

# Series 3700A System Switch/Multimeter

## User's Manual

3700AS-900-01 Rev. A / July 2011

**KEITHLEY**

A GREATER MEASURE OF CONFIDENCE

# Series 3700A

## System Switch/Multimeter

### User's Manual

© 2011, Keithley Instruments, Inc.  
Cleveland, Ohio, U.S.A.  
All rights reserved.

Any unauthorized reproduction, photocopy, or use the information herein, in whole or in part,  
without the prior written approval of Keithley Instruments, Inc. is strictly prohibited.

All Keithley Instruments product names are trademarks or registered trademarks of Keithley  
Instruments, Inc. Other brand names are trademarks or registered trademarks of their respective  
holders.

The Lua 5.0 software and associated documentation files are copyright © 1994-2008, Tecgraf,  
PUC-Rio. Terms of license for the Lua software and associated documentation can be accessed at  
[the Lua licensing site \(<http://www.lua.org/license.html>\).](http://www.lua.org/license.html)

Keithley's standard terms and conditions of sale in effect at the time of acceptance of buyer's order by Keithley  
shall apply to all purchase of goods and performance of services from Keithley, to the exclusion of any additional  
or different terms and conditions, including any terms or conditions which buyer may purport to apply under any  
buyer's request for quotation, purchase order or similar document, or which buyer may offer in response to these  
terms. A copy of Keithley's current terms can be accessed at  
<http://www.keithley.com/company/termsandconditions> (these "Terms"). To obtain a printed copy of these Terms,  
please contact your local sales office or send an email to [orders@keithley.com](mailto:orders@keithley.com). Buyer's assent to these Terms,  
and only these Terms, shall be conclusively presumed from buyer's acceptance of delivery of the products and/or  
services provided by Keithley.

# Safety Precautions

The following safety precautions should be observed before using this product and any associated instrumentation. Although some instruments and accessories would normally be used with nonhazardous voltages, there are situations where hazardous conditions may be present.

This product is intended for use by qualified personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product. Refer to the user documentation for complete product specifications.

If the product is used in a manner not specified, the protection provided by the product warranty may be impaired.

The types of product users are:

**Responsible body** is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring that operators are adequately trained.

**Operators** use the product for its intended function. They must be trained in electrical safety procedures and proper use of the instrument. They must be protected from electric shock and contact with hazardous live circuits.

**Maintenance personnel** perform routine procedures on the product to keep it operating properly, for example, setting the line voltage or replacing consumable materials. Maintenance procedures are described in the user documentation. The procedures explicitly state if the operator may perform them. Otherwise, they should be performed only by service personnel.

**Service personnel** are trained to work on live circuits, perform safe installations, and repair products. Only properly trained service personnel may perform installation and service procedures.

Keithley Instruments products are designed for use with electrical signals that are rated Measurement Category I and Measurement Category II, as described in the International Electrotechnical Commission (IEC) Standard IEC 60664. Most measurement, control, and data I/O signals are Measurement Category I and must not be directly connected to mains voltage or to voltage sources with high transient over-voltages. Measurement Category II connections require protection for high transient over-voltages often associated with local AC mains connections. Assume all measurement, control, and data I/O connections are for connection to Category I sources unless otherwise marked or described in the user documentation.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30 V RMS, 42.4 V peak, or 60 VDC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000V, no conductive part of the circuit may be exposed.

Do not connect switching cards directly to unlimited power circuits. They are intended to be used with impedance-limited sources. NEVER connect switching cards directly to AC mains. When connecting sources to switching cards, install protective devices to limit fault current and voltage to the card.

Before operating an instrument, ensure that the line cord is connected to a properly-grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before: connecting or disconnecting cables or jumpers, installing or removing switching cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

The instrument and accessories must be used in accordance with its specifications and operating instructions, or the safety of the equipment may be impaired.

Do not exceed the maximum signal levels of the instruments and accessories, as defined in the specifications and operating information, and as shown on the instrument or test fixture panels, or switching card.

When fuses are used in a product, replace with the same type and rating for continued protection against fire hazard.

Chassis connections must only be used as shield connections for measuring circuits, NOT as safety earth ground connections.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

If a  screw is present, connect it to safety earth ground using the wire recommended in the user documentation.

The  symbol on an instrument means caution, risk of danger. The user should refer to the operating instructions located in the user documentation in all cases where the symbol is marked on the instrument.

The  symbol on an instrument means caution, risk of electric shock. Use standard safety precautions to avoid personal contact with these voltages.

The  symbol on an instrument shows that the surface may be hot. Avoid personal contact to prevent burns.

The  symbol indicates a connection terminal to the equipment frame.

If this  symbol is on a product, it indicates that mercury is present in the display lamp. Please note that the lamp must be properly disposed of according to federal, state, and local laws.

The **WARNING** heading in the user documentation explains dangers that might result in personal injury or death. Always read the associated information very carefully before performing the indicated procedure.

The **CAUTION** heading in the user documentation explains hazards that could damage the instrument. Such damage may invalidate the warranty.

Instrumentation and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

To maintain protection from electric shock and fire, replacement components in mains circuits — including the power transformer, test leads, and input jacks — must be purchased from Keithley Instruments. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. Other components that are not safety-related may be purchased from other suppliers as long as they are equivalent to the original component (note that selected parts should be purchased only through Keithley Instruments to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call a Keithley Instruments office for information.

To clean an instrument, use a damp cloth or mild, water-based cleaner. Clean the exterior of the instrument only. Do not apply cleaner directly to the instrument or allow liquids to enter or spill on the instrument. Products that consist of a circuit board with no case or chassis (e.g., a data acquisition board for installation into a computer) should never require cleaning if handled according to instructions. If the board becomes contaminated and operation is affected, the board should be returned to the factory for proper cleaning/servicing.

# Table of Contents

---

<b>Introduction .....</b>	<b>1-1</b>
Welcome .....	1-1
Contact information .....	1-1
Extended warranty .....	1-1
Introduction to this manual .....	1-2
CD-ROM contents.....	1-2
Measuring capabilities.....	1-3
Safety precautions for connections.....	1-3
 <b>Using the front-panel interface.....</b>	 <b>2-1</b>
Introduction .....	2-1
Front panel overview for Series 3700A models .....	2-2
(1) The USB port.....	2-2
(2) The display .....	2-3
(3) The navigation wheel .....	2-5
(4) The POWER key .....	2-5
(5) The status lights.....	2-5
(6) The setup and control keys.....	2-5
Channel identification .....	2-7
Matrix card channel specifiers.....	2-7
Analog backplane relay channel specifiers .....	2-7
Multiplexer, digital I/O, totalizer, and DAC channel specifiers .....	2-8
Operating a channel from the front panel .....	2-8
Viewing the close or open status of a channel .....	2-9
Emulate a Model 3706 .....	2-10
Channel patterns.....	2-10
Create a channel pattern.....	2-10
Exercise: Create channel pattern "TEST01" from closed channels .....	2-11
Performing close and open operations on channel patterns .....	2-11
Exercise: Close channel pattern "TEST01" from the front panel .....	2-12
Set up row, column, and channel labels.....	2-12
Exercise: Create a channel label using the front panel .....	2-12
Front-panel scanning .....	2-12
Exercise: Perform a scan and save to a USB flash drive .....	2-13
Taking measurements without a switch card .....	2-14
Pseudocards .....	2-14
Using the analog backplane connector to take measurements .....	2-14
 <b>Using the web interface.....</b>	 <b>3-1</b>
Introduction .....	3-1
Connect to the instrument web interface .....	3-1

Web interface home page .....	3-2
Log in to the instrument.....	3-2
Card pages.....	3-3
Open and close channels from the card pages .....	3-3
Set up channel patterns from the web interface .....	3-6
Scan Builder page.....	3-8
Create a scan list .....	3-8
Run the scan.....	3-9
Stop the scan .....	3-9
Clear the scan list from the web interface .....	3-9
Set up simple triggers .....	3-9
TSB Embedded.....	3-11
Create a script using TSB Embedded .....	3-11
<b>Applications .....</b>	<b>4-1</b>
About the Series 3700A examples.....	4-1
Scanning examples.....	4-1
Example: Scan the card with no measurement (switch-only).....	4-1
Example: Scan the card and take measurements.....	4-2
Example: Take 4-wire ohm measurements using a background scan .....	4-3
Example: Scan, measure and store .....	4-3
Example: Optimize scan for speed.....	4-4
Example: Fast DC volt measurement.....	4-5
Example: Fast alternating scan, DC volts, and two-wire ohms.....	4-8
Example: Thermocouple temperature measurement from the front panel .....	4-9
Example: Thermocouple measurement through the remote interface.....	4-10
Example: TSP-Link and interactive triggers.....	4-10
Set up communication.....	4-11
Program code.....	4-12
Program code to run the test.....	4-14
Example: Using the scanning and triggering model .....	4-15
Set up communication.....	4-15
Program code.....	4-16
Program code to run the test.....	4-18
Using background scans for longer scan lists .....	4-18
Example: IEEE-1588 in Series 3700A-based systems.....	4-20
Scheduling alarms.....	4-20
Scheduling alarms on a stand-alone Series 3700A.....	4-21
Example: External DMM and switch triggering .....	4-22
Example: DMM buffer statistics with interactive operation .....	4-25
Example: Commonsode ohm measurement with Model 3721 .....	4-27
<b>Troubleshooting FAQs .....</b>	<b>5-1</b>
About this section.....	5-1
Need different line frequency or voltage .....	5-1
Switch matrix does not recognize my card .....	5-1
The internal web page of the instrument is not accessible .....	5-2

