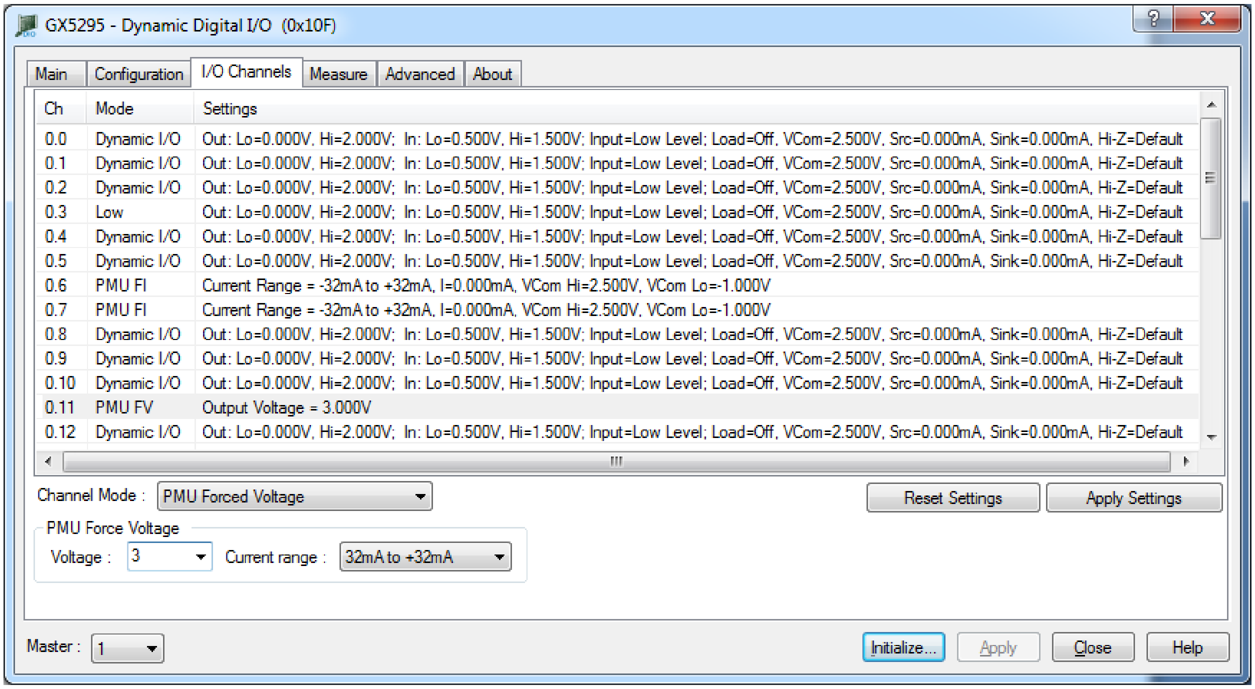
Group Direction: Sets/displays the direction for each group of 8-channels (enabled only for the Gx5280 boards).

Group Termination Sets/displays the on-board termination resistors for each group of 8-channels.

Channels Output Enable: sets/displays for each channels if its output is enabled or disabled.

Virtual Panel Gx5295: I/O Channels Page

Clicking on the I/O Channels Page tab (Gx5290/Gx5280 only) will show the I/O Channels Page as shown in Figure 4-6



Ch	Mode	Settings
0.0	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.1	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.2	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.3	Low	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.4	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.5	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.6	PMU FI	Current Range = -32mA to +32mA, I=0.000mA, VCom Hi=2.500V, VCom Lo=-1.000V
0.7	PMU FI	Current Range = -32mA to +32mA, I=0.000mA, VCom Hi=2.500V, VCom Lo=-1.000V
0.8	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.9	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.10	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default
0.11	PMU FV	Output Voltage = 3.000V
0.12	Dynamic I/O	Out: Lo=0.000V, Hi=2.000V; In: Lo=0.500V, Hi=1.500V; Input=Low Level; Load=Off, VCom=2.500V, Src=0.000mA, Sink=0.000mA, Hi-Z=Default

Channel Mode : PMU Forced Voltage

PMU Force Voltage

Voltage : 3 Current range : 32mA to +32mA

Master : 1

Figure 4-6: Gx5295 Virtual Panel –I/O Channels Page

Resets all settings (Button): Resets all settings back to default values

Apply settings (Button): Apply settings to the selected channels in the list control.

Channel Mode: Displays/set channel operating mode, mdes are: Enabled (default operating mode) channel's output is enabled. Disabled: channel's output is disabled (Tri-State). Low: channel's output is set to low; the output low voltage corresponds to the output low voltage settings. High: channel's output is set to high; the output high voltage corresponds to the output high voltage settings. PMU FI: channel's output is set to PMU (Parametric Measurement Unit) forced current mode. The channel's forced current is set to zero. PMU FV: channel's output is set to PMU (Parametric Measurement Unit) forced voltage mode. The channel's forced voltage is set to zero volts.

Output voltage group:

High (Edit/ Listbox): Displays/set the channel output high voltage, voltage can be set from -2V to +7.0V and need to be higher than the output low voltage settings.

Low (Edit/ Listbox): Displays/set the channel output low voltage, voltage can be set from -2V to +7.0V and need to be less than the output high voltage settings.

Imput voltage group:

High (Edit/ Listbox): Displays/set the channel input high threshold voltage, voltage can be set from -2V to +7.0V and need to be higher than the input low threshold voltage settings.

Low (Edit/ Listbox): Displays/set the channel input low threshold voltage, voltage can be set from -2V to +7.0V and need to be less than the input high threshold voltage settings.

Imput Active Load group:

State (Listbox): Displays/set the channel input active load state; the load state can be turned On or Off.

Commutating Voltage (Edit/ Listbox): Displays/set the channel input active load sink and source currents commutating voltages value, voltage can be set from --2V to +7V.

I-Source (Edit/ Listbox): Displays/set the channel input active load source current settings; source current can be set form 0 to 24mA.

I-Sink (Edit/ Listbox): Displays/set the channel input active load sink current settings; sink current can be set form 0 to 24mA.

Imput Source (Listbox): Displays/set the channel input data comparator source, Input data comparator can be as follows: Low Threshold Comparator: data will be the result of the comparison done on the input data by the low threshold comparator, logic levels are as follows: Logic Low: whenever the input signal is below the low comparator threshold Logic High: whenever the input signal is above the low comparator threshold High Threshold Comparator: data will be the result of the comparison done on the input data by the high threshold comparator, logic levels are as follows: Logic Low: whenever the input signal is below the high comparator threshold. Logic High: whenever the input signal is above the high comparator threshold.

PMU Forced Current group:

Current Range**(Listbox):** Displays/set the channel PMU forced current value\n\rPMU forced current value depends on the current range settings PMU forced current range can as follow: -32mA to +32mA, -8mA to +8mA, -2mA to +2mA, -512uA to +512uA, -128uA to +128uA, -32uA to +32uA, -8uA to +8uA, -2uA to +2uA.

Current (Edit/ Listbox): Displays/set the channel PMU forced current value PMU forced current value depends on the current range settings PMU forced current range can as follows: -32mA to +32mA, -8mA to +8mA, -2mA to +2mA, -512uA to +512uA, -128uA to +128uA, -32uA to +32uA, -8uA to +8uA, -2uA to +2uA.

Commutating Voltage Hi (Edit/ Listbox): Displays/set the channel high commutating voltages value, voltage can be set from -2V to +7V. Each PMU has a set of programmable voltage clamps that limit the voltage swing when the PMU is forcing current. These clamps protect the DUT when current is being forced into a high impedance node at the DUT. If the sensed voltage exceeds the high or low voltage commutating the PMU reduces the output current in order for the output voltage to not exceed the clamp. If the voltage subsequently returns back to within the Commutating thresholds, the PMU resumes forcing the programmed current.

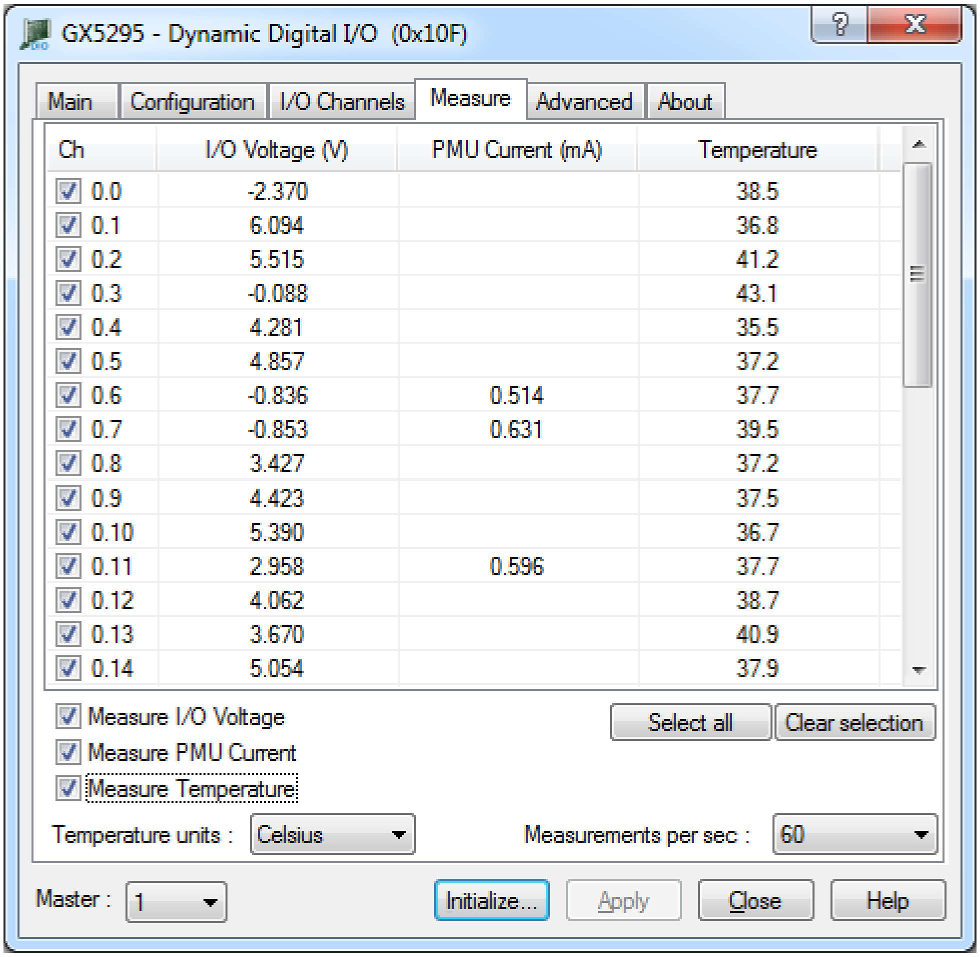
Commutating Voltage Lo (Edit/ Listbox): Displays/set the channel low commutating voltages value, voltage can be set from -2V to +7V. Each PMU has a set of programmable voltage clamps that limit the voltage swing when the PMU is forcing current. These clamps protect the DUT when current is being forced into a high impedance node at the DUT. If the sensed voltage exceeds the high or low voltage commutating the PMU reduces the output current in order for the output voltage to not exceed the clamp. If the voltage subsequently returns back to within the Commutating thresholds, the PMU resumes forcing the programmed current.

PMU Forced Voltage group:

Voltage (Edit/ Listbox): Displays/set the channel PMU forced voltage value. PMU forced voltage value can be between -2V to +7V.

Virtual Panel Gx5295 Measure Page

Clicking on Measure Page tab will show the Measure Page as shown in Figure 4-7:



Ch	I/O Voltage (V)	PMU Current (mA)	Temperature
☑ 0.0	-2.370		38.5
☑ 0.1	6.094		36.8
☑ 0.2	5.515		41.2
☑ 0.3	-0.088		43.1
☑ 0.4	4.281		35.5
☑ 0.5	4.857		37.2
☑ 0.6	-0.836	0.514	37.7
☑ 0.7	-0.853	0.631	39.5
☑ 0.8	3.427		37.2
☑ 0.9	4.423		37.5
☑ 0.10	5.390		36.7
☑ 0.11	2.958	0.596	37.7
☑ 0.12	4.062		38.7
☑ 0.13	3.670		40.9
☑ 0.14	5.054		37.9

☑ Measure I/O Voltage

☑ Measure PMU Current

☑ Measure Temperature

Temperature units : Celsius

Measurements per sec : 60

Master : 1

Figure 4-7: Gx5295 Virtual Panel –Measure Page

Temperarures group:

Measure (Check Box): When checked, start measuring channels temperatures; when unchecked stop all temperatures measurements. All temperatures measurements are automatically disabled when moving to another tab.

Rate **(Listbox):** Displays/set channels temperatures measurements rate in measurements per sec.

Units (Listbox): Displays/set channels temperatures units."

I/O Voltages group:

Measure (Check Box): When checked, start measuring channels I/O voltages, when unchecked stop all I/O voltages measurements. All I/O voltages measurements are automatically disabled when moving to another tab. All temperatures measurements are automatically disabled when moving to another tab.

Rate **(Listbox):** Displays/set channels I/O voltages measurements rate in measurements per sec.

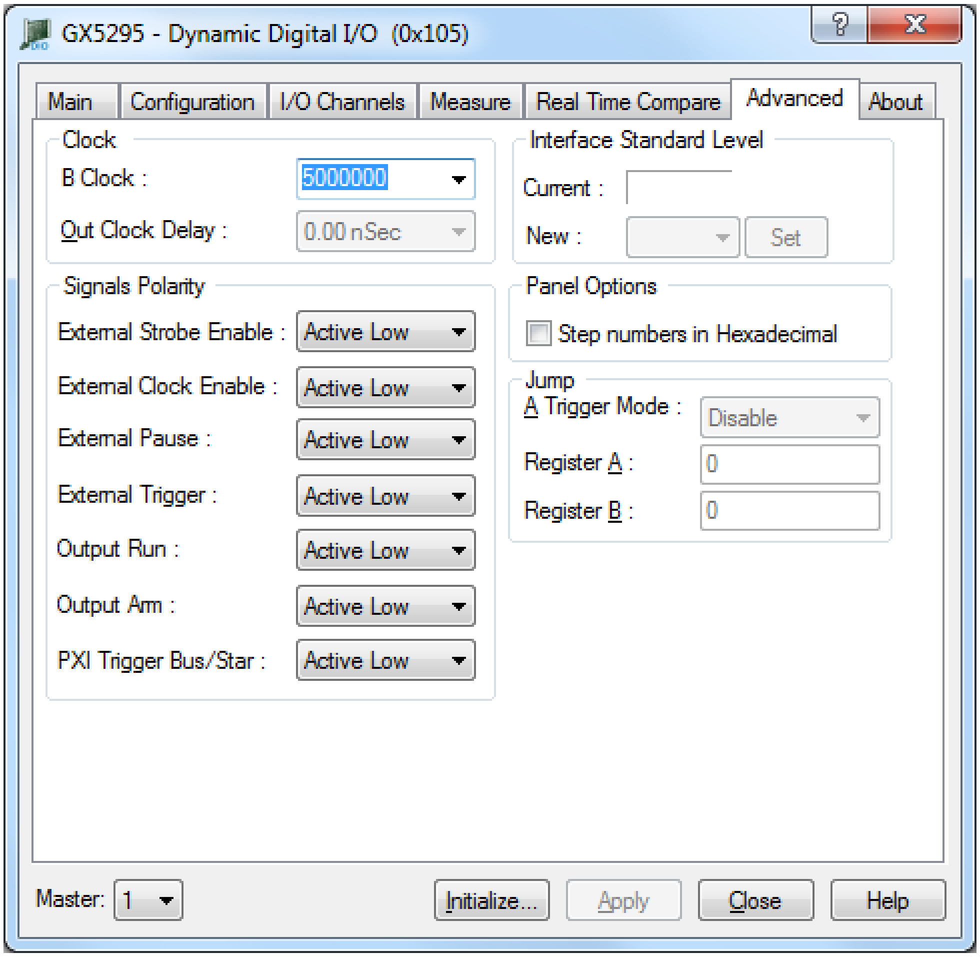
PMU Curretns group:

Measure (Check Box): When checked, start measuring channels PMU currents, only for those channels which are set to be in PMU Forced Current Mode. When unchecked stop all currents measurements. All measurements are automatically disabled when moving to another tab.

Rate (Listbox): Displays/set channels PMU forced measurements rate in measurements per sec.

Virtual Panel Advanced Page

Clicking on Advanced Page tab will show the Advanced Page as shown in Figure 4-8:



Clock

B Clock : 5000000

Out Clock Delay : 0.00 nSec

Interface Standard Level

Current :

New :

Signals Polarity

External Strobe Enable : Active Low

External Clock Enable : Active Low

External Pause : Active Low

External Trigger : Active Low

Output Run : Active Low

Output Arm : Active Low

PXI Trigger Bus/Star : Active Low

Panel Options

□ Step numbers in Hexadecimal

Jump

A Trigger Mode : Disable

Register A : 0

Register B : 0

Master: 1

Figure 4-8: Virtual Panel –I/O Channels Settings Page

Clock group:

B Clock : Sets/display the B clock frequency.

OutClock Delay: **:** Sets/display the delay added between the main clock and the out clock signal on the timing connector.

Jump group (Gx5150 only):

Jump A Trigger Mode : Sets/display the trigger mode for the external Jump A line.

Jum Register A: Sets/display the jump address for register A.

Jum Register B: Sets/display the jump address for register B.

Interface Standard Level group (Gx5290/Gx5280 only) :

Current: Display the current programmed Interface Standard Level.

New: Select the Interface Standard Level, Interface Standard Level will only be applied after rebooting the system.

Panel Options group:

Steps in hexadecimal: When checked the main tab shows steps in hexadecimal.

Signals Polarity group:

External Strobe Enable (Dropdown list): external strobe enables input signal Active mode (Timing connector) can be set as one of the following:

Active Low: signal is active when low level is present (default).

Active High: signal is active when high level is present.

External Clock Enable (Dropdown list): external clock enables input signal Active mode (Timing connector) can be set as one of the following:

Active Low: signal is active when low level is present (default).

