Chroma

Programmable
Power Supply
PXI /cPCI 52912/52914
User's Manual

Edition
March 2004

Legal Notices

The information in this document is subject to change without notice.

Chroma ATE INC. makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Chroma ATE INC. shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

CHROMA ATE INC. 43 Wu-Chuan Road, Wu-Ku Industrial Park, Wu-Ku, Taipei, Taiwan

Trademarks

MAX®, LabVIEW ®, LabWindows®/CVI 6.0, IVI® and VISA® are registered trademarks of National Instruments Corporation.

Microsoft® Visual C++® 6.0 and Microsoft Visual Basic® 6.0 are registered trademarks of Microsoft Corporation.

Borland® C++ Builder is a registered trademark of Borland Corporation.

All other brand and product names are trademarks or registered trademarks of their respective holders.

Warranty

All Chroma instruments are warranted against defects in material and workmanship for a period of one year after date of shipment. Chroma agrees to repair or replace any assembly or component found to be defective, under normal use during this period. Chroma's obligation under this warranty is limited solely to repairing any such instrument, which in Chroma's sole opinion proves to be defective within the scope of the warranty when returned to the factory or to an authorized service center. Transportation to the factory or service center is to be prepaid by purchaser. Shipment should not be made without prior authorization by Chroma.

This warranty does not apply to any products repaired or altered by persons not authorized by Chroma, or not in accordance with instructions furnished by Chroma. If the instrument is defective as a result of misuse, improper repair, or abnormal conditions or operations, repairs will be billed at cost.

Chroma assumes no responsibility for its product being used in a hazardous or dangerous manner either alone or in conjunction with other equipment. High voltage used in some instruments may be dangerous if misused. Special disclaimers apply to these instruments. Chroma assumes no liability for secondary charges or consequential damages and in any event, Chroma's liability for breach of warranty under any contract or otherwise, shall not exceed the purchase price of the specific instrument shipped and against which a claim is made.

Any recommendations made by Chroma for use of its products are based upon tests believed to be reliable, but Chroma makes no warranty of the results to be obtained. This warranty is in lieu of all other warranties, expressed or implied, and no representative or person is authorized to represent or assume for Chroma any liability in connection with the sale of our products other than set forth herein.

CHROMA ATE INC.
43 Wu-Chuan Road, Wu-Ku Industrial Park,
Wu-Ku, Taipei Hsien, Taiwan
Tel: 886 -2-2298-3855
Fax: 886-2-2298-3596
http://www.chromaate.com

Table of Contents

1. Introduction

1.1 Product Overview

Chroma's 52912/52914 Programmable power supply is the industry's first embedded power supply packaged on standard single-slot 3U PXI card format. The advantages of using embedded PXI power supplies over conventional rack-and-stack power sources include: significantly faster execution speed for great ATE throughput, size and weight reductions, and greater modularity thereby reducing spares and maintenance costs and increasing system up-time.

Power Levels

The 52912/52914 provides two independent 60W channels each programmable from 0-48VDC to a maximum of 2 amps from two 56V DC supplies. If output isolation is a requirement, then isolated 56V DC supplies must be used. The instrument include programmable current limit to protect critical UUT's from excessive current as well as built in isolation and remote sense relays. For greater power or voltage applications channels maybe used in parallel, an optional front panel mounted adapter module also available.

Input Power

To avoid excess power draw from the PXI backplane, the 52912 draws input power (+56VDC) via front panel connections. This approach not only minimizes power required from the backplane but also maintains complete isolation between backplane logic and power conversion circuitry for noise immunity. Chroma provides an optional AC-DC adapter(A529102), which allows the instrument to be operated from 115VAC/ 220VAC mains.

Compliant to PXI and cPCI Standards

The 52912/52914 power supply cards comply with the latest PXI Revision 1.0 specifications of the PXI System Alliance (PXISA) as well as the CompactPCI specifications as defined by the PCI Industrial Computer Manufacturing Group (PICMG). Thus, the 52912/52914 maybe be used in either PXI or CompactPCI mainframes.

Conclusion

If you have selected PXI or cPCI for your next generation of ATE, considering taking full advantage of its benefits by eliminating antiquated rack-and-stack chassis and replacing them with embedded power supplies. The end result will be a more fully integrated system, which is fast, elegant and cost effective.

1.2 Opening the Power Supply

1.2.1 Opening the Package

Open the **52912/52914 Programmable Power Supply** package carefully and inspect if all hardware are in good condition without any damages and if the Compact Disc is broken or unreadable. If any damage is found and caused the hardware or software application unable to execute, return the product in its original package and contact us.

1.2.2 Inspecting the Goods

The PXI/cPCI Programmable Power Supply kit contains the following items:

- A PXI 52912 Programmable Power Supply card
- A CD containing test software and drivers, and user's manual

1.3 Features of Programmable Power Supply

Key Features

- 1-Slot, 3U PXI card For 52912; 3-Slot, 3U PXI card For 52914
- Dual Isolated outputs; 0-48VDC/2A/60W per output
- Programmable current limit
- Includes over voltage, over current, and short circuit protection
- On-board isolation and remote sense relays
- 16Bit read back of output voltage and current
- Outputs maybe connected in parallel
- 32-bit DLLs provided

2. Drivers Installation

2.1 Driver CD

The driver CD contains the following programs and files:

- Windows 2000, Window 9x, and Windows NT4 device drivers
- 52912/52914 Programmable Power Supply soft front panel and program examples
- The instrument drivers and libraries are supported as listed below:
  - NI LabVIEW® Ver. 6.0, 6.1 & 7.0
  - NI LabWindows/CVI® Ver. 6.0
  - Microsoft Visual C++® Ver. 6.0
  - Microsoft Visual Basic® Ver. 6.0
  - Borland C++ Builder® Ver. 5.0

2.2 Installing the Software

The 52912/52914 instrument drivers are base on National Instruments VISA and IVI. Please install NI-VISA 2.6 and NI-IVI 1.83 (or newer version) before installing 52912/52914 instrument drivers.

Insert the Driver CD into the CDROM drive and the following program should show up automatically. If it does not run, select **Start** and choose **RUN**. Click **Browse** and locate the file "**`X:\setup.exe`**" (where 'X' is the drive the CD is in), then execute it.

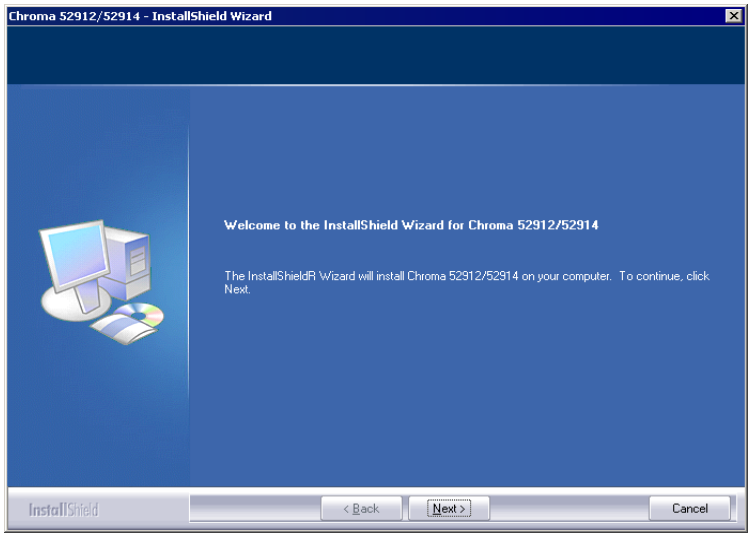


Figure 2-1 Installation Menu

Following the instruction of setup program to install Soft Front Panel. The pre-defined installation directory is `C:\Program Files\Chroma\PXI\52912`. It is recommended to use this path as installation directory. Please follow the prompts to complete the installation procedure.

Note: To install the device driver on Windows 2000 or Windows NT4, you should log on as the "administrator".

2.3 Installing the Hardware

Step 1. Make sure the chassis is powered off.
Step 2. Select a slot in the chassis and install the 52912/52914 PXI card carefully, then secure it with the screws on the panel.
Step 3. Connect the A529102 Adapter with an 8 pins of terminal blocks.
Step 4. Power the chassis up and boot Windows.

2.3.1 Device Driver Installation on Windows 2000

Step 5. "Found New Hardware Wizard" will show up after Windows 2000 booting. Click **Next >**.

Step 6. Choose “Search for a suitable driver for my device (recommended).”. Click **Next >**.

Step 7. Choose “Specify a location”. Browse the drive path and choose `C:\Program Files\Chroma\PXI\52912\inf\chrni2k.inf` (or the directory you specified in section 2.2). Click **Next >**.

Step 8. The wizard found “52912/52914 Programmable Power Supply”. Click **Next >**.

Step 9. Click **Finish** to finish device driver installation on Windows 2000.

2.3.2 Device Driver Installation on Windows 98

Step 5. “Add New Hardware Wizard” will show up after Windows 98 booting. Click **Next >**.

Step 6. Choose “Search for a best driver for your device (Recommended).”. Click **Next >**.

Step 7. Choose “Specify a location”. Browse the drive path and choose `C:\Program Files\Chroma\PXI\52912\inf\chrni98.inf` (or the directory you specified in section 2.2). Click **Next >**.

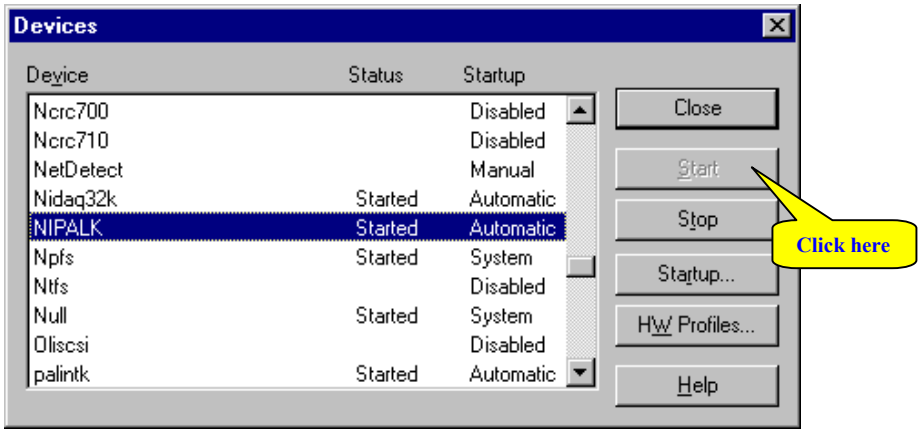
Step 8. The wizard found “52912/52914 Programmable Power Supply”. Click **Next >**.

Step 9. Click **Finish** to finish device driver installation on Windows 98.

2.3.3 Device Driver Installation on Windows NT4

Step 5. To install `chrnint4.inf` file, please locate the file at `C:\Program Files\Chroma\PXI\52912\inf` directory (or the directory you specified in section 2.2). Right-click the file then select **Install** from the options menu.

Step 6. From **Control Panel**, double click **Devices** icon. Select the device named “NIPALK” then click **Start** to start it.



2.4 Hardware Verification

When the chassis with 52912 PXI card installed is powered on and all signal LEDs are off, input the +56Vdc external power source, the signal LED will light up in green. You can use the control software for 52912 PXI card to set the V and I of output channel. When the channel output signal is activated, the ON signal on the panel will light up in green. If both of the PWR and ON signal LEDs are lit up, it indicates the 52912 PXI card is in normal operation. Once there is an OVP or OTP situation occurred, the ERR signal LED on the panel will light up in red.

3. Software

3.1 Block Diagram

As the following figure shows, the application is on top of the system. It calls the supported function, such as:

chr52912_init(ViRsrc resourceName,…,ViPSession instrumentHandle);

The instrument driver communicates with the hardware via the standard VISA call function.

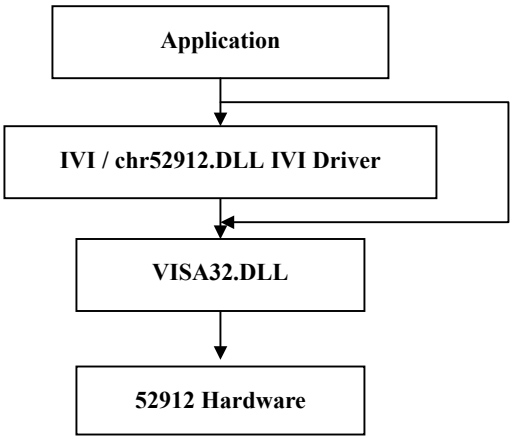


Figure 3-1 Block Diagram of Software

3.2 User Applications

User applications can be written by standard programming languages such as LabVIEW, LabWindows/CVI, Visual Basic, Visual C^{++}, and Borland C^{++} Builder. An example of it can be seen in section 5.3 *Example Programs* at the end of this manual.

3.3 Software Control

Following are the easy control routines for read, write and PXI configuration log.

```
ViStatus chr52912_init (
                      ViRsrc resourceName,
                      ViBoolean IDQuery,
                      ViBoolean resetDevice,
                      ViPSession instrumentHandle);

ViStatus chr52912_InitWithOptions (
                      ViRsrc resourceName,
                      ViBoolean IDQuery,
                      ViBoolean resetDevice,
                      ViString optionString,
                      ViPSession instrumentHandle};

ViStatus chr52912_close (ViSession instrumentHandle)

ViStatus chr52912_ConfigureOperationMode (
                      ViSession instrumentHandle,
                      ViChar _VI_FAR channelName[],
                      ViInt32 operationMode);

ViStatus chr52912_ConfigureOutputEnabled (
                      ViSession instrumentHandle,
                      ViChar _VI_FAR channelName[],
                      ViBoolean enabled);

ViStatus chr52912_ConfigureCurrentLimit (
                      ViSession instrumentHandle,
                      ViChar _VI_FAR channelName[],
                      ViInt32 behavior,
                      ViReal64 limit );

ViStatus chr52912_ConfigureVoltageLevel (
                      ViSession instrumentHandle,
                      ViChar _VI_FAR channelName[],
                      ViReal64 level );

ViStatus chr52912_ConfigureTriggerSource (
                      ViSession instrumentHandle,
                      ViInt32 stratSource,
                      ViInt32 stopSource,
                      ViInt32 slope);

ViStatus chr52912_ConfigureTriggeredOutput (
```

```
                    ViSession instrumentHandle, ViChar
                    _VI_FAR channelName[],
                    ViReal64 triggeredLevel,
                    ViReal64 triggeredLimit);

ViStatus chr52912_Measure (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViInt32 measurementType,
                    ViPReal64 measurement);

ViStatus chr52912_ConfigureSequencerTable (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViUInt16 tableSize,
                    const SEQUENCER_TABLE *tablePointer,
                    ViUInt16 errorIndex);

ViStatus chr52912_ClearSequencerTable (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[]);

ViStatus chr52912_IsRunningSequencer (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViPBoolean isRunning);

ViStatus chr52912_ConfigureMeasure (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViReal64 level, ViReal64 limit,
                    ViInt32 measurementType,
                    ViInt32 delayTb,
                    ViInt32 delayTbUnit,
                    ViInt32 delayTc,
                    ViInt32 delayTcUnit,
                    ViInt32 delayTe,
                    ViInt32 delayTeUnit,
                    ViInt32 measureCount);

ViStatus chr52912_ReadMeasuredData (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViPInt16 num_ofData,
                    MEASURED_DATA measuredData[]);

ViStatus chr52912_Initiate (
                    ViSession instrumentHandle,
```

```
                    ViInt32 channel_select);

ViStatus chr52912_ Abort (ViSession instrumentHandle);

ViStatus chr52912_SendSoftwareTrigger (
                    ViSession instrumentHandle);

ViStatus chr52912_IsWaitingTrigger (
                    ViSession instrumentHandle,
                    ViPBoolean inState);

ViStatus chr52912_reset (ViSession instrumentHandle);

ViStatus chr52912_ResetOutputProtection (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[]);

ViStatus chr52912_error_message (
                    ViSession instrumentHandle,
                    ViStatus errorCode,
                    ViChar _VI_FAR errorMessage[]);

ViStatus chr52912_error_query (
                    ViSession instrumentHandle,
                    ViPInt32 errorCode,
                    ViChar _VI_FAR errorMessage[]);

ViStatus chr52912_QueryOutputState (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViInt32 outputState,
                    ViPBoolean inState);

ViStatus chr52912_QueryMaxCurrentLimit (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViReal64 voltageLevel,
                    ViPReal64 maxCurrentLimit);

ViStatus chr52912_QueryMaxVoltageLevel (
                    ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViReal64 currentLimit,
                    ViPReal64 maxVoltageLevel);

ViStatus chr52912_QueryCalibrationDate (
                    ViSession instrumentHandle,
                    ViChar errorMessage[]);
```

3.4 Operating procedure

Each channel in a Chroma PXI 52912/52914 Power Supply Card has one unique Sequencer table and Measure table. The concept of sequencer tables and Measure tables is shown in Figure 3-2.

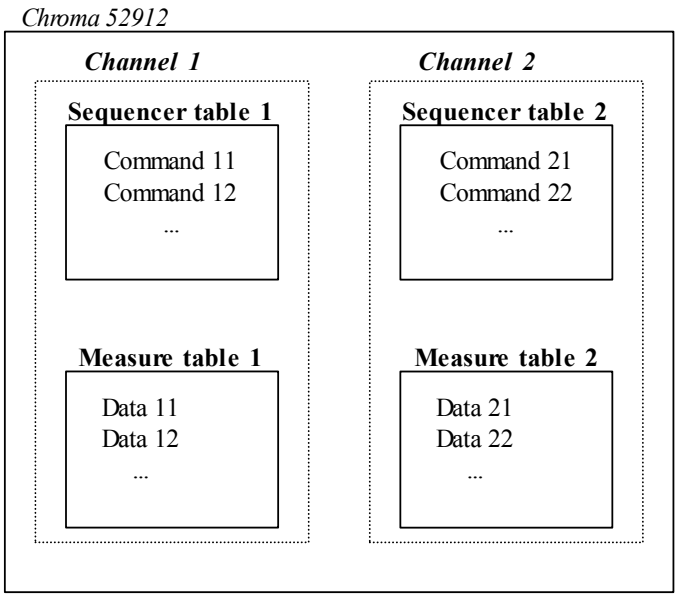


Figure 3-2 the concept of sequencer tables and Measure tables

There are ten commands you can utilize to configure the Sequencer table. Refer to section 5.1.11 chr52912_ConfigureSequencertable for details on these commands.

The Sequencer tables can be configured and activated separately, or they can also be activated simultaneously via the trigger events. If the command, which takes measurement of the voltage or current on the channel, appears anywhere in the sequencer table, the measured values will be stored in the corresponding Measure table. For instance, the measured value for channel 1 will be stored in the Measure table belonging to channel 1.