Error when I try to close a channel .....	5-2
Switch relays will not close .....	5-3
Save the present state of the instrument .....	5-3
LabVIEW driver for the Series 3700A .....	5-3
<b>Next steps.....</b>	<b>6-1</b>
Next steps .....	6-1
<b>Index.....</b>	<b>Index-1</b>

# Section 1

---

## Introduction

### In this section:

---

Welcome .....	1-1
Contact information .....	1-1
Extended warranty .....	1-1
Introduction to this manual .....	1-2
CD-ROM contents .....	1-2
Measuring capabilities .....	1-3
Safety precautions for connections .....	1-3

## Welcome

Thank you for choosing a Keithley Instruments product. The Series 3700A System Switch/Multimeter features scalable, instrument grade switching and multi-channel measurement solutions that are optimized for automated testing of electronic products and components. The Series 3700A includes four versions of the Model 3706A system switch mainframe, along with a growing family of plug-in switch and control cards. When the Model 3706A mainframe is ordered with the high performance multimeter, you receive a tightly-integrated switch and measurement system that can meet the demanding application requirements in a functional test system or provide the flexibility needed in stand-alone data acquisition and measurement applications.

## Contact information

If you have any questions after reviewing this information, please contact your local Keithley Instruments representative or call Keithley Instruments corporate headquarters (toll-free inside the U.S. and Canada only) at 1-888-KEITHLEY (1-888-534-8453), or from outside the U.S. at +1-440-248-0400. For worldwide contact numbers, visit the [Keithley Instruments website](http://www.keithley.com) (<http://www.keithley.com>).

## Extended warranty

Additional years of warranty coverage are available on many products. These valuable contracts protect you from unbudgeted service expenses and provide additional years of protection at a fraction of the price of a repair. Extended warranties are available on new and existing products. Contact your local Keithley Instruments representative for details.

## Introduction to this manual

This manual provides detailed tutorials to help you achieve success with your Keithley Instruments Series 3700A System Switch/Multimeter. In addition, the basics of the two simplest interfaces, the front panel and the web interface, are provided to familiarize you with the instrument. Select and run all examples in Sections 5, 6, and 7 that are relevant to your intended use and to the equipment you are using.

Some of the examples in this manual may use unfamiliar commands and concepts. For detailed information about these, refer to the Reference Manual (part number 3700AS-901-01) on the Product Information CD-ROM that came with your instrument.

## CD-ROM contents

Two CD-ROMs are shipped with each Series 3700A order. The Series 3700A Quick Start Guide, User's Manual, Reference Manual, and Switch Card Manual are provided in PDF format on the Series 3700A Product Information CD-ROM.

- **Quick Start Guide:** Provides unpacking instructions, describes basic connections, and reviews basic operation information. If you are new to Keithley Instruments equipment, refer to the Quick Start Guide to take the steps needed to unpack, set up, and verify operation.
- **User's Manual:** Provides application examples. If you need a starting point to begin creation of applications, refer to the User's Manual for a variety of specific examples.
- **Reference Manual:** Includes advanced operation topics and maintenance information. Programmers looking for a command reference, and users looking for an in-depth description of the way the instrument works (including troubleshooting and optimization), should refer to the Reference Manual.
- **Switching and Control Cards Reference Manual:** Contains information specific to the switch cards that can be used with the Series 3700A.
- **Additional product information:** The product data sheet, product specifications, and rack-mount kit instructions are also included on the CD-ROM.

A second CD-ROM contains the Test Script Builder script development software (Keithley Instruments part number KTS-850). Use this CD-ROM to install the Test Script Builder Integrated Development Environment. This software provides an environment to develop a test program and the ability to load the test program onto the instrument. Running a program loaded on the instrument eliminates the need to send individual commands from the host computer to the instrument when running a test.

The second CD-ROM also includes:

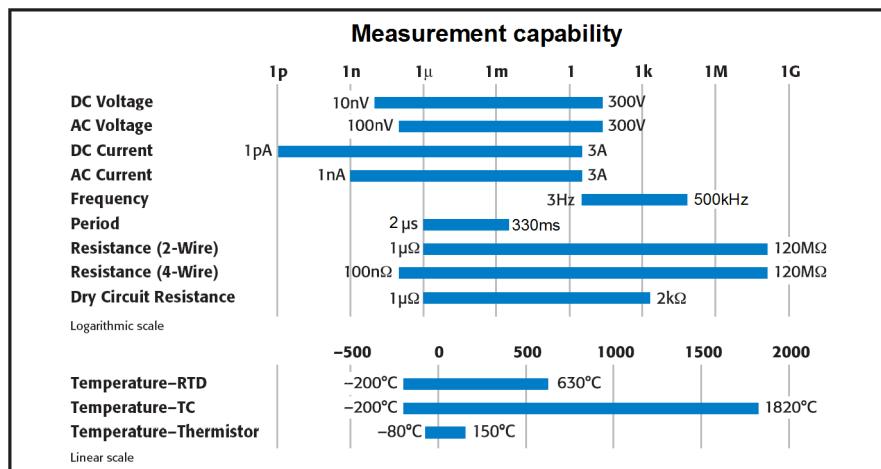
- The 3700A TSB Add-in, which is a software tool you can use to create, modify, debug, and store Test Script Processor (TSP®) test scripts.
- IVI Instrument Driver, driver for National Instruments LabVIEW™, and related release notes.
- J2SE™ Runtime Environment.
- Keithley I/O layer and release notes.
- Keithley LXI Discovery Utility.

For the latest drivers and additional support information, see <http://www.keithley.com> (<http://www.keithley.com>).

## Measuring capabilities

The basic measurement capabilities of Series 3700A systems are summarized in the following figure.

**Figure 1: DMM measurement capabilities**



## Safety precautions for connections

### **WARNING**

**Connection information for switching cards is intended for qualified service personnel. Do not attempt to connect DUT or external circuitry to a switching card unless qualified to do so.**

**To prevent electric shock that could result in serious injury or death, comply with these safety precautions:**

**Before making or breaking any connections to the switching card, make sure the Series 3700A is turned off and power is removed from all external circuitry.**

**Do not connect signals that will exceed the maximum specifications of any installed switching card.**

**If both the rear analog backplane connector of the Series 3700A and the switching card terminals are connected at the same time, the test lead insulation must be rated to the highest voltage that is connected. For example, if 300V is connected to the analog backplane connector, the test lead insulation for the switching card must also be rated for 300V.**

**Dangerous arcs of an explosive nature in a high-energy circuit can cause severe personal injury or death if contacted. If the multimeter is connected to a high-energy circuit when set to a current range, low resistance range, or any other low-impedance range, the circuit is virtually shorted.**

**Dangerous arcing can result (even when the multimeter is set to a voltage range) if the minimum voltage spacing is reduced in the external connections. For details about how to safely make high energy measurements, see *High-energy circuit safety precautions in the Series 3700A Reference Manual*.**

**As described in the International Electrotechnical Commission (IEC) Standard IEC 664, the Series 3700A is Installation Category I and must not be connected to mains.**

## Section 2

# Using the front-panel interface

### In this section:

Introduction .....	2-1
Front panel overview for Series 3700A models .....	2-2
Channel identification .....	2-7
Operating a channel from the front panel .....	2-8
Viewing the close or open status of a channel .....	2-9
Emulate a Model 3706 .....	2-10
Channel patterns .....	2-10
Front-panel scanning .....	2-12
Taking measurements without a switch card.....	2-14

## Introduction

Before starting this section, complete the tasks outlined in the Model 3706A Quick Start Guide. Once you have completed those tasks, read this section, which provides enough basic information about the Model 3706A front-panel interface to work through the examples provided in this manual.

The Keithley Instruments Models 3706A and 3706A-S front panels contain the following items:

- [\(1\) The USB port](#) (on page 2-2)
- [\(2\) The display](#) (on page 2-3)
- [\(3\) The navigation wheel](#) (on page 2-5)
- [\(4\) The POWER key](#) (on page 2-5)
- [\(5\) The status lights](#) (on page 2-5)
- [\(6\) The setup and control keys](#) (on page 2-5)

For these models, you can use the keys, display, and the navigation wheel ☺ to access, view, and edit the menu items and set values.

### NOTE

The Models 3706A-NFP and 3706A-SNFP front panels do not have a display, navigation wheel, and setup and control keys. These models are configured and controlled through a remote interface. For more information, see [Using the web interface](#) (on page 3-1) in this manual, and refer to the Model 3706A Reference Manual for detailed information about instrument programming and remote commands.

## Front panel overview for Series 3700A models

The Series 3700A includes several models that support different features. The following figures show the front panels of each of the models; a brief description of the features follows the figures.

For more detailed information about the Model 3706A front panel, see the “General operation” section in the Model 3706A Reference Manual.