Active High: signal is active when high level is present.

External Pause (Dropdown list): external pause enables input signal Active mode (Timing connector) can be set as one of the following:

Active Low: signal is active when low level is present (default).

Active High: signal is active when high level is present.

Active Rising Edge: signal logic when low to high transient occurs.

Active Falling Edge: signal is active when high to low transient occur.

External Trigger (Dropdown list): external trigger input signal Active mode (Timing connector) can be set as one of the following:

Active Low: signal is active when low level is present (default).

Active High: signal is active when high level is present.

Active Rising Edge: signal logic when low to high transient occurs.

Active Falling Edge: signal is active when high to low transient occur.

Output_Run (Dropdown list): output run signal Active mode (Timing connector) can be set as one of the following:

Active Low: signal is active when low level is present (default).

Active High: signal is active when high level is present.

Output Arm (Dropdown list): external arm input signal Active mode (Timing connector) can be set as one of the following:

Active Low: signal is active when low level is present (default).

Active High: signal is active when high level is present.

PXI_Trigger Bus/Start (Dropdown list): external star trigger input signal Active mode (Timing connector) can be set as one of the following:

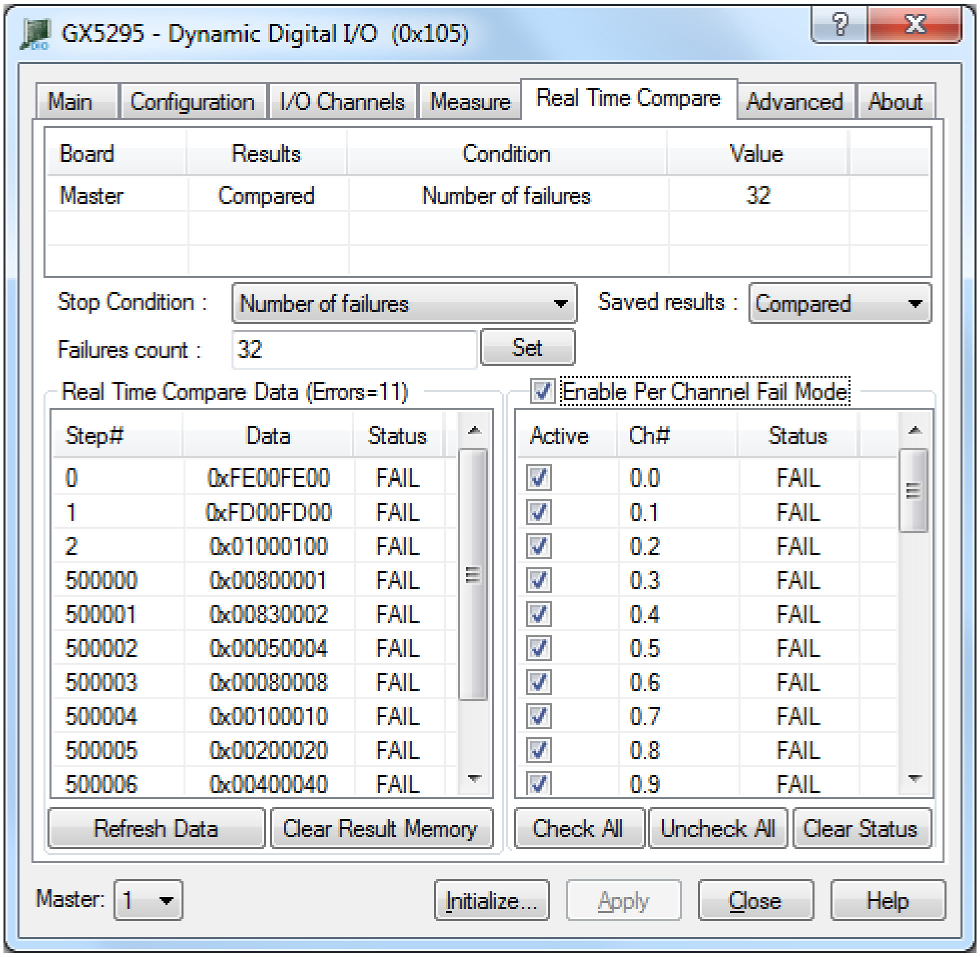
Active Low: signal is active when low level is present (default).

Active Rising Edge – signal logic when low to high transient occur.

Virtual Panel Gx529x Real Time Compare Page

The **Real Time Compare Page** will be visible only if there is an initialized Gx529x board that was set to Real Time Compare operation mode.

Clicking on **Real Time Compare Page** will show the Gx529x Real Time Compare Page as shown in Figure 4-9:



Board	Results	Condition	Value
Master	Compared	Number of failures	32

Stop Condition : Number of failures

Saved results : Compared

Failures count : 32

Real Time Compare Data (Errors=11)

Step#	Data	Status
0	0xFE00FE00	FAIL
1	0xFD00FD00	FAIL
2	0x01000100	FAIL
500000	0x00800001	FAIL
500001	0x00830002	FAIL
500002	0x00050004	FAIL
500003	0x00080008	FAIL
500004	0x00100010	FAIL
500005	0x00200020	FAIL
500006	0x00400040	FAIL

☑ Enable Per Channel Fail Mode

Active	Ch#	Status
☑	0.0	FAIL
☑	0.1	FAIL
☑	0.2	FAIL
☑	0.3	FAIL
☑	0.4	FAIL
☑	0.5	FAIL
☑	0.6	FAIL
☑	0.7	FAIL
☑	0.8	FAIL
☑	0.9	FAIL

Master: 1

Figure 4-9: Gx5295 Virtual Panel –Real Time Compare Page

Real Time Compare List (ListControl): Displays the Real Time Compare domain configuration.

Real Time Compare Data (List Control): Display the last Real Time Compare run results.

Refresh Data (Button): Read a block of data from the onboard comparison result memory. The board has to be in Real Time Compare operating mode prior calling this function. See the **DioDomainSetupOperatingMode** function for more details.

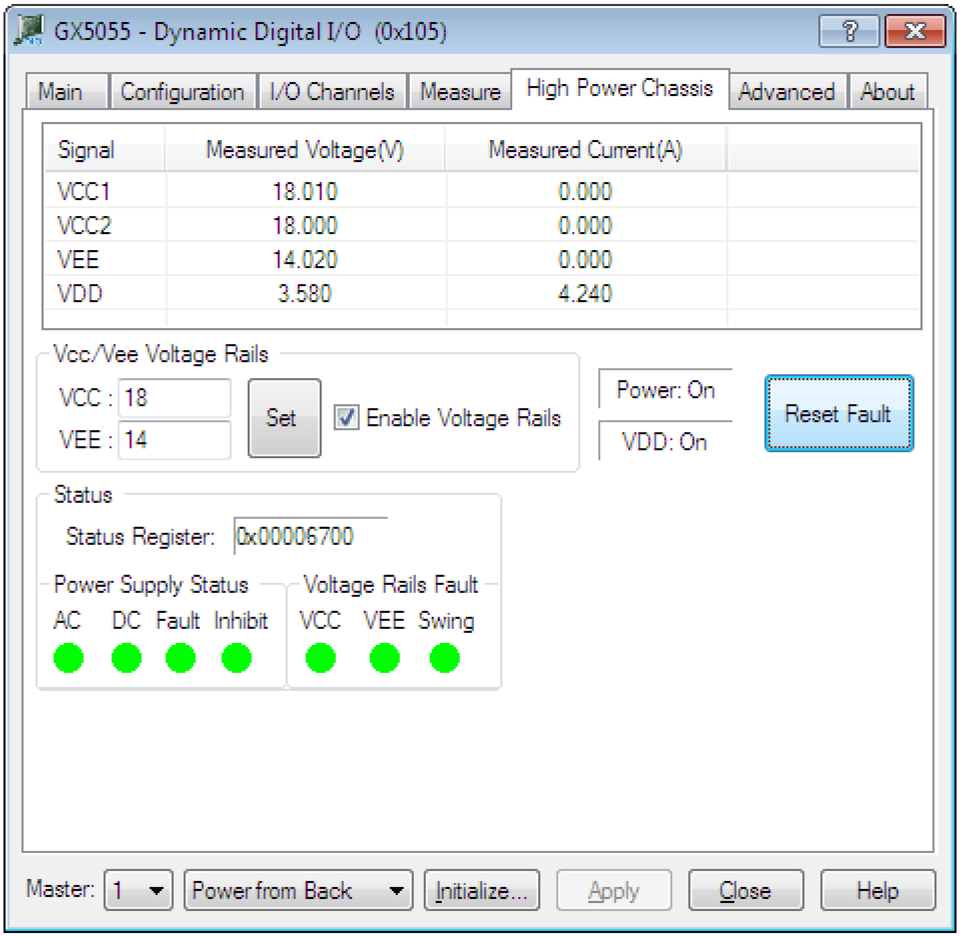
Clear Result Memory (Button): Clears all the onboard comparison result memory. This function is called by the driver internally whenever the **DioArm** command is called. The onboard comparison result memory is cleared each time the board is triggered.
Note: the Real Time Compare functionality is supported by GX529x boards with firmware versions 0x8A00 and above.
Per Channel Fail Mode (List Control): Sets/display all the channels in the domain status.
Enable Per Channel Fail Mode (Check Box Button): Sets/display the Real Time Compare mode. UnCheck=Default Real Time Compare mode operating mode. Supported by supported only by GX529x boards with

firmware versions 0x8A00 and above. Check=In this mode all default Real Time Compare operating mode are supported, but in addition when running vectors only the first failure per channels will be logged. Supported by supported only by GX5295 boards with firmware versions 0xF605 and above.
Check All (Button): Sets all the channels in the domain to active channels when in real time compare per channel fail mode.
Uncheck All (Button): Resets all the channels in the domain to active channels when in real time compare per channel fail mode.
Clear Status (Button): Clears all the channels in the domain per channel failed status. Arming clears all the channels in the domain per channel failed status.

Virtual Panel Gx5055 High Power Chassis Page

The **High Power Chassis Page** is visible only if there is a Gx5055 board in the chassis and the chassis is a Marvin Test high power chassis.

Clicking on **High Power Chassis Page** tab will show the Gx5055 High Power Chassis Page as shown in Figure 4-10:



Signal	Measured Voltage(V)	Measured Current(A)
VCC1	18.010	0.000
VCC2	18.000	0.000
VEE	14.020	0.000
VDD	3.580	4.240

VCC : 18

VEE : 14

☑ Enable Voltage Rails

Power: On

VDD: On

Status Register: 0x00006700

Master: 1

Power from Back

Figure 4-10: Gx5055 Virtual Panel –High Power Chassis Page

The following controls are shown in the High Power Chassis page:

Signal List (ListControl): Displays the VCC1, VCC2, VEE, and VDD measured voltage and measured currents.

Vcc/Vee Voltage Rails (GroupBox):

VCC (EditBox): Sets/displays the VCC voltage.

VEE (EditBox): Sets/displays the VEE voltage.

Set Button: Sets both VCC and VEE specified voltages.

Enable Voltage Rails (CheckBox): When checked, the Vcc and Vee voltage rails are enabled.

Power: On status: Displays the Power supply state.

VDD: On status: Displays the VDD voltage rail state.

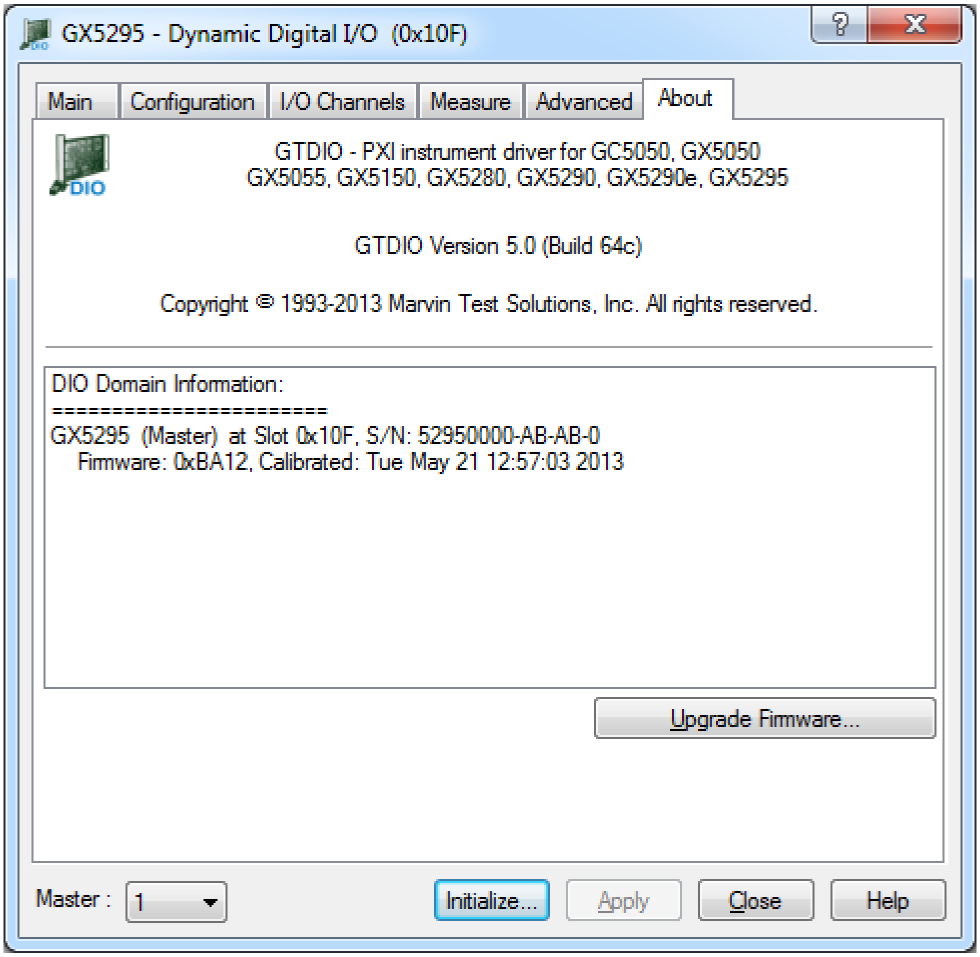
Reset Fault (Button): When pressed it will clear all pending power supply faults, disable both Vcc and Vee voltage rails.

Status (GroupBox):

Displays the power supply different components status.

Virtual Panel About Page

Clicking on the **About** tab will show the **About** page as shown in Figure 4-11:



GTDIO - PXI instrument driver for GC5050, GX5050
GX5055, GX5150, GX5280, GX5290, GX5290e, GX5295

GTDIO Version 5.0 (Build 64c)

Copyright © 1993-2013 Marvin Test Solutions, Inc. All rights reserved.

```
DIO Domain Information:
=======================
GX5295  (Master)  at  Slot 0x10F, S/N: 52950000-AB-AB-0
   Firmware: 0xBA12, Calibrated: Tue May 21 12:57:03 2013
```

Figure 4-11: Virtual Panel – About Page

The following controls are shown in the About page:

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

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 Marvin Test Solutions support. The upgrade requires a firmware file (.jam) that is written to the board FPGA. After the upgrade is complete, you must shut down the computer to recycle power to the board.

Chapter 5 - Using DIOEasy with the GC5050/GX5050

Introduction

Using ***DIOEasy*** with the GC5050/GX5050 section introduces fundamental commands and features of DIOEasy software. A more comprehensive discussion of menu commands can be found in chapter 4 - **Using *DIOEas*y**.

A tutorial is presented first, to demonstrate how to produce a test vector for a simple UUT example. This is followed by procedures for verifying and saving results. Other topics include the Virtual Panel, procedures for running and saving vectors and methods for comparing actual to anticipated results.

Planning and Writing Example Testware

The first section of this chapter develops a test procedure to test a simple unit under test (UUT). The UUT is a simple, example circuit board. The reader is advised to first focus on the approach, not the details of the ***DIOEasy*** syntax. The best way to become familiar with the syntax is to write some testware code for a simple UUT.

Choosing the DIO Board

The project requires testware for a simple UUT with limited TTL-level I/O. The example test procedure must both generate test vectors and acquire UUT responses. Should the GC5050/GX5050 or GX5150 be used for this example project?

It takes both a GX5150 and GX5151, two boards, to generate vectors and acquire responses. A single GC5050/GX5050 board can do both, acquire or generate vectors. In addition, the GC5050/GX5050 is TTL-compatible by default so that no I/O module is needed. Therefore, a GC5050/GX5050 is the more efficient solution for the example.

The user should be aware that some ***DIOEasy*** functions work with only one DIO family. Therefore, function availability must also be considered when choosing a board for a test project. See the section “Functions Reference” in the book “Programmer’s Reference” for more information about function support by DIO family.

An Example UUT

The example schematic of Figure 5-1 shows a simple, one-board UUT.

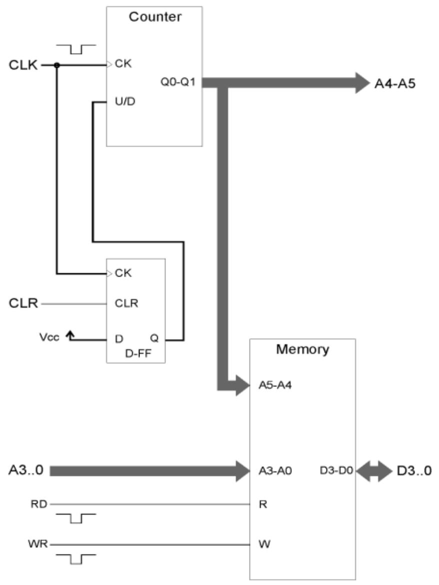


Figure 5-1: Example UUT Schematic

The first step is to map various UUT states and modes. A counter provides two MSBs (most significant bits) for addressing a 64x4 bit static RAM (random access memory). RAM I/O (input/output) data share lines D0 – D3. The four address LSBs (least significant bits), A0 – A3, are inputs.

When UUTs are powered on, they can be in an unpredictable state. They must be initialized to a known state before testing can begin. Procedures for initialization are specific to each UUT design. Generally, all memory elements that affect the test must be initialized to set the UUT to a known state. In this example, initializing the board is achieved by initializing the counter. A vector is then applied to the inputs to test all board functions.