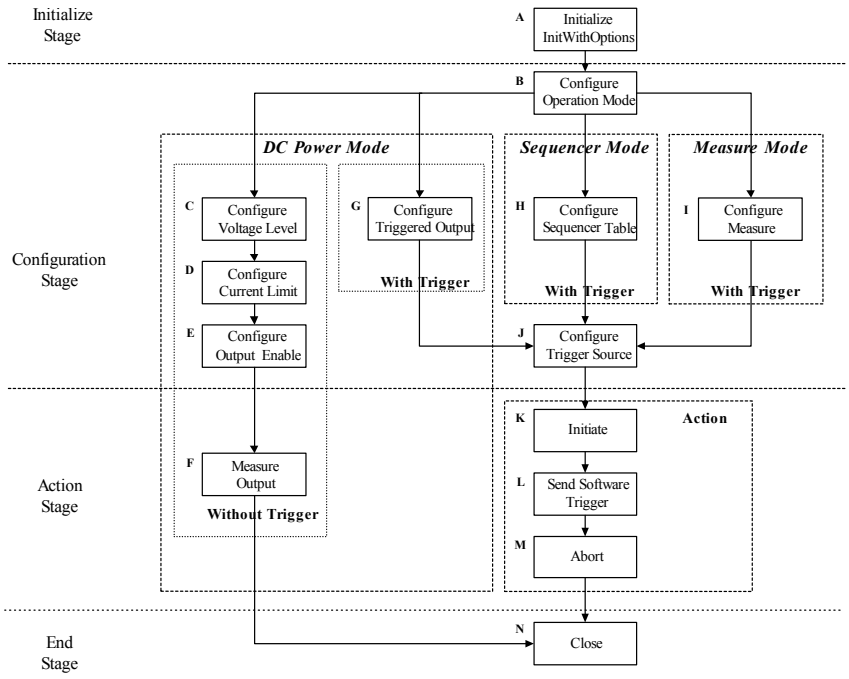


Figure 3-3 Operating procedure of 52912

The operating procedure of 52912/52914 is shown in Figure 3-3. More detailed descriptions are discussed in later sections. Each block in the figure above is labeled with a bold-faced alphabet and the APIs that will be called in these blocks are listed below.

APIs Called in Figure 3-3

A.
1.chr52912_init
2.chr52912_InitWithOptions
B.chr52912_ConfigureOperationMode
C.chr52912_ConfigureVoltageLevel
D.chr52912_ConfigureCurrentLimit
E.chr52912_ConfigureOutputEnabled
F. chr52912_Measure
G. chr52912_ConfigureTriggeredOutput
H. chr52912_ConfigureSequencerTable
I. chr52912_ConfigureMeasure
J. chr52912_ConfigureTriggerSource

K. chr52912_Initiate
L. chr52912_SendSoftwareTrigger
M. chr52912_Abort
N. chr52912_close

3.5 Operation Modes

There are three operation modes supplied with Chroma PXI 52912/52914 Power Supply Card: DC Power mode, Sequencer mode and Measure mode.

3.5.1 DC Power mode

3.5.1.1 Operating Without Trigger

This is the default operation mode as the power supply is initialized or reset. In this mode, almost all of the driver APIs can be called (unlike in the other modes, some APIs might be locked), and by doing so, they can achieve the common functionalities of a power supply, such as configuring the voltage level and current limit.

3.5.1.2 Operating With Trigger

The power supply can also be activated by trigger events in the DC Power mode. First, you have to configure the triggered level and triggered limit that the power supply should generate as receiving the trigger; i.e. by inputting the commands such as SET_V and SET_I via chr52912_ConfigureTriggeredOutput function into the sequencer table. Then, you have to determine the start/stop triggers for a 52912/52914 to start/stop outputting, as well as determining the trigger slope.

The last step is to initiate the power supply to wait for triggers. After receiving a start trigger signal, the power supply starts to generate output. It will keep outputting until receiving a stop trigger signal, or if the 'abort' API is called.

3.5.2 Sequencer mode

The Sequencer table is a powerful tool. Users can predefine several commands and write them into the power supply for execution.

In this mode, once when the power supply receives the start trigger signal, it will start to execute the commands in the sequencer table line by line. The outputting remains even if the 'abort' API is called. In section 5.3.6.4, there are two simple examples showing how a sequencer table is used.

Notice that there are several restrictions when using a sequencer table.

1. All values in a sequencer table should be within the valid range (using the pre-defined constants in the header files is recommended; e.g. CHR52912_VAL_SET_V).

2. There must be at least one pair of SET_V and SET_I commands executed before any OUTPUT_ON command, in order to protect the external circuit connecting to the 52912/52914 (UUT) from burned-out. This rule applies for both channel 1 and 2.

3. 52912/52914 has a power limit of 60 Watts on each channel.

4. Nested REPEAT commands are allowed, but the number of nested layers cannot exceed 7.

5. The last command will always be END and each sequencer table must have at least one END command. The power supply will keep executing through the sequencer table until it encounters an END command.

3.5.3 Measure mode

The Measure mode provides a simple way to take periodical measurements on the power supply's output. By calling the chr52912_ConfigureMeasure function, the driver will insert the necessary commands into the sequencer table for periodical measurements. Notice that in this mode, users do not have to inputs commands manually. Then, just like in the Sequencer mode, you must configure the trigger source (Note: in this mode, only the start trigger is configurable), as well as initiating the power supply to wait for it. After receiving the start trigger signal, the power supply starts to generate output and takes periodical measurements on the output. The outputting stops as 52912/ 52914 takes the specified times of measurements, or if the 'abort' API is called. In section 4.3.7.3, there are two simple examples showing how a sequencer table is used in the Measure Mode.

4. Soft Front Panel

4.1 Software Control Panel

Features and functions of this product can be controlled via a software control panel.

4.2 Block Diagram

The following figure shows the flow of device operations. Users rely on the application programs to control the 52912/52914.

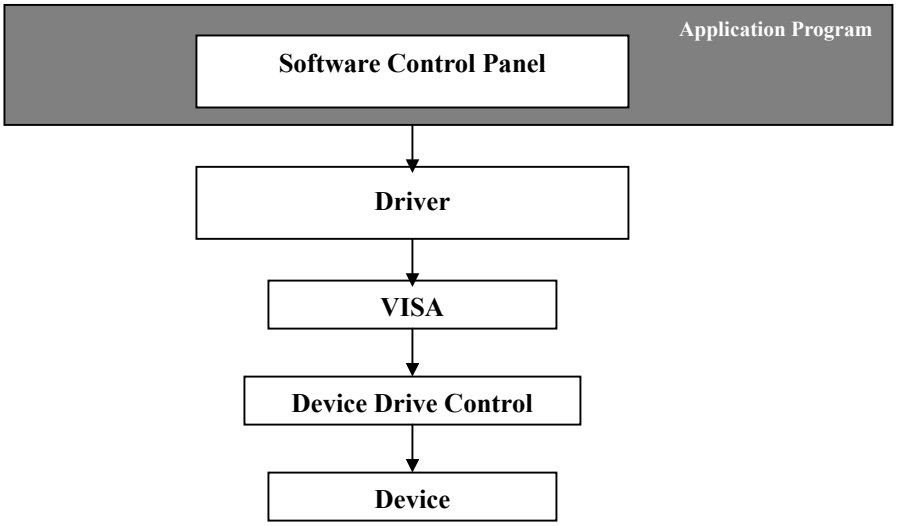


Figure 4-1 Process Flow

4.3 Using the Software Control Panel

4.3.1 Executing SFP5291214

To execute the application program, run **SFP5291214.exe** under the directory destined during installation. (The default path for this file is **C:\Program Files\Chroma\PXI\52912\bin**)

4.3.1.1 Starting up 52912

1. Run **SFP5291214.exe** or double click on the icon "**SFP5291214**".
2. The main screen will appear as the figure below (Figure 4-2).
3. A "**Select 52912/52914 Programmable Power Supply**" dialog box will also be shown as follows.

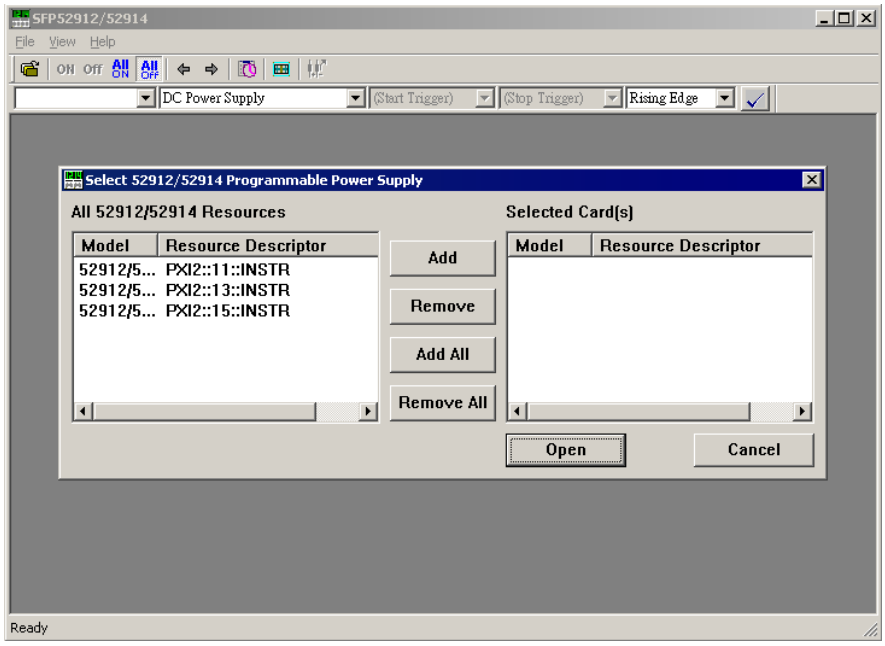


Figure 4-2 Main Window

4.3.1.2 Selecting 52912/52914

1. On the left-hand side, all 52912/52914's detected by SFP5291214 are listed.
2. On the right-hand side, all 52912/52914's selected for operations are listed.
3. Click on **Add** or **Add All** button to add card(s) to the list on the right, or click on **Remove** or **Remove All** button to remove card(s) from the list on the right.
4. After all desired card(s) are added to the "Selected Card(s)" list, click on the **Open** button to start up **SFP5291214** with all selected card(s) or click on the **Cancel** button to enter the main program without opening any 52912/52914. (You can open 52912/52914 later by clicking on the icon, or by choosing **File** (on the menu bar) → **Open** to launch the "**Choose 52912/52914**" dialog box again.)

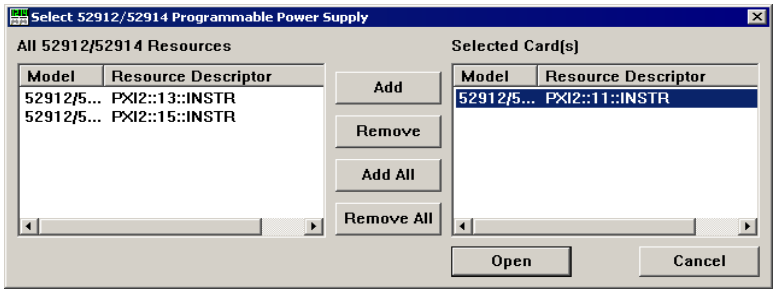


Figure 4-3 Device Select

4.3.2 Operation Mode

4.3.2.1 Introducing Operation Mode

SFP5291214 has four operation modes: "**DC Power Supply**", "**DC Power Supply with trigger",** "**Sequencer**" and "**Measure**" modes.

In the "**DC Power Supply**" mode, a 52912/52914 acts as a conventional DC power supply source. (For more details on this mode, please refer to **Section 4.3.4** ***Introduction of DC Power Supply Control Panel***.)

In the "**Sequencer**" mode, a 52912/52914 acts as a sequence generator with some command tables pre-set via "**Sequencer Editor**". (For more details on "**Sequencer Editor**", please refer to **Section 4.3.6** ***Introduction of Sequencer Mode***.)

In the "**Measure**" mode, you can configure the time-interval between each measurement precisely. (For more details on "***Measure Attributes Editor***", please refer to **Section 4.3.7** "***Measure Attributes Editor*** ".)

In the "**DC Power Supply with trigger**" mode, the power supply can also wait for trigger signal. (For more details on this mode, please refer to **section 4.3.5** "***Introduction of DC Power Supply with Trigger Mode***".)

4.3.2.2 Configuring Operation Mode

Before you manipulate a 52912/52914, you need to configure operation mode in advance.
The default mode is "**DC Power Supply**".

To select modes other than the default one, select **View →Configure Mode Trigger Bar** at the menu bar. Then, a trigger bar will appear as shown in Figure 4-4. This bar allows you configure "Resource Name", "Operation Mode", "Start Trigger" and "Stop Trigger". The available selections in each scrollbox of the trigger bar are listed below:

Figure 4-4 Configuring operation mode

(Select a 52912/52914): This is the left-most scrollbox in the trigger bar. Default is set to be the first 52912/52914 that is opened(empty if no 52912/52914 is opened). The format of the name of 52912/52914's is as follows:

PXI[board]::<logical address>::INSTR

For more details on this, please refer to **Section 5.1.1 "*chr52912_init*"**.

(Select Mode): **DC Power Supply**
Sequencer
Measure
DC Power Supply with trigger

(Start Trigger): External
Software
TTL0
TTL1
TTL2
TTL3
TTL4
TTL5
TTL6
TTL7

(Stop Trigger): Same as (Start Trigger)

 : Clicking on this button will apply all the selections in the scrollboxes to the selected 52912/52914. (Note: no change will be made until this "Apply" button is toggled.)

Note: According to different operation modes, you must configure appropriate triggers for each one. Then, before executing the mode-dependent actions, the power supply will be suspended until a (Start Trigger) signal is received.

For example, in the "**DC Power Supply with trigger"** mode, you must configure both the (Start Trigger) and the (Stop Trigger). The power supply will generate output as soon as it receives the (Start Trigger) signal, and terminates the outputting when the (Stop Trigger) signal occurs.

In the "**Sequencer"** and the "**Measure"** modes, you can only specify the (Start Trigger). And the power supply executes the command sets in the sequencer table as soon as it receives the (Start Trigger) signal.

P.S. One 52912/52914 can only be in one mode at a time.

4.3.3 Introduction of Tool Bars

Tool buttons:

Figure 4-5 General tool buttons

 *Opens a 52912/52914:* clicking on this icon will launch the "**Choose 52912/52914**" dialog box. (For more details on the "**Choose 52912/52914**" dialog box, please refer to **Section 4.3.1.2 *"Selecting 52912/52914"***)

 *Turns On The Power:* clicking on this icon will turn on the power of the currently-selected 52912/52914. In each mode, 52912/52914 won't start functioning until this icon is toggled. In the "**DC Power Supply**" mode, the conventional panel is activated after this icon is toggled; and in the "**Sequencer**" mode, after the power is turned on and you have finished editing correct commands via Sequencer Editor, the sequencer can be ready for receiving run by toggling the 

button (For more details on this mode, please refer to **Section 4.3.6 *"Introduction of Sequencer Mode"***); and in the **"Measure"** mode, after the power is turned on and you have finished editing correct attributes settings via Measure Attributes Editor, the mode can be ready for receiving run by toggling the 

button (For more details on this mode, please refer to **Section 4.3.7** *"**Introduction of Measure Mode"***).

 *Turn Off The Power:* clicking on this icon will turn off the power of the currently-selected 52912/52914. In all modes, toggling this icon will

disable all the functionalities. In the "**DC Power Supply**" mode, the conventional panel is deactivated after this icon is toggled.

Turn On The Power of All 52912/52914's: clicking on this icon will turn on the power of all 52912/52914's. (Also see 

for details of turning on powers.)

Turn Off The Power of All 52912/52914's: clicking on this icon will turn off the power of all 52912/52914's. (Also see 

for details of turning off powers.)


Select Previous Card: clicking on this icon will move the active window to the previous Soft Front Panel (SFP).


Select Next Card: clicking on this icon will move the active window to the next Soft Front Panel.


Get Last Calibration Date: clicking on this icon will show the latest date of calibration.


Switch to Status Table: clicking on this icon will show the status of all 52912/52914's. This status table is for observations only and has no controls and effects on 52912/52914's.


View Measured Data: clicking on this icon will launch a dialog of "View Measured Data" for viewing the currently-active 52912/52914's measured data in channel 1 or channel 2.

4.3.4 Introduction of DC Power Supply Control Panel

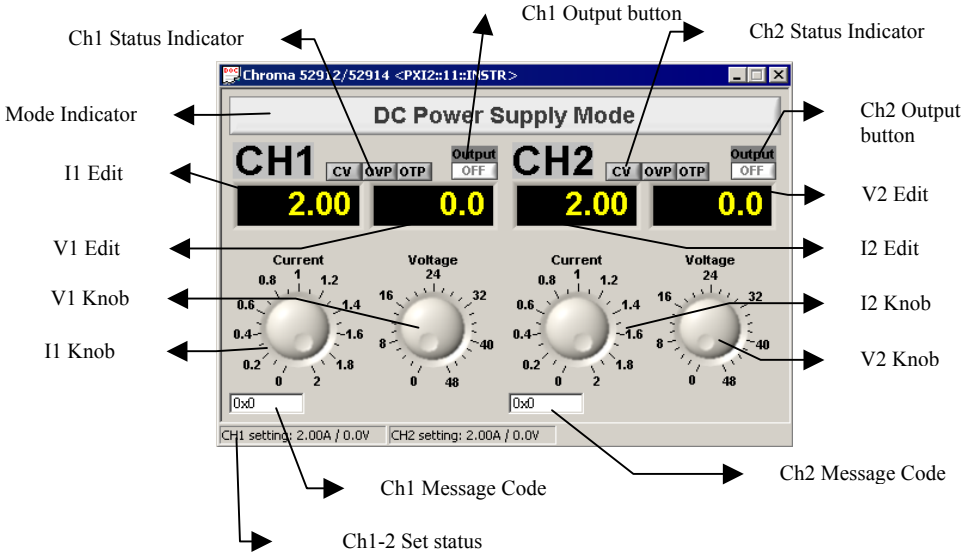


Figure 4-6 Layout of DC Power Supply SFP

4.3.4.1 Components of DC Power Supply SFP

There are three types of component in the DC Power Supply SFP.

Controller type: this type of components is used to adjust/enable/disable outputs.

I1 Knob: Sets channel 1's current limit
V1 Knob: Sets channel 1's voltage level
I2 Knob: Sets channel 2's current limit
V2 Knob: Sets channel 2's voltage level
Ch1 Output button: Enables channel 1's output
Ch2 Output button: Enables channel 2's output

Indicator type: this type of components is used for displaying status only (unmodifiable).

Ch1 Status and Ch2 Status: Indicates the output state of channel 1 and channel 2, respectively:

CV - constant voltage
CC - constant current
OVP - over voltage protection
OTP - over temperature protection

Ch1 and Ch2 Message Code: Indicates the status code of channel 1 and channel 2, respectively.

(For more details on warnings and errors of NI-VISA® and NI-IVI®, please refer to **section 5.2 *"Error Code"***)

Ch1 and Ch2 Set Status: Indicates the set values of current and voltage of channel 1 and channel 2, respectively.

Mode: Indicates the current mode of operation.

Controller and Indicator type: this type of components works either as a controller (text will be in yellow) or as an indicator (text will be in green).

I1 Edit, I2 Edit,V1 Edit,V2 Edit: Indicates the input current/voltage of channel 1 and 2 while working as a controller, and indicates the output current/voltage of channel 1 and 2 while working as a indicator.

As a 52912/52914 is just powered on, all of *I1 Edit, I2 Edit, V1 Edit, V2 Edit* work initially as controllers. After the output relay is activated, *I1 Edit, I2 Edit, V1 Edit, V2 Edit* will work as indicators unless manually **set** again. (For more details on "Setting Voltage/Current on Edit", please refer to **Section 4.3.4.2 *"Setting Voltage/Current on Edits"***)

4.3.4.2 Setting Voltage/Current on Edits

- As a 52912/52914 is just powered on, all of *I1 Edit, I2 Edit, V1 Edit, V2 Edit* work initially as controllers (the text will be in yellow). Click on *Edits* and then the area will become editable. After entering the desired values, press the Output OFF button. SFP5291214 will export those entries to 52912/52914, and then show the responding values (the read-back values) in the *Edits*. Now the values in the *Edits* are shown in green, which means *Edits* now work as an indicator.