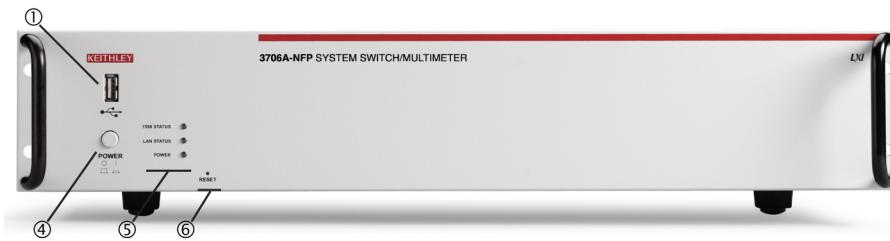
**Figure 2: Model 3706A with DMM front panel**



**Figure 3: Model 3706A-S front panel (no DMM)**



**Figure 4: Models 3706A-NFP (with DMM) and 3706A-SNFP (no DMM) front panel**



### (1) The USB port



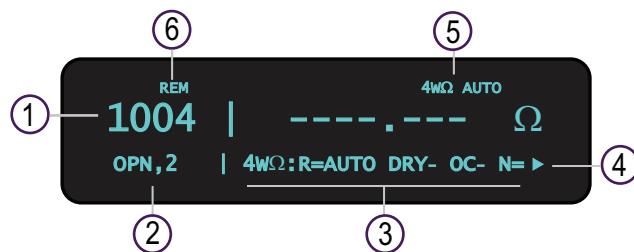
Use the front-panel USB port to connect a USB flash drive. The USB flash drive can be used to store reading buffer data, scripts, and user setup options.

## (2) The display

During setup, the display shows menu choices that you can use to configure the instrument. See "Menu overview" in the Series 3700A Reference Manual for more information about Series 3700A menus.

During operation, the display provides information about the selected channel, channel pattern, channel state, and errors. It also shows the control status (local or remote). If REM is displayed, the instrument is presently controlled through a remote interface (GPIB, LAN, or USB). If REM is not displayed, control is through the front panel. The following figure shows an example of the Series 3700A during operation.

**Figure 5: Series 3700A display during operation**



Series 3700A display during operation	
1	Active channel (slot 1, channel 004)
2	Channel state (open, 2-pole operation)
3	Present state of the DMM attributes for displayed channel: <ul style="list-style-type: none"> <li>The 4-WΩ and autorange are enabled</li> <li>Dry-circuit ohms is disabled (DRY-)</li> <li>Offset compensation is off (OC-)</li> </ul> For detailed descriptions of the DMM attribute symbols, see the table labeled "DMM attribute symbols" in the "Front-panel operation" topic in the Series 3700A Reference Manual
4	Arrow indicating that more menu items exist; turn the navigation wheel  to the left or right to see the additional items.
5	The 4-WΩ and autorange is enabled
6	Indicates the instrument is in remote operation rather than local operation

The table below lists the display indicators and what they mean.

Indicator	Meaning
AUTO	Measure autorange is selected
EDIT	Instrument is in the editing mode
ERR	Questionable reading or invalid calibration step
FILT	Digital filter is enabled
LSTN	Instrument is addressed to listen
MATH	Enabled for mX+b, percent, or reciprocal (1/X) calculation
REL	Relative mode is enabled
REM	Instrument is in remote mode
SRQ	Service request is asserted
TALK	Instrument is addressed to talk
TRIG	Instrument is processing a front-panel reading request
4W	Four-wire resistance or RTD temperature reading
* (asterisk)	Readings are being stored in the buffer

The bottom left line of the display contains the DMM attribute symbols. The symbols that appear are dependent on whether the attribute exists for the selected function. The following table indicates the DMM attribute symbols that may appear on the front panel. If the symbol has a value associated with it, the third column in the table indicates the value definition.

#### DMM attribute symbols

Front-panel DMM attribute	Symbol	Values
range	R=	AUTO or n, where n equals the range
nplc	N=	n, where n equals the NPLC
auto delay	AD	+ for ON, 1 for ONCE, or 0 for OFF
auto zero	AZ	+ for ON or – for OFF
line sync	LS	+ for ON or – for OFF
limit	LIM	+ for a limit enabled or – for limits disabled
detector bandwidth	DBW	3, 30, or 300
threshold	THR=	n, where n indicates the threshold
aperture	A=	n, where n indicates the aperture setting
dry circuit	DRY	+ for ON or – for OFF
offset compensation	OC	+ for ON or – for OFF
thermocouple sensor K	K_T/C	N/A
thermocouple sensor T	T_T/C	N/A
thermocouple sensor E	E_T/C	N/A
thermocouple sensor R	R_T/C	N/A
thermocouple sensor S	S_T/C	N/A
thermocouple sensor B	B_T/C	N/A
thermocouple sensor N	N_T/C	N/A
theristor	THRM	N/A
three-wire RTD	3RTD	N/A
4-wire RTD	4RTD	N/A
simulated reference junction	RJ_SIM	N/A
internal reference junction	RJ_INT	N/A
external reference junction	RJ_EXT	N/A

### (3) The navigation wheel



Turn the navigation wheel to scroll to the desired menu option or to change the value of the selected numeric parameter. Pressing the navigation wheel has the same functionality as pressing the **ENTER** key.

When changing a multiple character value, such as an IP address or channel pattern name, press the navigation wheel to enter edit mode, rotate the navigation wheel to change the characters value as desired, but do not leave edit mode. Use the **CURSOR** keys to scroll to the other characters and use the navigation wheel to change their value as needed. Press the **ENTER** key when finished changing all the characters.

### (4) The POWER key



**POWER** Power switch. The in position turns the Series 3700A on (I); the out position turns it off (O).

### (5) The status lights

The Series 3700A has three status lights on the front panel.



The 1588 status light indicates 1588 operation. If this light is off, the 1588 feature is disabled or improperly configured. If the light blinks at a one second rate, the instrument is the 1588 master. If the instrument is a slave, the light will not blink.



The LAN status light is lit when the instrument is connected through the local area network (LAN) with no errors. If this is not lit, the instrument is not connected through the LAN or there is a connection problem.



If you are using the web interface, the LAN Status light blinks when you click **ID**.

The POWER light is lit when power is applied to the instrument.

### (6) The setup and control keys

The setup and control keys provide front panel control and configuration. The following figure illustrates each key's location. The table following the figure contains a definition of each key. For detailed information about front-panel keys, refer to the "Menu overview" topic in the Model 3706A Reference Manual.

Figure 6: Model 3706A with DMM setup and control keys



**Key descriptions**

<b>Key</b>	<b>Description</b>
DISPLAY	Toggles between the channel measure display, closed channels list, and the user message mode
CONFIG	Configures a function or operation
RESET	Restores factory default LAN settings
OPEN ALL	Opens all closed channels
STEP	Steps through each step in a predefined scan list
OPEN	Opens the selected channel or channel pattern
CLOSE	Closes the selected channel or channel pattern
STORE	Selects, clears, and saves reading buffer data and creates and deletes reading buffers
REC	Displays stored readings for selected reading buffer
RATE	Sets measurement speed (fast, medium, or slow) for the active or selected function
FUNC	Changes to the next function in a factory-defined list of available DMM functions
CHAN	If a channel is displayed, opens and closes channels; if a pattern is displayed, switches to channel view
PATT	If a pattern is displayed, manages, opens and closes, and resets patterns; if a channel is displayed, changes to display a pattern
SLOT	Displays information about the installed cards and the instrument, including firmware revision, model name, and model number
SCAN	Allows you to run, manage, view, and reset scan lists
DMM	Opens the DMM ACTION menu
LIMIT	Puts cursor in the compliance limit field for editing and selects the limit value to edit
REL	Controls relative measurements; allows baseline value to be subtracted from reading
FILTER	Enables or disables the digital filter; you can use this filter to reduce reading noise
LOAD	Loads test for execution
RUN	Runs the last selected user-defined test code
INSERT	Appends the present channels to the scan list
DELETE	Deletes channels or channel patterns from the scan list
TRIG	Triggers readings
MENU	Accesses the main menu, which can be used to configure instrument operation
EXIT (LOCAL)	Cancels selection and backs out of the menu structure; also used as a <b>LOCAL</b> key to take the unit out of remote operation
ENTER	Accepts the current selection or brings up the next menu option; pressing this key is often the same as pressing the navigation wheel
AUTO	Enables or disables autorange for the selected function
RANGE (up and down arrows)	Selects the next higher or lower measurement range on the measurement display for the selected function.
CURSOR	Moves the cursor left or right for editing and scrolls through menu options

## Channel identification

### Matrix card channel specifiers

The channels on the matrix cards are referred to by their slot, bank, row, and column numbers:

- **Slot number:** The number of the slot in which the card is installed.
- **Bank number:** The bank number, if used by your card. See your card documentation.
- **Row number:** The row number is either 1 to 8 or A to Z. See your card documentation.

**Column number:** Always two digits. For columns greater than 99, use A, B, C and so on to represent 10, 11, 12, ...; the resulting sequence is: 98, 99, A0, A1, ..., A8, A9, B0, B1, ...

#### Matrix channel examples

Specifier	Slot number	Bank number	Row number	Column number
1104	1	N/A	1	04
11104	1	1	1	04
1203	1	N/A	2	03
213A4	2	1	3	104
3112	3	N/A	1	12
62101	6	2	1	01

### Analog backplane relay channel specifiers

The channels for slots with analog backplane relays are referred to by their slot, backplane, bank, and relay numbers:

- **Slot number:** The number of the slot.
- **Backplane number:** Always 9.
- **Bank number:** The bank number, if used by your card. See your card documentation for detail.
- **Analog backplane relay number:** The number of the backplane relay. Typically 1 to 6. See your card documentation for detail.