Initializing the UUT

Counter outputs A4 and A5 are initially in an unknown state and are mapped to DIO channels 8 and 9. They also connect to the two LSBs on the DIO External Event input bus. (Other External Event bus inputs are grounded). DIO channel 0 (CLK) generates the board clock input. There is no way to reset the counter. When initializing the board to a known state, the counter is clocked and its output sampled until the desired starting count, A4 and A5 both low, is present.

Table 5-1 displays a GC5050/GX5050 board setup to initialize the counter. The table specifies channels, channel groups, channel names and direction along with commands, labels and values for the first 20 steps. It is followed by a detailed explanation.

											I/O Ch	0	1	2	3	4	5	6	7	8	9	16	17	18	19
		Ch. Group Direction									Group	1	1	1	1	1	1	1	1	2	2	3	3	3	3
Step	**Label**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Cmd**	**Op / Label**	**CLK**	**CLR**	**RD**	**WR**	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**D0**	**D1**	**D2**	**D3**
0	CNTINIT	O	I	O								1	1	1	1	0	0	0	0	X♥	X	X	X	X	X
1		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
2		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
3		O	I	O						Set A	0	1	1	1	1	0	0	0	0	X	X	X	X	X	X
4		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
5		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
6		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
7		O	I	O						Set B	4	1	1	1	1	0	0	0	0	X	X	X	X	X	X
8	LOOPINIT	O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
9		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
10		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
11		O	I	O						JE A	RAMTEST	1	1	1	1	0	0	0	0	X	X	X	X	X	X
12		O	I	O								0	1	1	1	0	0	0	0	X	X	X	X	X	X
13		O	I	O								0	1	1	1	0	0	0	0	X	X	X	X	X	X
14		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
15		O	I	O						LOOP B	LOOPINIT	1	1	1	1	0	0	0	0	X	X	X	X	X	X
16		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
17		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
18		O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
19		O	I	O						HALT		1	1	1	1	0	0	0	0	X	X	X	X	X	X
20	RAMTEST	O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X

Notes: O = Output, I = Input; X = Don’t care

Table 5-1: Initializing the Counter

The following steps explain the commands in Table 5-1.

1. Set Register A to zero 0 (Step 3).
2. Set Register B to 4 (Step 7).
3. A conditional JUMP to the RAMINIT label occurs if EXT=A. This ensures that the program will terminate that loop (by jumping to Step 20) when the two LSB outputs of the Counter reach the “00” state (Step 11).
4. The CLK input goes low to provide the Counter with a clock signal (Step 12).
5. The CLK input returns to high (Step 14) and the sequencer performs an unconditional jump to LOOPINIT (Step 8).
6. Set a trap to HALT the program in the event the Counter cannot be initialized. Regardless of its initial state, the Counter should be initialized in 1 to 3 clock signals. The described LOOP will provide a maximum of 4 clock signals before entering the trap (Step 19).

Testing the RAM

Writing to the RAM

When the test program reaches Step 20, the counter outputs A4 and A5 are both low and the UUT is initialized. Now Group 2 channels (including channels 8 - 9) are redefined as receivers with well-defined inputs.

At Step 20, Group 3 (D0, D3, including channels 16-19) is now reversed and outputs 0000 (0x0) to the board. LSB Address lines A0 - A3 are set to 0x0.

At Step 21 (Table 5-2), WR is driven low at Channel 3 to enable writing to RAM. In Step 22, WR is returned high. Three steps are used for each WR cycle: (1) to avoid "races" within the UUT, (2) to assure that data and address inputs are stable when the WR signal is applied, and (3) to assure that data and address inputs are static while WR is active.

These three steps repeat 64 times. Data is written to all memory locations. For the first 16 locations, data written to RAM is set to the address on lines A0-A3. In the second 16, it is inverted. In the third 16, 0xAA is written and in the final 16 locations, 0x55 is written.

After the first 16 write cycles, channel 0 is clocked to increment the counter to state 01 while writing continues for the next 16 cycles, etc. Step 214 completes writing to RAM.

Table 6-2 below describes DIO channels and steps that have been implemented to write data to RAM.

											I/O	0	1	2	3	4	5	6	7	8	9	16	17	18	19
		Ch. Group Direction									Group	1	1	1	1	1	1	1	1	2	2	3	3	3	3
Step	**Label**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Cmd**	**Op / Label**	**CLK**	**CLR**	**RD**	**WR**	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**D0**	**D1**	**D2**	**D3**
20	RAMTEST	O	I	O								1	1	1	1	0	0	0	0	X	X	X	X	X	X
21	WR0	O	I	O								1	1	1	0	0	0	0	0	0	0	0	0	0	0
22		O	I	O								1	1	1	1	0	0	0	0	0	0	0	0	0	0
23		O	I	O								1	1	1	1	1	0	0	0	0	0	1	0	0	0
24	WR1	O	I	O								1	1	1	0	1	0	0	0	0	0	1	0	0	0
25		O	I	O								1	1	1	1	1	0	0	0	0	0	1	0	0	0
26		O	I	O								1	1	1	1	0	1	0	0	0	0	0	1	0	0
27	WR2	O	I	O								1	1	1	0	0	1	0	0	0	0	0	1	0	0
28		O	I	O								1	1	1	1	0	1	0	0	0	0	0	1	0	0
Step	**Label**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Cmd**	**Op / Label**	**CLK**	**CLR**	**RD**	**WR**	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**D0**	**D1**	**D2**	**D3**
64		O	I	O								1	1	1	1	0	1	1	1	0	0	0	1	1	1
65		O	I	O								1	1	1	1	1	1	1	1	0	0	1	1	1	1

											I/O	0	1	2	3	4	5	6	7	8	9	16	17	18	19
		Ch. Group Direction									Group	1	1	1	1	1	1	1	1	2	2	3	3	3	3
Step	**Label**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Cmd**	**Op / Label**	**CLK**	**CLR**	**RD**	**WR**	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**D0**	**D1**	**D2**	**D3**
66	WR15	O	I	O								1	1	1	0	1	1	1	1	0	0	1	1	1	1
67		O	I	O								1	1	1	1	1	1	1	1	0	0	1	1	1	1
68		O	I	O								0	1	1	1	1	1	1	1	0	0	1	1	1	1
69		O	I	O								1	1	1	1	1	1	1	1	1	0	0	0	0	0
70	WR16	O	I	O								1	1	1	0	1	1	1	1	1	0	0	0	0	0
71		O	I	O								1	1	1	1	1	1	1	1	1	0	0	0	0	0
72		O	I	O								1	1	1	1	0	1	1	1	1	0	1	0	0	0
Step	**Label**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Cmd**	**Op / Label**	**CLK**	**CLR**	**RD**	**WR**	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**D0**	**D1**	**D2**	**D3**
204	WR60	O	I	O								1	1	1	0	1	1	0	0	0	0	1	0	1	0
205		O	I	O								1	1	1	1	0	0	1	1	1	1	0	1	0	1
206		O	I	O								1	1	1	1	1	0	1	1	1	1	0	1	0	1
207	WR61	O	I	O								1	1	1	0	1	0	1	1	1	1	0	1	0	1
208		O	I	O								1	1	1	1	1	0	1	1	1	1	0	1	0	1
209		O	I	O								1	1	1	1	0	1	1	1	1	1	0	1	0	1
210	WR62	O	I	O								1	1	1	0	0	1	1	1	1	1	0	1	0	1
211		O	I	O								1	1	1	1	0	1	1	1	1	1	0	1	0	1
212		O	I	O								1	1	1	1	1	1	1	1	1	1	0	1	0	1
213	WR63	O	I	O								1	1	1	0	1	1	1	1	1	1	0	1	0	1
214		O	I	O								1	1	1	1	1	1	1	1	1	1	0	1	0	1

Table 5-2: Writing to RAM

Reading from RAM

To complete the test, all RAM locations are read back. In Step 215, the direction of Group 3 (channels 16-19) is changed to input. The expected data is defined only when the RD signal is set low.

After reading the first 16 locations, the Counter advances to the position (Step 264) to initiate an additional cycle to read the next 16 locations, etc.

Table 5-3 below describes DIO channels and steps for reading the RAM. The result of this reading can then be compared to the data actually written to RAM.

											I/O Ch.	0	1	2	3	4	5	6	7	8	9	16	17	18	19
		Ch. Group									Group	1	1	1	1	1	1	1	1	2	2	3	3	3	3
Step	Label	1	2	3	4	5	6	7	8	Cmd	Op / Label	CLK	CLR	RD	WR	A0	A1	A2	A3	A4	A5	D0	D1	D2	D3
215	STRTRD	O	I	I								0	1	1	1	0	0	0	0	1	1	X	X	X	X
216		O	I	I								1	1	1	1	0	0	0	0	0	0	X	X	X	X
217	RD0	O	I	I								1	1	0	1	0	0	0	0	0	0	0	0	0	0
218		O	I	I								1	1	1	1	0	0	0	0	0	0	X	X	X	X
219	WR3	O	I	I								1	1	1	1	1	0	0	0	0	0	X	X	X	X
220	RD1	O	I	I								1	1	0	1	1	0	0	0	0	0	1	0	0	0
221		O	I	I								1	1	1	1	1	0	0	0	0	0	X	X	X	X
222		O	I	I								1	1	1	1	0	1	0	0	0	0	X	X	X	X
223	RD2	O	I	I								1	1	0	1	0	1	0	0	0	0	0	1	0	0
224		O	I	I								1	1	1	1	0	1	0	0	0	0	X	X	X	X
Step	Label	1	2	3	4	5	6	7	8	Cmd	Op / Label	CLK	CLR	RD	WR	A0	A1	A2	A3	A4	A5	D0	D1	D2	D3
7		O	I	I								1	1	1	1	1	0	1	1	1	1	X	X	X	X
405		O	I	I								1	1	1	1	0	1	1	1	1	1	X	X	X	X
406	RD62	O	I	I								1	1	0	1	0	1	1	1	1	1	0	1	0	1
407		O	I	I								1	1	1	1	0	1	1	1	1	1	X	X	X	X
408		O	I	I								1	1	1	1	1	1	1	1	1	1	X	X	X	X
409	RD63	O	I	I								1	1	0	1	1	1	1	1	1	1	0	1	0	1
410		O	I	I								1	1	1	1	1	1	1	1	1	1	X	X	X	X

Table 5-3: Reading from RAM

Testing the D Flip-flop

The last test is designed to verify functionality of the D flip-flop. In Step 411 (Table 5-4), CLR is applied to force the U/D input to logic high. CLK at Step 412 causes the Counter to count down to A4=0 and A5=1. The RD cycle in Step 414 will read address 0 from the 3rd bank (AA).

At Step 415, the sequencer "End Of Program" command ends the test.

Table 5-4 demonstrates a program for testing the D flip-flop.

											I/O Ch.	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**16**	**17**	**18**	**19**
		Ch. Group Direction									**Group**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**2**	**3**	**3**	**3**	**3**
Step	**Label**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Cmd**	**Op / Label**	**CLK**	**CLR**	**RD**	**WR**	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**D0**	**D1**	**D2**	**D3**
411		O	I	I								1	0	1	1	1	1	1	1	1	1	X	X	X	X
412		O	I	I								0	0	1	1	1	1	1	1	1	1	X	X	X	X
413		O	I	I								1	0	1	1	0	0	0	0	0	1	X	X	X	X
414	RD0	O	I	I								1	0	0	1	0	0	0	0	0	1	1	0	1	0
415	END	O	I	I						HALT		1	0	1	1	0	0	0	0	0	1	X	X	X	X

Table 5-4: Testing the D Flip-flop

Adding Data and Commands to Vector Files

The methods described below are discussed in more detail in "Chapter 4 - Using ***DIOEasy***".

Defining Channels

The first step in entering vector file data is to assign new channel names to channels in Vector View as specified in Table 5-1.

1. Double-click any name in the Vector View name column (for example, CH0) to make the name editable.
2. Edit or type a new name (for example, CLK). Press **Enter** or click the left mouse button to save the name. Click on the next channel name to be edited and continue.
3. Repeat this procedure and continue selecting channels and assigning names until new names are assigned to I/O channels 0-19.
4. Channel names - channels for this example are: CLK - 0, CLR - 1, RD - 2, WR - 3, A0 - 4, A1 - 5, A2 - 6, A3 - 7, A4 - 8, A5 - 9, D0 - 16, D1 - 17, D2 - 18 and D3 - 19. Channel names should match the channel numbers as specified in the example tables.

Notice that not all channel numbers need channel names.

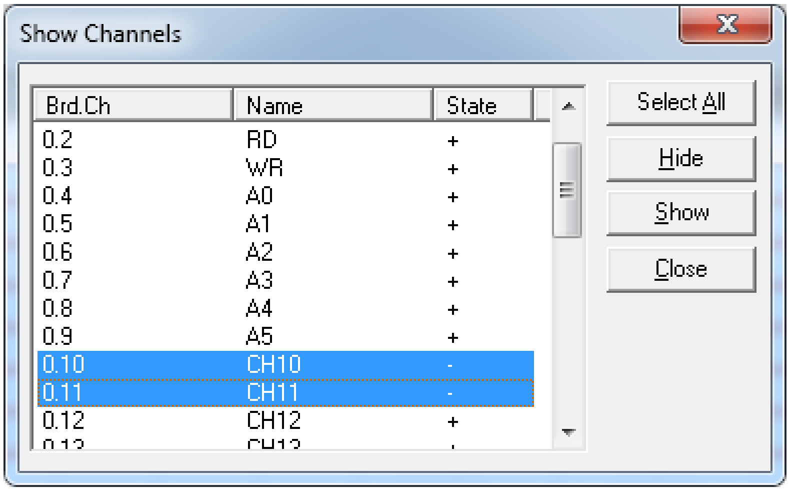
Hiding Unused Channels

Because CH10 - CH15 are not used by our vector, we can hide these channels to help reduce clutter on the Vector View display.

1. To hide one or more channels, select them by clicking a channel name in Vector View (for example, CH10). In the name column, drag the mouse over a group of channels to hide.

2. Select **Channel** from the Main menu and click the **Hide** command. Selected channel(s) no longer display in Vector View.

3. To redisplay a hidden channel, select Channel/Show from the Main menu. A dialog box (Figure 5-2) displays. Viewable channels display with a "+" and hidden channels display a "–" in the *State* column. Select the channels to display and click **Show** (multiple selections are supported). The selected state(s) display "+". Now click **OK**. Selected channels that were hidden now display in Vector View. Channels, which are selected but not hidden, are unaffected.



Brd.Ch	Name	State
0.2	RD	+
0.3	WR	+
0.4	A0	+
0.5	A1	+
0.6	A2	+
0.7	A3	+
0.8	A4	+
0.9	A5	+
0.10	CH10	-
0.11	CH11	-
0.12	CH12	+
0.13	CH13	+

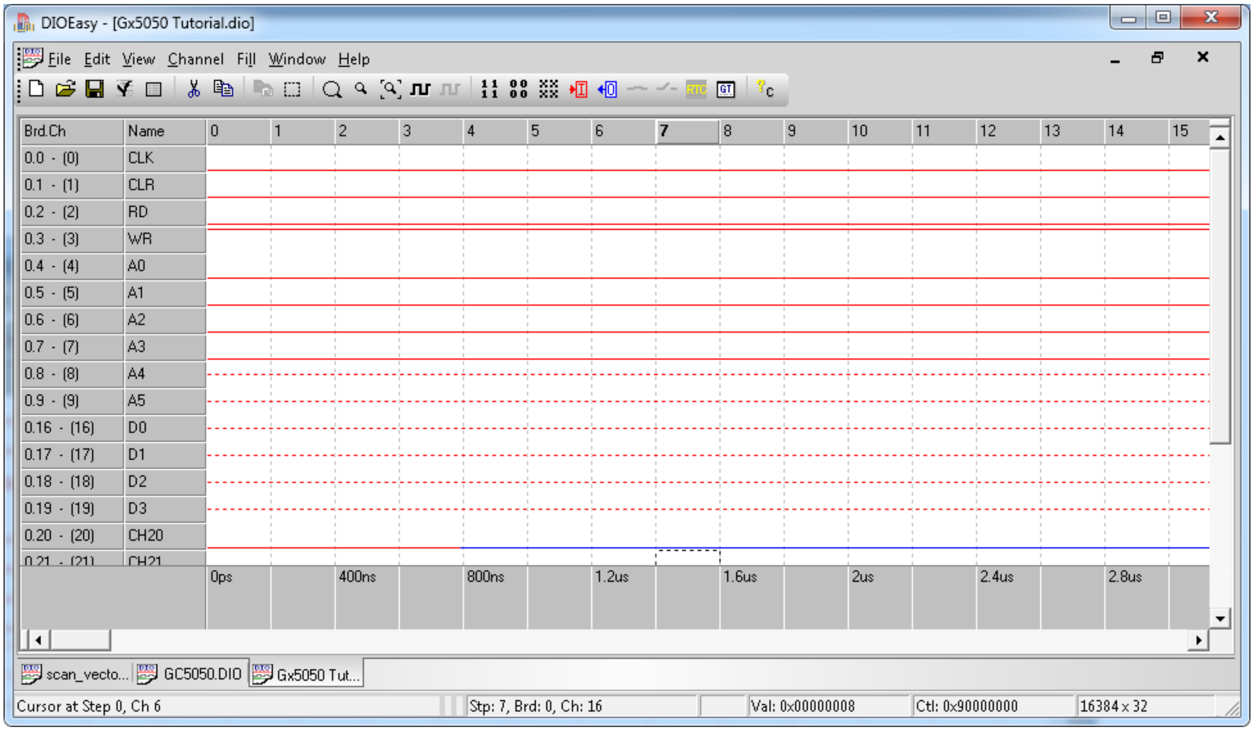
Figure 5-2: Show Channels Dialog Box

Adding Data and Controlling GC5050/GX5050 I/O Direction

Channels 0-3, which are at logic "one" for most steps, can be set to all ones. To do that, select Channels 0-3 (click 'CLK' and drag to 'WR'), then click the 'One' toolbar button. The step channels now display a '1' level. Click the 'Output' toolbar button. This designates the direction of all channels in the group (channels 0-7) as output. (Note: Input and Output buttons for GX5150 DIO files are not available). For this example, channels must be in numerical order.