- As a 52912/52914 works as an indicator, if you wish to export a new set of current/voltage, you can again click on *Edits*, and then two buttons, SET and CANCEL, will appear. (Note: the text color will become yellow, which means *Edits* now work as controllers.) After entering the new export values, clicking on SET will cause SFP5291214 to export the new values to the 52912/52914 again, and read back the responding values, showing them in *Edits*. Note that the text color of the values in *Edits* now becomes green again. If you wish to discard the changes after some new values are entered, just simply click on the CANCEL button.

4.3.4.3 The Progress of Exporting Current/Voltage

1. You can use either *Knobs* or *Edits* to set up the output voltage and the current limit. (See Figure 4-7)

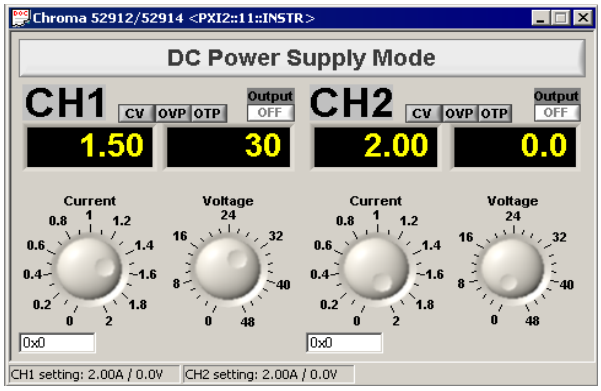


Figure 4-7 Setting Voltage/Current on DC Power Supply

2. After current/voltage are set, click on the Output OFF button to enable the output relay. Then, all the readings will be displayed in the *Edits* in the color of green. (See Figure 4-8)

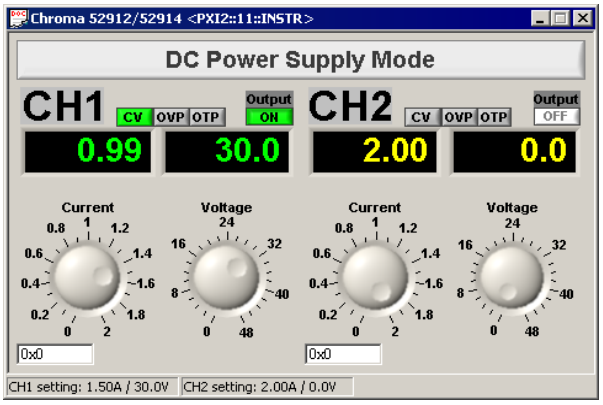


Figure 4-8 Channel 1 Output condition I

3. While outputting, SFP will try its best to remain in the constant voltage (CV) state. However, when the load condition surpasses the current limit, SFP will change into the constant current (CC) state. In this case, the voltage output will be reduced, accordingly.

4. When abnormal load conditions (such as external voltage discharged on the card) made the output voltage exceed the control range, the over voltage protection (OVP) LED will light up in red, and the output relay will be disabled. In addition, an Error Message dialog box will appear.

***WARNING*: To Prevent personal injury or damage to 52912/14, do not attempt to manipulate it again via SFP5291214 before eliminating OVP condition.**

5. You can manually enter a new value for voltage or current during measurement. After the value is entered, click on SET to output the value, or click on CANCEL to return to the original readings. (See Figure 4-9)

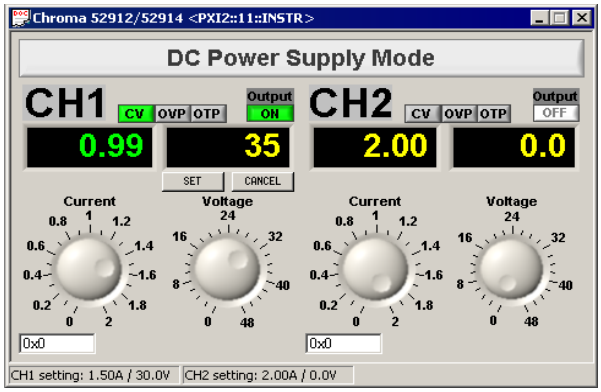


Figure 4-9 Channel 1 Output condition II

6. Input values are always checked to ensure proper operations. Warning messages will be given if incorrect values or misformatted inputs are given.

4.3.4.4 Tool Tips and Status Bar

SFP5291214 has a user-friendly design for monitoring the status of 52912/52914's. Resting your mouse on the Knobs will exhibit a tool tip, which tells you the current set value of current/voltage. (See Figure 4-10) This value will be the same as the reading displayed in the status bar.

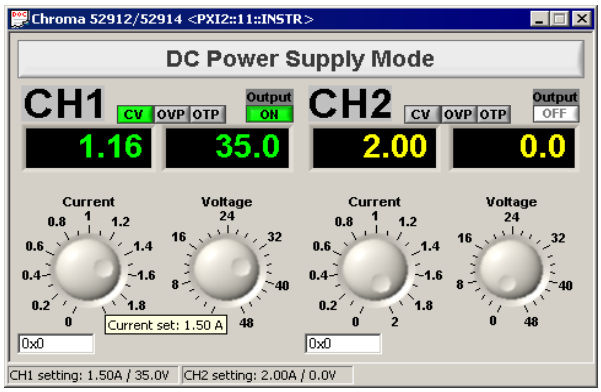


Figure 4-10 Tool Tips - 1

If an error occurs, resting your mouse on the *Message Code* indicator will show you a translated version of the message code shown in the indicator.
(See Figure 4-11)

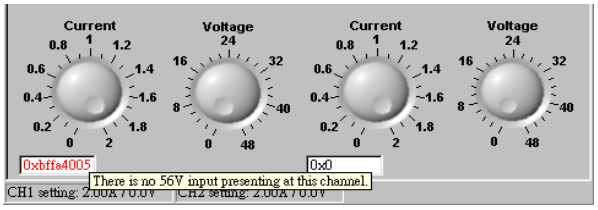


Figure 4-11 Tool Tips - 2

4.3.5 Introduction of DC Power Supply with Trigger Mode

52912/52914 can also wait for a trigger signal: It will start to generate outputs as soon as a (Start Trigger) signal occurs, and won't stop until a (Stop Trigger) signal is received.

4.3.5.1 Setting Up The Triggers

To activate the triggers, select **View →Configure Mode Trigger Bar** at the menu bar. A trigger bar will then appear under the tool bar. Under the (Select a 52912/52914) scrollbox, choose which 52912/52914 you want to manipulate. Under the (Select Mode) scrollbox, select "**DC Power Supply with Trigger**". Select the (Start Trigger) and (Stop Trigger) you desire. At last, select the slope (Rising Edge or Falling Edge) of trigger and confirm all the selections by clicking on the button. (For more details on the available selections in the scrollboxes, please refer to **Section 4.3.2.2. *"Configuring Operation Mode"***)

After all the selections are done and confirmed, you will see the appearance of the control panel changed as shown in Figure 4-12.

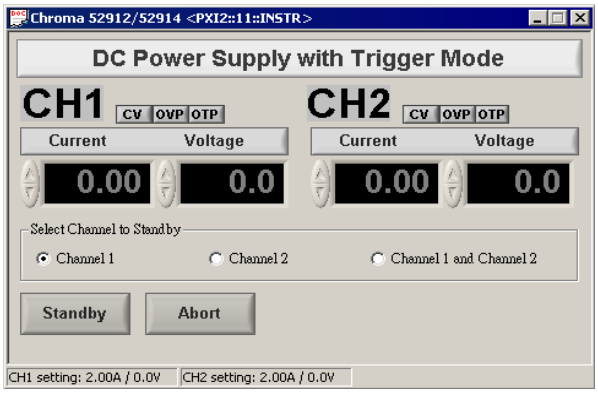


Figure 4-12 Control Panel for DC Power Supply with Trigger

4.3.5.2 Running in DC Power Supply with Trigger Mode

Once when you have done the settings in **4.3.5.1**, the next step is to set up the values of voltage and current for both channels. Then, after that, select the channel you want by clicking one of the three check buttons: Channel 1, Channel 2 and Channel 1 and Channel 2.

Only a condition can be selected to stand-by for the trigger signal.

Then, you can click the 

button to force 52912/52914 to wait for a trigger signal (became 

). If the 52912/52914 receives a start trigger (became 

), it will try its best to remain in the programmed settings of voltage and current, and stop when receiving a stop trigger signal. It goes back to the stand-by status and wait for the next start trigger again.

If you have selected the software trigger as a start trigger, the 

button will appear. Clicking on this button allows you to send a software trigger for starting and stopping.

4.3.5.3 Examples

The following is an example of 52912/52914 in DC Power Supply with Triggers mode. Be careful that every channel has a power limit of 60 watts, therefore, be aware of the actual load at the instant when 52912/52914 starts outputting.

Example: Controlling Channel 1 and 2 with TTL0 as (Start Trigger) and TTL7 as (Stop Trigger).

Step 1: At the menu bar, select **View →Configure Mode Trigger Bar**. Choose the 52912/52914 that you want to manipulate, and then under the (Select Mode) scrollbox, select "**DC Power Supply with trigger**". Then, select TTL0 under the (Start Trigger) scrollbox, followed by selecting TTL7 under the (Stop Trigger) scrollbox. At last, select the slope type of the triggers, and then click on the  button to confirm.

Step2: The appearance of soft control panel will be changed as in Figure 4-13. Enter your values of the voltage and current for each channel. Select the checkbox of "Channel 1 and Channel 2". Then click on the 

button. The 52912/52914 now starts waiting for a TTL0 trigger, and the button becomes 

(see Figure 4-14.)

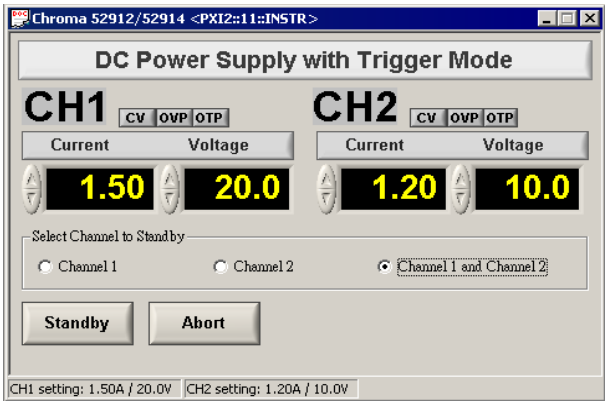


Figure 4-13 Example for DC Power Supply mode

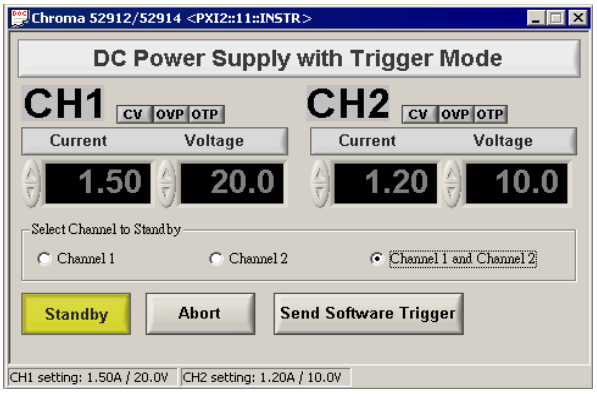


Figure 4-14 Example for Waiting a TTL0 trigger

Step3: At this time, you need to generate a TTL0 trigger. After receiving the TTL0 trigger. The skin changed as Figure 4-15.

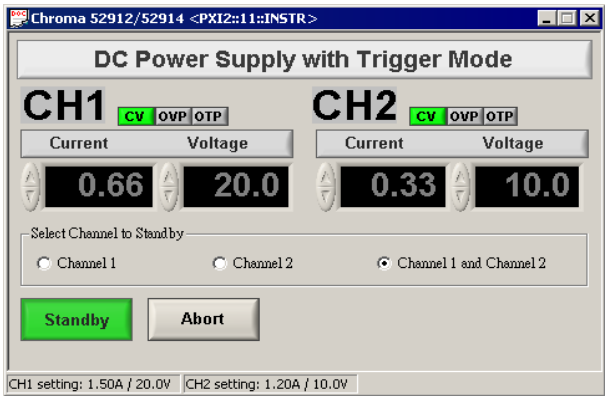


Figure 4-15 Example After Receiving a TTL0 trigger

Step3: Then you need to generate a TTL7 trigger. After receiving the TTL7 trigger. The 52912/52914 goes back to wait for the TTL0 just like what is shown in Figure 4-14.

Note: During a standby condition, you can click the Abort button to abort the stand-by state. By doing so, 52912/52914 will disable the output relays and go back to the default setting.

4.3.6 Introduction of Sequencer Mode

There are two kinds of sequencer tables you can configure. When the start trigger signal is received, the power supply sequentially executes the command sets in the sequencer table.

4.3.6.1 What is A Sequencer Table?

A sequencer table is a series of commands that you want 52912/52914 to execute in a sequential manner. There are a variety of commands you can use in a table, however, the maximum number of commands in each sequencer table is 256. Some columns may be redundant for some commands. In this case, the title of the useless columns will become "Not Used".

For example: If the command is "Set Voltage Level", then only the first two columns to the right of "Command Type", i.e. "**Select Channel**" and "**Value/Times**", are used. The right-most column is redundant and becomes "**Not Used**".

The following is a table showing which columns are used for each command:

Command Type	Select Channel	Value/Times	Duty/StartIndex
Set Voltage Level	v	v	-
Set Current Limit	v	v	-
Output On	v	-	-
Output Off	v	-	-
Delay Time	-	-	v
Measure Voltage	v	-	-
Measure Current	v	-	-
Repeat	-	v	v
NonStop Repeat	-	-	v
End	-	-	-

Table 1 Command Types vs Editor Columns

"v" *means that you must give an appropriate choice or value; otherwise, an "Command Error" message will appear.*
"-" *means that this column is not used for this command.*

Additional Notes:

1) Each table must have at least the following 4 commands appearing in the order they are shown below (otherwise, an "Command Error" will appear):

   Set Current Limit
   Set Voltage Level
   Output On
   End

2) Be careful of the power limit of 60 watts for each channel in 52912/52914.

3) For additional notes on sequencer tables, please refer to **Section 5.1.11 *"chr52912_ConfigureSequencerTable"***.

4.3.6.2 Using Sequencer Editor

At the menu bar, select **Editor →Sequencer**. Then, a dialog box titled "**Sequencer Editor -> PXI[board]::<logical address>::INSTR**" will be launched as below:

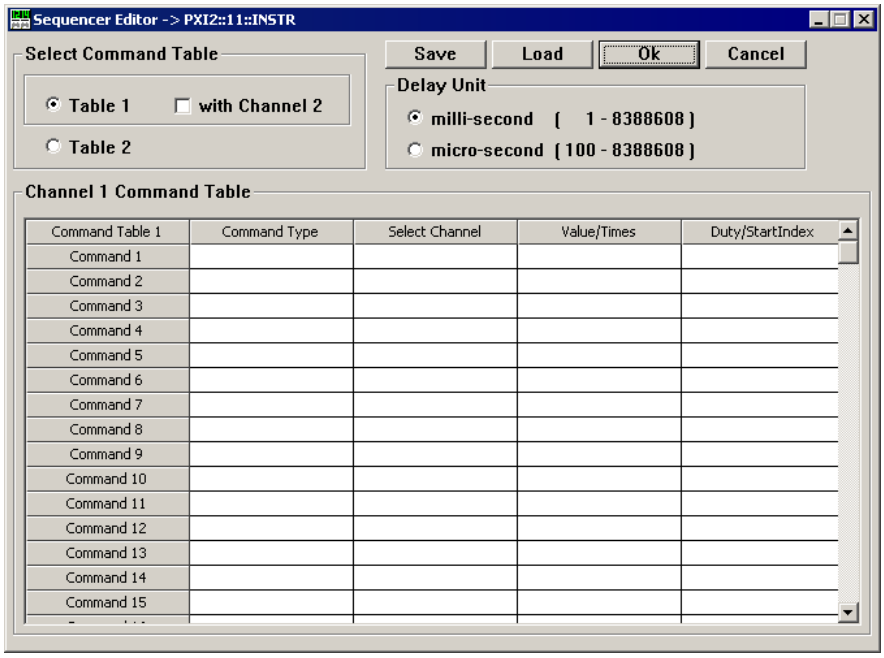


Figure 4-16 Sequencer Editor

There are two configurable sequencer tables, Table 1 & Table 2. Usually, Table 1 is used for channel 1, and Table 2 for channel 2, respectively. However, it is also possible to control both channels in Table 1. Just simply check the "with Channel 2" box and then you will have the options of "Channel 1 and 2", "Channel 1 only" and "Channel 2 only" in the "**Select Channel**" column. Only Table 1 can be used to control both channels but Table 2 cannot.

Click on "OK" to confirm when you have finished editing the table(s).

4.3.6.3 Running Sequencer Mode

Once you have already finished editing the sequencer table(s) and configured 52912/52914 same as Section 4.3.5.1 to set up the Start trigger. In this mode, only Start trigger you need to select. That means 52912/52914 auto starts to run sequential commands if it received the specified trigger. Then it will stop runs of

specified channel if executes an "End" command or never stop if executes a "NonStop Repeat" command.
If you configured a "**Sequencer with trigger**" mode, the skin of soft control panel changed as Figure 4-17.

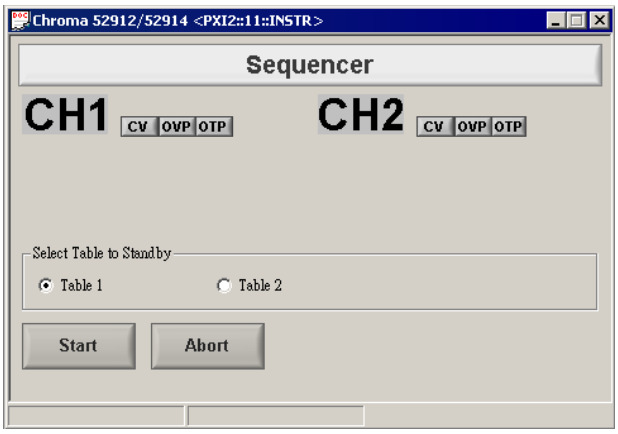


Figure 4-17 Control Panel for Sequencer with trigger Mode

At first, selecting which Table you want to manipulate. Notice that the control panel allows 52912/52914 to run Table 1 with Channel 2, and only one Table can be run at one time.

To execute in this mode, click on the 

button and the 52912/52914 will start to function. It will wait for the (Start Trigger) signal, and as soon as it receives the signal, it starts to execute the commands in the selected sequencer table. The 52912/52914 stops when it comes to execute the "End" command, or will never stop as it comes to execute the "NonStop Repeat" command in the sequencer table.