#### Backplane relay examples

Specifier	Slot number	Backplane number	Bank number	Backplane relay number
1914	1	9	1	4
1922	1	9	2	2
2924	2	9	2	4
3916	3	9	1	6

## Multiplexer, digital I/O, totalizer, and DAC channel specifiers

The channels for multiplexer (MUX), digital I/O, totalizer, and digital to analog converter channels are referred to by their slot and channel numbers:

- **Slot number:** The number of the slot in which the card is installed.
- **Channel number:** The number of the channel (always three digits).

Specifier	Slot number	Channel number
1004	1	004
2050	2	050
3012	3	012
3003	3	003
2007	2	007
1020	1	020

## Operating a channel from the front panel

### CAUTION

Hot switching can dry-weld reed relays, causing them to always be on. Hot switching is recommended only when external protection is provided.

You can perform operations on a single channel from the front panel.

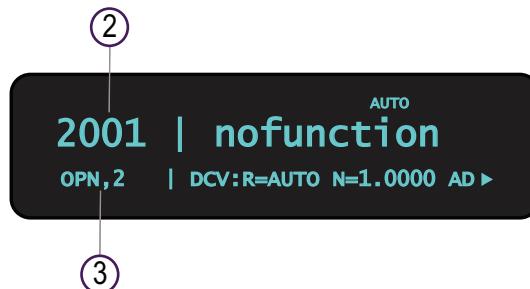
**To select a channel:**

1. If the instrument is being controlled remotely, press **EXIT** to allow control from the front panel.
2. Press the navigation wheel  to select the first digit of the channel specifier, which is the slot number . The digit flashes, which indicates that it can be edited.

Figure 7: Select a channel from the Series 3700A front panel



3. To change to a different slot number, turn the navigation wheel ① until the slot number you want is displayed.
4. Press navigation wheel ①.
5. If your card supports banks, the next number you can select is the bank number. Set this as needed using the navigation wheel ①.
6. Set the channel number (or rows and columns for installed matrix cards) as needed using the navigation wheel ② ②.
7. The display shows the current state of the selected channel in the bottom row ③. In this example, the channel is open and 2-pole (if you see : followed by a channel specifier, you selected a range; press **EXIT** to return to the main display and reselect your channel).

**Figure 8: Series 3700A selected channel state**

8. To:
  - Close a channel without affecting any other channels: Select **CLOSE**.
  - Open the channel: Press **OPEN**.
  - Close a channel and open any other closed channels on the instrument: Select **CHAN** and select **EXCLOSE**. Press **ENTER** to close the selected channels.
  - Close a channel and open any other closed channels on the slot that contains the selected channel: Select **CHAN**, and then select **EXSLOTCLOSE**. Press **ENTER** to close the selected channels.

### NOTE

Once a channel is selected, it is the selected channel for any subsequent front-panel operations.

## Viewing the close or open status of a channel

Closed channels are shown separated by commas after the **CLS:** characters on the display of the instrument. If no channels are closed, **<none>** is displayed here. If the list of closed channels extends past one screen, **...** is displayed at the end of the line. Press **DISPLAY** to display the full list of closed channels. Use the navigation wheel ① to scroll through the list. After viewing the list, press **DISPLAY** twice to return to the main display.

**Figure 9: 3700A multiple closed channels**

## Emulate a Model 3706

If you replace a Model 3706 with a Model 3706A in a system where computer drivers may be querying the model, you may need to emulate a Model 3706. This can occur if you replace a Model 3706 with a Model 3706A in an existing system, or if you duplicate a system but use a Model 3706A instead of a Model 3706.

When this is set to emulate a Model 3706, the model number is reported as a 3706 when you send a request with a command such as `localnode.model` or `*idn?`. This allows drivers that query the model number to continue to operate normally.

This setting is maintained through an instrument reset and through power cycles.

**To emulate a Model 3706 using the front panel menus:**

1. If the instrument is being controlled remotely, press **EXIT** to allow control from the front panel.
2. Press **MENU**.
3. Use navigation wheel  to display and select **EMULATION**.
4. Select **3706**.
5. Press **ENTER**.

You can also use the remote command `localnode.emulation` to set emulation.

### NOTE

All other Model 3706A behavior is the same. Emulation mode does not affect the changes to the IEEE-1588 features or the response times that occurred with the update from the Model 3706 to the Model 3706A.

## Channel patterns

You can use channel patterns as a convenient way to refer to a group of switching channels and backplane relays with a single alphanumeric name. When you perform close or open operations on a channel pattern, only the channels and analog backplane relays that are in the channel pattern are affected.

There is no speed difference when performing close and open operations on channel patterns compared to performing the same operations on individual channels or a list of channels.

### Create a channel pattern

When you create a channel pattern, make sure to:

- Include all the channels and backplane relays that are needed for that channel pattern.
- Check that channels contained in the pattern are correct.
- Check that channels contained in the pattern create the desired path connection.
- Make sure that channels that you want to include in the pattern are not set to forbidden to close.

When naming the channel pattern, be aware:

- The first character of the name must be alphabetic (upper or lower case)
- Names are case sensitive
- Pattern names must be different than channel labels

## Exercise: Create channel pattern "TEST01" from closed channels

**To create channel pattern "TEST01" from channels currently closed:**

1. Close the channels you want to include in the channel pattern. For this example, close channels **1001**, **1015**, and **1031**. Refer to [Operating a channel from the front panel](#) (on page 2-8) for instructions on how to close individual channels.
2. Press the **PATT** key.
3. From this menu, select the **CREATE** menu item.
4. From this menu, select the **SNAPSHOT** menu item.
5. At the prompt, enter a pattern name using the navigation wheel . For this example, enter the name **TEST01**. Use the navigation wheel to select each character of the name.
6. Press the **ENTER** key to apply the selection.
7. Press the **EXIT** key to leave the menu.

### NOTE

Refer to "Channel patterns" in the Series 3700A Reference Manual for more details.

## Performing close and open operations on channel patterns

### WARNING

**Careless channel pattern operation could create an electric shock hazard that could result in severe injury or death. Improper operation can also cause damage to the switching cards and external circuitry. The control of multiple channels using channel patterns should be restricted to experienced test engineers who recognize the dangers associated with multiple channel closures.**

You can close and open channel patterns the same way you do for individual channels.

When you request a close or open operation, the Series 3700A verifies that the channels exist for a pattern, but does not verify that the switch path connection is correct. You must ensure the requested operation is safe for a channel pattern and that a good connection will result for your application with the channel pattern.

## Exercise: Close channel pattern "TEST01" from the front panel

*To close channel pattern "TEST01" from the front panel:*

1. Press the **PATT** key to display a channel pattern.
2. Use the navigation wheel  $\odot$  to select **TEST01**, the channel pattern you want to close.
3. Perform any of the following actions:
  - Open the channels in the channel pattern: Press **OPEN**.
  - Close the channels in the channel pattern without affecting any other channels: Press **CLOSE**.
  - Close the channels in the channel pattern and open any other closed channels on the instrument: Select **PATT** and select **EXCLOSE**. Press **ENTER** to open or close the channels.
  - Close the channels in the channel pattern and open any other closed channels on the slot: Press **PATT** and select **EXSLOTCLOSE**. Press **ENTER** to open or close the channels.

## Set up row, column, and channel labels

You can define labels for channels. Using labels is a more descriptive way to refer to switching paths than the default channel identifiers.

Labels must be unique; they cannot have the same as the name of another channel or channel pattern. Labels cannot contain spaces, and they do not persist through a power cycle.

Channel labels can be up to 19 characters. You can set labels only for channels that are installed in the instrument.

## Exercise: Create a channel label using the front panel

*To create a channel label using the front panel:*

1. Press the **CONFIG** key, and then press the **CHAN** key. The CHANNEL ATTR MENU opens.
2. Select **LABEL**, and then press the navigation wheel  $\odot$ .
3. Press the navigation wheel  $\odot$  again to enter EDIT mode.
4. At the blinking cursor, turn the navigation wheel  $\odot$  to the right to select an alphanumeric character, or turn it to the left to select a numeral.

### NOTE

The cursor blinks for several seconds when you enter the EDIT mode; if the cursor stops blinking before you are done, press the navigation wheel again to resume EDIT mode.

5. Press the right cursor key to move to the next character to set.
6. Repeat steps 4 and 5 until you have added all the desired characters for the label.
7. Press the navigation wheel  $\odot$  to save the label, and then press the **EXIT** key to return to the default display.

## Front-panel scanning

A scan is a series of steps that opens and closes switches sequentially for a selected group of channels. During each step, actions occur, such as waiting for a trigger, taking a measurement, and completing a step count. Scans automate actions that you want to perform consistently and repeatedly on a set of channels.

## Exercise: Perform a scan and save to a USB flash drive

This exercise creates a ten-channel scan operation to measure voltage on ten sequential channels. A buffer is created to store the scan data to save to a standard USB flash drive. For this exercise, assume a known voltage (or short) is connected across channels 1-5 and 17-21 on a multiplexing switch card installed in slot 1.