Continue the process. Select channels 6-7, but this time, click the 'Zeros' toolbar button.

Continuing, fill channels 8-19 with "Don't Care" by clicking the "Don't Care" toolbar button. This displays a half-level dashed line. Designate Channels 8-9 as inputs by clicking the "Input" toolbar button. Channels 8 -15 are now designated inputs. Similarly, designate Channels 16 - 19 as outputs. Channels 16 - 23 are all outputs now. Figure 5-3 shows Vector View after all these operations are completed.


DIOEasy - [Gx5050 Tutorial.dio]
File Edit View Channel Fill Window Help
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0ps 400ns 800ns 1.2us 1.6us 2us 2.4us 2.8us
scan_vecto... GC5050.DIO Gx5050 Tut...

Brd.Ch	Name
0.0 - (0)	CLK
0.1 - (1)	CLR
0.2 - (2)	RD
0.3 - (3)	WR
0.4 - (4)	A0
0.5 - (5)	A1
0.6 - (6)	A2
0.7 - (7)	A3
0.8 - (8)	A4
0.9 - (9)	A5
0.16 - (16)	D0
0.17 - (17)	D1
0.18 - (18)	D2
0.19 - (19)	D3
0.20 - (20)	CH20
0.21 - (21)	CH21

Cursor at Step 0, Ch 6 | Stp: 7, Brd: 0, Ch: 16 | Val: 0x00000008 | Ctl: 0x90000000 | 16384 x 32

Figure 5-3: Vector View after the Basic Steps

Note: Commands introduced here can also be implemented using the Value selection in the Fill menu. See "Chapter 6 - Using ***DIOEasy***".

To change Steps 12 and 13 to ‘0’ in the CLK Channel (Channel 0), select Channel 0, Step 12 and click the right mouse button until the step is set to zero. Repeat for Step 13.

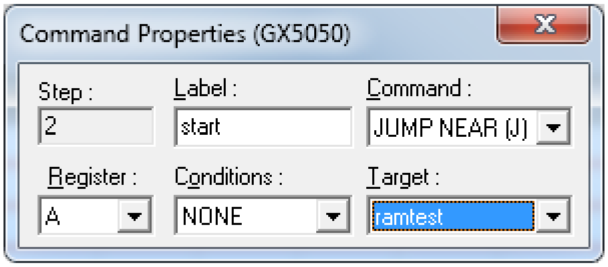
Commands and Labels

The next procedure programs commands using the **Command Properties** dialog box.

Use this dialog box to select a **Command,** add a step **Label**, and specify a **Register**, **Conditions** and jump **Labels** for flow control.

You can open this dialog box by first clicking anywhere in a Vector View step, then select **Command Properties** from the **View** menu or press **Alt** + **Enter**. The easiest way is to double-click the Command Bar at the step.

The Command Properties dialog box always displays Step, Label and Command fields (Figure 5-4). Depending on the Command selected, additional fields can display. These are Register, Conditions and Target. These additional fields are used to enter modifiers that reference a specific register, condition or target label.



Command Properties (GX5050)

Step : 2

Label : start

Command : JUMP NEAR (J)

Register : A

Conditions : NONE

Target : ramtest

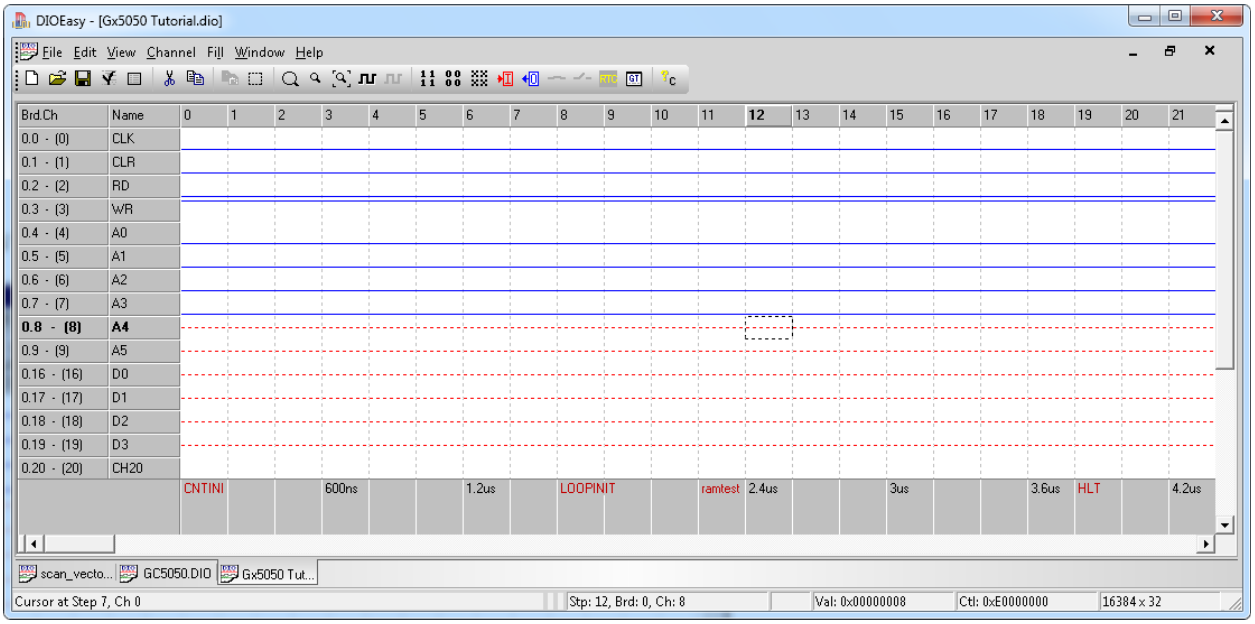
Figure 5-4: Adding Commands and Labels

Command properties are stored and displayed on the Command Bar when the mouse is clicked outside the Command Properties dialog box (that is, when the dialog box loses focus).

The following describes how to enter commands and properties called for in Table 5-1:

1. Select Step 0. Open the **Command Properties** dialog box and type **cntinit** in the Label field.
2. Select Step 3. Open the **Command Properties** dialog box. Select **SET** from the Command list. Select **A** from the Register list and leave the Value at '**0**'. No label is required for this step.
3. Select Step 7. Open the **Command Properties** dialog box and select **SET** from the Command list. Select **B** from the Register list and type in a **Value** of **4**.
4. Select Step 8. Open the **Command Properties** dialog box and type **loopinit** in the Label text box.
5. Select Step 20. Open the **Command Properties** dialog box and type the **ramtest** in the Label text box. This label must be defined for the command in Step 11.
6. Select Step 11. Open the **Command Properties** dialog box and select **JUMP NEAR (J)** from the Command list. Select **= (E)** (Equal) from the Conditions list and select register **A**. Select **RAMTEST** in the Target list. (Table 5-4).
7. Select Step 14. Open the **Command Properties** dialog box. Then select **LOOP** from the Command list and **B** from the Register list ("Condition" should be set to **None**). Label with **loopinit**.
8. Select Step 19. Open the **Command Properties** dialog box and select **HALT** from the Command list.
9. Continue filling data and entering labels as specified in Table 5-1 through Table 5-4.

Figure 5-5 shows the Vector View window after commands and labels are inserted.



Brd.Ch	Name
0.0 - (0)	CLK
0.1 - (1)	CLR
0.2 - (2)	RD
0.3 - (3)	WR
0.4 - (4)	A0
0.5 - (5)	A1
0.6 - (6)	A2
0.7 - (7)	A3
0.8 - (8)	**A4**
0.9 - (9)	A5
0.16 - (16)	D0
0.17 - (17)	D1
0.18 - (18)	D2
0.19 - (19)	D3
0.20 - (20)	CH20

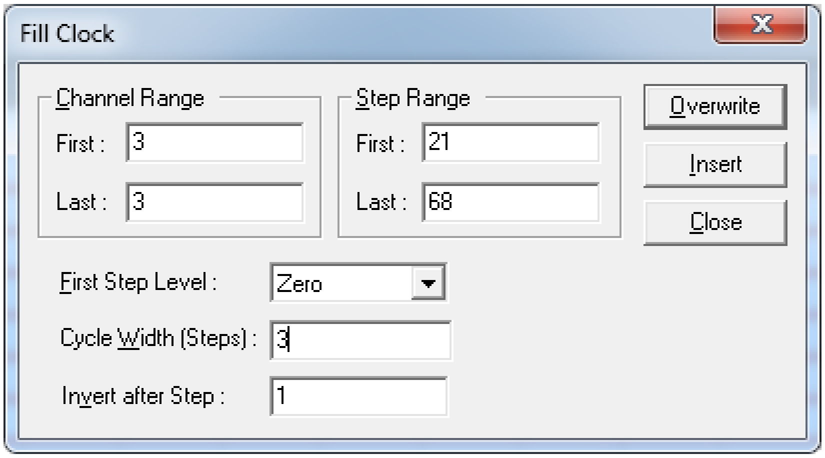
Cursor at Step 7, Ch 0 | Stp: 12, Brd: 0, Ch: 8 | Val: 0x00000008 | Ctl: 0xE0000000 | 16384 x 32

Figure 5-5: Vector View after Adding Commands

Using the Fill Clock Command

Steps 21 - 68 of Channel 3 have a repeating clock pattern of 0, 1, 1, 0, 1, 1, etc. The **Fill Clock** command can be used to write this pattern in one operation. To use the **Fill Clock** command:

1. Click on Fill and select the Clock command. The Fill Clock dialog box displays (Figure 5-6).
2. Select Channel Range (3 and 3) and Step Range (21 and 68).
3. Select Zero as the Start level.
4. Enter ‘3’ in the cycle width and ‘1’ in the invert after step.
5. Click **Overwrite!**



Channel Range
First : 3
Last : 3

Step Range
First : 21
Last : 68

First Step Level : Zero

Cycle Width (Steps) : 3

Invert after Step : 1

Figure 5-6: Fill Clock Dialog

Using the Fill Ramp Command

Channels 16-19 contain a repeating pattern. The value is incremented by 1 every third step. We can use the **Fill Ramp** command to write this data to memory in one operation, starting with steps 21 through 68. To fill using the Fill Ramp command:

1. Click **Fill** and select **Ramp**. This displays the **Fill Ramp** Values dialog box.
2. Select the Channel Range (16 through 19) and the Step Range (21 through 68).
3. Select a First Step value of '0,' a Limit Value of '15' and an Increment by of '1'. Select '3' as the Steps increment.
4. Click **Overwrite!**

Using Copy and Paste to Duplicate Data

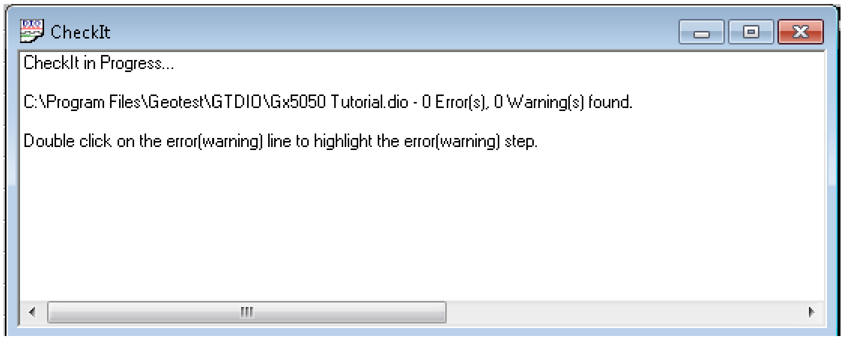
The ramp data on channels 16-19, steps 21-68, can be copied to Channels 6-7, Steps 21-68 using **Copy** and **Paste** edit commands:

1. Select Channels 16-19, Steps 21-68, by using the mouse or the **Edit/Select** command.
2. Select **Copy** from the Edit menu or use the toolbar button.
3. Click on Channel 4, Step 21.
4. Select **Paste** from the Edit menu or use the toolbar button.

Continue the vector described in Table 5-1 through Table 5-4 using techniques covered in this chapter.

Checking Your Work

It is time now to check for errors (for example, a branch to an undefined label). Select **CheckIt** from the **File** menu or use the toolbar button. The **CheckIt** window displays error messages while checking results (Figure 5-7).



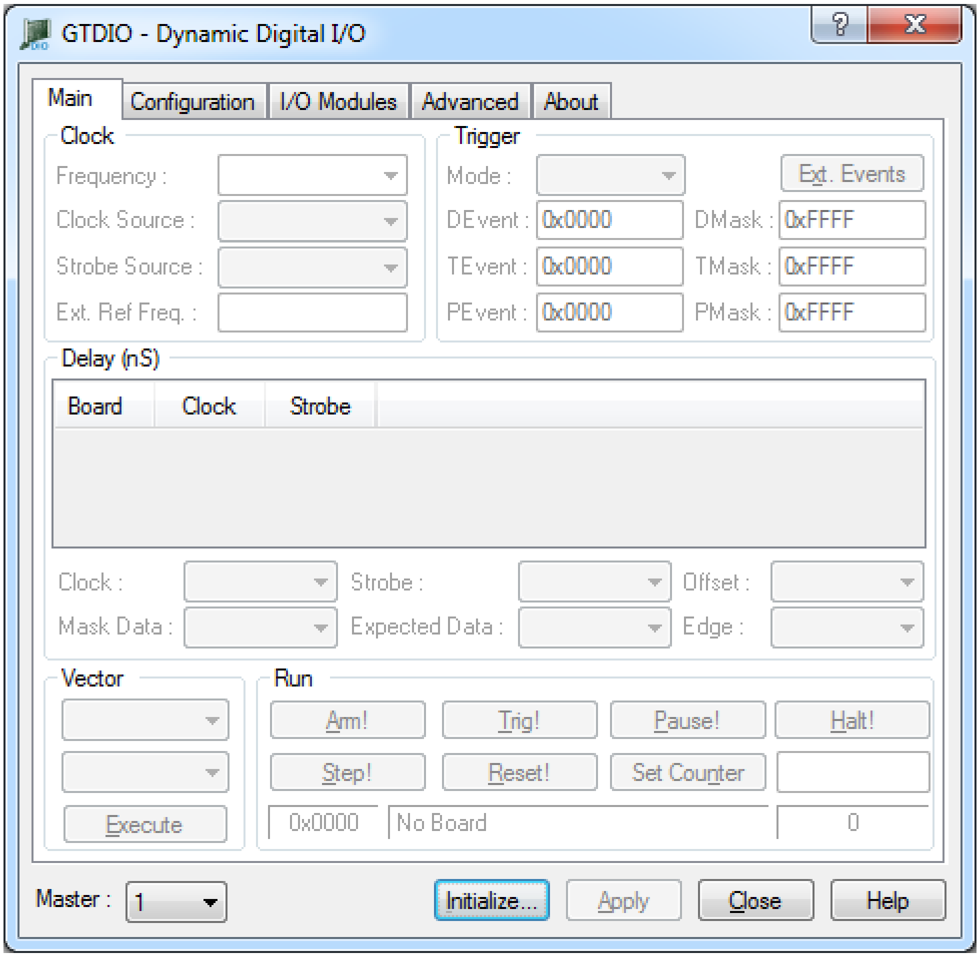
```
CheckIt in Progress...

C:\Program Files\Geotest\GTDIO\Gx5050 Tutorial.dio - 0 Error(s), 0 Warning(s) found.

Double click on the error(warning) line to highlight the error(warning) step.
```

Figure 5-7: Checking the Vector File

CheckIt displays errors by listing each step containing an error with an explanation of what caused that error. Errors can then be corrected using the **Command Properties** dialog box. The total number of errors found displays in the list.

Using the Panel

After the vector file is created, the GTDIO Panel can be used to test it interactivity. To open the panel, click **GTDIO Panel** in the **Window** menu. A panel similar to Figure 5-8 appears.



GTDIO - Dynamic Digital I/O

DEvent : 0x0000
DMask : 0xFFFF
TEvent : 0x0000
TMask : 0xFFFF
PEvent : 0x0000
PMask : 0xFFFF

0x0000 No Board 0

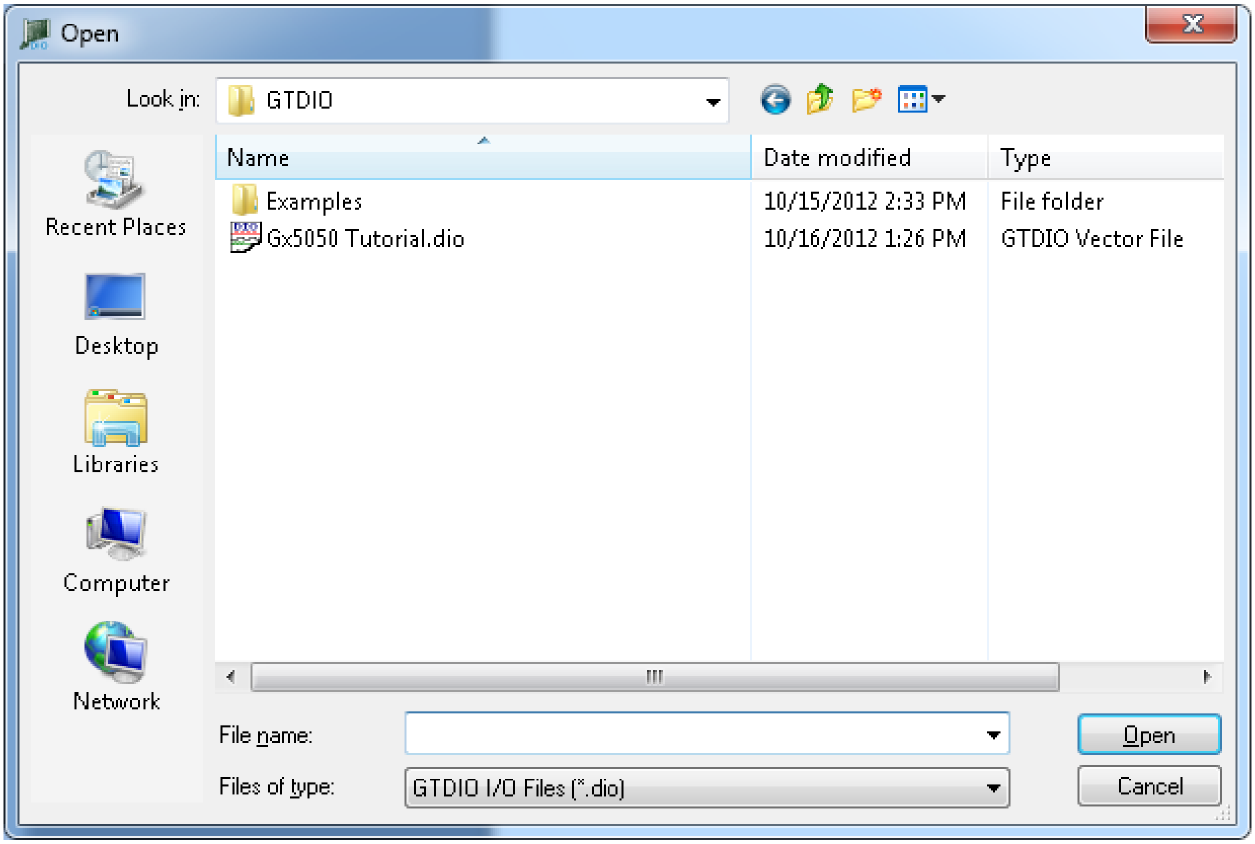
Master : 1

Figure 5-8: The GTDIO Panel

This panel can be used to configure the domain, change setup parameters, initialize the board and load a vector file. It can arm (ARM) and trigger (TRIG) the board. During HALT, it saves captured data to a file. "Chapter 6 - Using ***DIOEasy***" contains a more detailed description of this window.