***CAUTION*: While editing the command table, if the time to switch between CV and CC is less than 0.5 sec., The light shown in the Status Indicator might not be the actual current status.**

4.3.6.4 Examples

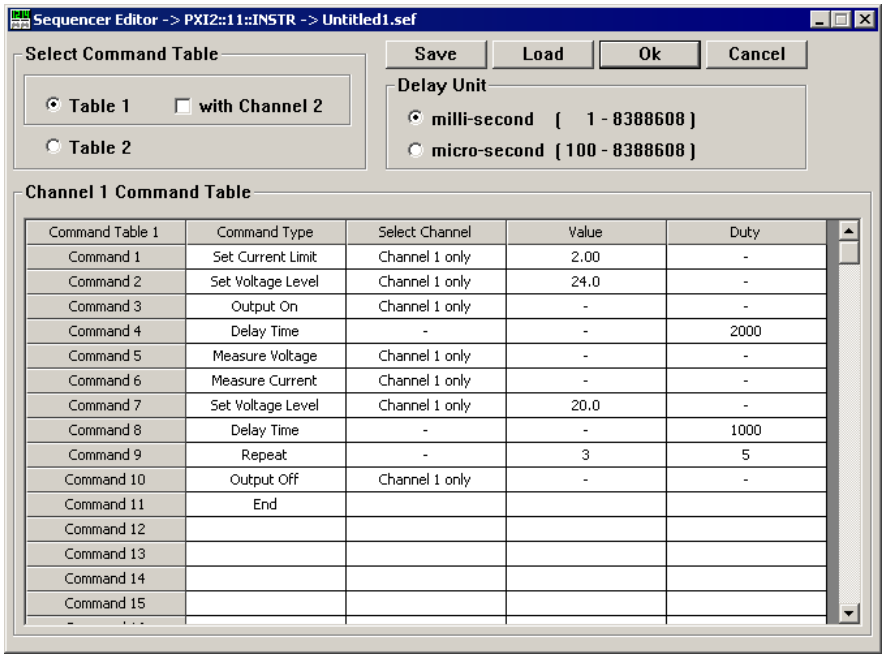
Here are two examples, one with Table 1 used to control channel 1 only, and the other with Table 1 controlling both channels. The loads for each channel in these examples are both 30Ω(50 watts). Be aware of the power limit of 60 watts in

your own practice.

Example 1: **Table 1 Controlling Channel 1 Only with Software trigger**

Step 1: At the menu bar, select **View →Configure Mode Trigger Bar**. Choose the 52912/52914 you want to manipulate, and then under the (Select Mode) scrollbox, select "**Sequencer**". Under the (Start Trigger) scrollbox, select Software. At last, select the slope type of the triggers, and then click on the button to confirm.

Step2: At the menu bar, select **Editor→Sequencer** to edit the sequencer table as in Figure 4-18, and then click on the "Ok" button. (For more details on editing the sequencer table, please refer to **Section 4.3.6.2 *"Using Sequencer Editor"***)



Command Table 1	Command Type	Select Channel	Value	Duty
Command 1	Set Current Limit	Channel 1 only	2.00	-
Command 2	Set Voltage Level	Channel 1 only	24.0	-
Command 3	Output On	Channel 1 only	-	-
Command 4	Delay Time	-	-	2000
Command 5	Measure Voltage	Channel 1 only	-	-
Command 6	Measure Current	Channel 1 only	-	-
Command 7	Set Voltage Level	Channel 1 only	20.0	-
Command 8	Delay Time	-	-	1000
Command 9	Repeat	-	3	5
Command 10	Output Off	Channel 1 only	-	-
Command 11	End			
Command 12				
Command 13				
Command 14				
Command 15				

Figure 4-18 Sequencer Table of Example 1

Step3: On the Control Panel, choose Table 1. Then, click on the 

button. The 52912/52914 now start to wait software trigger. Then click on the 

button to start to run Table1. After the 52912/52914 receives the necessary start trigger, it runs through all the commands in Table 1, one by one. Note that the output voltages and/or currents will change according to the commands. As the 52912/52914 is outputting voltages/ currents, the LED of channel 1 in the front panel of 52912/52914 will light in green if Table1 have "Output On" command. And when the 52912/52914 finishes executing the commands of Table 1.

The  button is now available. Click on the  button to view the measured data if you have the commands, "**Measure Voltage**" and/or "**Measure Current**", in Table 1.

Figure 4-19 is what you will see when you click on the  button after the 52912/52914 stops from executing the sequencer table in example 1.

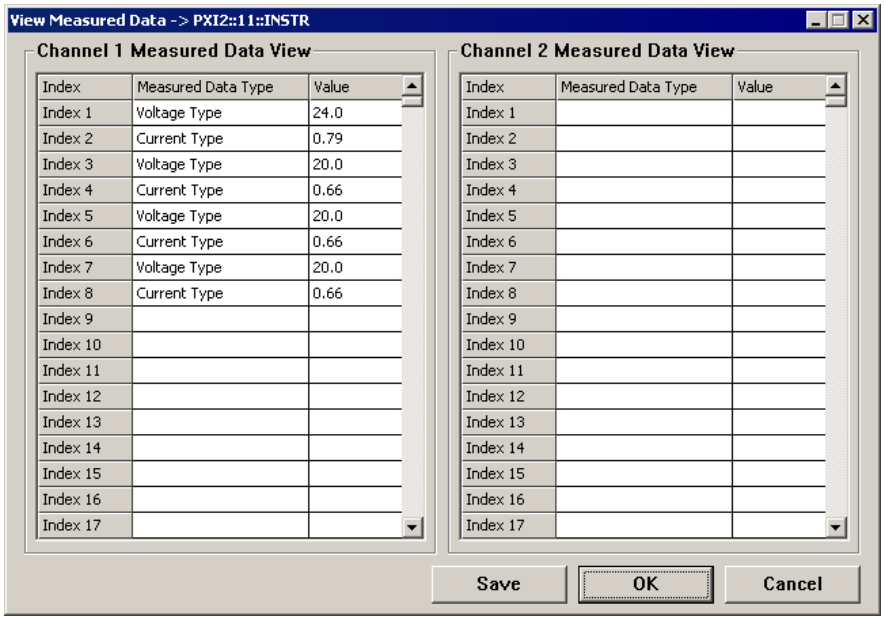
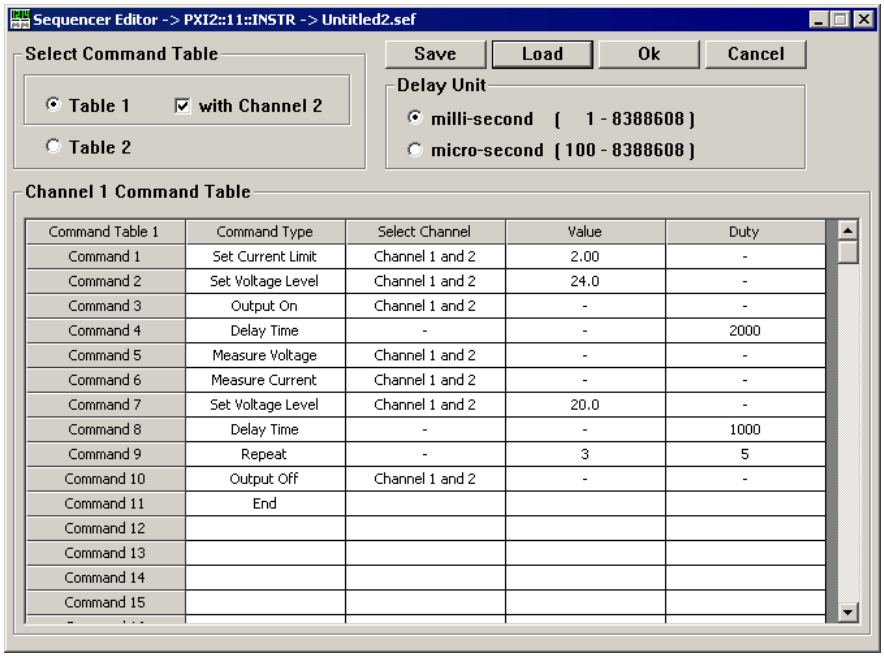


Figure 4-19 Viewing Measured Data of Example 1

Example 2: Table 1 with channel 2 control with software trigger

Step1: Same as Step1 in Example 1.

Step2: At the menu bar, select **Editor→Sequencer** to edit the sequencer table as in **Figure 4-16**, and then click on the "Ok" button. (For more details on editing the sequencer table, please refer to **Section 4.3.6.2 *"Using Sequencer Editor"***)



Command Table 1	Command Type	Select Channel	Value	Duty
Command 1	Set Current Limit	Channel 1 and 2	2.00	-
Command 2	Set Voltage Level	Channel 1 and 2	24.0	-
Command 3	Output On	Channel 1 and 2	-	-
Command 4	Delay Time	-	-	2000
Command 5	Measure Voltage	Channel 1 and 2	-	-
Command 6	Measure Current	Channel 1 and 2	-	-
Command 7	Set Voltage Level	Channel 1 and 2	20.0	-
Command 8	Delay Time	-	-	1000
Command 9	Repeat	-	3	5
Command 10	Output Off	Channel 1 and 2	-	-
Command 11	End			
Command 12				
Command 13				
Command 14				
Command 15				

Figure 4-20 Sequencer Table of Example 2

Step3: On the Control Panel, choose Table 1. Then click on the Start button. The 52912/52914 now starts to wait for a software trigger. Click on the Send Software Trigger button will make 52912/52914 start to run Table 1. After the 52912/52914 receives the necessary start trigger, it runs through all the commands in Table 1, one by one. Note that the output voltages and/or currents will change according to the commands. As the 52912/52914 is outputting voltages/ currents, the LED of channel 1 in the front panel of 52912/52914 will light in green if Table1 have "Output On" command. And when the 52912/52914 finishes executing the commands of Table 1.

The button is now available. Click on the button to view the measured data if you have the commands, "**Measure Voltage**" and/or "**Measure Current**", in Table 1.

Figure 4-21 is what you will see when you click on the  button after the 52912/52914 stops from executing the sequencer table in example 2. Note that voltage/current for each channel can be measured and shown separately in Figure 4-21.

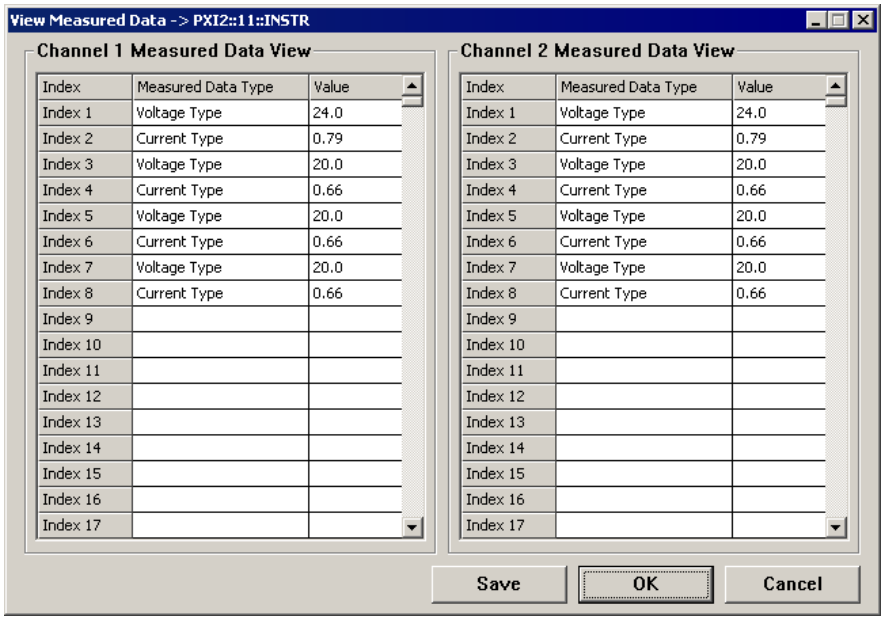


Figure 4-21 Viewing Measured Data of Example 2

4.3.7 Introduction of Measure Mode

A 52912/52914 can also serve as an instrument which will measure the value of output periodically. The time interval between every measurement is adjustable.

The 52912/52914 in this mode is designate to measure the amount of output in the

specified channel. Thus, it cannot be used as a meter. If the 52912/52914 is set to measure the output of channel 1, then you can't use it to measure that of channel 2.

4.3.7.1 Using Measure Attributes Editor

Before using the Measure Attributes Editor, please refer to **Section 5.1.14 *"chr52912_ConfigureMeasure"*** for setting up the 6 parameters.

The Measure Attributes Editor dialog box will be shown as follow:

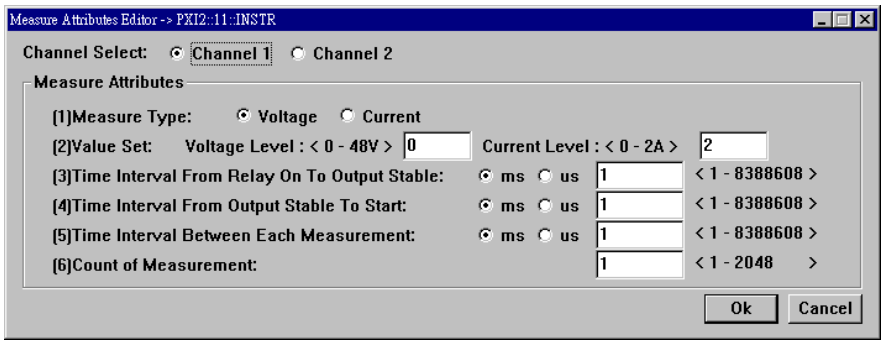


Figure 4-22 Measure Attributes Editor

This editor manipulates the currently-active 52912/52914 only. At first, choose the channel you want to control, i.e. channel 1 or 2. Then, fill in the appropriate Measure Attributes from (1) to (6).

After the "Ok" button is clicked, you are now in the Measure Mode (see Figure 4-23). Select the channel and click on the Standby button. This will bring the 52912/52914 into the waiting state. If the (Start Trigger) is set to be "Software", a Send Software Trigger button will show up to let you toggle. If the (Start Trigger) is set to be anything else, 52912/52914 will wait for the actual signal sent by the hardware, and then start measuring. All the measurements will cease after the 52912/52914 runs "(6)Count of Measurement" times.

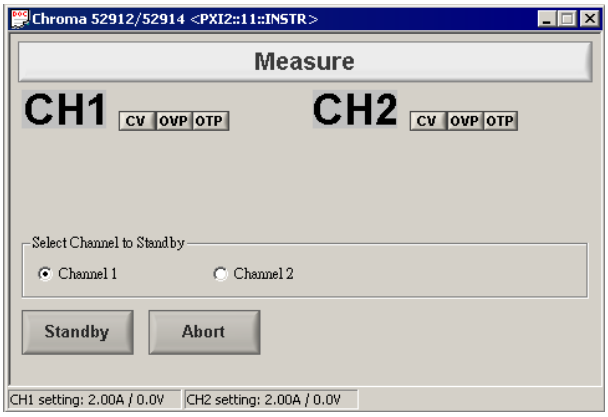


Figure 4-23 Control Panel for Measure with trigger Mode

Note:

1) After the "Ok" button is clicked while editing Measure Attributes, the editor will check if all the entry values are within the range. If any error is detected, a corresponding error message will appear. Please refer to **Section 4.3.8** "***List of Messages"*** for a list of all the error messages.

2) Each time when a new Measure Attribute is edited and run, the measurements has to be done on the same channel throughout this process.

4.3.7.2 Running Measure Mode

Standby *Standby to Measure:* clicking on this icon will engage currently-active 52912/52914 to be ready for receiving a trigger signal to measure channel 1's output. This measurement depends on you have already edited Measure Attributes via "**Measure Attributes Editor**". (For more details on "**Measure Attributes Editor**", please refer to **Section 4.3.7.1 *"Using Measure Attributes Editor"***)

Abort *Abort Measurement:* clicking on this icon will abort the measuring.

Once you have finished editing "**Measured Attributes Editor**", click on the

Standby button and the 52912/52914 will start to function. It will wait for the (Start Trigger) signal, and as soon as it receives the signal, it starts to perform periodical measurements on the specified channel. The 52912/52914 will cease after it runs the total count of measurements times.

4.3.7.3 Examples

Here are two examples, one is edited with voltage measurement on channel 1, and the other is edited with current measurement on channel 2. The loads for each channel in these examples are both 30Ω(50 watts). Be aware of the power limit of 60 watts in your own practice.

Example 1: Voltage Measurement on Channel 1

Step 1: At the menu bar, select **View →Configure Mode Trigger Bar**. Choose the 52912/52914 you want to manipulate, and then under the (Select Mode) scrollbox, select "**Measure**". Then, under the (Start Trigger) scrollbox, select Software. At last, select the slope type of the trigger, and then click on the button to confirm.

Step2: At the menu bar, select **Editor→Measure** to edit the attributes for this measurement set as shown in Figure 4-24, and then click on the "Ok" button.

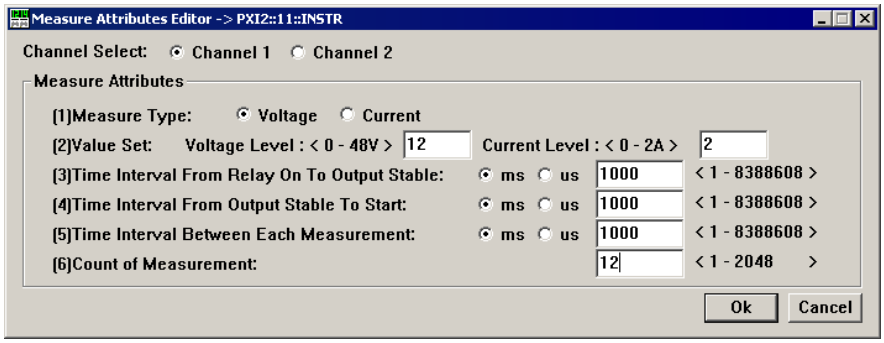


Figure 4-24 Editing Voltage Measurement of Channel 1 for Example 1

Step3: On the control panel, click on Standby, and after the 52912/52914 receives the necessary start trigger, it will start to perform periodical measurements on channel 1. As the 52912/52914 is outputting voltages/ currents, the LED of channel 1 in the front panel of 52912/52914 will light in green. And after the 52912/52914 runs the total count of measurements times, it will cease and put out the LED.

After the LED is off, you can click the button to view the measured data and Figure 4-25 is what you will see when clicking on the button.

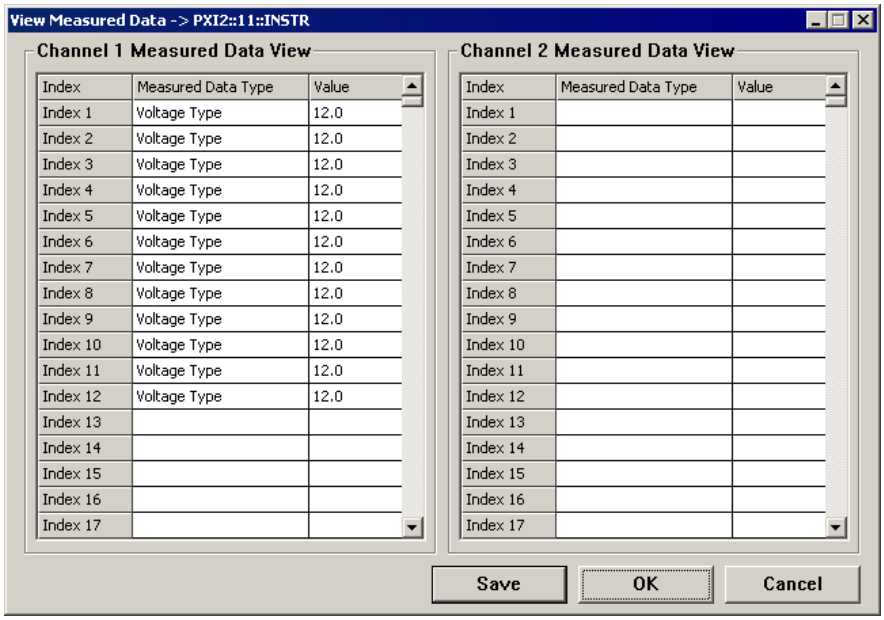


Figure 4-25 Viewing Measured Voltage Data of Example 1

Example 2: Current Measurement of Channel 2

Step1: Same as Step 1 in Example 1.