### **To perform a scan operation and save data to a USB flash drive:**

1. Create a scan list.
  - a. Use the navigation wheel  $\odot$  to select and modify the start channel to **001**.
  - b. Use the navigation wheel  $\odot$  to select and modify the end channel to **005**.
  - c. Press the **INSERT** key to add more channels to the scan list.
2. Assign the default voltage DC DMM configuration to the scan list:
  - a. Use the navigation wheel  $\odot$  to select **dcvolts** as the measurement configuration.
  - b. Repeat the above steps, selecting channels **1017** (step a) to **1021** (step b).

### NOTE

In addition to the default DMM configurations, the scan list will also show the names of any user-created DMM configurations. For more information about user-created DMM configurations, see "Save DMM configurations" in the Series 3700A Reference Manual.

3. Create a buffer (required to save readings on the USB flash drive):

- a. Press the **STORE** key on the front panel.
  - b. Use the navigation wheel  $\odot$  to select **CREATE > fpbuf**.

### NOTE

The default buffer name is **fpbuf**, but can be modified using the navigation wheel  $\odot$  in edit mode. This example assumes the buffer name is **fpbuf**.

- c. Press **Enter** until CAPACITY 000010 is displayed.
  - d. Press **Enter**. The fpbuf MENU is displayed.
  - e. Use the navigation wheel  $\odot$  to choose **SELECT > fpbuf**, and then press **Enter**.
  - f. Press the **EXIT** key to return to the active channel display.

4. Run the scan:
  - a. Press the **SCAN** key.
  - b. Use the navigation wheel  $\odot$  to select **BACKGROUND**.
  - c. Press **Enter**.
5. Store the buffer data on a USB flash drive:
  - a. Insert the USB flash drive (not included) into the front-panel USB connector.
  - b. Press the **STORE** key to access the buffer menu.
  - c. Use the navigation wheel  $\odot$  to select **SAVE > USB**.
  - d. Use the navigation wheel  $\odot$  to select the default buffer name **fpbuf** and change it to a more meaningful name for storage on the USB flash drive.
  - e. Push the navigation wheel  $\odot$  until the Buffer menu is displayed. The data file is now stored in the root directory on the USB flash drive.

## Taking measurements without a switch card

### Pseudocards

You can perform open, close, and scan operations and configure your system without having an actual switching card installed in your instrument. Using the remote interface, you can assign a pseudocard to an empty switching card slot, allowing the instrument to operate as if a switching card were installed.

A pseudocard cannot be configured from the front panel. However, once the remote configuration is complete, you can take the instrument out of remote mode and use the front panel. Press the **EXIT** key to take the instrument out of remote mode.

When the instrument is turned off, the pseudocard is no longer assigned to the slot.

#### NOTE

A saved setup or created configuration script retains the model number of the card installed in each slot. The model number of a pseudocard is the same as the model number of an actual card (except for Model 3732 cards; see the "Pseudocard support for the Model 3732" topic in the Series 3700 Switch and Control Cards Reference Manual for details). This allows a saved setup or created configuration script to be recalled if the installed card (or pseudocard) matches the model number for the slot in the saved setup or created configuration script.

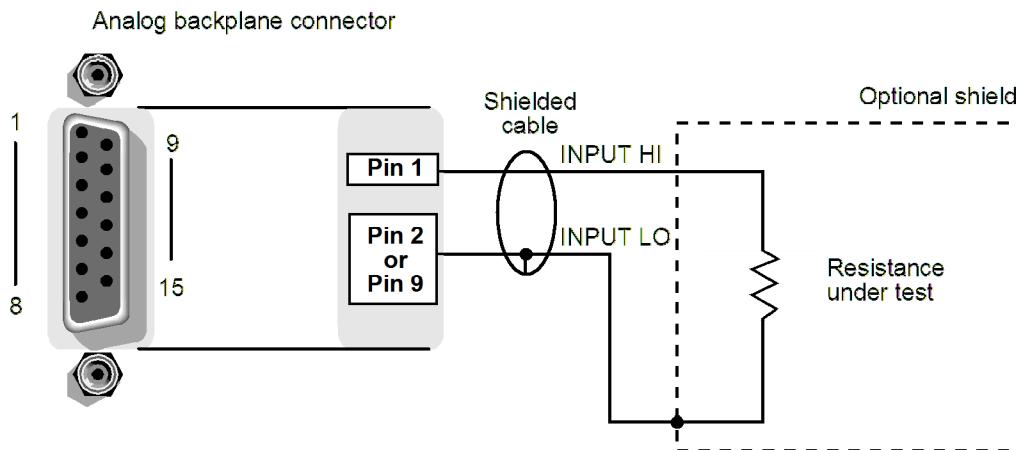
## Using the analog backplane connector to take measurements

You can perform measurements without an installed switch card by using the rear-panel analog backplane connection, which is always connected to the digital multimeter (DMM) input. To connect to the analog backplane as shown below, use a standard DB-15 mating connector (not supplied).

#### ⚠ WARNING

*The switch card and analog backplane connector cannot be used at the same time, because they are connected in parallel.*

*There is no relay that connects to the DB-15 connector.*

**Figure 10: Analog backplane connector**

**NOTE** Source current flows from the connector HI to the connector LO pins.

**To take a measurement using the analog backplane connector:**

1. Connect the resistor to the input terminals.
2. Configure the measurement by pressing the **FUNC** key until  $2\text{W}\Omega$  is displayed.
3. Press the **AUTO** key to enable the autorange feature. **AUTO** is displayed in upper right corner.
4. Press the **TRIG** key to trigger a reading.

## Section 3

# Using the web interface

### In this section:

Introduction .....	3-1
Connect to the instrument web interface .....	3-1
Web interface home page .....	3-2
Log in to the instrument.....	3-2
Card pages.....	3-3
Scan Builder page .....	3-8
TSB Embedded.....	3-11

## Introduction

The Series 3700A web interface allows you to review instrument status, control the instrument, and upgrade the instrument over a LAN connection.

The instrument web page resides in the firmware of the instrument. Changes you make through the web interface are immediately made in the instrument.

Many examples in this manual and in the Series 3700A Reference Manual can be run through the TSB Embedded page of the instrument web interface.

## Connect to the instrument web interface

### NOTE

The instrument web interface requires the web browser plug-in Java™ SE Runtime Environment Version 6 or later. The latest version of the plug-in is available from <http://www.java.com/en/download/manual.jsp> (<http://www.java.com/en/download/manual.jsp>).

Installation files are also available on the Series 3700A Product Information CD-ROM that came with your instrument.

The instrument web interface uses Java applets and, depending on your browser security settings, may require your permission to download and install them.

To connect to the instrument web interface, you must have a LAN connection from the computer to the instrument. See "LAN concepts and settings" in the Series 3700A Reference Manual for more information about configuring the Series 3700A for a LAN connection, connecting the Series 3700A to the LAN, and establishing a LAN connection to the instrument.

Once the Series 3700A is configured correctly and connected to the LAN, you can use the Keithley LXI® Discovery Utility to identify the IP addresses of LXI certified instruments that are set up for automatic IP address selection. Once identified, you can double-click the IP address in the Utility to open the web interface for the instrument. The Keithley LXI Discovery Utility is available on the [Keithley Instruments website](http://www.keithley.com) (<http://www.keithley.com>).

**To locate the Keithley LXI Discovery Utility on the Keithley website:**

1. Select the **Support** tab.
2. In the model number box, type **3700A**.
3. From the list, select **Software** and click the search icon. A list of software applications for the instrument is displayed.
4. See the readme file included with the application for more information.

For more information about the LXI Consortium, see the [LXI Consortium website](http://www.lxistandard.org/) (<http://www.lxistandard.org/>).

**Alternatively, use the following instructions to connect to the instrument web interface:**

1. Confirm the LAN Status light on the instrument is illuminated.
2. Open an Internet browser, such as Microsoft® Windows® Internet Explorer® (version 6.0 or later only).
3. If you do not know the IP address, press the **MENU** key on the instrument front panel and then select **LAN > STATUS > IP-ADDRESS**.
4. In the Address box of the Internet browser, enter the IP address of the instrument and press **Enter**.

The home page of the instrument web interface is displayed.

## Web interface home page

The home page of the web interface gives you basic information about the instrument, including:

- The instrument model, serial number, firmware revision, and LXI information
- A list of slots and the switch cards that are installed in each slot
- An **ID** button to help you locate the instrument
- Links to the instrument web applications, including TSB Embedded.

## Log in to the instrument

The web interface has both interactive and read-only pages. These pages are always listed in the navigation panel on the left side of the web interface. You can review information on any of the pages without logging in, but to change information, you must log in.

Pages that contain information you can change include a **Login** button. Once you have logged in to one page of the web interface, you do not need to log in again unless you reload the page.

**To log into the instrument:**

1. Open a page that contains a **Login** button, such as one of the Cards pages, Scan Builder, or TSB Embedded.

**Figure 11: Web interface login**



2. Click **Login**. The login dialog box is displayed.

3. Enter the password (the default is **admin**).

**Figure 12: 3700A Enter web interface password**



4. Click **Login**.

### NOTE

The default password is **admin**. If the password has been changed, it is available from the front panel of the instrument. Press **MENU > LAN > STATUS > PASSWORD**.

## Card pages

The card pages are interactive pages where you can work with channels in each slot.

To open a card page, on the left navigation, click the slot number.