Saving Your Work

To save the vector, select **Save** from the **File menu**. A **Save As** dialog box opens. Type the name of the file (for example, ***MyDioFile***) as shown in Figure 5-9 and click **Save**. ***DIOEasy*** saves the named file to the current directory and appends a **DIO** extension.



Open

Look in: GTDIO

Name	Date modified	Type
Examples	10/15/2012 2:33 PM	File folder
Gx5050 Tutorial.dio	10/16/2012 1:26 PM	GTDIO Vector File

File name:

Files of type: GTDIO I/O Files (*.dio)

Figure 5-9: Saving the Vector File

The **File Open…** command can load and display a result file (**DI** extension) in Vector View. A DI result file can be compared to a DIO vector file or DI result file using the **Compare** command in the **File** menu.

Results can be inspected against the vector file. The DI file can be used to explain why an unexpected result occurs.

See the book Programmer's Reference User's Guide for details about the ***DIO*** driver and board programming.

Chapter 6 - Using *DIOEasy* with the GX515X

Introduction

This chapter introduces fundamental commands and features of ***DIOEasy*** software. A more comprehensive discussion of menu commands can be found in “Chapter 4 - Using ***DIOEasy***”.

A tutorial is presented first, to demonstrate how to produce a test vector for a simple UUT example. This is followed by procedures for verifying and saving results. Other topics include the Virtual Panel, procedures for running and saving vectors and methods for comparing actual to anticipated results.

UUT Description

In our example we will demonstrate how to test a specific UUT using the DIO board and creating and editing a vector file using ***DIOEasy***.

UUT Environment

The example UUT (Figure 6-1) is a circuit board that contains an 8-bit DAC and a local bus controller (LBC). The local bus controller connects to a larger distributed bus system.

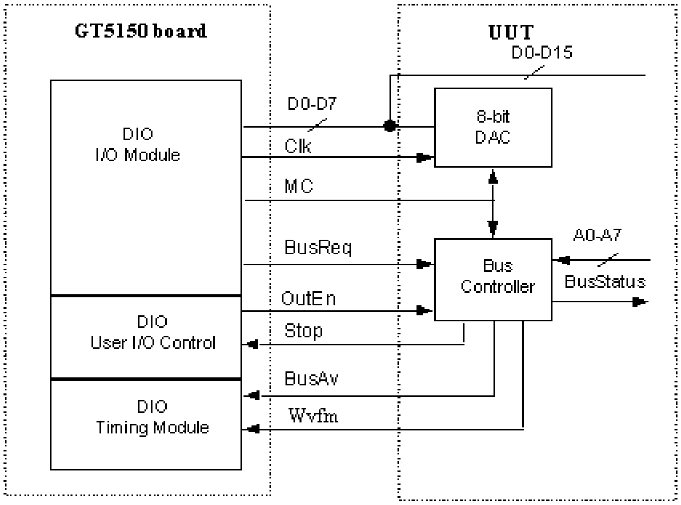


Figure 6-1: GX5150 and UUT

A Master Controller Unit (MCU, not shown) coordinates all bus access by granting bus access to an LBC specified by an 8-bit address on the Address Line. The LBC, in turn, grants access to the DIO. When the LBC is finished using the bus, it relinquishes control to the MCU.

The LBC uses the Status and Control Bus to report back to the MCU.

DIO/UUT Interface

The DIO enables the 8-bit DAC and generates the clock (**Clk**) as the data strobe. Since the DIO and the UUT share the same data bus, the DIO signals to the UUT that it is ready to RUN. The UUT controls when the DIO run and when break out of the program.

The DIO generates a Master Clear (**MC**), active low, to initialize the DAC and the LBC upon startup. It also switches the data lines (D0-D7) between Tri-State to active when the DIO is in RUN. Since it's critical for the UUT not to have conflicts on the data bus, a low on the **OutEn** line will signal to the LBC that the data bus is now occupied by the DIO.

When the DIO is ready to run, the **BusReq** is set to low and the DIO is ARMed. When the LBC is able to grant access to the DIO, it responds by lowering **BusAv**, which in return triggers the DIO to run state.

Depending on the selected waveform (**Wvfm**), the DIO generates one of two signals: a saw or a triangular wave.

The UUT can terminate the program at any time by lowering the **Stop** line.

UUT Flow of Events

The DAC's linear range is a ±10.0 volt. Our objective is to generate either of two waveforms in accordance with the LBC waveform line (**Wvfm**), a continuos saw or a triangular wave between 0 and 5 V.

The DAC is initialized before the main test program starts. A Master Clear (**MC**) signal resets the DAC and the LBC.

When the system wants the DAC to generate a waveform, it issues an address that is interpreted by the LBC as a request for the indicated value. When the DIO is ARMed, after initializing the DAC and LBC, the **BusReq** is set to low and the DIO is now in PAUSE state. The LBC has control of the bus; it responds by lowering the **BusAv** line which triggers the DIO. The DIO then changes to the RUN state. While in RUN state the 8-bits of data are enabled. At that point the **OutEn** line will be set to low, signaling to the LBC that the data bus is now occupied by the DIO. That line will stay low as long as the DIO is in the RUN state.

When the **Wvfm** is low then a saw waveform is selected. The DIO will produce a series of steps starting from 0 to 127, after each new value a **Clk** is issued to strobe the voltage level into the DAC's input register. Waveforms will be output continuously until a **Stop** signal is received from the UUT.

When the **Wvfm** is high then a triangle waveform is selected. The DIO starts to output a sequence of values stating from 0 V to 5 V and back to 0 V. For each waveform value the DIO issues a **Clk** to strobe the new value into the DAC's input register. Waveforms will be output continuously until a **Stop** signal is received from the UUT.

Waveform selection can be change dynamically by setting the waveform line to either low or high at any point.

The UUT can stop the DIO by setting the **Stop** to low. The DIO immediately will be set to the HALT state.

Implementing the example

This section will show step by step of how to create and edit the vector file using ***DIOEasy*** in order to implement the example.

The vector has two sections: the first is initializing the system and the second is Generating Waveforms. In our example, the DIO will change states as listed below:

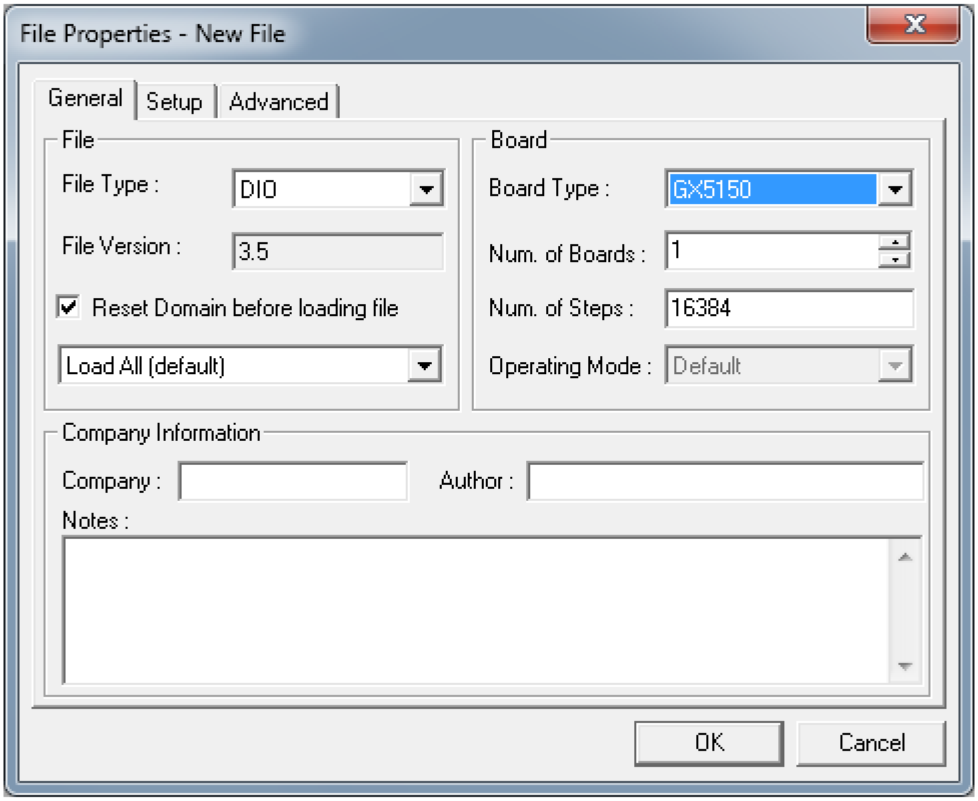
PAUSE The program starts, executes the first two steps, then DIO goes to the PAUSE state with output data lines disabled.

RUN In RUN state, the output data lines are enabled.

HALT At the end of the program, the DIO is set to HALT state and output lines are set to Tri-State.

Creating the Example Vector File

- Open ***DIOEasy***.
- Click on the File menu and then click on New. Dialog with the “Create a New File” caption appears. In the “Board Type” group selects GX5150 and unchecks the “Show File Property”. Make sure the fields in Figure 6-2 are entered and click OK.



File
File Type : DIO
File Version : 3.5
☑ Reset Domain before loading file
Load All (default)

Board
Board Type : GX5150
Num. of Boards : 1
Num. of Steps : 16384
Operating Mode : Default

Company Information
Company :
Author :
Notes :

Figure 6-2: Creating a New DIO File

- A new vector file will fill ***DIOEasy*** vector view with a default name of *Dio1* in the caption.
- Click File menu and **Save As…** then save the file as WaveformsExample.dio.

Assigning Vector Channels

We first need to assign the channels in the new vector that implement the example block diagram (see "Chapter 4 - Using ***DIOEasy***" on how to rename).

D0-D7: DAC 8 Bit Data Lines

We assign the first eight channels to D0-D7. In order to do that, rename CH0 to D0, CH1 to D1 and so on, up to D7.

Clk: DAC Clock

The **Clk** will be channel number 8, rename CH8 to **Clk**.

MC: Master Clear DAC and Bus Controller

The **MC** will be channel 9, rename CH9 to **MC**.

BusReq: Is DIO Ready to Output the Next DAC Value?

The **BusReq** will be channel number 10; rename CH10 to **BusReq**.

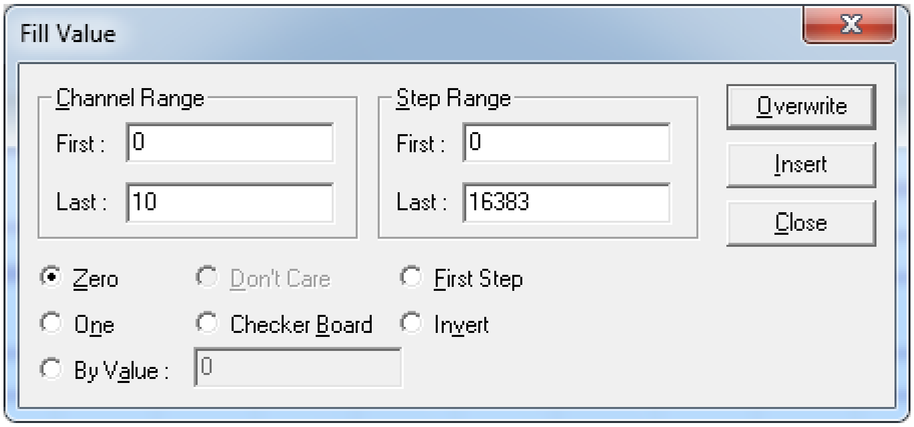
Editing the vector file

This section shows how to edit the example vector file in ***DIOEasy***. Editing the vector includes the following:

- Assigning labels to specific steps.
- Inserting commands and conditions to control the vector flow of events.
- Edit individual step data in order to generate the correct data output for each step.

Set All Channels Values to Zero

To ensure that the channels 0 through 10 have a value of zero, click the **Fill** menu and then **Value...** The Fill Value dialog box displays. Enter dialog values in Figure 6-3 and click the **Overwrite!** button. Channels 0 through 10 have the value set to zero.



Channel Range
First : 0
Last : 10

Step Range
First : 0
Last : 16383

● Zero ○ Don't Care ○ First Step
○ One ○ Checker Board ○ Invert
○ By Value : 0

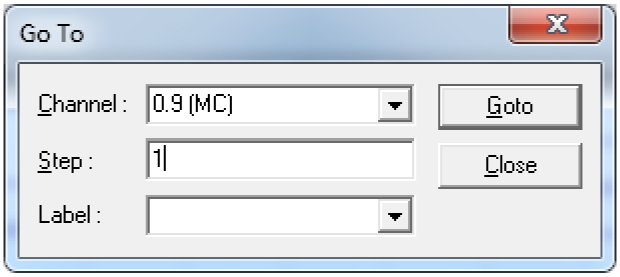
Figure 6-3: Using the Fill Value Dialog to Initialize Data

Initialize the system

To set the system to its default state we first need to reset both the 8-bit DAC and the Local Bus Controller. Channel 9 was assigned to control the reset portion of the vector (see "Chapter 6 - Using ***DIOEasy***" on how to select steps and add commands).

Select step 0 and double-click the **Command Property** area. The Command Property dialog appears. In the label edit-box type “Start” and press **Enter**.

Click **Edit** and **Go To…** The Go To dialog appears. Set the fields as displayed in Figure 6-4.



Go To

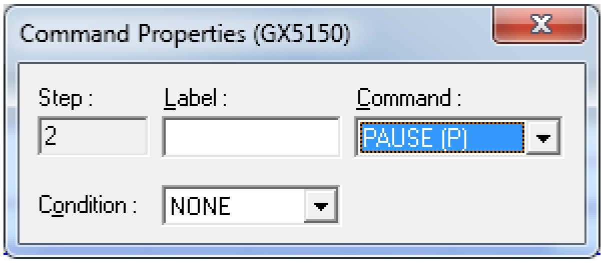
Channel : 0.9 (MC)

Step : 1

Label :

Figure 6-4: Using the Go To Dialog to Change Channels

Click the **GoTo** button. The Go To dialog box is closed. Step 1 of Channel 9 (named MC) is selected in Vector View. In the tool bar click the button. After the initalization phase, we need to wait until the UUT triggers the DIO. To do that we will need to add a PAUSE command in the next step. Double-click the **Command Property** area under step 2. The Command Property dialog (Figure 6-5) appears.



Command Properties (GX5150)

Step : 2

Label :

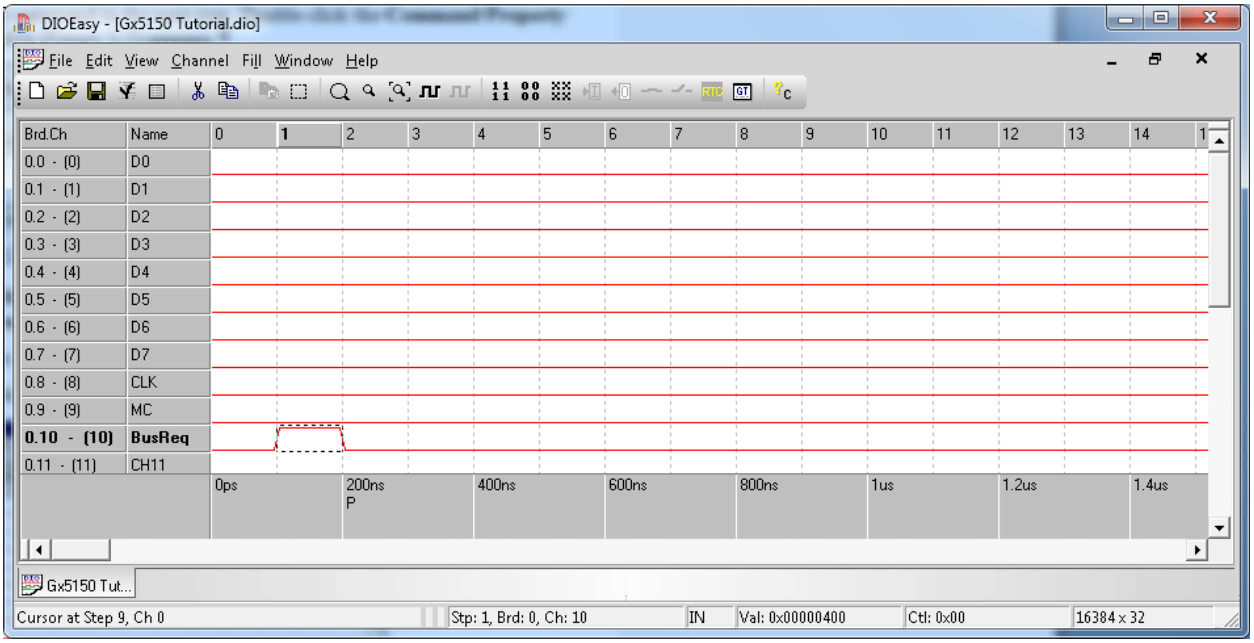
Command : PAUSE (P)

Condition : NONE

Figure 6-5: Using Command Properties Dialog to Add a Pause

Apply the displayed settings, then press **Enter**, **Esc**.