Step2: At the menu bar, select **Editor→Measure** to edit the attributes for this measurement set as shown in Figure 4-26, and then click on the "Ok" button.

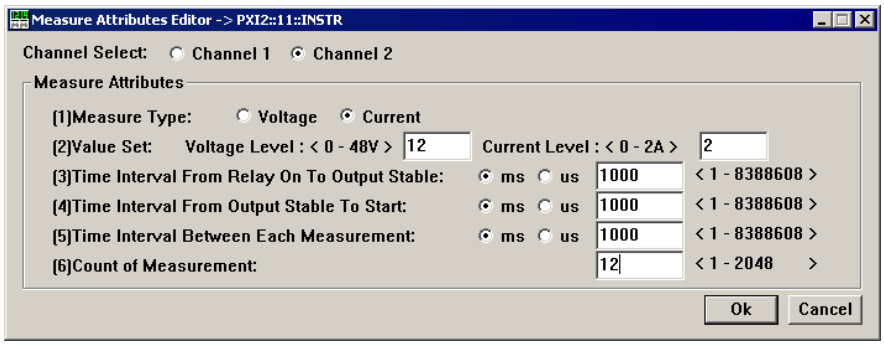


Figure 4-26 Editing Current Measurement of Channel 2 for Example 2

Step3: On the control panel, click on 

, and after the 52912/52914 receives the necessary start trigger, it will start to perform periodical measurements on channel 2. As the 52912/52914 is outputting voltages/ currents, the LED of channel 1 in the front panel of 52912/52914 will light in green. And after the 52912/52914 runs the total count of measurements times, it will cease and put out the LED.

After the LED is off, you can click the  button to view the measured data, and Figure 4-27 is what you will see after clicking on the button.

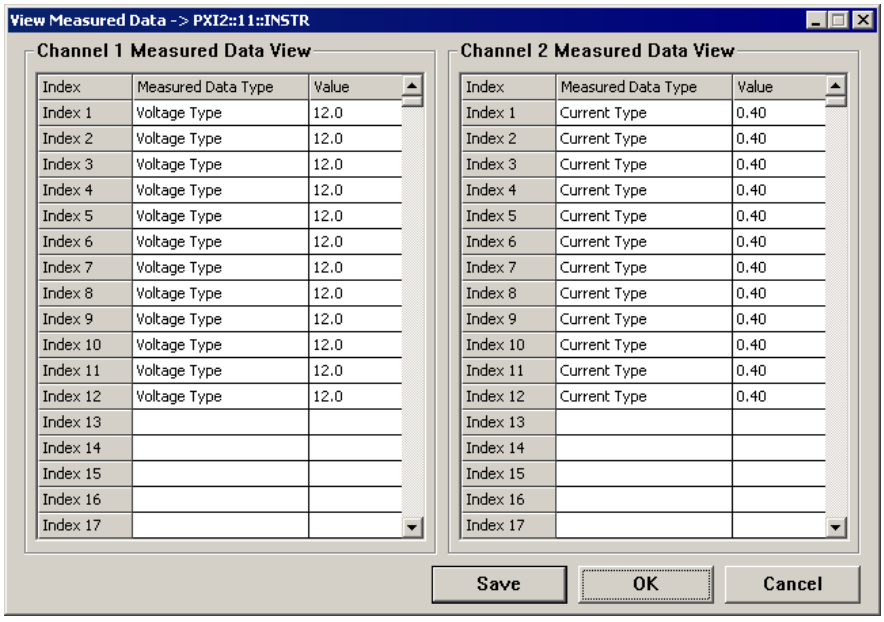


Figure 4-27 Viewing Measured Current Data of Example 2

4.3.8 List of Messages

All possible messages that will appear in the message box are listed in the following table. For warnings and errors of NI-VISA® and NI-IVI®, please refer to **section 5.2**.

Title	Message	Description
SFP52912/52914	SFP5291214 Finds Resources error! Please check your system.	Program cannot find the hardware. Please insert card or check its seating.
SFP52912/52914	<PXI?::??::INSTR>Are you sure to exit?	A warning prompt before exiting the selected 52912/52914.
SFP52912/52914	Channel 1 of < PXI?::??::INSTR> has caused: Error Code : 0xbffa4005 Error Message : There is no 56V input preset in this channel. Please check your system! After checking your system. Do you want to reset output?	An error occurred after toggling the Output button. This message shows that Channel 1 of < PXI?::??::INSTR> has no input present. You must check whether an input of +56Vdc is available or not.
SFP52912/52914	PXI?::??::INSTR Cannot initial Power	May resource is not present in the system.
SFP52912/52914	Channel 1 of <PXI?::??::INSTR> has Status Error occurred Status : 0x4C Please check your system!	An error occurred when measure. (For more details on this., please refer to **section5.1.24** ***"chr52912_error_query"***.)
Cwui	Type Mismatch	Input data is not decimal figures.
SFP52912/52914	Caution: Your set over power limit<60 Watts>	The multiplied result of Current and Voltage is over the power limit

SFP52912/52914	Channel 1: Read Measured Data Error.	Not edited measure attributes for channel1
SFP52912/52914	Channel 2: Read Measured Data Error.	Not edited measure attributes for channel2
SFP52912/52914	There is no measured data in both channels.	No measured data stored in buffer of both channels at currently active 52912/52914.
SFP52912/52914	Channel 1 Running.	Channel 1 is still running. After 52912/52914 finishing sequential commands, you just take more actions.
SFP52912/52914	Channel 2 Running.	Channel 2 is still running. After 52912/52914 finishing sequential commands, you just take more actions.
Configure	Please turn 52912/52914 off first.	May a different operation mode will be changed, SFP5291214 show you need to turn power off first for coming change.
Configure	Configuration Changed Ok.	A different operation changed Ok.
Select Trigger	Please select a valid start trigger.	Selecting a valid start trigger.
Select Mode	Please select a valid mode.	Selecting a valid mode.
Command Error	The voltage level exceeds the valid range.	You must enter a valid value of range from 0V to 48V.
Command Error	The value of the delay is invalid.	You must enter a valid value of range from 1 to 8388608 in millisecond or range from 100 to 8388608 in microsecond.
Command Error	The repeat Start Index can't be the index of repeat command itself.	The Start Index of **Repeat** or **NonStop Repeat** command should less than the location of itself.
Command Error	The repeat index is invalid.	The Start Index of **Repeat** or **NonStop Repeat** command is greater then the location of itself occurred, see above description.
Command Error	The repeat times exceeds the valid range.	The value should only from 0 to 524288.

Command Error – Table *	The current limit is invalid at Command #	The value of column of Set Current Limit is invalid at the location of command # in Table*.
Command Error – Table *	The voltage level is invalid at Command #	The value of column of Set Voltage Level is invalid at the location of command # in Table*.
Command Error – Table *	The value of the delay is invalid at Command #	The value of column of Delay Time unit is invalid at the location of command # in Table*.
Command Error – Table *	There is no End command in the sequencer table.	The Sequencer Editor will recognize the End command existed or not for confirming table available, it's a must command at the last you select.
Command Error – Table *	There is over one End command in this table.	Only one End command can be existed in a sequencer table.
Command Error – Table *	Table size is less than 4.	For more details, please refer to section 4.3.6.1
Command Error – Table *	There exists unknown command at Command # .	There exists unknown command at Command # in Table*.
Command Error – Table *	There is no Output On command in the sequencer table.	There is no Output On command in Table*. This is warning you that the output of 52912/52914 won't turn the output relay on ever.
Command Error – Table *	The channel selected is invalid at Command #.	You didn't select a valid channel of column of Select Channel at Command # in Table*.
Command Error – Table *	There is over one NonStop Repeat command in this table.	A Sequencer table must only hold one NonStop Repeat command in Table*.
Command Error – Table *	Exceed power limit. To set this current limit, lower voltage level first at Command#.	Please check the power limit at Command# in Table*, the maximum power is 60 watts.
Command Error – Table *	There is no Set Current Limit command before Output On command. This check is a protection for the circuit.	Please check whether a valid command of Set Current Limit before the Output On command in Table*.

Command Error – Table *	There is no Set Voltage Level command before Output On command. This check is a protection for the circuit.	Please check whether a valid command of Set Voltage Level before the Output On command in Table*.
Command Error – Table *	The nested layer of repeat can't be bigger than 7 at Command#.	Please check your table about the nested layer of repeat, 52912/52914 can only run 7 layers with overlapping a command.
Edit Measure Error	Your set over power limit<60 Watts>. SFP5291214 will auto change your set value.	The product of voltage and current value is over 60 watts.
Measure	Channel 1: Not Edited Measure Attributes existed, please edit Measure Attributes first.	Please edit the measure attributes for channel 1 first.
Measure	Channel 2: Not Edited Measure Attributes existed, please edit Measure Attributes first.	Please edit the measure attributes for channel 2 first.
Configuring Sequencer Error	The Table1 is not available, please edit Table1 first.	Please use Sequencer Editor to edit Table 1 first.
Configuring Sequencer Error	The Table2 is not available, please edit Table2 first.	Please use Sequencer Editor to edit Table 2 first.

Note:

* , Table 1 or Table 2

#, the index of command in the table

5. DLL Calls and Examples

5.1 DLL Calls

5.1.1 chr52912_init

ViStatus chr52912_init (ViRsrc resourceName,
ViBoolean IDQuery,
ViBoolean resetDevice,
ViPSession instrumentHandle);

Purpose

This function performs the following initialization actions:
- Creates a new IVI instrument driver session.
- Opens a session to the specified device using the interface and address you specify for the Resource Name parameter.
- If the ID Query parameter is set to VI_TRUE, this function queries the instrument ID and checks that it is valid for this instrument driver.
- If the Reset parameter is set to VI_TRUE, this function resets the instrument to a known state.
- Sends initialization commands to set the instrument to the state necessary for the operation of the instrument driver.
- Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.

Note: This function creates a new session each time you invoke it. Although you can open more than one IVI session for the same resource, it is best not to do so. You can use the same session in multiple program threads. You can use the chr52912_LockSession and chr52912_UnlockSession functions to protect sections of code that require exclusive access to the resource.

Parameter List

resourceName

Variable Type ViRsrc

Pass the resource name of the device to initialize.

You can also pass the name of a virtual instrument or logical name that you configure with the IVI Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical Name identifies a particular virtual instrument.

Refer to the following table below for the exact grammar to use for this parameter. Optional fields are shown in square brackets ([]).

Syntax

PXI[board]::<logical address>::INSTR

If you do not specify a value for an optional field, the following values are used:

Optional Field - Value

---
board - 0
secondary address - none (31)

The following table contains example valid values for this parameter.

"Valid Value" - Description

"PXI::64::INSTR" - VXI board 0, logical address 64
"PXI1::64::INSTR" - VXI board 1, logical address 64

Default Value: "PXI::10::INSTR"

Note: You specify the resource name with the "VInstr->" if you have the logical name that is the same as the virtual instrument name and you want to explicitly use the virtual instrument name. Otherwise, the driver uses the logical name.

IDQuery
Variable Type ViBoolean
Specify whether you want the instrument driver to perform an ID Query.
Valid Range:
VI_TRUE (1) - Perform ID Query (Default Value)
VI_FALSE(0) - Skip ID Query

resetDevice
Variable Type ViBoolean
Specify whether you want the to reset the instrument during the initialization procedure.

Valid Range:
VI_TRUE (1) - Reset Device (Default Value)
VI_FALSE (0) - Don't Reset

instrumentHandle

Variable Type ViSession (passed by reference)
Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.
Notes:
(1) This function creates a new session each time you invoke it. This is useful if you have multiple physical instances of the same type of instrument.
(2) Avoid creating multiple concurrent sessions to the same physical instrument. Although you can create more than one IVI session for the same resource, it is best not to do so. A better approach is to use the same IVI session in multiple execution threads. You can use functions chr52912_LockSession and chr52912_UnlockSession to protect sections of code that require exclusive access to the resource.

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.2 chr52912_InitWithOptions

ViStatus chr52912_InitWithOptions (ViRsrc resourceName,
ViBoolean IDQuery,
ViBoolean resetDevice,
ViString optionString,
ViPSession instrumentHandle);

Purpose

This function performs the following initialization actions:

- Creates a new IVI instrument driver and optionally sets the initial state of the following session attributes:
  - CHR52912_ATTR_RANGE_CHECK
  - CHR52912_ATTR_QUERY_INSTR_STATUS
  - CHR52912_ATTR_CACHE
  - CHR52912_ATTR_SIMULATE
  - CHR52912_ATTR_RECORD_COERCIONS
- Opens a session to the specified device using the interface and address you specify for the Resource Name parameter.
- If the ID Query parameter is set to VI_TRUE, this function queries the instrument ID and checks that it is valid for this instrument driver.
- If the Reset parameter is set to VI_TRUE, this function resets the instrument to a known state.
- Sends initialization commands to set the instrument to the state necessary for the operation of the instrument driver.
- Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.

Note: This function creates a new session each time you invoke it. Although you can open more than one IVI session for the same resource, it is best not to do so. You can use the same session in multiple program threads. You can use the chr52912_LockSession and chr52912_UnlockSession functions to protect sections of code that require exclusive access to the resource.

Parameter List

resourceName

Variable Type ViRsrc

Pass the resource name of the device to initialize.

You can also pass the name of a virtual instrument or logical name that you configure with the IVI Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical Name identifies a particular virtual instrument.

IDQuery
Variable Type ViBoolean
Specify whether you want the instrument driver to perform an ID Query.
Valid Range:
VI_TRUE (1) - Perform ID Query (Default Value)
VI_FALSE(0) - Skip ID Query

resetDevice
Variable Type ViBoolean
Specify whether you want the to reset the instrument during the initialization procedure.
Valid Range:
VI_TRUE (1) - Reset Device (Default Value)
VI_FALSE (0) - Don't Reset

optionString
Variable Type ViString
You can use this control to set the initial value of certain attributes for the session. The following table lists the attributes and the name you use in this parameter to identify the attribute.

Name	Attribute Defined Constant
RangeCheck	CHR52912_ATTR_RANGE_CHECK
QueryInstrStatus	CHR52912_ATTR_QUERY_INSTR_STATUS
Cache	CHR52912_ATTR_CACHE
Simulate	CHR52912_ATTR_SIMULATE
RecordCoercions	CHR52912_ATTR_RECORD_COERCIONS

The format of this string is, "AttributeName=Value" where AttributeName is the name of the attribute and Value is the value to which the attribute will be set. To set multiple attributes, separate their assignments with a comma.
If you pass NULL or an empty string for this parameter and a VISA resource descriptor for the Resource Name parameter, the session uses the default values for the attributes. The default values for the attributes are shown below:

Attribute Name	Default Value
RangeCheck	VI_TRUE
QueryInstrStatus	VI_TRUE

Cache	VI_TRUE
Simulate	VI_FALSE
RecordCoercions	VI_FALSE

You can override the values of the attributes by assigning a value explicitly in a string you pass for this parameter. You do not have to specify all of the attributes and may leave any of them out. If you do not specify one of the attributes, its default value or the value that you configure with the IVI Configuration utility will be used.

The following are the valid values for ViBoolean attributes:
True: 1, TRUE, or VI_TRUE
False: 0, False, or VI_FALSE

Default Value:
"Simulate=0,RangeCheck=1,QueryInstrStatus=1,Cache=1"

instrumentHandle

Variable Type ViSession (passed by reference)
Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.
Notes:
(1) This function creates a new session each time you invoke it. This is useful if you have multiple physical instances of the same type of instrument.
(2) Avoid creating multiple concurrent sessions to the same physical instrument. Although you can create more than one IVI session for the same resource, it is best not to do so. A better approach is to use the same IVI session in multiple execution threads. You can use functions chr52912_LockSession and chr52912_UnlockSession to protect sections of code that require exclusive access to the resource.

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.3 chr52912_close

ViStatus chr52912_close (ViSession instrumentHandle);

Purpose

This function performs the following operations:
- Closes the instrument I/O session.
- Destroys the instrument driver session and all of its attributes.
- Deallocates any memory resources the driver uses.

Notes:
(1) You must unlock the session before calling chr52912_close.
(2) After calling chr52912_close, you cannot use the instrument driver again until you call chr52912_init or chr52912_InitWithOptions.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.4 chr52912_ConfigureOperationMode

ViStatus chr52912_ConfigureOperationMode (ViSession instrumentHandle, ViInt32 operationMode);

Purpose

This function configures the operation mode of the power supply. There are three operation modes: DC Power, Sequencer and Measure.

-DC Power Mode
Supports functionalities that traditional power supplies provide. The power supply can also wait for trigger signals. As soon as the start trigger signal occurs, the power supply will start to generate output. Until receiving the stop trigger signal, the power supply will stop outputting.

-Sequencer Mode
Provides two configurable sequencer tables in the power supply. When the start trigger signal occurs, the power supply sequentially executes the command sets in the sequencer table, line by line. Refer to chr52912_ConfigureSequencerTable for details on configuring a sequencer table.

-Measure Mode
Provides periodical measurements on output. Precise time intervals between every two measurements are configurable. Refer to chr52912_ConfigureMeasure for details on configurations in the Measure mode.

Parameter List

instrumentHandle

Variable Type ViSession
The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

operationMode

Variable Type ViInt32
The operation mode of the power supply.

Valid Values:
CHR52912_VAL_DC_POWER
CHR52912_VAL_SEQUENCER_TABLE

CHR52912_VAL_SOURCE_METER

Default value: CHR52912_VAL_DC_POWER

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.5 chr52912_ConfigureCurrentLimit

ViStatus chr52912_ConfigureCurrentLimit (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViInt32 behavior,
ViReal64 limit);

Purpose

This function configures the current limit. You specify the output current limit value and the behavior of the power supply when the output current is greater than or equal to that value.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel where the current limit is configured.
Valid Channel Names: ch1,ch2

behavior

Variable Type ViInt32

Pass the behavior you want the power supply to exhibit when the output current is greater than or equal to the value of the limit parameter. The driver uses this value to set the CHR52912_ATTR_CURRENT_LIMIT_BEHAVIOR attribute.

Valid Values:
CHR52912_VAL_CURRENT_REGULATE - Regulatory limit

limit

Variable Type ViReal64

Pass the current limit you want to use. The driver uses this value to set the CHR52912_ATTR_CURRENT_LIMIT attribute.

Units: amps
Valid Range: 0.0 - 2.0 amps (depends on current voltage level)

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.6 chr52912_ConfigureVoltageLevel

ViStatus chr52912_ConfigureVoltageLevel (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViReal64 level);

Purpose

This function configures the DC voltage level the power supply attempts to generate. In order to protect power supply, if you set voltage level higher than 30V, it will automatically adjust current limit to comply with maximal power output (60W).

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel where the voltage level is configured.
Valid Channel Names: ch1,ch2

level

Variable Type ViReal64

Pass the DC voltage you want the power supply to attempt to generate. The driver uses this value to set the CHR52912_ATTR_VOLTAGE_LEVEL attribute.

Units: volts
Valid Range: 0.0 - 48.0 volts

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.7 chr52912_ConfigureOutputEnabled

ViStatus chr52912_ConfigureOutputEnabled (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViBoolean enabled);

Purpose

Configures whether the signal that the power supply produces on a channel appears at the output connector.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel you want to enable or disable.

Valid Channel Names: ch1,ch2

enabled

Variable Type ViBoolean

Pass whether you want the signal the power supply produces on a channel to appear at the output connector. The driver uses this value to set the CHR52912_ATTR_OUTPUT_ENABLED attribute.