There is a specific page for each card installed in the mainframe. The page displays a grid that shows the relay configuration of the switch card.

**Figure 13: Series 3700A web interface Cards page**



## Open and close channels from the card pages

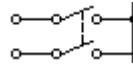
You can open and close channels from the card pages in several ways.

The simplest method is to click a connection. The channel changes state to open or closed. When the channel is open, the connection will look similar to one of the following graphics (the actual item on the web interface depends on the installed card):

**Figure 14: Series 3700A web interface relay open**



**Figure 15: Series 3700A web interface relay open**

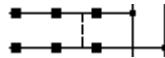


When the channel is closed, the connection will look similar to one of the following:

**Figure 16: Web interface closed channel**



**Figure 17: Web interface relay closed**



To specify the type of close, select a Channel Action Type from the box in the upper right before closing a channel. The options are:

- **Channel Close:** Close the selected channel without affecting the state of any other channels.
- **Exclusive Slot Close:** Close the selected channel and open any closed channels in the same slot.
- **Exclusive Close:** Close the selected channel and open any closed channels in the instrument (the only closed channel is the selected channel).

You can open all channels in a slot by clicking **Open Slot**.

You can open all channels in the instrument by clicking **Open All**.

For more information on opening and closing channels, see "Working with channels" in the Series 3700A Reference Manual.

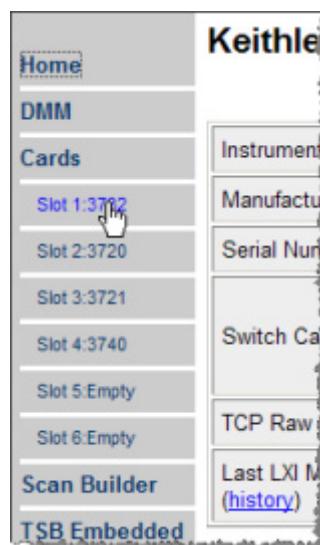
## Exercise: Close multiple channels using the web interface

This exercise describes how to close channels and channel patterns using the web interface.

### To close multiple channels using the web interface:

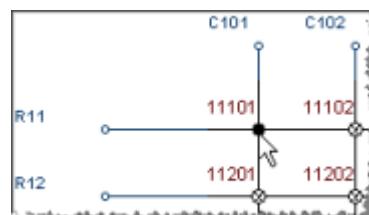
- The instrument home page is displayed. From the navigation on the left, select the slot that contains the channels you want to close.

Figure 18: 3700A web interface select slot



- You must log into the instrument to work with the channels. See [Log in to the instrument](#) (on page 3-2). After logging in, you can access the channel controls.
- To close channel "11101", click the channel.

Figure 19: 3700A web interface close channel



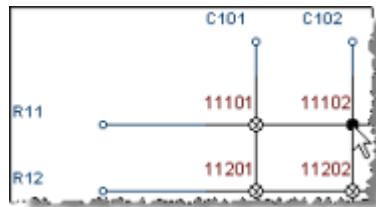
- To perform an exclusive close on channel "11102", select **Exclusive Slot Close** in the Channel Action Type box.

Figure 20: 3700A web interface exclusive slot close

Channel Action Type:			
<input checked="" type="radio"/> Channel Close	<input type="radio"/> Exclusive Close	<input type="radio"/> Exclusive Slot Close	<input type="radio"/> DMM Close

5. Click "11102" to close that channel and open all other channels.

**Figure 21: 3700A web interface exclusive close**



## Set up channel patterns from the web interface

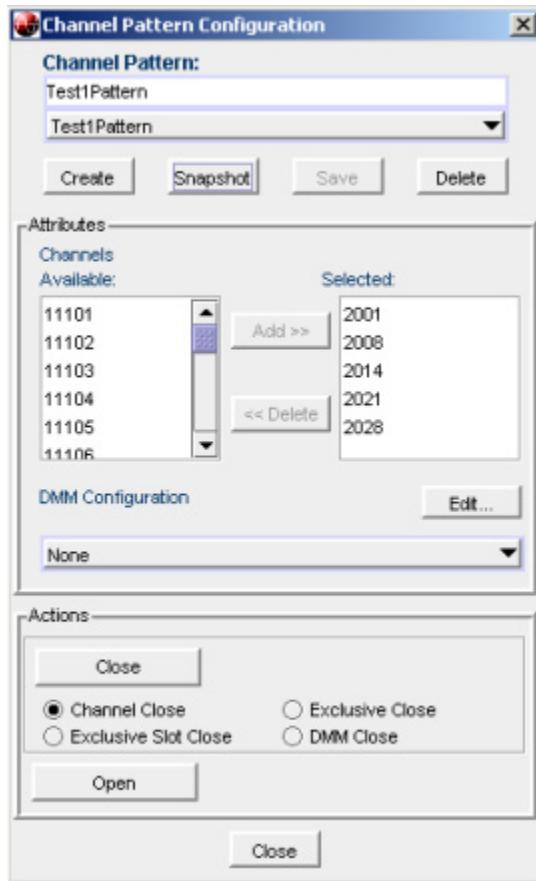
You can use channel patterns as a convenient way to refer to a group of switching channels with a single alphanumeric name. When you perform close or open operations on a channel pattern, only the channels that are in the channel pattern are affected.

There is no speed advantage in performing close and open operations on channel patterns compared to performing the same operations on individual channels or a list of channels.

### **Exercise: Create channel pattern "Test1Pattern" using the Snapshot feature**

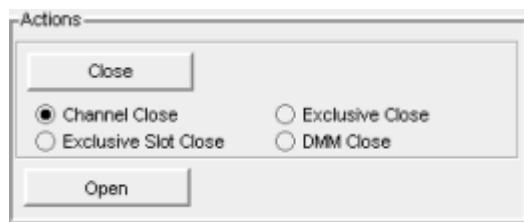
**To create a channel pattern "Test1Pattern" from the web interface using the Snapshot feature:**

1. From the left navigation, click **Slot 2**.
2. Close the channels that you want to include in the pattern. For this example, close channels **2001, 2008, 2014, 2021, and 2028**.
3. Click **Pattern** (above the Channel Action Type box). The Channel Pattern Configuration dialog box is displayed.
4. Type **Test1Pattern** for the pattern name in the box at the top of the dialog box.
5. Click **Snapshot**. A new pattern is created that contains the closed channels.

**Figure 22: Series 3700A Channel pattern configuration dialog box****Exercise: Close channels in "Test1Pattern" from the web interface**

**To close channels in channel pattern "Test1Pattern" from the web interface:**

1. Click **Pattern** (above the Channel Action Type box). The Channel Pattern Configuration dialog box is displayed.
2. Select **Test1Pattern** from the list at the top of the dialog box.
3. In the Actions area, select the type of close you want to perform.

**Figure 23: Close channels in a pattern**

4. Click **Close** to close the channels.
5. When you are done acting on this channel pattern, click **Close** at the bottom of the dialog box to exit.

**To delete a channel pattern from the web interface:**

1. Select the name of the pattern that you want to delete in the Channel Pattern Configuration dialog box.
2. Click **Delete**.
3. For more information about patterns, including opening and closing the channels that are in patterns, see [Channel patterns](#) (on page 2-10).

## Scan Builder page

The Scan Builder page allows you to set up and run scans and triggers.

A scan is a series of steps that opens and closes switches sequentially for a selected group of channels. During each step, actions occur, such as waiting for a trigger, taking a measurement on an external instrument, and completing a step count. Scans automate actions that you want to perform consistently and repeatedly on a set of channels.

Triggers are events that prompt the instrument to move from one step to another in a scan. Triggers can come from a variety of sources, such as a key press, digital input, or expiration of a timer. The sequence of actions and events that occur during the scan is called the trigger model, described in Trigger model.

Scanning and triggering allow you to synchronize actions across channels. You can set up a scan using the trigger model to precisely time and synchronize the Series 3700A between channels and multiple instruments. You can also use triggers without the triggering model to set up a scan to meet the needs of a specific application that does not fit the triggering model.

### NOTE

If you use Scan Builder to create a scan, use the options in the Scan Builder page to run the scan. Using the TSB Embedded page may not give you the expected results.

## Create a scan list

Before you can run a scan, you must create a scan list. A scan list is a set of steps that runs in order during a scan. Each step contains a channel, channels, or channel patterns that you want to measure in that step. Each step is acted on separately during the scan.

You can mix channel patterns and individual channels in a scan list. Note that the steps are executed in the order in which they are added to the scan.

### NOTE

Before setting up a scan list, make sure your channels and channel patterns are configured. See "Working with channels" in the Series 3700A Reference Manual for detail.

If you change the channel configurations or channel patterns after the scan list has been set up, you may not see expected results. If the change prevents the scan from functioning properly (such as deleting something referenced by the scan), an error message is logged.

## Exercise: Create a scan list from the web interface

### **To create a scan list from the web interface:**

In this exercise, create a scan list containing channels **2001**, **2008**, **2014**, **2021**, **2028** and the pattern **Test1Pattern**.

1. From the left navigation of the web interface home page, select **Scan Builder**.
2. In the Add Channel By list on the right, select **Number** to add the channels. Select the channel numbers from the list. You can use **Ctrl+click** to select multiple channels and **Shift+click** to select a range of channels. To remove your selections from the Add Channel By list, click **Clear Channel Selection**.
3. Click **Add Step**. The channels are added to the Steps list.
4. In the Add Channel By list on the right, select **Pattern**.
5. Select **Test1Pattern** from the Channel Pattern list.
6. Click **Add Step**. The pattern is added to the Steps list.
7. In the Scan Count box, enter the number of times you want to repeat the steps in the scan. For example, enter **5**.