The resulting Vector View display should look like Figure 6-6.



DIOEasy - [Gx5150 Tutorial.dio]

Brd.Ch	Name
0.0 - (0)	D0
0.1 - (1)	D1
0.2 - (2)	D2
0.3 - (3)	D3
0.4 - (4)	D4
0.5 - (5)	D5
0.6 - (6)	D6
0.7 - (7)	D7
0.8 - (8)	CLK
0.9 - (9)	MC
0.10 - (10)	**BusReq**
0.11 - (11)	CH11

Cursor at Step 9, Ch 0 | Stp: 1, Brd: 0, Ch: 10 | IN | Val: 0x00000400 | Ctl: 0x00 | 16384 x 32

Figure 6-6: Vector View Display of Current Vector Build

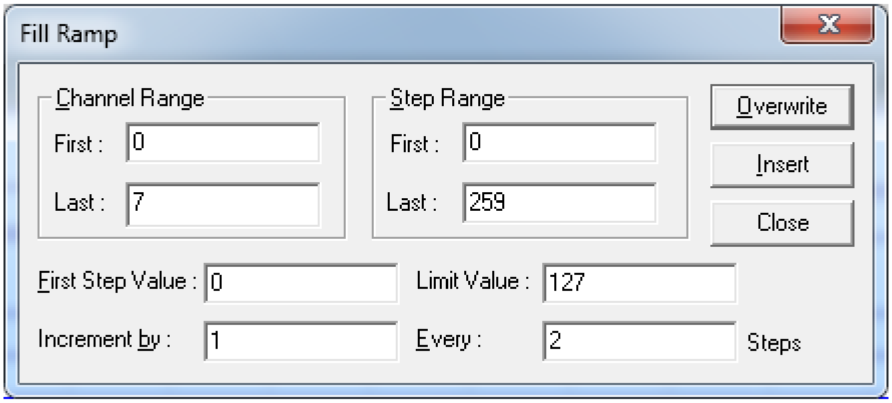
Generating Waveform Data

This section demonstrates step by step how to edit the vector file in order to create waveforms and commands that control waveforms in the RUN mode.

The UUT can select two waveforms. The UUT should be able to switch between the two waveforms every time the DIO completes a waveform. We will first set the data for the triangular waveform.

Click the **Fill** menu, and then **Ramp**. The Fill Ramp dialog box now displays. Each ramp value needs to be on the output lines before we can clock it in. The clock alternates every step. The ramp increments on the positive-going clock transition, incrementing every second step. Each value must remain stationary for one clock interval. While the clock is active (high) in the second step, the next ramp value is set up.

The last range step value is now calculated. The total number of DAC values is 128 (0 to 127). Since each value needs two steps and there is a 4-step offset, the last step value is 128*2+4 -1=259. Set the Fill Ramp dialog values that display in Figure 6-7. Click **Overwrite!** to enter the steps.



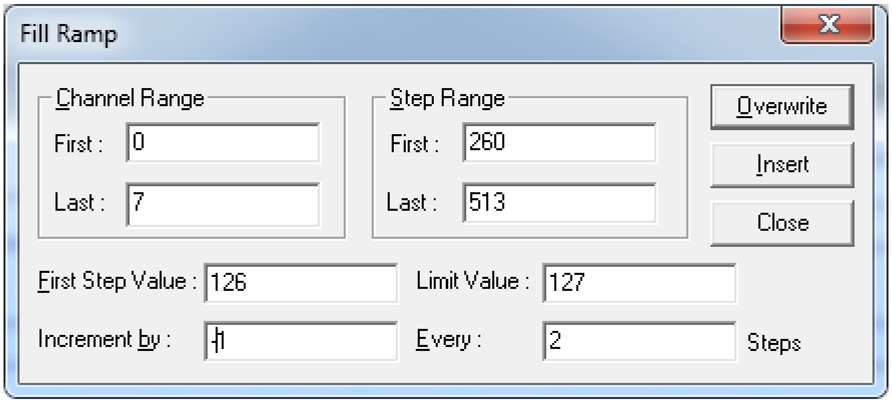
Channel Range
First : 0
Last : 7

Step Range
First : 0
Last : 259

First Step Value : 0
Limit Value : 127
Increment by : 1
Every : 2 Steps

Figure 6-7: Using Fill Ramp to Generate the Waveform

To complete the triangular waveform, a negative ramp is needed. Again, click the **Fill** menu and select **Ramp** to display the Fill Ramp dialog. To create a negative ramp, set the First Step Value to 126 (0x7E), so that peak value will appear only once in vector steps 258-259. There are a total of 127 DAC values, 0 to 126, and 127*2=254 steps. Set First Step Value to 126 and set "Increment by" to –1. This sets up a down-sloping ramp. Set the other values as displayed in Figure 6-8.



Channel Range
First : 0
Last : 7

Step Range
First : 260
Last : 513

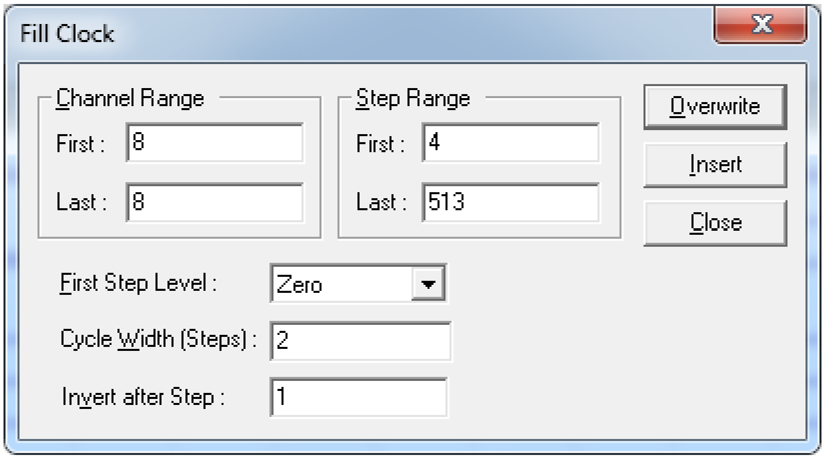
First Step Value : 126
Limit Value : 127
Increment by : -1
Every : 2 Steps

Figure 6-8: Using Fill Ramp to Generate Waveform Values

Generating the DAC clock

To clock data into the DAC, we need to fill the **Clk** channel with appropriate clock values.

Click the **Fill** menu, then **Ramp**. The Fill Clock dialog box displays. In our case, DAC values change every second step. In the first step, the data is placed on the bus. In the second step, the data is clocked to the DAC. The total number of steps to fill was calculated earlier. Set the values as displayed in Figure 6-9:



Channel Range
First : 8
Last : 8

Step Range
First : 4
Last : 513

First Step Level : Zero

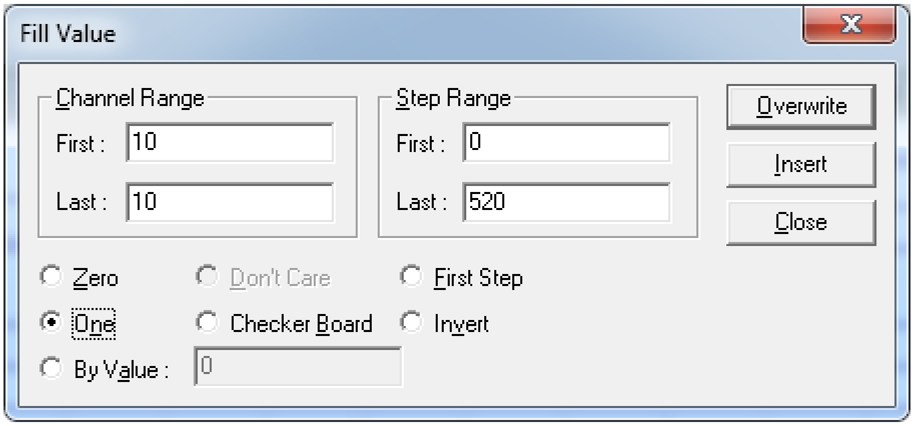
Cycle Width (Steps) : 2

Invert after Step : 1

Figure 6-9: Using Fill Clock to Generate the Clock

Bus Request Line (BusReq)

BusReq is usually set to inactive (high). It is active low, but we need to first set it high to assure we output in a known state. Click the **Fill** menu and select **Fill Value**. Set parameters as displayed in Figure 6-10, then click **Overwrite!**



Channel Range
First : 10
Last : 10

Step Range
First : 0
Last : 520

○ Zero ○ Don't Care ○ First Step

● One ○ Checker Board ○ Invert

○ By Value : 0

Figure 6-10: Using Fill Value to Set the BusReq Line

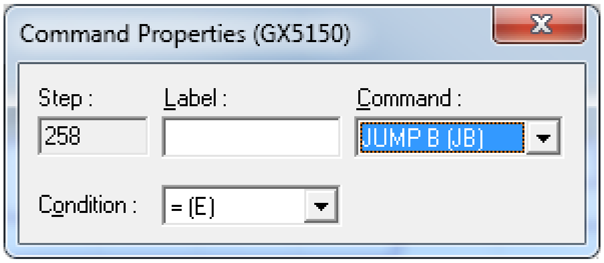
In Vector View select Step 2, Channel 10. In the Tool Bar click 00 to set that cell to zero.

Controlling Waveform Generation in the RUN State

To continuously output waveforms in the RUN state, Registers A and B are used to control operations. A Jump command is added at the last step of each waveform.

For the first waveform, the jump is conditional. We want to jump to the beginning of the waveform at step 4. The address is loaded in Register B. The jump will be to the address in Register B on condition that External Event line 0 matches bit 0 of Register D. For the second waveform, the jump is unconditional because it's the end of both waveforms. We jump to the predefined address in Register B, Step 4, at the beginning of the waveform.

Double-click the **Command Property** area under Step 258. The "Command Property dialog appears. Apply the settings displayed in Figure 6-11 and press **Enter**.



Command Properties (GX5150)

Step : 258

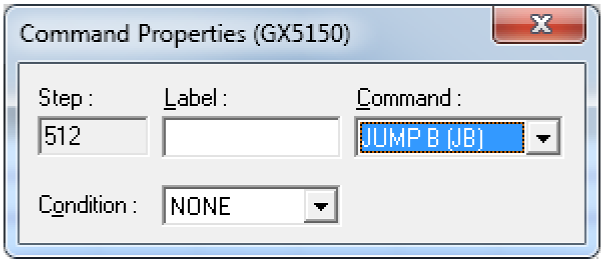
Label :

Command : JUMP B (JB)

Condition : = (E)

Figure 6-11: Using Command Properties to Set a Conditional Jump

The second jump is unconditional and occurs at Step 512 in the last step of the second waveform.



Command Properties (GX5150)

Step : 512

Label :

Command : JUMP B (JB)

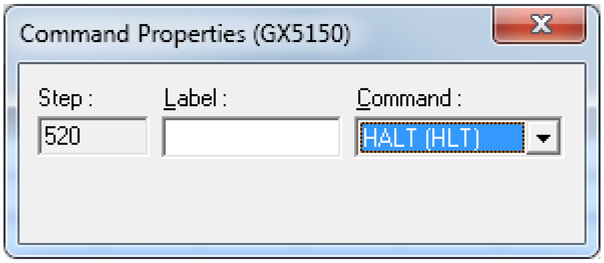
Condition : NONE

Figure 6-12: Using Command Properties to Set an Unconditional Jump

To stop the program, program an external unconditional jump on Register A. Use the user I/O Control located on J9, Pin 10. A low on this pin causes an immediate jump to the address loaded in Register A.

Registers A and B can be set in the Registers group located in the File Properties dialog box, under the **Setup** tab. To open the File Properties dialog box, click **File** then **Properties…** If the DIO file is saved after setting properties, the next time the vector is loaded the registers will preload with the saved values.

In order to insure proper program termination we will insert to step 520 HALT command, as shown in Figure 6-13.



Command Properties (GX5150)

Step : 520

Label :

Command : HALT (HLT)

Figure 6-13: Using Command Properties to Insert HALT Command

Set Vector Properties

The **File Properties** window contains all DIO board presets. Settings are loaded into DIO boards with the vector file.

On the Tool Bar, click the ▤ icon. The File Properties dialog box appears. Click the **Setup** tab.

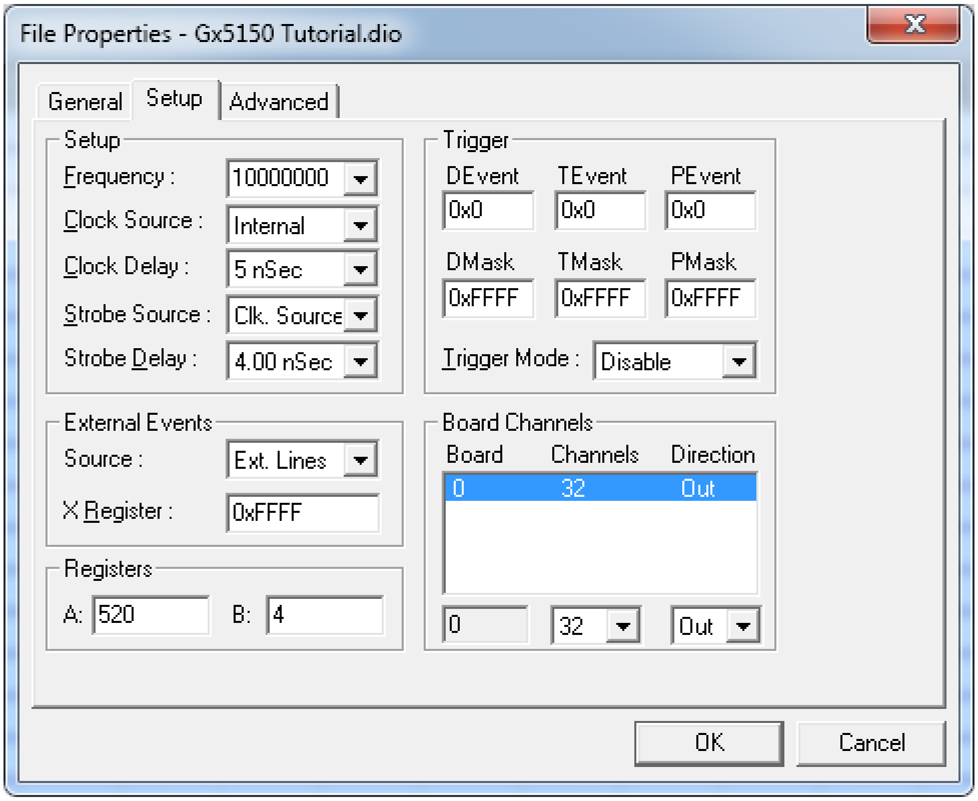
Set DEvent register to 0. This register value is compared to the jump condition while running. If the UUT selects the first waveform, the **WvFm** line is low. When this is compared to the D register, a jump back is executed to the beginning of the waveform.

The target of the conditional/unconditional jump in Register B is step 4. To set the Register B content for the jump to step 4, type *4* in the Register B text box in the Registers group.

Set the Register A content to 520 the same way to implement an unconditional jump to the last step that ends the program. Step 520 executes the HALT command to ensure proper program termination.

Set the vector direction to output by selecting Output in the Direction group.

The File Properties window now looks like Figure 6-14.



File Properties - Gx5150 Tutorial.dio

Setup

Frequency : 10000000

Clock Source : Internal

Clock Delay : 5 nSec

Strobe Source : Clk. Source

Strobe Delay : 4.00 nSec

Trigger

DEvent: 0x0 TEvent: 0x0 PEvent: 0x0

DMask: 0xFFFF TMask: 0xFFFF PMask: 0xFFFF

Trigger Mode : Disable

External Events

Source : Ext. Lines

X Register : 0xFFFF

Registers

A: 520 B: 4

Board Channels

Board	Channels	Direction
0	32	Out

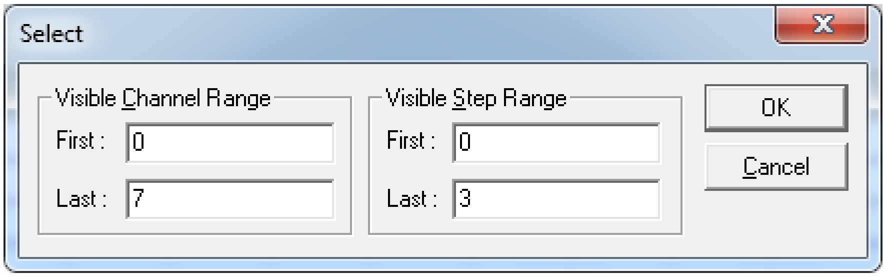
0 32 Out

Figure 6-14: File Properties Window Displaying Setup Values

Click **OK** to apply changes and close the dialog box. Graticle lines now display as blue in Vector View, indicating the board outputs.

Enable/Disable Output Data Lines

Output data lines are enabled only when the DIO board is in the RUN state. Other times, those lines are in the Tri-State high impedance state. Therefore, steps 0 to 3 and 514 to 520 need to be disabled. From the **Edit** menu click **Select...** The Select dialog box (Figure 6-15) displays. Set the fields as displayed and click **OK**.