Valid Values:

VI_TRUE - Enable the output

VI_FALSE - Disable the output

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.8 chr52912_ConfigureTriggerSource

ViStatus chr52912_ConfigureTriggerSource (ViSession instrumentHandle,
ViInt32 stratSource,
ViInt32 stopSource,
ViInt32 slope);

Purpose

Configures the start and stop trigger source. The power supply will wait for the start trigger signal before executing any command set in the sequencer table. In different operation modes, appropriate trigger source(s) will have to be configured. Refer to the following paragraphs for details on triggers in each mode.

In the DC power supply operation mode, you must configure both trigger sources. The power supply will not generate output until it receives a start trigger signal, and won't stop until the stop tgigger signal.

In Sequencer and Measure operation modes, only the start trigger source is enabled. The stop trigger source must be None. When the power supply receives a start trigger signal, it will execute the command sets in the sequencer table, line by line.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

stratSource

Variable Type ViInt32

Pass the start trigger source that the power supply will wait for before executing the command sets. The driver will set the CHR52912_ATTR_START_TRIGGER_SOURCE attribute to this given value.

Defined Values:

CHR52912_VAL_TRIG_EXTERNAL - External
CHR52912_VAL_SOFTWARE_TRIG - Software Trigger (Default)
CHR52912_VAL_TRIG_TTL0 - PXI TRIG0 or VXI TTL0

CHR52912_VAL_TRIG_TTL1	- PXI TRIG1 or VXI TTL1
CHR52912_VAL_TRIG_TTL2	- PXI TRIG2 or VXI TTL2
CHR52912_VAL_TRIG_TTL3	- PXI TRIG3 or VXI TTL3
CHR52912_VAL_TRIG_TTL4	- PXI TRIG4 or VXI TTL4
CHR52912_VAL_TRIG_TTL5	- PXI TRIG5 or VXI TTL5
CHR52912_VAL_TRIG_TTL6	- PXI TRIG6 or VXI TTL6
CHR52912_VAL_TRIG_TTL7	- PXI TRIG7 or VXI TTL7

Notes:

(1) CHR52912_VAL_TRIG_EXTERNAL
External Trigger. The power supply waits until it receives a trigger from an external source through the "trigger in" connector.

(2) CHR52912_VAL_SOFTWARE_TRIG
The power supply waits until you call the chr52912_SendSoftwareTrigger function.

(3) CHR52912_VAL_TRIG_TTL0
The power supply waits for a trigger on the PXI TRIG0 line (for PXI instruments) or the VXI TTL0 line (for VXI instruments).

(4) CHR52912_VAL_TRIG_TTL1
The power supply waits for a trigger on the PXI TRIG1 line (for PXI instruments) or the VXI TTL1 line (for VXI instruments).

(5) CHR52912_VAL_TRIG_TTL2
The power supply waits for a trigger on the PXI TRIG2 line (for PXI instruments) or the VXI TTL2 line (for VXI instruments).

(6) CHR52912_VAL_TRIG_TTL3
The power supply waits for a trigger on the PXI TRIG3 line (for PXI instruments) or the VXI TTL3 line (for VXI instruments).

(7) CHR52912_VAL_TRIG_TTL4
The power supply waits for a trigger on the PXI TRIG4 line (for PXI instruments) or the VXI TTL4 line (for VXI instruments).

(8) CHR52912_VAL_TRIG_TTL5
The power supply waits for a trigger on the PXI TRIG5 line (for PXI instruments) or the VXI TTL5 line (for VXI instruments).

(9) CHR52912_VAL_TRIG_TTL6
The power supply waits for a trigger on the PXI TRIG6 line (for PXI instruments) or the VXI TTL6 line (for VXI instruments).

(10) CHR52912_VAL_TRIG_TTL7
The power supply waits for a trigger on the PXI TRIG7 line (for PXI instruments) or the VXI TTL7 line (for VXI instruments).

stopSource
Variable Type ViInt32

Pass the stop trigger source that the power supply will wait for before stopping. The driver will set the CHR52912_ATTR_STOP_TRIGGER_SOURCE attribute to this given value.

Defined Values:

CHR52912_VAL_TRIG_NONE - No Trigger
CHR52912_VAL_TRIG_EXTERNAL - External
CHR52912_VAL_SOFTWARE_TRIG - Software Trigger (Default)
CHR52912_VAL_TRIG_TTL0 - PXI TRIG0 or VXI TTL0
CHR52912_VAL_TRIG_TTL1 - PXI TRIG1 or VXI TTL1
CHR52912_VAL_TRIG_TTL2 - PXI TRIG2 or VXI TTL2
CHR52912_VAL_TRIG_TTL3 - PXI TRIG3 or VXI TTL3
CHR52912_VAL_TRIG_TTL4 - PXI TRIG4 or VXI TTL4
CHR52912_VAL_TRIG_TTL5 - PXI TRIG5 or VXI TTL5
CHR52912_VAL_TRIG_TTL6 - PXI TRIG6 or VXI TTL6
CHR52912_VAL_TRIG_TTL7 - PXI TRIG7 or VXI TTL7

Notes:

(0) CHR52912_VAL_TRIG_NONE
The power supply does not produce a Scan Advanced Output trigger.

(1) CHR52912_VAL_TRIG_EXTERNAL
External Trigger. The power supply waits until it receives a trigger from an external source through the "trigger in" connector.

(2) CHR52912_VAL_SOFTWARE_TRIG
The power supply waits until you call the

chr52912_SendSoftwareTrigger function.

(3) CHR52912_VAL_TRIG_TTL0
The power supply waits for a trigger on the PXI TRIG0 line (for PXI instruments) or the VXI TTL0 line (for VXI instruments).

(4) CHR52912_VAL_TRIG_TTL1
The power supply waits for a trigger on the PXI TRIG1 line (for PXI instruments) or the VXI TTL1 line (for VXI instruments).

(5) CHR52912_VAL_TRIG_TTL2
The power supply waits for a trigger on the PXI TRIG2 line (for PXI instruments) or the VXI TTL2 line (for VXI instruments).

(6) CHR52912_VAL_TRIG_TTL3
The power supply waits for a trigger on the PXI TRIG3 line (for PXI instruments) or the VXI TTL3 line (for VXI instruments).

(7) CHR52912_VAL_TRIG_TTL4
The power supply waits for a trigger on the PXI TRIG4 line (for PXI instruments) or the VXI TTL4 line (for VXI instruments).

(8) CHR52912_VAL_TRIG_TTL5
The power supply waits for a trigger on the PXI TRIG5 line (for PXI instruments) or the VXI TTL5 line (for VXI instruments).

(9) CHR52912_VAL_TRIG_TTL6
The power supply waits for a trigger on the PXI TRIG6 line (for PXI instruments) or the VXI TTL6 line (for VXI instruments).

(10) CHR52912_VAL_TRIG_TTL7
The power supply waits for a trigger on the PXI TRIG7 line (for PXI instruments) or the VXI TTL7 line (for VXI instruments).

slope
Variable Type ViInt32

Specify the polarity of the trigger slope. The power supply triggers on the rising or falling edge of the trigger source depending on the value of this parameter.

Valid value:

CHR52912_VAL_POSITIVE
CHR52912_VAL_NEGATIVE

Default value:
CHR52912_VAL_POSITIVE

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.9 chr52912_ConfigureTriggeredOutput

ViStatus chr52912_ConfigureTriggeredOutput (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViReal64 triggeredLevel,
ViReal64 triggeredLimit);

Purpose

This function configures the voltage level, current limit of the power supply after receiving the start trigger signal. The power supply will keep outputting until receiving the stop trigger signal. This function inputs necessary commands into the sequencer table to achieve its functionality.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel where the triggered voltage output is configured.

Valid Channel Names: ch1,ch2

triggeredLevel

Variable Type ViReal64

Pass the DC voltage level you want the power supply to generate after it receives a start trigger.

Units: volts

Valid Range: 0.0 - 48.0 volts

triggeredLimit

Variable Type ViReal64

Pass the current limit of the power supply.

Units: amps

Valid Range: 0.0 - 2.0 amps

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.10 chr52912_Measure

ViStatus chr52912_Measure (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViInt32 measurementType,
ViPReal64 measurement);

Purpose

This function takes a measurement on the output and returns the measured value.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel where a measurement takes place.

Valid Channel Names: ch1,ch2

measurementType

Variable Type ViInt32

Pass the measurement you want the power supply to take.

Valid Values:

CHR52912_VAL_MEASURE_VOLTAGE - DC voltage measurement

CHR52912_VAL_MEASURE_CURRENT - DC current measurement

measurement

Variable Type ViReal64 (passed by reference)

Returns the measured value.

Units: volts (for voltage measurement)
amps (for current measurement)

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.11 chr52912_ConfigureSequencerTable

ViStatus chr52912_ConfigureSequencerTable (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViUInt16 tableSize,
const SEQUENCER_TABLE *tablePointer,
ViUInt16 errorIndex);

Purpose

This function configures the sequencer table in the 52912. A sequencer table is a list of commands that the 52912 will execute sequentially. There are many commands available in the table such as OUTPUT_ON, OUTPUT_OFF, SET_V, SET_I, DELAY and etc.

Before writing the sequencer table into the 52912, the driver will check if every command/value in the table is valid.

There are two configurable sequencer tables, table 1 & table 2. In sequencer table 1, it is possible to control both channel 1 & 2, either separately or together, by using several adequate sequencer tables.

In sequencer table 2, you can only control channel 2, but not channel 1.

If there is any unacceptable condition, this function will return one of the following errors and an error index to indicate the position of the error in the table:

CHR52912_ERROR_NO_END_COMMAND
If there is no END command in the sequencer table.

CHR52912_ERROR_TABLE_SIZE
If the table size is less than 3 or greater than 256.

CHR52912_ERROR_INVALID_DELAY
If the value of the delay is invalid.

CHR52912_ERROR_REPEAT_INDEX
If the repeat index is invalid.

CHR52912_ERROR_INVALID_COMMAND_TYPE
If there exists unknown command.

CHR52912_ERROR_TABLE2_SELECT_CH1
If the channel select is CHR52912_VAL_SELECT_CH1 or

CHR52912_VAL_SELECT_CH1_CH2 in the table 2.

CHR52912_ERROR_INVALID_CH_SELECT
If the channel select exceeds the valid range.

CHR52912_ERROR_INVALID_VOLTAGE_RANGE
If the voltage level exceeds the valid range.

CHR52912_ERROR_INVALID_CURRENT_RANGE
If the current limit exceeds the valid range.

CHR52912_ERROR_NOT_SET_V_I_BEFORE_OUTPUT_ON
If an OUTPUT_ON command appears before the voltage level & current limit are set. This examination is for the protection of the circuit.

CHR52912_ERROR_INVALID_REPEAT_TIMES
If the repeat times exceeds the valid range.

Parameter List

instrumentHandle

Variable Type ViSession
The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]
Pass the name of the channel for which the sequencer table is configured.
Valid Channel Names: ch1,ch2

tableSize

Variable Type ViUInt16
The size of the sequencer table. The maximum size of the table is 256.

tablePointer

Variable Type const SEQUENCER_TABLE *
A pointer to the structure of the sequencer table. This pointer is usually an

array, and each element in the array is a structure of a sequencer table. The maximum size of the array is 256. The structure of a sequencer table is defined in the chr52912.h.

Some member variables in the structure may be redundant with some commands. In such situation, VI_NULL is assigned to these members. For example: If the command type is CHR52912_VAL_SET_V, the only used members in the structure is channelSelect & value. The others will be assigned as VI_NULL.

the following is the struct definition & member variables description:

```
typedef tagSequencerTable
{
    ViInt32 commandType;
    ViInt32 channelSelect;
    ViReal64 value;
    ViInt32 delayUnit;
    ViInt32 delay;
    ViInt32 repeatTimes;
    ViInt32 repeatStartIndex;
} SEQUENCER_TABLE;
```

-commandType
Determines which function to be accomplished

Valid value: CHR52912_VAL_SET_V
CHR52912_VAL_SET_I
CHR52912_VAL_RELAY_ON
CHR52912_VAL_RELAY_OFF
CHR52912_VAL_DELAY
CHR52912_VAL_MEASURE_V
CHR52912_VAL_MEASURE_I
CHR52912_VAL_REPEAT
CHR52912_VAL_NONSTOP_REPEAT
CHR52912_VAL_END

-channelSelect
Selects which channel to be controlled. You can control both channels in one command.

Valid value: CHR52912_VAL_SELECT_CH1

CHR52912_VAL_SELECT_CH2
CHR52912_VAL_SELECT_CH1_CH2

-value
The voltage level or current limit to be set.

Valid value: voltage level: 0.0 - 48.0 Volts
current limit: 0.0 - 2.0 Amps

-delayUnit
Unit of the delay. Two possible values are millisecond or microsecond.

Valid value: CHR52912_VAL_DELAY_UNIT_MILLI_SECOND
CHR52912_VAL_DELAY_UNIT_MICRO_SECOND

-delay
The value of delay.

Valid value: 1 - 8388608 (for millisecond)
100 - 8388608 (for microsecond)

-repeatTimes:
In the sequencer table, the command CHR52912_VAL_REPEAT can repeat the commands between table indexes from repeatStartIndex to the current index of the repeat command. The repeated times declares the times that these commands is going to be repeated.

Valid value: 0 - 524288

-repeatStartIndex
The zero-based index in the table. The repeatStartIndex can't be the index of repeat command itself.

errorIndex

Variable Type ViInt32 (passed by reference)
Before the driver sends the sequencer table into the 52912, a check routine will ensure the validity of the table. If there is any invalid entry in the table, the error index will show you where it is.

Return Value

If the function succeeds, status code is 0. To obtain a text description of the

status code, call the chr52912_error_message function.

5.1.12 chr52912_ClearSequencerTable

ViStatus chr52912_ClearSequencerTable (ViSession instrumentHandle,
ViChar _VI_FAR channelName[]);

Purpose

This function clears the sequencer table in the 52912.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel where the sequencer table is cleared.

Valid Channel Names: ch1,ch2

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.13 chr52912_IsRunningSequencer

ViStatus chr52912_IsRunningSequencer (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViPBoolean isRunning);

Purpose

This function returns the state of the 52912. It will indicate if the instrument is currently running a sequencer table.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel where a measurement takes place.
Valid Channel Names: ch1,ch2

isRunning

Variable Type ViBoolean (passed by reference)

Indicates the state of the sequencer table. The driver returns the value of CHR52912_ATTR_IS_RUNNING_SEQUENCER attribute.

The value VI_TRUE indicates that the sequencer table is running.

The value VI_FALSE indicates that the 52912 is not running the sequencer table.

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.14 chr52912_ConfigureMeasure

ViStatus chr52912_ConfigureMeasure (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViReal64 level, ViReal64 limit,
ViInt32 measurementType,
ViInt32 delayTb,
ViInt32 delayTbUnit,
ViInt32 delayTc,
ViInt32 delayTcUnit,
ViInt32 delayTe,
ViInt32 delayTeUnit,
ViInt32 measureCount);

Purpose

This function configures the measure functionality of the power supply. You can only take the measurement on the output of the power supply. This function inputs necessary commands into the sequencer table to achieve the measure functionality.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel where the measure functionality is configured.

Valid Channel Names: ch1, ch2

level

Variable Type ViReal64

Pass the DC voltage that the power supply is about to generate.

Units: volts

Valid Range: 0.0 - 48.0 volts

limit

Variable Type ViReal64
Pass the current limit.

Units: amps

Valid Range: 0.0 - 2.0 amps (depends on current voltage level)

measurementType
Variable Type ViInt32
Pass the measurement the power supply is about to take.

Valid Values:
CHR52912_VAL_MEASURE_VOLTAGE - DC voltage measurement
CHR52912_VAL_MEASURE_CURRENT - DC current measurement

delayTb
Variable Type ViInt32
The value of delay Tb. After turning on the output relay, wait for a time period Tb. Then, the output should be stable.

Valid value: 1 - 8388608 (for millisecond)
100 - 8388608 (for microsecond)

delayTbUnit
Variable Type ViInt32
Unit of the delay Tb.

Valid value: CHR52912_VAL_DELAY_UNIT_MILLI_SECOND
CHR52912_VAL_DELAY_UNIT_MICRO_SECOND

delayTc
Variable Type ViInt32
The value of delay Tc. Before any measurement is taken, wait for a time period Tc.

Valid value: 1 - 8388608 (for millisecond)
100 - 8388608 (for microsecond)

delayTcUnit
Variable Type ViInt32
Unit of the delay Tc.

Valid value: CHR52912_VAL_DELAY_UNIT_MILLI_SECOND

CHR52912_VAL_DELAY_UNIT_MICRO_SECOND

delayTe
Variable Type ViInt32
The value of delay Te. Delay Te is the time interval between each measurement, i.e., take a measurement once after every Te.

Valid value: 1 - 8388608 (for millisecond)
100 - 8388608 (for microsecond)

delayTeUnit
Variable Type ViInt32
Unit of the delay Te.

Valid value: CHR52912_VAL_DELAY_UNIT_MILLI_SECOND
CHR52912_VAL_DELAY_UNIT_MICRO_SECOND

measureCount
Variable Type ViInt32
The total number of times that the power supply will take the measurements. The maximum count is 2048.

Valid value: 1 - 2048

Return Value
If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.15 chr52912_ReadMeasuredData

ViStatus chr52912_ReadMeasuredData (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViPInt16 num_ofData,
MEASURED_DATA measuredData[]);

Purpose

This function acquires the measured data stored in the register, after the 52912 completes the sequential measurements.

Parameter List

instrumentHandle

Variable Type ViSession
The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]
Pass the name of the channel where the measured data is read.
Valid Channel Names: ch1, ch2

num_ofData

Variable Type ViInt32 (passed by reference)
The amount of measured data that the power supply will acquire. If the total number of data in the power supply is not equal to the number you request, this argument will be set to be the actual number of data in the power supply when returned.

measuredData

Variable Type MEASURED_DATA []
An array that consisted of MEASURED_DATA structure. This structure is defined in the chr52912.h.

struct definition & member variables description:

typedef struct tagMeasuredData
{
ViInt32 index;

ViInt32 measurementDataType;
ViReal64 measuredValue;
} MEASURED_DATA;

-index
A zero-based index indicates the time order of the measured data.

-measurementDataType
Indicate whether the data is voltage or current.
current – 0
voltage – 1

-measuredValue
The measured value.

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.16 chr52912_Initiate

ViStatus chr52912_Initiate (ViSession instrumentHandle, ViInt32 channel_select);

Purpose

This function makes the power supply wait for a trigger.

After this function is called, if the power supply is not currently waiting for a trigger, it will wait for the trigger you specify via the chr52912_ConfigureTriggerSource function. After the power supply detects the specified trigger, it will start to run the sequencer table.

If the power supply is already waiting for a trigger, then this function will return a CHR52912_ERROR_WAITING_TRIGGER error.

Once you set the power supply to wait for a trigger signal, no operation can then be performed except Abort, SendSWTrigger, IsWaitingTrigger, IsRunningSequencer. Executing any other functions will cause a CHR52912_ERROR_WAITING_TRIGGER error.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channel_select

Variable Type ViInt32

Indicate which sequencer table to be initiated. You can initiate both sequencer table simultaneously.

Valid value: CHR52912_VAL_SELECT_CH1
CHR52912_VAL_SELECT_CH2
CHR52912_VAL_SELECT_CH1_CH2

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.17 chr52912_Abort

ViStatus chr52912_ Abort (ViSession instrumentHandle);

Purpose

If the power supply is currently waiting for a trigger to run the sequencer table, this function aborts the waiting state.