## Run the scan

You can run a scan in one of several ways:

- **Background:** Runs the scan in the background so that you can perform other tasks while the scan is running. You can use the Query State to check scan status.
- **Step by Step:** Steps through the scan.

### **To run the scan from the web interface:**

Click **Execute Background** or **Step by Step**.

## Stop the scan

### **To stop the scan from the web interface:**

On the Build & Run tab, click **Abort**.

## Clear the scan list from the web interface

Clearing the scan list deletes all channels and channel patterns from the scan list.

### **To clear the scan list from the web interface:**

1. From the left navigation area of the web interface home page, select **Scan Builder**.
2. Click **Scan Clear**.

## Set up simple triggers

You can set up triggers to control your scan using the options on the Simple Trigger tab. You can set:

- The event that starts the scan
- The time interval event that controls the channel action for each step of the scan
- The time interval event that controls how measurements are taken during the scan

To see these options, click the **Simple Trigger** tab from the top of the Scan Builder page.

## Selecting triggers

You can choose the triggers that will be used to start the scan. The options to start the scan are:

**Immediate:** When Immediate is selected, the scan starts as soon as you click **Execute Background** on the Build & Run tab. Select Immediate when you do not have trigger requirements that must be met before the scan starts. This is the default selection.

**Digital Input:** When selected, you select the digital line (1 to 14) that is used to start a scan. You can select falling or rising for the digital input. Falling selects the falling edge trigger. Rising selects the rising edge of the trigger.

### NOTE

If Other is displayed in the mode list, a different mode (other than falling or rising) is already selected. Other is not a mode and cannot be selected. It is only an indicator that the digital triggering is already set up for a different mode. See the Series 3700A Reference Manual, "Using the web interface" section, and the "Advanced triggering" topic for other options.

**Time:** When selected, you can select options that define when the scan starts and at what rate triggers will occur.

You can select the trigger to use to continue channel action for each scan. The options to continue channel action are:

**Immediate:** When immediate is selected, the scan immediately steps to the next channel in the scan list. This is the default setting.

**Digital Input:** When selected, you select a digital line (1 to 14) that is used to trigger the instrument to step to the next channel. You can select falling or rising for the digital input. Falling selects the falling edge trigger. Rising selects the rising edge of the trigger.

**Every N seconds:** This parameter adds a fixed delay between each channel. The delay occurs before the next channel in the scan list is closed.

You can also select select the trigger to use to take a measurement for each scan step.

**Immediate:** When immediate is selected, the measurement is taken as soon as the channel is closed. This is the default setting.

**Digital Input:** When selected, you select a digital line (1 to 14) that triggers the instrument to take a measurement. You can select falling or rising for the digital input. Falling selects the falling edge trigger. Rising selects the rising edge of the trigger.

**Every N seconds:** This parameter adds a fixed delay after the channel is closed and before the measurement is taken.

## TSB Embedded

TSB Embedded is a web application that includes a command-line interface that you can use to issue commands and interact with the instrument. TSB Embedded also provides a convenient way to create and manage user scripts. TSB Embedded resides in the instrument.

You can use TSB Embedded to run some of the application examples in other sections of this manual.

If you can access the web interface home page, but cannot use TSB Embedded (or TSP Express), make sure you have the Java™ SE Runtime Environment (JRE) Version 6 or later installed on your computer.

### Create a script using TSB Embedded

#### NOTE

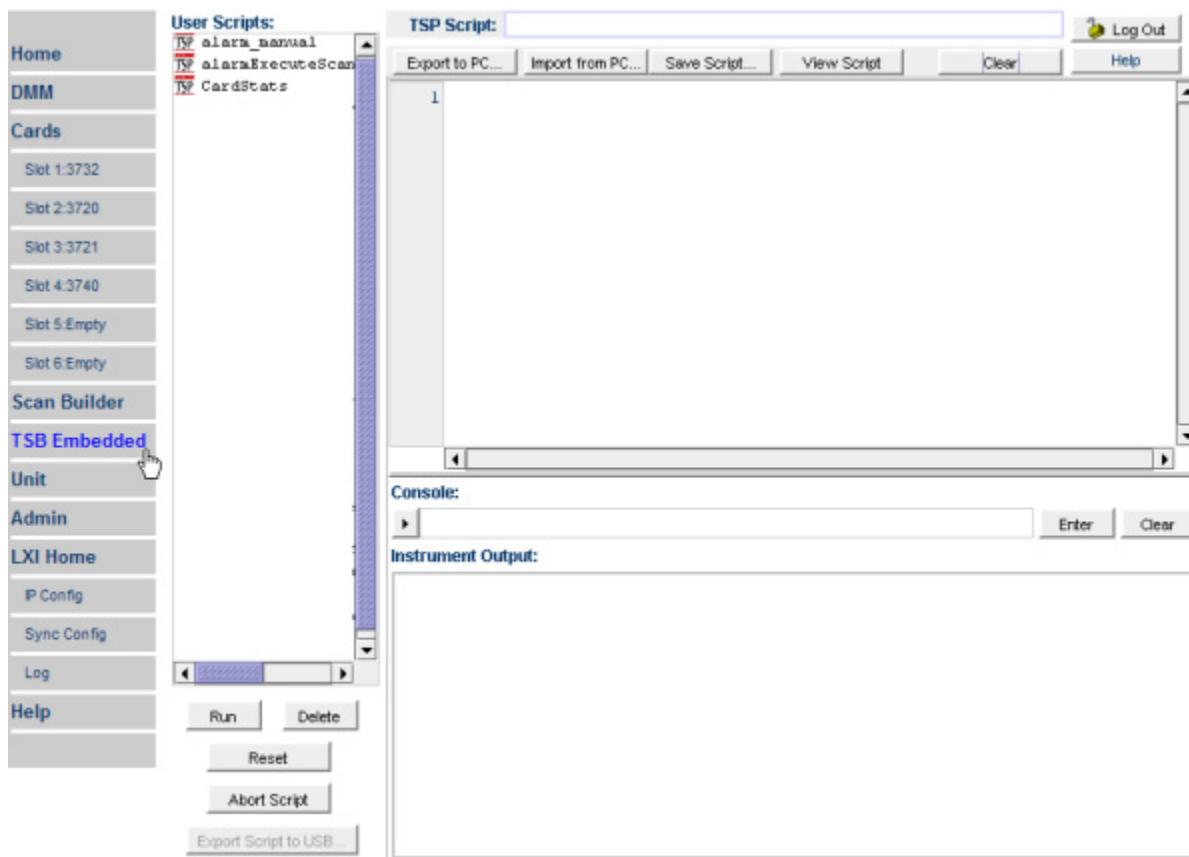
If you are using TSB Embedded to create scripts, you do not need to use the commands `loadscript` or `loadandrunscript` and `endscript`. For information about using TSB Embedded, see the help on the instrument's web page.

#### Exercise: Create and run a script with TSB Embedded

This exercise script checks each slot to see what channels are available.

***To create and run a script with TSB Embedded:***

1. From the navigation area on the left side of the web interface, select TSB Embedded. The TSB Embedded page is displayed.

**Figure 24: Web interface: Select TSB embedded**

2. You must log in to the instrument to use TSB Embedded. After logging in, you can access the options on the TSB Embedded page.
3. To create the example script, enter the name, **CardStats**, in the TSP Script box.

**Figure 25: Enter script name**

4. Enter the code below in the TSB box below the buttons:

```

CardChannels = function(SlotNumber)
    if slot[SlotNumber].idn == "Empty Slot" then
        print("    Slot is Empty")
    else
        if (slot[SlotNumber].startchannel.voltage == nil) and
            (slot[SlotNumber].endchannel.voltage == nil) then
            print("    no voltage channels")
        else
            print("    Start voltage channel is " ..
                  slot[SlotNumber].startchannel.voltage)
            print("    End voltage channel is " ..
                  slot[SlotNumber].endchannel.voltage)
        end
        if (slot[SlotNumber].startchannel.amps == nil) and
            (slot[SlotNumber].endchannel.amps == nil) then
            print("    no amp channels")
        else
            print("    Start amp channel is " ..
                  slot[SlotNumber].startchannel.amps)
            print("    End amp channel is " ..
                  slot[SlotNumber].endchannel.amps)
        end
        if (slot[SlotNumber].digio == 1) then
            print("    Start digital i/o channel is " ..
                  slot[SlotNumber].startchannel.digitalio)
            print("    End digital i/o channel is " ..
                  slot[SlotNumber].endchannel.digitalio)
        else
            print("    no digio channels")
        end
    if (slot[SlotNumber].totalizer == 1) then
        print("    Start totalizer channel is " ..
              slot[SlotNumber].startchannel.totalizer)
        print("    End totalizer channel is " ..
              slot[SlotNumber].endchannel.totalizer)
    else
        print("    no totalizer channels")
    end

    if (slot[SlotNumber].startchannel.analogoutput == nil) and
        (slot[SlotNumber].endchannel.analogoutput == nil) then
        print("    no analog output channels")
    else
        print("    Start analog output channel is " ..
              slot[SlotNumber].startchannel.analogoutput)
        print("    End analog output channel is " ..
              slot[SlotNumber].endchannel.analogoutput)
    end
    if (slot[SlotNumber].matrix == 1) then
        print("    Channels on card are matrix type")
    end
end
for x = 1,6 do
    print("Checking card channels in slot " .. x)
    CardChannels(x)
end

```

## NOTE

Commands and parameters for Series 3700A are case-sensitive. It is important to type in the commands exactly as shown to avoid any syntax or execution errors.