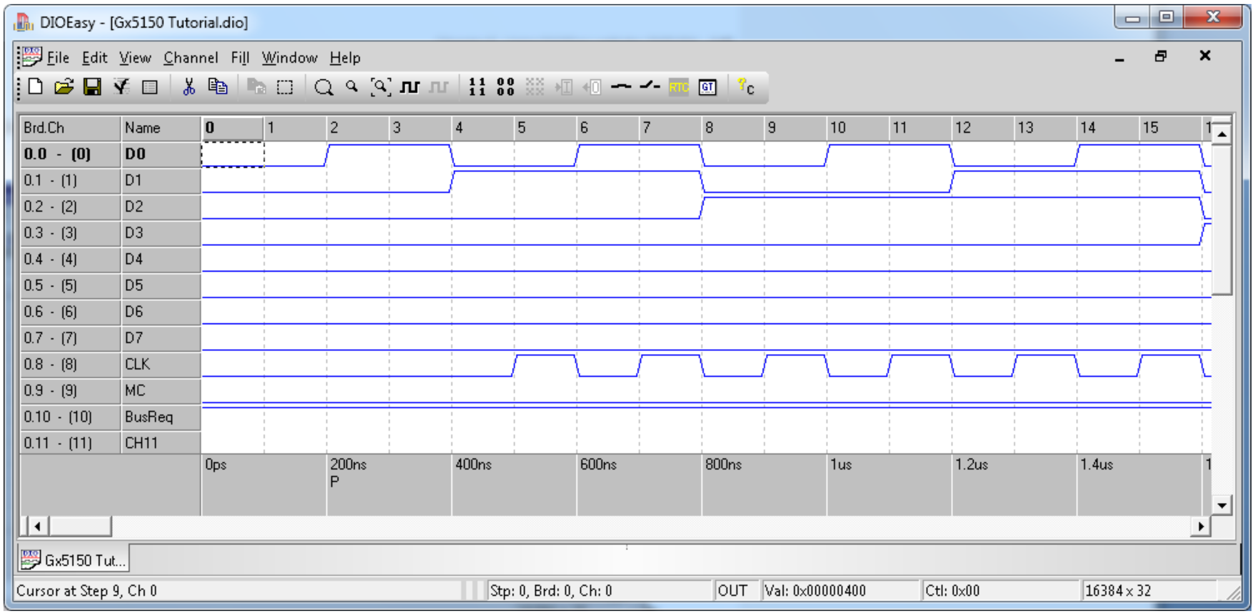
Select

Visible Channel Range
First : 0
Last : 7

Visible Step Range
First : 0
Last : 3

Figure 6-15: Selecting a Range of Channels and Steps

Channels 0 to 7 and Steps 0 to 3 are disabled by clicking in the **Tool Bar**. The selected area displays in Vector View, Figure 6-16.



DIOEasy - [Gx5150 Tutorial.dio]

Brd.Ch	Name
0.0 - (0)	D0
0.1 - (1)	D1
0.2 - (2)	D2
0.3 - (3)	D3
0.4 - (4)	D4
0.5 - (5)	D5
0.6 - (6)	D6
0.7 - (7)	D7
0.8 - (8)	CLK
0.9 - (9)	MC
0.10 - (10)	BusReq
0.11 - (11)	CH11

Cursor at Step 9, Ch 0 | Stp: 0, Brd: 0, Ch: 0 | OUT | Val: 0x00000400 | Ctl: 0x00 | 16384 x 32

Figure 6-16: Disabled Range of Channels and Steps

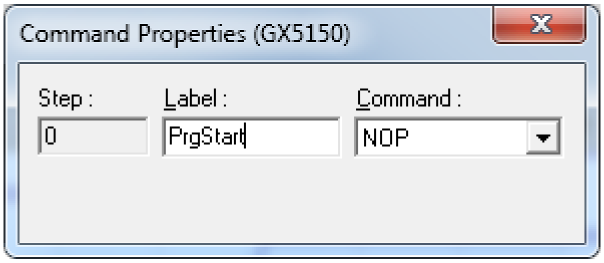
Disable steps 514 through 520 at Channels 0 to 7. There is no need to specifically enable Steps 4 through 513 because all channels are enabled by default.

Adding Labels

Labels at key steps make it easier to follow vector sequences and do troubleshooting. Labels are added to key steps as follows:

1. Label the waveform starting point, Step 4 “Wvfrm Start”.
2. Label the end of first waveform, Step 259 “End Wvfrm1”.
3. Label the end of second waveform, Step 513 “End Wvfrm2”.
4. Label the program end, Step 520 “Prg End”.

Double-click the **Command Property** area under Step 0. The Command Properties dialog appears, as shown in Figure 6-17.



Step : 0
Label : PrgStart
Command : NOP

Figure 6-17: Using Command Properties to Add a Label

“PrgStart” in the Label: text box, then press **Enter**. Add the other labels to the vector.

UUT / DIO Interface Signals

The following table summarizes various UUT interface signals.

Signal	**I/O Dir.**	**DIO Module**	**Pin & Mnemonic**	**Comments**
BusAv	In	Timing	J1-26 EXTrig	Generated by the LBC, triggers the DIO into RUN mode.
Stop	In	I/O Control	J9-10	Terminates the program. Issued by the system
Wvfm	In	Timing	J1-3 Ext2	Selects the waveform: DC or triangular
BusReq	Out	I/O Data	J4-9	DIO is ready to output the next DAC value.
Clk	Out	I/O Data	J4-10	DAC clock
D0 – D7	Out	I/O Data	J4-1 – J4-8	8 bit data to the DAC
MC	Out	I/O Data	J4-10	Master Clear for the DAC and Bus Controller.
OutEn	Out	I/O Control	J9-11	Signals the state of channels 0-7. Low = Enable, High = Tri-State, high Z.

Chapter 7 - Programming the Board

This chapter contains information about how to program the switching instruments using the GTDIO driver. The GTDIO 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. The **Programmer's Reference** User's Guide contains a complete and detailed description of the available programming functions.

The driver supports many development tools. Using these tools with the driver is described in this chapter. In addition, the GTDIO directory contains examples written for these development tools.

An example using the DLL driver with Microsoft Visual C++ 6.0 and 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 GTDIO driver types.

The GTDIO Driver

The GTDIO driver is a 32-bit/64-bit Windows DLL file: GTDIO32.DLL for 32 bit applications or GTDIO64.DLL for 64 bit applications. The DLL uses a device driver to access the board resources. The device driver HW.SYS/HW64.SYS is installed by the setup program and is shared by other Marvin Test Solutions products (e.g. ATEasy).

The DLL can be used with various development tools such as Microsoft Visual C++, Borland C++ Builder, Microsoft Visual Basic, Borland Pascal or Delphi, ATEasy, LabView, LabWindows/Cvi 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 GTDIO driver with C/C++ development tools:

- Include the GTDIO.H header file in the C/C++ source file that uses the GTDIO 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 GTDIO32.LIB/GTDIO64.LIB for Microsoft Visual C++ (Also LabWindows/CVI) Use the GTDIO32BC.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 GTDIO 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 GTDIO32.BAS to the project. For Visual Basic .NET use the GTDIO32.VB.

The file can be loaded using *Add File* from the Visual Basic *File menu*. The GTDIO32.BAS/ GTDIO32.VB contain function declarations for the DLL driver.

Programming Using Pascal/Delphi

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

Programming DIO Boards Using ATEasy®

The GTDIO package is supplied with a separate ATEasy driver for each of board types. The ATEasy driver uses the GTDIO32.DLL to program the board. In addition, each driver is supplied with an example that contains a program and a system file pre-configured with the ATEasy driver. Use the driver shortcut property page from the System Drivers sub-module to change the PCI slot number before attempting to run the example.

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

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

- The *nHandle* parameter is omitted. The driver handles this parameter automatically. ATEasy uses driver logical names instead i.e. DIO1, DIO2 for GTDIO.
- The *nStatus* parameter was omitted. Use the Get Status commands instead of checking the status. After calling a DLL function the ATEasy driver will check the returned status and will call the error statement (in case of an error status) to generate exception that can be easily trapped by the application using the **OnError** module event or using the **try-catch** statement.

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

Programming Using LabView and LabView/Real Time

To use the driver with LabView use the provided LabView library GTDIO.llb. The library is located in the GTDIO folder. An example for LabView is also provided in the Examples folder. Check the ReadMe.txt file if your GTDIO board is supported under LabView/Real-Time.

Using and Programming under Linux

Marvin Test Solutions provides a separate package with a Linux driver (Marvin Test Solutions Driver Pack for Linux). The software package can be download from the Marvin Test Solutions website. See the ReadMe.txt in that package if your board is currently supported under Linux and for more information regarding using and programming the driver under Linux.

Using the GTDIO driver functions

The GTDIO driver contains a set of functions for all the supported DIO boards. The GTDIO functions are designed with consistent set of arguments and functionality. All boards have a function that initializes the GTDIO driver for a specific board, reset the board, and display the virtual panel. All the functions use handles to identify and reference a specific board and all functions return status and share the same functions to handle error codes.

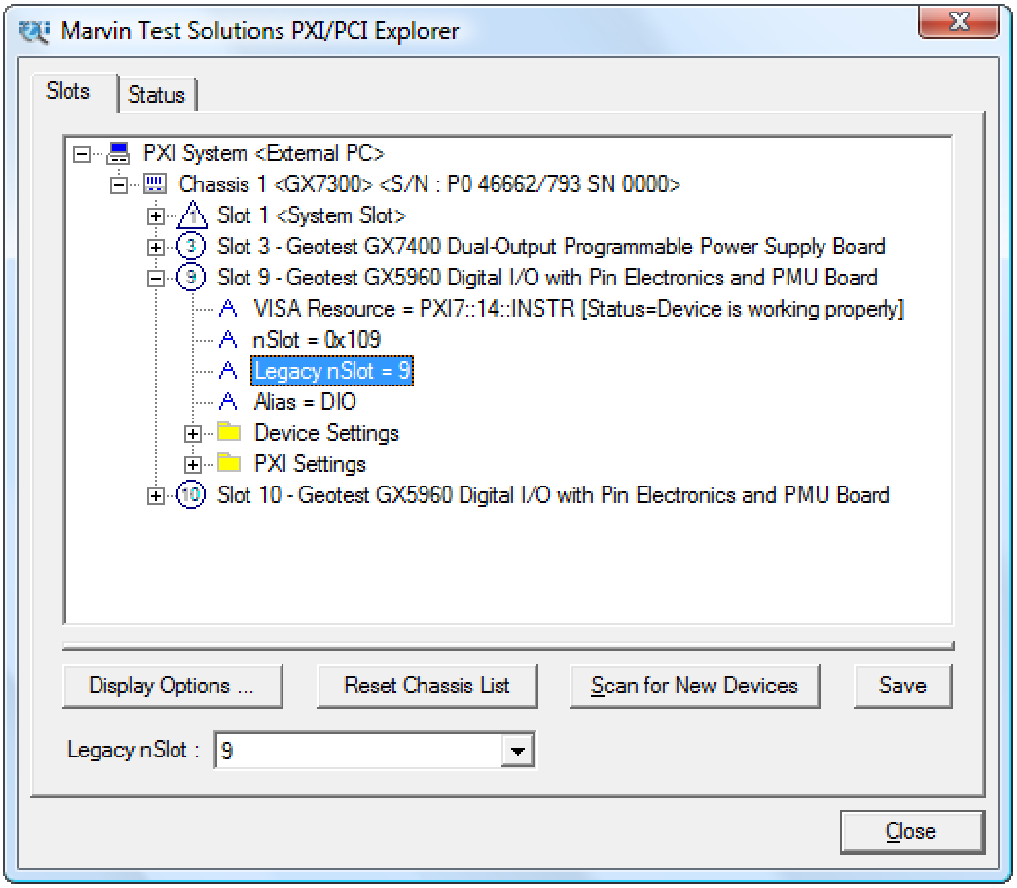
Initialization, HW Slot Numbers and VISA Resource

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

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

    - A combination of chassis number (chassis # x 256) with the chassis slot number, e.g. 0x105 for chassis 1 and slot 5. Chassis number can be set by the **PXI/PCI Explorer** applet.

    - Legacy nSlot as used by earlier versions of HW/VISA. The slot number contains no chassis number and can be changed using the **PXI/PCI Explorer** applet: 23 in this example.



PXI System <External PC>
Chassis 1 <GX7300> <S/N : P0 46662/793 SN 0000>
Slot 1 <System Slot>
Slot 3 - Geotest GX7400 Dual-Output Programmable Power Supply Board
Slot 9 - Geotest GX5960 Digital I/O with Pin Electronics and PMU Board
VISA Resource = PXI7::14::INSTR [Status=Device is working properly]
nSlot = 0x109
Legacy nSlot = 9
Alias = DIO
Device Settings
PXI Settings
Slot 10 - Geotest GX5960 Digital I/O with Pin Electronics and PMU Board

Legacy nSlot : 9

Figure 7-1: PXI/PCI Explorer

2. **VISA** – this is a third party library usually by National Instruments (NI-VISA). You must ensure that the VISA installed supports PXI and PCI devices (not all VISA providers supports PXI/PCI). GTDIO setup installs a VISA compatible driver for the DIO board in-order to be recognized by the VISA provider. Use the **DioSetupInitializationVisa**(*szVisaResource*, *pnHandle*, *pnStatus*) function to initialize the driver board using VISA. The first argument *szVisaResource* is a string that is displayed by the VISA resource manager such as NI **Measurement and Automation** (NI_MAX). It is also displayed by Marvin Test Solutions **PXI/PCI Explorer** as shown in the prior figure. The VISA resource string can be specified in several ways as the following examples:

    - Using chassis, slot: "PXI0::CHASSIS1::SLOT5"
    - Using the PCI Bus/Device combination: "PXI9::13::INSTR" (bus 9, device 9).
    - Using alias: "COUNTER1". Use the PXI/PCI Explorer to set the device alias.

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

The **DioSetupInitialization** function returns a handle that is required 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 settings.

Board Handle

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

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

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

Reset

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

Error Handling

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

Driver Version

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

Panel

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

The **DioPanel** function is also used by the DioPanel.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 (GTDIO32.DLL or GTDIO64.DLL and the HW device driver files located in the HW folder) can be shipped with the application. Typically, the GTDIO32.DLL should be copied to the Windows System directory (System32). The HW device driver files should be installed using a special setup program HWSETUP.EXE that is provided with GTDIO driver files. Alternatively, you can provide the GTDIO.EXE setup program to be installed along with the board. The setup program can be invoked in silent mode (with /s) to install the driver using the command line specified settings with no user interface. Marvin Test Solutions provides permission to re-distribute the driver software providing it is in support of an application using the target hardware.

Sample Programs

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

The example bellow works with several DIO boards. Check the Examples folder for additional examples for other boards.

To run, enter the following command line:

GtDioExampleC <*Slot*> <function> <`value` >

Where:

<Slot> PXI Explorer slot number where the board resides.