If the power supply is not waiting for a trigger, this function returns a CHR52912_ERROR_NOT_WAITING_TRIGGER error.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.18 chr52912_SendSoftwareTrigger

ViStatus chr52912_SendSoftwareTrigger (ViSession instrumentHandle);

Purpose

This function sends a trigger signal to the 52912. Call this function if you configure the start trigger of the power supply to be software triggers. If this function is called while the power supply is not configured to respond to software triggers, this function will return a CHR52912_ERROR_TRIGGER_NOT_SOFTWARE error.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.19 chr52912_IsWaitingTrigger

ViStatus chr52912_IsWaitingTrigger (ViSession instrumentHandle,
ViPBoolean inState);

Purpose

This function returns the state of the 52912. It indicates if the instrument is currently waiting for a trigger signal.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

inState

Variable Type ViBoolean (passed by reference)

This parameter returns VI_TRUE if the power supply is currently in the state you specify with the outputState parameter, and VI_FALSE if it is not.

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.20 chr52912_reset

ViStatus chr52912_reset (ViSession instrumentHandle);

Purpose

This function resets the instrument to a known state and sends initialization commands to the instrument. This function also clears the sequencer table in the instrument. The initialization commands set instrument settings such as Headers Off, Short Command form, and Data Transfer Binary to the state necessary for the operation of the instrument driver.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.21 chr52912_ResetOutputProtection

ViStatus chr52912_ResetOutputProtection (ViSession instrumentHandle,
ViChar _VI_FAR channelName[]);

Purpose

This function resets the power supply's output protection after an over-voltage, over-temperature, or other output protection condition occurs.

When an over-voltage or over-temperature condition occurs, the output protection of the power supply disables the output. This function calls the chr52912_reset function to reset output protection. After the output protection is reset, the power supply resumes generating a power signal only on the channel you specified.

You use the chr52912_error_query or chr52912_QueryOutputState function to determine if the power supply is in an output-protection state.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel you want to reset.

Valid Channel Names: ch1,ch2

Return Value

If the function succeeds, status code is 0. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.22 chr52912_error_message

ViStatus chr52912_error_message (ViSession instrumentHandle,
ViStatus errorCode,
ViChar _VI_FAR errorMessage[]);

Purpose

This function converts a status code returned by an instrument driver function into a user-readable string.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session. You can pass VI_NULL for this parameter. This is useful when one of the initialize functions fail.

errorCode

Variable Type ViStatus

Pass the Status parameter that is returned from any of the instrument driver functions.

errorMessage

Variable Type ViChar[]

Returns the user-readable message string that corresponds to the status code you specify. You must pass a ViChar array with at least 256 bytes.

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.23 chr52912_error_query

ViStatus chr52912_error_query (ViSession instrumentHandle,
ViPInt32 errorCode,
ViChar _VI_FAR errorMessage[]);

Purpose

This function reads an error code and a message from 52912.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

errorCode

Variable Type ViInt32 (passed by reference)

Returns the error code (0xBFFA4061) read from 52912 if any 52912 status error occurred.

errorMessage

Variable Type ViChar[]

Returns the error message string with status error code read from 52912. You must pass a ViChar array with at least 256 bytes.

52912 status error code (1-byte)

bit	7	6	5	4	3	2	1	0
Status error	Ch2 unstable input	Ch2 56V input	Ch2 OVP	Ch2 OTP	Ch1 unstable input	Ch1 56V input	Ch1 OVP	Ch1 OTP

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.24 chr52912_QueryOutputState

ViStatus chr52912_QueryOutputState (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViInt32 outputState,
ViPBoolean inState);

Purpose

This function returns whether the power supply is in a particular output state.

A constant voltage condition occurs when the output voltage is equal to the value of the CHR52912_ATTR_VOLTAGE_LEVEL attribute and the current is less than or equal to the value of the CHR52912_ATTR_CURRENT_LIMIT attribute.

A constant current condition occurs when the output current is equal to the value of the CHR52912_ATTR_CURRENT_LIMIT attribute and the CHR52912_ATTR_CURRENT_LIMIT_BEHAVIOR attribute is set to CHR52912_VAL_CURRENT_REGULATE.

An unregulated condition occurs when the output voltage is less than the value of the CHR52912_ATTR_VOLTAGE_LEVEL attribute and the current is less than the value of the CHR52912_ATTR_CURRENT_LIMIT attribute.

An over-voltage condition occurs when the output voltage is equal to or greater than the value of the CHR52912_ATTR_OVP_LIMIT attribute and the CHR52912_ATTR_OVP_ENABLED attribute is set to VI_TRUE.

An over-temperature condition occurs when the temperature of the power supply is over 80 degrees.

When either an over-voltage condition or an over-temperature condition occurs, the power supply's output protection disables the output. If the power supply is in an over-voltage or over-temperature state, it does not produce power until the output protection is reset. The chr52912_ResetOutputProtection function resets the output protection. Once the output protection is reset, the power supply resumes generating a power signal.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel on which to query for an output state.
Valid Channel Names: ch1,ch2

outputState

Variable Type ViInt32

Pass the output state for which you want to query.

Valid Values:
CHR52912_VAL_OUTPUT_CONSTANT_VOLTAGE - Constant Voltage State
CHR52912_VAL_OUTPUT_UNREGULATED - Unregulated State
CHR52912_VAL_OUTPUT_OVER_VOLTAGE - Over-voltage State
CHR52912_VAL_OUTPUT_OVER_TEMEPRATURE - Over-temperature State

inState

Variable Type ViBoolean (passed by reference)

This parameter returns VI_TRUE if the power supply is currently in the state you specify with the outputState parameter, and VI_FALSE if it is not.

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.25 chr52912_QueryMaxCurrentLimit

ViStatus chr52912_QueryMaxCurrentLimit (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViReal64 voltageLevel,
ViPReal64 maxCurrentLimit);

Purpose

This function returns the maximum programmable current limit that the power supply accepts for a particular voltage level on a channel for the output range to which the power supply is currently configured.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle obtained from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel on which to query for the maximum programmable current limit.

Valid Channel Names: ch1,ch2

voltageLevel

Variable Type ViReal64

Pass the voltage level for which to determine the maximum programmable current limit.

Units: volts (V).

Valid Range: 0.0 - 48.0 volts

maxCurrentLimit

Variable Type ViReal64 (passed by reference)

This parameter returns the maximum programmable current limit for the voltage level you specify.

Units: amps (A)

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.26 chr52912_QueryMaxVoltageLevel

ViStatus chr52912_QueryMaxVoltageLevel (ViSession instrumentHandle,
ViChar _VI_FAR channelName[],
ViReal64 currentLimit,
ViPReal64 maxVoltageLevel);

Purpose

This function returns the maximum programmable voltage level that the power supply accepts for a particular current limit on a channel for the output range to which the power supply is currently configured.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

channelName

Variable Type ViChar[]

Pass the name of the channel on which to query for the maximum programmable voltage level.

Valid Channel Names: ch1, ch2

currentLimit

Variable Type ViReal64

Pass the current limit for which to determine the maximum programmable voltage level.

Units: amps (A).

Valid Range: 0.0 - 2.0 amps

maxVoltageLevel

Variable Type ViReal64 (passed by reference)

This parameter returns the maximum programmable voltage level for the current limit you specify.

Units: volts (V)

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.27 chr52912_QueryCalibrationDate

ViStatus chr52912_QueryCalibrationDate (ViSession instrumentHandle,
ViChar errorMessage[]);

Purpose

This function returns the last calibration date.

Parameter List

instrumentHandle

Variable Type ViSession

The ViSession handle that you obtain from the chr52912_init or chr52912_InitWithOptions function. The handle identifies a particular instrument session.

errorMessage

Variable Type ViChar[]

Returns the last calibration date string.

You must pass a ViChar array with at least 64 bytes.

Return Value

Status code 0 is for success. To obtain a text description of the status code, call the chr52912_error_message function.

5.1.28 Other Driver Functions

The 52912 Power Supply does not support the functions of self-test, configure-output-range, and configure-OVP. When calling **chr52912_self_test**, **chr52912_ConfigureOutputRange**, or **chr52912_ConfigureOVP**, you will get the return value of VI_SUCCESS (0) without real action. The detail description of each driver function see the file of chr52912.HLP.

5.2 Error Code

52912 Status Codes:

WARNINGS:

3FFA4001	Calibration expired. Re-calibration is recommended.

EERRORS:

BFFA4001	52912 ch1 init fail.
BFFA4002	52912 ch2 init fail.
BFFA4003	52912 init fail.
BFFA4004	Exceed power limit. To set this current limit, lower voltage level first.
BFFA4005	There is no 56V input presenting at this channel.
BFFA4006	Unstable 56V input occurred. Call reset or resetOutputProtection first.
BFFA4007	Invalid operation mode.
BFFA4008	Invalid command type in the sequencer table.
BFFA4009	Invalid channel select in the sequencer table.
BFFA400A	Invalid voltage range in the sequencer table.
BFFA400B	Invalid current range in the sequencer table.
BFFA400C	Invalid delay value in the sequencer table.
BFFA400D	Invalid repeat times in the sequencer table.
BFFA400E	Invalid repeat start index in the sequencer table.
BFFA400F	The nested layer of repeat can't exceed 7.
BFFA4010	No END command in the sequencer table.
BFFA4011	Not set Voltage Level & Current Limit yet before turn on output relay.
BFFA4012	Can't select channel 1 in the sequencer table of channel 2.
BFFA4013	Sequencer table is empty, configure Sequencer table first.
BFFA4014	Sequencer table 2 is unusable.
BFFA4015	The Sequencer table is currently running.
BFFA4016	The instrument is not in waiting trigger state.
BFFA4017	The instrument is in waiting trigger state.
BFFA4018	Invalid trigger.
BFFA4019	No measured data.
BFFA4061	52912 Status error.

IviPower Status Codes:

ERRORS:

BFFA1001	The trigger source is not software trigger.

IVI Engine Status Codes:

ERRORS:

BFFA0001	Instrument error. Call chr52912_error_query.
BFFA0002	Cannot open file.
BFFA0003	Error reading from file.
BFFA0004	Error writing to file.
BFFA0005	Driver module file not found.
BFFA0006	Cannot open driver module file for reading.
BFFA0007	Driver module has invalid file format or invalid data.
BFFA0008	Driver module contains undefined references.
BFFA0009	Cannot find function in driver module.
BFFA000A	Failure loading driver module.
BFFA000B	Invalid path name.
BFFA000C	Invalid attribute.
BFFA000D	IVI attribute is unable to write.
BFFA000E	IVI attribute is not readable.
BFFA000F	Invalid parameter.
BFFA0010	Invalid value.
BFFA0011	Function not supported.
BFFA0012	Attribute not supported.
BFFA0013	Value not supported.
BFFA0014	Invalid type.
BFFA0015	Types do not match.
BFFA0016	Attribute already has a value waiting to be updated.
BFFA0017	Specified item already exists.
BFFA0018	Not a valid configuration.
BFFA0019	Requested item does not exist or value not available.
BFFA001A	Requested attribute value not known.
BFFA001B	No range table.
BFFA001C	Range table is invalid.
BFFA001D	Object or item is not initialized.
BFFA001E	Non-interchangeable behavior.
BFFA001F	No channel table has been built for the session.
BFFA0020	Channel name specified is not valid.
BFFA0021	Unable to allocate system resource.
BFFA0022	Permission to access file was denied.
BFFA0023	Too many files are already open.
BFFA0024	Unable to create temporary file in target directory.
BFFA0025	All temporary filenames already used.
BFFA0026	Disk is full.
BFFA0027	Cannot find configuration file on disk.
BFFA0028	Cannot open configuration file.
BFFA0029	Error reading configuration file.

BFFA002A Invalid ViInt32 value in configuration file.
BFFA002B Invalid ViReal64 value in configuration file.
BFFA002C Invalid ViBoolean value in configuration file.
BFFA002D Entry missing from configuration file.
BFFA002E Initialization failed in driver DLL.
BFFA002F Driver module has unresolved external reference.
BFFA0030 Cannot find CVI Run-Time Engine.
BFFA0031 Cannot open CVI Run-Time Engine.
BFFA0032 CVI Run-Time Engine has invalid format.
BFFA0033 CVI Run-Time Engine is missing required function(s).
BFFA0034 CVI Run-Time Engine initialization failed.
BFFA0035 CVI Run-Time Engine has unresolved external reference.
BFFA0036 Failure loading CVI Run-Time Engine.
BFFA0037 Cannot open DLL for read exports.
BFFA0038 DLL file is corrupt.
BFFA0039 No DLL export table in DLL.
BFFA003A Unknown attribute name in default configuration file.
BFFA003B Unknown attribute value in default configuration file.
BFFA003C Memory pointer specified is not known.
BFFA003D Unable to find any channel strings.
BFFA003E Duplicate channel string.
BFFA003F Duplicate virtual channel name.
BFFA0040 Missing virtual channel name.
BFFA0041 Bad virtual channel name.
BFFA0042 Unassigned virtual channel name.
BFFA0043 Bad virtual channel assignment.
BFFA0044 Channel name required.
BFFA0045 Channel name not allowed.
BFFA0046 Attribute not valid for channel.
BFFA0047 Attribute must be channel based.
BFFA0048 Channel already excluded.
BFFA0049 Missing option name (nothing before the '=').
BFFA004A Missing option value (nothing after the '=').
BFFA004B Bad option name.
BFFA004C Bad option value.
BFFA004D Operation only valid on a class driver session.
BFFA004E "ivi.ini" filename is reserved.
BFFA004F Duplicate run-time configuration entry.
BFFA0050 Index parameter is one-based.
BFFA0051 Index parameter is too high.
BFFA0052 Attribute is not cacheable.
BFFA0053 You cannot export a ViAddr attribute to the end-user.

BFFA0054 Bad channel string in channel string list.
BFFA0055 Bad prefix name in default configuration file.

VISA Status Codes:

WARNINGS:

3FFF0002 Event enabled for one or more specified mechanisms.
3FFF0003 Event disabled for one or more specified mechanisms.
3FFF0004 Successful, but queue already empty.
3FFF0005 Specified termination character was read.
3FFF0006 Number of bytes transferred equals input count.
3FFF0077 Configuration non-existent or could not be loaded.
3FFF007D Open successful, but the device not responding.
3FFF0080 Wait successful, but more event objects available.
3FFF0082 Specified object reference is not initialized.
3FFF0084 Attribute value not supported.
3FFF0085 Status code could not be interpreted.
3FFF0088 Specified I/O buffer type not supported.
3FFF0098 Successful, but invoke no handlers for this event.
3FFF0099 Successful but session has nested shared locks.
3FFF009A Successful but session has nested exclusive locks.
3FFF009B Successful but operation not asynchronous.

ERRORS:

BFFF0000 Unknown system error (miscellaneous error).
BFFF000E Session or object reference is invalid.
BFFF000F Resource is locked.
BFFF0010 Invalid expression specified for search.
BFFF0011 Resource is not present in the system.
BFFF0012 Invalid resource reference specified. Parsing error.
BFFF0013 Invalid access mode.
BFFF0015 Timeout expired before operation completed.
BFFF0016 Unable to deallocate session data structures.
BFFF001B Specified degree is invalid.
BFFF001C Specified job identifier is invalid.
BFFF001D Attribute is not supported by the referenced object.
BFFF001E Attribute state not supported by the referenced object.
BFFF001F Specified attribute is read-only.
BFFF0020 Lock type lock not supported by this resource.
BFFF0021 Invalid access key.
BFFF0026 Specified event type not supported by the resource.
BFFF0027 Invalid mechanism specified.
BFFF0028 A handler was not installed.

BFFF0029 Handler reference either invalid or was not installed.
BFFF002A Specified event context invalid.
BFFF002D Event queue for specified type has overflowed.
BFFF002F Event type must be enabled in order to receive.
BFFF0030 User abort during transfer.
BFFF0034 Violation of raw write protocol during transfer.
BFFF0035 Violation of raw read protocol during transfer.
BFFF0036 Device reported output protocol error during transfer.
BFFF0037 Device reported input protocol error during transfer.
BFFF0038 Bus error during transfer.
BFFF0039 Unable to queue asynchronous operation.
BFFF003A Unable to start operation because setup is invalid.
BFFF003B Unable to queue the asynchronous operation.
BFFF003C Insufficient resources to perform memory allocation.
BFFF003D Invalid buffer mask specified.
BFFF003E I/O error.
BFFF003F Format specifier invalid.
BFFF0041 Format specifier not supported.
BFFF0042 Trigger line is currently in use.
BFFF004A Service request not received for the session.
BFFF004E Invalid address space specified.
BFFF0051 Invalid offset specified.
BFFF0052 Invalid access width specified.
BFFF0054 Offset not accessible from this hardware.
BFFF0055 Source and destination widths are different.
BFFF0057 Session not currently mapped.
BFFF0059 Previous response still pending.
BFFF005F No listeners condition detected.
BFFF0060 Interface not currently the controller in charge.
BFFF0061 Interface not the system controller.
BFFF0067 Session does not support this operation.
BFFF006A A parity error occurred during transfer.
BFFF006B A framing error occurred during transfer.
BFFF006C An overrun error occurred during transfer.
BFFF0070 Offset not properly aligned for operation access width.
BFFF0071 Specified user buffer not valid.
BFFF0072 Resource valid, but VISA cannot access it.
BFFF0076 Width not supported by this hardware.
BFFF0078 Invalid parameter value, parameter unknown.
BFFF0079 Invalid protocol.
BFFF007B Invalid window size.
BFFF0080 Session currently contains a mapped window.

BFFF0081 Operation not implemented.
BFFF0083 Invalid length.
BFFF0091 Invalid mode.
BFFF009C Session did not have a lock on the resource.
BFFF009D The device does not export any memory.
BFFF009E VISA-required code library not located or not loaded.

VXIPnP Driver Status Codes:

WARNINGS:

3FFC0101 Instrument does not have ID Query capability.
3FFC0102 Instrument does not have Reset capability.
3FFC0103 Instrument does not have Self-Test capability.
3FFC0104 Instrument does not have Error Query capability.
3FFC0105 Instrument does not have Revision Query capability.

ERRORS:

BFFC0001 Parameter 1 out of range, or error trying to set it.
BFFC0002 Parameter 2 out of range, or error trying to set it.
BFFC0003 Parameter 3 out of range, or error trying to set it.
BFFC0004 Parameter 4 out of range, or error trying to set it.
BFFC0005 Parameter 5 out of range, or error trying to set it.
BFFC0006 Parameter 6 out of range, or error trying to set it.
BFFC0007 Parameter 7 out of range, or error trying to set it.
BFFC0008 Parameter 8 out of range, or error trying to set it.
BFFC0011 Instrument failed the ID Query.
BFFC0012 Invalid response from instrument.

5.3 Example Programs

5.3.1 C Sample Program

Following is a sample program written in C. It is a simple example to initializes the power supply, then demonstrates how to operate the power supply under different operation mode.

```
#include <stdio.h>
#include <userint.h>
#include <utility.h>
#include "chr52912.h"

void BuildErrorString (ViSession chr52912, ViStatus error, ViString
errStr);

main ()
{
     ViSession      chr52912;
     ViStatus       error = VI_SUCCESS;
     ViChar         msg[256];
     ViInt32        errorIndex;
     ViInt32        numOfData;
     SEQUENCER_TABLE ST1[6], *pST1;
     MEASURED_DATA  MD1[2048];

     //Definition of sequencer table ST1
     ST1[0].commandType = CHR52912_VAL_SET_I;
     ST1[0].channelSelect = CHR52912_VAL_SELECT_CH1;
     ST1[0].value = 1;

     ST1[1].commandType = CHR52912_VAL_SET_V;
     ST1[1].channelSelect = CHR52912_VAL_SELECT_CH1;
     ST1[1].value = 20;

     ST1[2].commandType = CHR52912_VAL_OUTPUT_ON;
     ST1[2].channelSelect = CHR52912_VAL_SELECT_CH1;

     ST1[3].commandType = CHR52912_VAL_DELAY;
     ST1[3].delayUnit = CHR52912_VAL_DELAY_UNIT_MILLI_SECOND;
     ST1[3].delay = 3000;

     ST1[4].commandType = CHR52912_VAL_OUTPUT_OFF;
     ST1[4].channelSelect = CHR52912_VAL_SELECT_CH1;

     ST1[5].commandType = CHR52912_VAL_END;

      /*
         If you want to run this sample program and the instrument is
          not present, set the Simulate flag to 1.  (Example:  "Simulate
```

```
     = 1")
 */
checkErr( chr52912_InitWithOptions ("PXI3::10::INSTR", VI_TRUE,
          VI_TRUE, "Simulate=0,RangeCheck=1,QueryInstrStatus=1,
          Cache=1", &chr52912));

//DC Power Mode
checkErr( chr52912_ConfigureOperationMode(chr52912,
          CHR52912_VAL_DC_POWER));
checkErr( chr52912_ConfigureCurrentLimit(chr52912, "ch1", 0,
          2.0));
checkErr( chr52912_ConfigureVoltageLevel(chr52912, "ch1", 15.0));
checkErr( chr52912_ConfigureOutputEnabled(chr52912, "ch1",
          VI_TRUE));
//chr52912_Measure to get the reading of current/voltage.
//chr52912_reset or chr52912_ResetOutputProtection to reset 52912
//if any error occurred.
//chr52912_close to close.
Delay(2);
checkErr( chr52912_ConfigureOutputEnabled(chr52912, "ch1",
          VI_FALSE));

//DC Power Mode with trigger signal
checkErr( chr52912_ConfigureTriggeredOutput(chr52912, "ch1", 15,
          1));
checkErr( chr52912_ConfigureTriggerSource(chr52912,
          CHR52912_VAL_SOFTWARE_TRIG,
          CHR52912_VAL_SOFTWARE_TRIG));
checkErr( chr52912_Initiate(chr52912, CHR52912_VAL_SELECT_CH1));
MessagePopup("Message", "Press OK to start generating triggered
          output.");
checkErr( chr52912_SendSoftwareTrigger(chr52912));
MessagePopup("Message", "Press OK to stop triggered output.");
checkErr( chr52912_SendSoftwareTrigger(chr52912));
checkErr( chr52912_Abort(chr52912));

//Sequencer Mode
checkErr( chr52912_ConfigureOperationMode(chr52912,
          CHR52912_VAL_SEQUENCER));
pST1 = ST1;
checkErr( chr52912_ConfigureSequencerTable(chr52912, "ch1", 6,
          pST1, &errorIndex));
checkErr( chr52912_ConfigureTriggerSource(chr52912,
          CHR52912_VAL_SOFTWARE_TRIG, CHR52912_VAL_TRIG_NONE));
checkErr( chr52912_Initiate(chr52912, CHR52912_VAL_SELECT_CH1));
MessagePopup("Message", "Press OK to start to run sequencer
          table.");
checkErr( chr52912_SendSoftwareTrigger(chr52912));
MessagePopup("Message", "Wait for the output off.");
checkErr( chr52912_Abort(chr52912));

//Measure Mode
checkErr( chr52912_ConfigureOperationMode(chr52912,
          CHR52912_VAL_MEASURE));
```

```
     checkErr( chr52912_ConfigureTriggerSource(chr52912,
               CHR52912_VAL_SOFTWARE_TRIG, CHR52912_VAL_TRIG_NONE));
     checkErr( chr52912_ConfigureMeasure(chr52912, "ch2", 10, 1,
               CHR52912_VAL_MEASURE_VOLTAGE,
               100, CHR52912_VAL_DELAY_UNIT_MICRO_SECOND,
               100, CHR52912_VAL_DELAY_UNIT_MICRO_SECOND,
               1, CHR52912_VAL_DELAY_UNIT_MILLI_SECOND,
               2048));
     checkErr( chr52912_Initiate(chr52912, CHR52912_VAL_SELECT_CH2));
     MessagePopup("Message", "Press OK to take measurement.");
     checkErr( chr52912_SendSoftwareTrigger(chr52912));
     MessagePopup("Message", "Wait for the output off then read the
               measured data.");
     checkErr( chr52912_Abort(chr52912));
     //Pass the amount of measured data you to read, then the driver API
     //gets the measured data.
     numOfData = 1000;
     checkErr( chr52912_ReadMeasuredData(chr52912, "ch2", &numOfData,
               MD1));
     sprintf(msg, "There are %d records of measured data in the power
               supply.", numOfData);
     MessagePopup("Message", msg);

Error:
     if (error != VI_SUCCESS) {
          ViChar  errStr[2048];

          BuildErrorString (chr52912, error, errStr);
          MessagePopup ("Error!", errStr);
     }

     chr52912_reset (chr52912);

     if (chr52912)
          chr52912_close (chr52912);}
}

void BuildErrorString (ViSession chr52912, ViStatus error, ViString
errStr)
{
     ViChar  tempStr[256], errElabStr[256];
     ViChar  *p = errStr;
     ViStatus primaryErr, secondaryErr;

     p += sprintf (p, "Driver Status:  (Hex 0x%x)", error);

     /* Get description of the returned status code */
     chr52912_error_message (chr52912, error, tempStr);
     p += sprintf (p, " %s\n\n", tempStr);

     /* Get any additional error information */
     errElabStr[0] = 0;
     chr52912_GetErrorInfo (chr52912, &primaryErr, &secondaryErr,
               errElabStr);
```

```
    if (primaryErr == error) {
         if (secondaryErr != VI_SUCCESS) {
              p += sprintf (p, "Secondary Error:  (Hex 0x%x)",
                  secondaryErr);

              /* Get description of the returned status code */
              chr52912_error_message (chr52912, secondaryErr,
                  tempStr);
              p += sprintf (p, " %s\n\n", tempStr);
         }
         if (errElabStr[0] != 0)
         sprintf (p, "Elaboration:  %s", errElabStr);
    }
}
```

5.3.2 Visual Basic Sample Program

Following is a sample program written in VB. It is a simple example to initializes the power supply, then demonstrates how to operate the power supply under different operation mode.

```
Option Explicit
Dim vi As ViSession
Dim status As ViStatus
Dim ST1(10) As SEQUENCER_TABLE
Dim MD1(2047) As MEASURED_DATA
Dim errorIndex As Integer
Dim numOfData As Integer

Private Sub Form_Load()
   'A simple sequencer table ST1 definition
   ST1(0).commandType = CHR52912_VAL_SET_V
   ST1(0).channelSelect = CHR52912_VAL_SELECT_CH1
   ST1(0).value = 20

   ST1(1).commandType = CHR52912_VAL_SET_I
   ST1(1).channelSelect = CHR52912_VAL_SELECT_CH1
   ST1(1).value = 1

   ST1(2).commandType = CHR52912_VAL_OUTPUT_ON
   ST1(2).channelSelect = CHR52912_VAL_SELECT_CH1

   ST1(3).commandType = CHR52912_VAL_DELAY
   ST1(3).delayUnit = CHR52912_VAL_DELAY_UNIT_MILLI_SECOND
   ST1(3).delay = 3000

   ST1(4).commandType = CHR52912_VAL_OUTPUT_OFF
   ST1(4).channelSelect = CHR52912_VAL_SELECT_CH1_CH2

   ST1(5).commandType = CHR52912_VAL_END

   status = chr52912_InitWithOptions("PXI3::10::INSTR", VI_TRUE,
       VI_TRUE, "Simulate=0", vi)

   MsgBox ("DC Power Mode")
   status = chr52912_ConfigureOperationMode(vi, CHR52912_VAL_DC_POWER)
   'Configure the power supply to generate 15V and 2A output when receive
    'the trigger signal.
   status = chr52912_ConfigureTriggeredOutput(vi, "ch1", 15, 2)
   status = chr52912_ConfigureTriggerSource(vi,
       CHR52912_VAL_SOFTWARE_TRIG, CHR52912_VAL_SOFTWARE_TRIG)
   status = chr52912_Initiate(vi, CHR52912_VAL_SELECT_CH1)
   MsgBox ("Push button to start to output.")
   status = chr52912_SendSoftwareTrigger(vi)
   MsgBox ("Push button to stop output.")
   status = chr52912_SendSoftwareTrigger(vi)
   status = chr52912_Abort(vi)
```

```
   MsgBox ("Sequencer Mode")
   status = chr52912_ConfigureOperationMode(vi,
       CHR52912_VAL_SEQUENCER)
   'Write the sequencer table ST1 into the power supply.
   status = chr52912_ConfigureSequencerTable(vi, "ch1", 6, ST1(0),
       errorIndex)
   status = chr52912_ConfigureTriggerSource(vi, _
       CHR52912_VAL_SOFTWARE_TRIG, CHR52912_VAL_TRIG_NONE)
   status = chr52912_Initiate(vi, CHR52912_VAL_SELECT_CH1)
   MsgBox ("Push button to start to output.")
   status = chr52912_SendSoftwareTrigger(vi)
   MsgBox ("Push button while the output is off.")
   status = chr52912_Abort(vi)

   MsgBox ("Measure Mode")
   status = chr52912_ConfigureOperationMode(vi, CHR52912_VAL_MEASURE)
   'Configure the power supply to generate 20V and 2A output then take
    'measurement of the output 2048 times. The time interval between each
    'measurement is 1 millisecond.
   status = chr52912_ConfigureMeasure(vi, "ch2", 20, 2,
       CHR52912_VAL_MEASURE_VOLTAGE, _
       100, CHR52912_VAL_DELAY_UNIT_MICRO_SECOND, _
      100, CHR52912_VAL_DELAY_UNIT_MICRO_SECOND, _
      1, CHR52912_VAL_DELAY_UNIT_MILLI_SECOND, _
      2048)
   status = chr52912_ConfigureTriggerSource(vi,
       CHR52912_VAL_SOFTWARE_TRIG, CHR52912_VAL_TRIG_NONE)
   status = chr52912_Initiate(vi, CHR52912_VAL_SELECT_CH2)
   MsgBox ("Push button to start to measure.")
   status = chr52912_SendSoftwareTrigger(vi)
   MsgBox ("Push button while the output is off.")
   status = chr52912_Abort(vi)
   numOfData = 2048
   'Get the measured data from the power supply and stored in MD1.
   status = chr52912_ReadMeasuredData(vi, "ch2", numOfData, MD1(0))

   status = chr52912_reset(vi)
   status = chr52912_close(vi)
End Sub
```

5.3.3 LabVIEW Sample Program

The sample program written in LabVIEW can be found in the directory **"C:\Program Files\Chroma\PXI\52912\drivers\examples"** after installing the driver. You can do several configurations such as **Triggered Output**, **Operation Mode**, **Sequencer Table**, **Measure** and **Trigger Source** and etc. After appropriately configuration, you can initiate the power supply to wait trigger signal. The power supply will run the sequencer table after receiving a trigger signal. This sample program demonstrates how to read the measured data stored in the power supply, too.

Also you can find a powerful subVI icon called "USE 52912" in the chr52912 submenu of the function palette of LabVIEW. This subVI can find all Chroma PXI 52912 properly installed in your system and let you choose which card to be initialized. All you have to do is that put this subVI in the Block diagram with other code you write. After executing the program, first there will be a pop up dialog which lists all 52912 in the ring control. And you can choose one then you get the instrument handle.

Note: With different versions of LabVIEW, there are different versions of sample program respectively. The prefix of the sample program indicates the correct version of suitable LabVIEW, for instance: **LV61_Exqample.vi** is for LabVIEW 6.1. Though you can open file of older version with newer version of LabVIEW, we can't ensure that the program can be executed correctly.

6. Hardware Specification

6.1 Application

52912/52914 card can used for any application where power supply voltage need to be programmable.

6.2 Block Diagram of Hardware

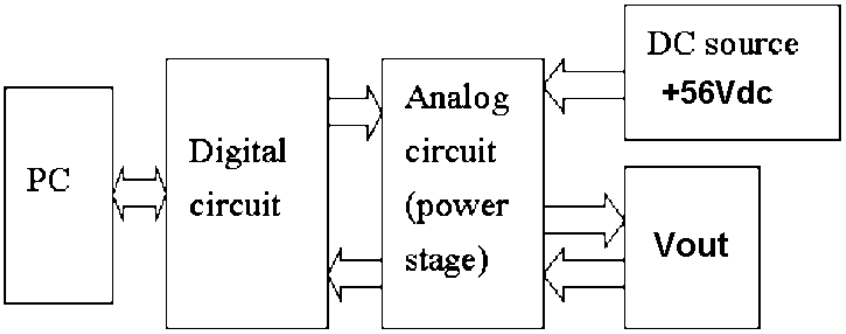
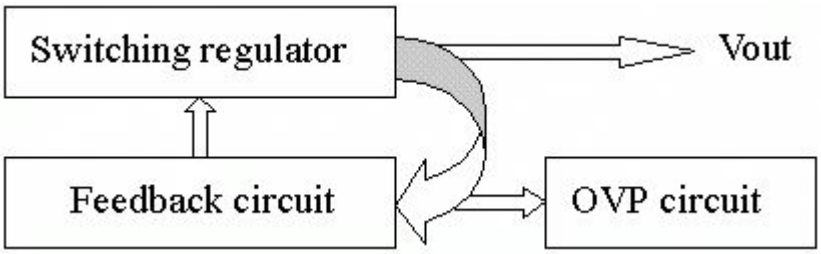


Figure 6-1 52912/52914 PXI Card Hardware Structure

Description of Each Function:

(1) PC: It uses control software to send commands to 52912/52914 and display the results on the screen.

(2) Digital circuit: Its main function is to control the output of V and I. It also uses the internal monitor circuit to read and monitor the actual size of V and I.



(3) Analog circuit (power stage): The core uses a switch with correct feedback circuit to control the actual output of V and I.

(4) DC source +56Vdc: Its main function is to provide the 52912 a stable DC voltage input source (+56Vdc), and DC source +56Vdc is built-in 52914 .

6.3 Front Panel Connector

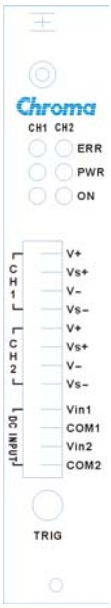


Figure 6-5 The Panel of 52912 PXI Card

Pin description of Front connector:

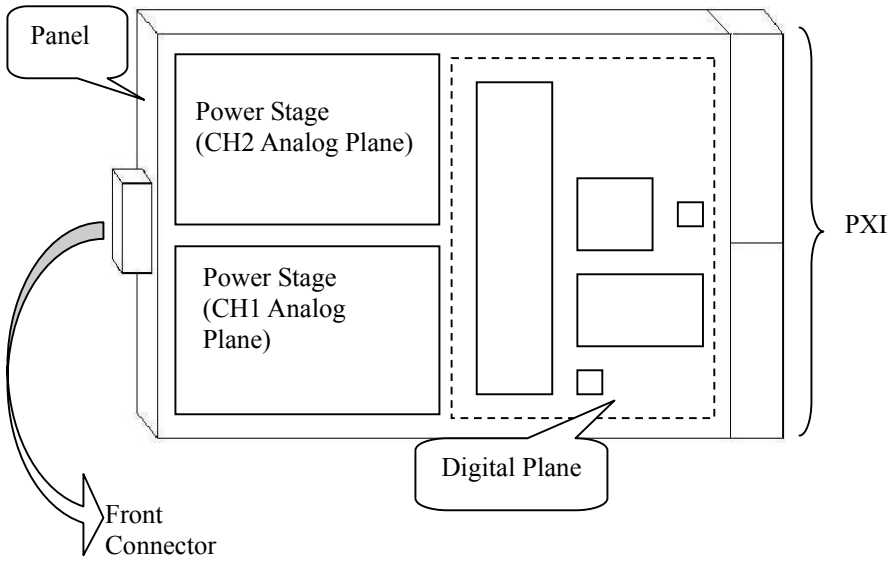
		Description
Channel 1	V+	channel 1 output V +
	Vs+	channel 1 output Vsense +
	V-	channel 1 output V -
	Vs-	channel 1 output Vsense -
Channel 2	V+	channel 2 output V +
	Vs+	channel 2 output Vsense +
	V-	channel 2 output V -
	Vs-	channel 2 output Vsense -
DC input	Vin1	56V input + of channel 1
	COM1	56V input ground of channel 1
	Vin2	56V input + of channel 2
	COM2	56V input ground of channel 2
TRIG		External Trigger Signal in

Description of each signal function:

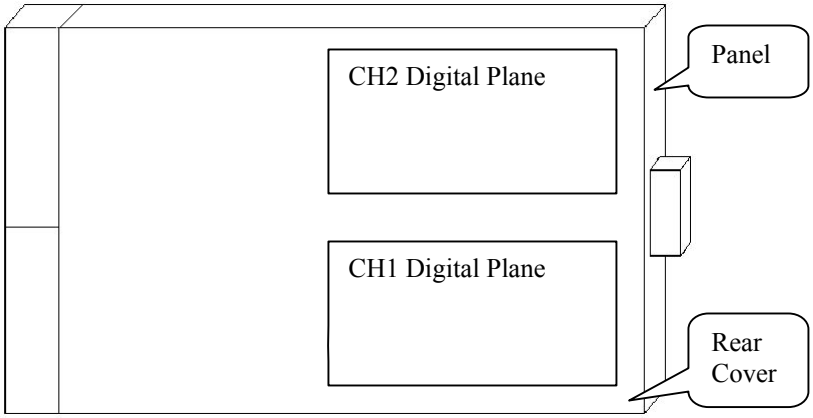
(1) When +56Vdc is input to 52912 PXI card, the PWR signal LED will light up in green.
(2) When the voltage is outputting, the ON signal LED will light up in green.
(3) When OVP or OTP occurs, the ERR signal LED will light up in red.

6.4 Outline

Front:



Rear:



6.5 Specification

Model	52912/52914
Dimensions	3U PXI (1 slot for 52912 / 3 slot for 52914)
Output Voltage	Channel # 1: 0-48VDC, 2 Amp, 60W Channel # 2: 0-48VDC, 2 Amp, 60W
Voltage Accuracy	0.5% of programmed value +/- 50mV
Voltage setting resolution	12 Bits
Fault Protection	Over-voltage Over-temperature Short Circuit
PARD (20MHz)	<200mV PK-PK
Line Regulation	0.01% +0.01%FS
Load Regulation	0.01% +0.02%FS
Current Limit Accuracy	0.5% +/- 50mA (12 Bits Resolution)
Read Back Voltage and Current	V: 0.2% +0.2%FS I: 0.5% +0.5%FS
Input	
DC Input	Isolated +56VDC (dual)

AC Input (w/Adapter A529102)	115V or 220VAC/50Hz or 60Hz
Software API	VISA compatible via National Instrument's VISA 2.5 or above 20 Windows DLL's API
PCI Data Bus	PCI V2.2 compliant, 33MHz, 32 Bits
Operating Temperature	-20 ~ 60 deg C (ambient air)
Operating Humidity	10% ~ 90 % relative
Storage Temperature	-30 ~ 70 deg C
Isolation	
Channel to Channel	500V DC
Channel to Chassis	500V DC
Standards	PXISA PXI® 1.0 PICMG 2.0 R3.0 CompactPCI®

6.6 Calibration

To ensure the accuracy of output of 52912/52914, calibration once in a year is strong recommended. To do the calibration, it is necessary to send 52912/52914 back to customer service of Chroma. Please contact your retailer for details.