Sample Program Listing

```
/****************************************************************************
FILE        : GtDioExampleC.cpp
PURPOSE     : WIN32 sample program for GX2470 board using the GtDio driver.
CREATED     : Feb. 2008
COPYRIGHT   : Copyright 2008 Marvin Test Solutions, Inc.
COMMENTS    :
To compile the WIN32 example:
      1. Microsoft VC++
            Load GtDioExampleC.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 GtDioExampleC.bpr from the Project/Open
                  Project... menu
            Select Project/Build all from the menu

****************************************************************************/
#include "windows.h"
#include "GtDio.h"
#include <time.h>
#include "stdio.h"

// Borland C++ Builder compat. block
#if defined(__BORLANDC__)
```

```
#pragma hdrstop
#include <condefs.h>
USELIB("GTDIO32B.lib");
USERC("GtDioExampleC.rc");
//---------------------------------------------------------
#endif // defined(__BORLANDC__)

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

//*********************************************************
//          DisplayUsage
//*********************************************************
void DisplayUsage(void)
{
DisplayMsg(
"This example shows the following:\r\n"
"\tInitialize a DIO board using the commands line parameters.\r\n"
"\tcreating a DIO file and fill it with a ramp patern.\r\n"
"\tFill the file control data with NOPs values and HALT command at step
1020\r\n"
"\tSet the file D, T and P events & mask registers.\r\n"
"\tSet the frequency to 2MHz and close and save the file.\r\n"
"\tLoad the DIO file, Arm, Trig Halt and save the result vector to another
DIO file (.DI)\r\n\r\n"

"Usage:\r\n"
"GtDioExampleC <slot>"

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

"\r\nExample:\r\n"
"GtDioExampleC 0x106\r\n"
"\r\nTo change command line under Windows Select File/Properties from
Program\r\n"
"Manager Menu and change the command line edit box as shown above."
);
exit(1);
}

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

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

//*********************************************************
//    MAIN
// This main function receives one parameters - slot number
//
//*********************************************************
int main(int argc, char **argv)
{
short nSlotNum;         // Board slot number
short nStatus;          // Returned status
short nHandle;          // DIO Master handle
short nDensity;         // Board density
short nBanks;           // Board density
clock_t ctStartTime;    // For timeout
WORD  wData;            // Status register value
short hFile;
DWORD dw, dwSize;
DWORD dwData[1024];     // data array
DWORD dwCtrl[1024];     // control array
BYTE  ucCtrl[1024];     // control array(GX5150)
short nType;            // board type
char  szFileNmae[128];

// Check number of arguments received
if (argc<3) DisplayUsage();

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

if (nSlotNum<0)
      DisplayUsage();
// Initialize Master DIO board
DioSetupInitialization(0, 1, nSlotNum, &nDensity, &nBanks, &nHandle,
&nStatus);
CheckStatus(nStatus);

DioGetBoardType (nHandle, &nType, &nStatus);
//creat DIO file
switch(nType)
{case DIO_BOARD_TYPE_GC5050:
      strcpy (szFileNmae, "GC5050.DIO");
      break;
case DIO_BOARD_TYPE_GX5050:
      strcpy (szFileNmae, "GX5050.DIO");
      break;
case DIO_BOARD_TYPE_GX5150:
      strcpy (szFileNmae, "GX5150.DIO");
      break;
case DIO_BOARD_TYPE_GX5280:
      strcpy (szFileNmae, "GX5280.DIO");
      break;
case DIO_BOARD_TYPE_GX5290:
      strcpy (szFileNmae, "GX5290.DIO");
      break;
```

```
default:
      MessageBox(0, "The board not supported", "GtDioExample", MB_OK);
      return 0;
}

// Create new file
DioFileOpen (szFileNmae, DIO_FILE_CREATE, &nType, &hFile, &nStatus);
CheckStatus(nStatus);

// Set number of steps to 1024
DioFileSetNumberOfSteps(hFile, 1024, &nStatus);
CheckStatus(nStatus);

// Create a ramp pattern and write it to the ouput data
for (dw=0; dw<1024; dw++)
      dwData[dw]=dw;
DioWriteOutMemory (hFile, dwData, 0, 1024, &nStatus);
CheckStatus(nStatus);

if (nType==DIO_BOARD_TYPE_GC5050 || nType==DIO_BOARD_TYPE_GX5050)
{
// Zero the control array (all groups are set to output)
for (dw=0; dw<1024; dw++)
      dwCtrl[dw]=0x0;

// Add commands to the control memory
dwCtrl[2]=0x01000064; // Set regiaster A to 100
dwCtrl[9]=0x00C00005; // Loop steps 5 to 9 for 100  (register A value)
      dwCtrl[1020]=0x01CE0000; // HALT to step 1020
            // Write Ctrl Memory
      DioWriteCtrlMemory (hFile, dwCtrl, 0, 1024, &nStatus);
      CheckStatus(nStatus);
}
else if (nType==DIO_BOARD_TYPE_GX5150)
{     // control array ( all outputs are enabled)
      for (dw=0; dw<1024; dw++)
            ucCtrl[dw]=0xF0;
            // Add HALT command to the control memory
      ucCtrl[1020]=0xFF; // HALT to step 1020
      // Write Ctrl Memory
      DioWriteCtrlMemory (hFile, ucCtrl, 1020, 1, &nStatus);
      CheckStatus(nStatus);
}
else if (nType==DIO_BOARD_TYPE_GX5280)
{     // Set channesl directions as follows:
      // group 0 (channels 0-7) to out
      // group 1 (channels 8-15) to in
      // group 2 (channels 16-23) to out
      // group 3 (channels 24-31) to in
      DioSetupIOConfiguration(hFile, 0, 0x5, &nStatus);
      // Insert HALT command at step 1020
      DioWriteCtrlCommand(hFile, 1020, DIO_COMMAND_HALT, 0, 0, 0, 0,
&nStatus);

}
else if (nType==DIO_BOARD_TYPE_GX5290)
{
```

```
// Fill an array to set the direction memory as follows:
// Set all odd channels numbers (1, 3 etc.) to be set to input on even steps
and output on odd steps numbers.
// Set all even channels numbers (0, 2 etc.) to be set to output on odd steps
and output on even steps numbers.
for (dw=0; dw<1024; dw++)
      dwCtrl[dw]= dw & 1? 0xAAAAAAAA: 0x55555555;
// Write to the direction Memory starting at step 0, 25000 steps
DioWriteDirectionMemory(hFile, dwCtrl, 0, 1024, &nStatus);
CheckStatus(nStatus);
      // Insert HALT command at step 1020
      DioWriteCtrlCommand(hFile, 1020, DIO_COMMAND_HALT, 0, 0, 0, 0,
&nStatus);

}

// Set D Events and Mask
DioSetupTriggerDEvent (hFile, 0x1234, 0xFF55, &nStatus);
CheckStatus(nStatus);

// Set T Events and Mask
DioSetupTriggerTEvent (hFile, 0xAAAA, 0xFFFF, &nStatus);
CheckStatus(nStatus);

// Set P Events and Mask
DioSetupTriggerPEvent (hFile, 0xABCD, 0xFFFF, &nStatus);
CheckStatus(nStatus);
// Set frequency to 2MHz
DioSetupFrequency (hFile, 2000000, &nStatus);
DioFileClose (hFile, &nStatus);

// Load data to the DIO board
DioLoadFile(nHandle, szFileNmae, 0, 0, &dwSize, &nStatus);
CheckStatus(nStatus);
// Arm the DIO card
DioArm(nHandle, &nStatus);
CheckStatus(nStatus);
// Trigger the DIO card
DioTrig(nHandle, &nStatus);
CheckStatus(nStatus);
// Wait up to 1000ms or until Halt state to complete
ctStartTime=clock();
do
{     DioReadStatusRegister(nHandle, &wData, &nStatus);}
while ((wData & 0x1C)!=0 && (clock()- ctStartTime)*CLOCKS_PER_SEC<1000000);

// Halt the DIO card
DioHalt(nHandle, &nStatus);
CheckStatus(nStatus);
// Read data from DIO card and save it into a file
switch(nType)
{     case DIO_BOARD_TYPE_GC5050:
            strcpy (szFileNmae, "GC5050.DI");
            break;
      case DIO_BOARD_TYPE_GX5050:
            strcpy (szFileNmae, "GX5050.DI");
            break;
      case DIO_BOARD_TYPE_GX5150:
            strcpy (szFileNmae, "GX5150.DI");
            break;
      case DIO_BOARD_TYPE_GX5280:
            strcpy (szFileNmae, "GX5280.DI");
            break;
      case DIO_BOARD_TYPE_GX5290:
            strcpy (szFileNmae, "GX5290.DI");
            break;
}
DioSaveFile(nHandle, szFileNmae, 0, &dwSize, &nStatus);
CheckStatus(nStatus);
MessageBox(0, "DIO example program was completed successfully",
"GtDioExample", MB_OK);
return 0;
}
//**********************************************************
//          End Of File
//**********************************************************
```

Index

A

About DIO
- Help Menu 45

Adding Commands and Labels
- Dialog 80

Arm!
- GTDIO Panel 56

ATEasy 97, 98

ATP 22, 51

ATP File 51

B

B Clock
- GTDIO Panel 64

Block
- Selecting 7

Board Handle 100

Borland 97, 98

Borland-Delphi 98

By Value
- Fill Value Dialog Box 39

C

C/C++ 97

C++ 97

Cancel
- Fill Value Dialog Box 38

Cascade
- Window Menu 45

Channel Menu 33
- Hide 36
- Import Names 34
- Name 33
- Order 35
- Remap 37
- Show 36

Channel Order
- Dialog 35

Channels
- Selecting in Vector View 7

Checker Board
- Fill Value Dialog Box 39

CheckIt!
- Dialog 15
- File Menu 15

Circular Rotate
- Fill Shift/Rotate Dialog Box 42

Clock
- Delay 55
- Fill Menu 43
- Source (GTDIO Panel) 54

Clock Group
- GTDIO Panel 54

Close
- File Menu 12

Command
- Bar, View Menu 28
- Properties Dialog 29
- Properties, View Menu 29

Compare
- File Menu 12

Comparison
- View Menu 29

Contents
- Help Menu 45

Control Menu
- *DIOEasy* Main Window 4

Conventions 2

Copy
- Edit Menu 23

Copyright i

Create a New File

Dialog 87

GX5290 8

Creating a New Vector File 5

Cut

Edit Menu 23

Cycle Width (Steps)

Fill Clock Dialog Box....................43

D

Definitions 2

Delay Group

GTDIO Panel....................55

Delete

Edit Menu 23

Delphi 97, 98

Dialog Screens

Adding Commands and Labels....................80

Channel Order....................35

CheckIt! 15

Command Properties 29

Create a New File 8

Creating a New DIO File....................87

File Open Command....................10

File Properties General 16

File Properties Setup....................18

Fill Clock....................43, 81

Fill Direction....................44

Fill Pseudo Random....................39

Fill Ramp 40

Fill Toggle 41

Fill Value Dialog Box for GX5050 38

GoTo....................25

Import Names 34

Insert Steps 24

New Command Properties....................28

Options Window....................32

Remap Channels....................37

Save 13

Save As....................14

Save Board Data 56

Saving the Vector File 84

Select 24

Selecting a Range of Channels and Steps 94

Shift/Rotate....................42

Show Channels 36, 78

Using Command Properties Dialog to Add a Pause 89

Using Command Properties to Add a Label 95

Using Command Properties to Insert HALT Command 92

Using Command Properties to Set a Conditional Jump 92

Using Command Properties to Set an Unconditional Jump....................92

Using Fill Clock to Generate the Clock....................91

Using Fill Ramp to Generate the Waveform 90

Using Fill Ramp to Generate Waveform Values .90

Using Fill Value to Set the BusReq Line....................91

Using the Fill Value Dialog to Initialize Data 88

Using the Go To Dialog to Change Channels....................89

Vector View Window....................5

DIO

Board

Definition....................2

Definition....................2

What is DIO?....................3

DIOEasy 3

Command Bar....................28

Concepts 3

Main Window 4

Starting 3

Status Bar....................28

Toolbar 27

DIOEasy Main Window

Control Menu....................4

Menu Bar 4

Status Bar..4

Title Bar...4

Tool Bar...4

DioGetDriverSummary 100

DioGetErrorString 100

DioPanel 100, 101

DioPanel.exe..101

DioReset 100

DioSetupInitialization...............................58, 99, 100

DioSetupInitializationVisa 58, 99, 100

Disclaimer..i

Distributing..101

Don't Care/Disable

Fill Value Dialog Box..39

Driver Version 100

E

Edge

Delay..56

Edit Menu 23

Copy 23

Cut 23

Delete...23

Find Next Failed Cell 26

Find Next Failed Step 26

GoTo..25

Insert Steps 24

Paste...23

Select 24

Error...100

Error Handling...100

eVCD...22, 52

Every ___ Steps

Fill Ramp Dialog Box...40

Exit

File Menu...23

Expected Data

Delay..55

External

Events 55

Frequency, GTDIO Panel 55

F

File

Import File dialog 22

Importing VCD/eVCD File 52

Importing WGL File 47

Open, Command Dialog 10

Properties General Dialog..................................16

Properties Gx5295 Channles Settings.................20

Properties Window 16

Properties, Setup Dialog 18

Type, Properties Window 17

Version, Properties Window...............................17

View Import Profiles 49

File Menu...8

CheckIt! 15

Close 12

Compare 12

Exit 23

New..8

Open 10

Properties...16

Properties Channels settings Window 20

Properties General Window................................17

Properties Setup Window 18

Save 13

Save As 14

File Type

Properties Window 17

File Version

Properties Window 17

Fill

Clock Dialog..43, 81

Direction, Dialog 44

Direction, Fill Menu 44

Function, Other Patterns 39

Pseudo Random Dialog 39

Ramp...40

Ramp Dialog..40

Toggle Dialog 41

Value, Dialog Box for GX5050 38

Fill Menu 37

Clock...43

Fill Direction...44

Other Fill Function Patterns.................................39

Value...38

Find Direction Forward 25

Find Next

Failed Cell 26

Failed Step 26

Find Next Command 25

Find Next Command/Label 26

Find Next Label...25

Find Next Signal Change...26

First Step Level

Fill Clock Dialog Box..43

First Step Value

Fill Ramp Dialog Box..40

Frequency

GTDIO Panel...54

G

GoTo

Dialog 25

Edit Menu 25

Group

Selecting in Vector View.......................................7

GTDIO

Driver-Description...97

Header-file 97

Supported-Development-Tools............................97

GTDIO Panel

Master Selecting Combo Box 53

GTDIO.EXE 101

GTDIO.H...97

GTDIO.PAS 98

GTDIO32.BAS 97

GTDIO32.DLL 97, 98

GTDIO32.LIB 97

GTDIO32B.LIB..97

GXFPGA.llb 98

H

Halt!

GTDIO Panel...56

Handle 99, 100

Help i

Help Menu 45

About DIO 45

Contents 45

Search 45

Hide

Channel Menu..36

HW 97, 101

I

Import File … 22

Import File Dialog 47

Import Names

Channel Menu..34

Dialog 34

Importing a File 51

Importing External Vector Files 47

Importing VCD/eVCD File 52

Importing WGL File..47

Increment by

Fill Ramp Dialog Box..40

Insert Steps

Dialog 24

Edit Menu 24

Insert!

Fill Value Dialog Box..38

Interface Standard Level

GTDIO Panel..64

Introduction 1

Invert

Fill Value Dialog Box..39

Invert after Step

Fill Clock Dialog Box..43

J

Jum Register A

GTDIO Panel..64

Jum Register B

GTDIO Panel..64

Jump A Trigger Mode

GTDIO Panel..64

L

LabView 98

LabView/Real Time 98

Linux 98

List Box

Selecting 7

Listing..101

M

Mask Data

Delay..55

Master Selection

GTDIO Panel..53

Menu Bar

DIOEasy Main Window 4

N

Name

Channel Menu..33

Navigation Shortcuts 6

New

Command Properties Dialog................................28

File Menu..8

Vector File 5

Window Menu 45

nHandle 98, 100

Notes

Properties Window 17

O

Offset

Delay..55

One

Fill Value Dialog Box..39

OnError..98

Open

File Menu...10

Open Windows Menu Area

Window Menu 45

Options

Cell 32

Colors 32

Don't care/Disabled Cell 33

New Cells Fill..33

Open Last File on Start..33

Step Value 33

View Menu 32

Window Dialog Box 32

Order

Channel Menu..35

OutClock Delay

GTDIO Panel..64

Overwrite!

Fill Value Dialog Box..38

P

Panel 53, 59, 60, 62, 64, 66, 68, 70, 99, 100

Pascal...97, 98

Paste

Edit Menu 23

Pause!

GTDIO Panel..56

pnStatus 100

Profile Tab 49

Programming
  Borland-Delphi 98
  Error..100
  Panel-Program 101
Programming the Board...97
Properties
  Channels settings Tab 20
  File Menu..16
  General Tab 17
  Setup Tab..18
Pseudo-Random
  Fill Menu 39
PXI...99
PXI/PCI Explorer 57, 99, 100
PXIeSYS.INI 57
PXISYS.INI..57

R

Remap
  Channel Menu..37
Remap Channels
  Dialog 37
Reset 100
Reset!
  GTDIO Panel..56
Run Group
  GTDIO Panel..56
  The Panel 56

S

Safety and Handling i
Sample 101
Sample Programs..101
Save
  Board Data Dialog 56
  File Menu..13
Save As
  Dialog 13, 14
  File Menu..14
Saving the Vector File
  Dialog 84
Search
  Help Menu 45
Select
  Dialog 24
  Edit Menu 24
Selecting
  a Range of Channels and Steps Dialog 94
  Cells...7
  Items 7
Set Counter
  GTDIO Panel..56
Shift
  Fill Shift/Rotate Dialog Box 42
Shift/Rotate
  Dialog 42
  Fill Menu 42
Shift/Rotate (number) Times
  Fill Shift/Rotate Dialog Box 42
Show
  Channel Menu...36
  Channels Dialog 36, 78
Signals Tab 48
Slot..57, 99
Software..3
Split
  View Menu 30
Status Bar
  DIOEasy Main Window 4
  View Menu 28
Status Display
  GTDIO Panel..56
Status Tab 50
Step!
  GTDIO Panel..56
Steps

Selecting in Vector View......................................7

STIL...22, 51

STIL File 51

Strobe

Delay..55

Source 55

System Overview..3

T

Technical support i

Testware

Definition...2

Tile

Horizontally (Window Menu) 45

Vertically (Window Menu).................................45

Title Bar

DIOEasy Main Window 4

Toggle

Fill Menu 41

Toolbar

DIOEasy Main Window 4

View Menu 27

Trademarks ii

Trig!

GTDIO Panel..56

Trigger Group

GTDIO Panel..55

U

Unselect

Vector View..7

Unsplit

View Menu 31

User Guide Organization 1

User Knowledge and Skills 1

Using Command Properties

to Add a Label 95

to Add a Pause Dialog 89

to Insert HALT Command Dialog.......................92

to Set a Conditional Jump Dialog 92

to Set an Unconditional Jump Dialog 92

Using DIOEasy

With the GC5050/GX5050.................................71

With the GX515X...85

Using Fill Clock to Generate the Clock Dialog.......91

Using Fill Ramp

to Generate the Waveform Dialog 90

to Generate Waveform Values Dialog.................90

Using Fill Value

to Set the BusReq Line Dialog 91

Using Marvin Test Solutions' HW Device Driver ..57

Using the Fill Value Dialog to Initialize Data Dialog 88

Using the Go To Dialog to Change Channels Dialog 89

Using the GTDIO driver functions..........................99

Using VISA Device Driver.....................................57

UUT

Definition...2

V

Value

Fill Menu 38

VCD..22, 48, 51, 52

Vector File

Creating a New 5

Vector Group

GTDIO Panel..56

Vector View

Board Number 6

Channel Names..6

Channel Number..6

Command Bar...5

Horizontal Scroll Bar..5

Selecting Cells 7

Steps 6

Title Bar..5

Vector Cells 6

Vertical Scroll Bar ... 5
Window ... 5
View Menu ... 27
Command Bar ... 28
Command Properties ... 29
Comparison ... 29
Options ... 32
Split ... 30
Status Bar ... 28
Toolbar ... 27
Unsplit ... 31
Zoom In / Out/ Fit ... 30
Virtual Panel ... 54, 57, 59, 60, 62, 64, 66, 68, 70, 99
About Page ... 70
Advanced Page ... 64
Gx5055 High Power Chassis Page ... 68
Gx5295 I/O Channels Settings Page ... 60
Gx5295 Measure Page ... 62
Gx529x Real Time Compare Page ... 66
I/O Channels Settings Page ... 59
Main Page ... 54, 57
Virtual Panel Description ... 53
VISA ... 57, 99, 100
Visual Basic ... 97
Visual C++ ... 97

W

Warranty ... i
WGL ... 22, 47, 51
WGL File ... 51
Window Menu ... 45
Cascade ... 45
New Window ... 45
Open Windows Menu Area ... 45
Tile Horizontally ... 45
Tile Vertically ... 45

Z

Zero
Fill Value Dialog Box ... 39
Zoom
View Menu ... 30