5. Click **Save Script**. The script is added to the User Scripts list.
6. To run the script, select the script in the User Scripts list and click **Run**.
7. The Instrument Output box displays the output from the script and error messages.

### Script management options

Existing scripts are listed in the User Scripts box on the left side of the web interface.

To run a script, click the name of the script and then click **Run**.

To delete a script, click the name of the script and click **Delete**. The script is deleted from the User Scripts list and from the nonvolatile memory of the instrument.

To stop operation of a script, click **Abort Script**.

To export the selected script to a flash drive, click **Export Script to USB**. Place a flash drive in the USB port on the front panel of the instrument. In TSP, enter the name as appropriate and click **OK**. Scripts are saved to a file with the extension `tsp`. TSP files are native to Test Script Builder or TSB Embedded, but they can be opened and edited in any text editor.

To import scripts from the computer, click **Import from PC**. Select the directory that contains the file. You can only import files with the extension `tsp`.

To clear the name box and the box that contains the script, click **Clear**.

To view the contents of a script, type the name of a script in the **TSP Script** box and click **View Script**.

## Exercise: Use the command line to close and open channels

### **Use the command line interface to perform close and open operations on channels:**

1. To send a single command to the instrument, you can enter it in the Console box.
2. To close channel 1002, enter `channel.close("1002")` in the Console box.
3. Click **Enter** to send the command.
4. To open all channels, enter `channel.open("allslots")` in the Console box.
5. Click **Enter** to send the command.

**Figure 26: Enter command in Console box**



6. To resend a command, click the arrow on the left side of the Console box.

**Figure 27: Web interface console**

The screenshot shows a web-based terminal window titled "Console". The input field contains the command "print(memory.used())". Below the input field, there is a scrollable list of previous commands and their outputs:

```
TSP> print(memory.available())
89.51, 98.86, 99.94
TSP>
TSP> print(memory.used())
10.50, 1.14, 0.06
TSP>
```

On the right side of the input field, there are two buttons: "Enter" and "Clear".

More information about commands that can be used to control the instrument can be found in the Series 3700A Reference Manual.

## Section 4

# Applications

### In this section:

About the Series 3700A examples .....	4-1
Scanning examples .....	4-1
Example: TSP-Link and interactive triggers .....	4-10
Example: Using the scanning and triggering model .....	4-15
Example: IEEE-1588 in Series 3700A-based systems.....	4-20
Example: External DMM and switch triggering.....	4-22
Example: DMM buffer statistics with interactive operation .....	4-25
Example: Commonside ohm measurement with Model 3721	4-27

## About the Series 3700A examples

This section shows you how to use the Series 3700A for a variety of uses.

## Scanning examples

### NOTE

In the examples, to clear a trigger stimulus after setting, set the stimulus to 0, which returns the stimulus setting back to its factory default value, which may or may not be 0.

### Example: Scan the card with no measurement (switch-only)

The following examples assume a Keithley Instruments Model 3720 card is installed in slot 3 of a Series 3700A. This example scans all channels in a slot in a switch-only application (no measurements are taken).

Digital I/O line 1 is used to initiate a background scan.

Example program code for the Series 3700A	
Code	Notes and comments
reset()	Reset the Series 3700A to factory defaults.
scan.create("slot3")	Create a scan for all channels on the card installed in slot 3.
digio.trigger[1].mode = digio.TRIG_FALLING	Set up digital I/O line 1 to detect a falling-edge trigger.
scan.trigger.arm.stimulus = digio.trigger[1].EVENT_ID	Use a digital I/O event as the stimulus of the arm layer of the trigger model.
scan.background()	Start the scan and runs the scan in the background.

## Example: Scan the card and take measurements

The following examples assume a Keithley Instruments Model 3720 card is installed in slot 3 of a Series 3700A. Command list to scan the entire card while measuring DC volts on each channel and storing readings in a buffer called `DCVbuffer` (see the "Notes and comments" for other specifics).

Example program code for the Series 3700A	
Code	Notes and comments
<code>reset()</code>	Reset the Series 3700A to factory defaults.
<code>dmm.func = "dcvolts"</code>	Set the DMM function to measure DC volts.
<code>dmm.range = 10</code>	Set the DC volt range to 10 volts.
<code>dmm.nplc = .1</code>	Sets the number of power line cycles (nplc) over which a measurement is integrated to 0.1. This helps improve measurement quality by canceling line noise.
<code>dmm.configure.set("measureDCV")</code>	Save the DMM configuration as "measureDCV".
<code>DCVbuffer = dmm.makebuffer(1000)</code>	Make a buffer named "DCVbuffer" and configure it to store up to 1000 readings.
<code>digio.trigger[1].mode = digio.TRIG_FALLING</code>	Set up digital I/O line 1 to detect a falling-edge trigger.
<code>scan.trigger.channel.stimulus = digio.trigger[1].EVENT_ID</code>	Use a digital I/O event as the stimulus to close each channel.
<code>scan.bypass = scan.OFF</code>	Set bypass to off so that first channel needs to see a trigger before closing.
<code>scan.create("3001:3060", "measureDCV")</code>	Create a scan for channels 1 to 60 on the card installed in slot 3.
<code>scan.background(DCVbuffer)</code>	Start the scan and runs the scan in the background. Readings are saved to the DCVbuffer that was created with <code>dmm.makebuffer</code> .

## Example: Take 4-wire ohm measurements using a background scan

The following examples assume a Keithley Instruments Model 3720 card is installed in slot 4 of a Series 3700A. This script scans all the channels in slot 4 and takes 4-wire ohm measurements using a background scan.

Example program code for the Series 3700A	
Code	Notes and comments
<code>reset()</code>	Reset the Series 3700A to factory defaults.
<code>dmm.func = "fourwireohms"</code>	Set the DMM function to four-wire ohm measurements.
<code>dmm.setconfig("slot4", "fourwireohms")</code>	Set the configuration for all channels in slot 4 to 4-wire ohms.
<code>scan.create("slot4")</code>	Create a scan for all channels on the card installed in slot 4.
<code>digio.trigger[1].mode = digio.TRIG_FALLING</code>	Set up digital I/O line 1 to detect a falling-edge trigger.
<code>digio.trigger[2].mode = digio.TRIG_FALLING</code>	Set up digital I/O line 2 to detect a falling-edge trigger.
<code>scan.trigger.channel.stimulus = scan.trigger.EVENT_MEASURE_COMP</code>	Set each channel so that it will close with a measurement complete event.
<code>digio.trigger[2].stimulus = scan.trigger.EVENT_CHANNEL_READY</code>	Set digital I/O line 2 to pulse when a channel ready event occurs.
<code>scan.trigger.measure.stimulus = digio.trigger[1].EVENT_ID</code>	Set each measurement to occur with a digital I/O line 1 event trigger.
<code>scan.bypass = scan.ON</code>	Set bypass to on so that the first channel closes without taking a measurement.
<code>4WBuffer = dmm.makebuffer(1000)</code>	Make a buffer named "4WBuffer" and configure it to store up to 1000 readings.
<code>scan.background(4WBuffer)</code>	Start the scan and runs the scan in the background. Readings are saved to the 4WBuffer that was created with <code>dmm.makebuffer</code> .

## Example: Scan, measure and store

The following examples assume a Keithley Instruments Model 3723 card is installed in slot 3 of a Series 3700A.

This program code:

- Scans the entire Model 3723 card
- Measures DC volts on each channel
- Stores readings in a buffer called `DCvoltBuffer`

### NOTE

For the Model 3723, the channels are reed relays, while the analog backplane relays are electromechanical relays. Therefore, to have the scan run faster, this example sets the scan mode to fixed ABR, which closes the backplane relays before scanning starts and keeps them closed during the entire scan.

<b>Example program code for the Series 3700A</b>	
<b>Code</b>	<b>Notes and comments</b>
<code>reset()</code>	Reset the Series 3700A to factory defaults.
<code>dmm.func = "dcvolts"</code>	Set the DMM function to DC volt measurements.
<code>dmm.range=10</code>	Select a range of 10 volts.
<code>dmm.nplc=.1</code>	Set the NPLC to 0.1.
<code>dmm.configure.set("DCVreadings")</code>	Save the current DMM settings as the configuration "DCVreadings".
<code>DCvoltBuffer = dmm.makebuffer(1000)</code>	Make buffer named "DCvoltBuffer" and configure it to store up to 1000 readings.
<code>digio.trigger[1].mode = digio.TRIG_FALLING</code>	Set up digital I/O line 1 to detect a falling edge trigger.
<code>scan.trigger.channel.stimulus = digio.trigger[1].EVENT_ID</code>	Set each channel so it closes with a digital I/O line 1 event trigger.
<code>scan.bypass = scan.OFF</code>	Set bypass to OFF so that the first channel needs to see the trigger before closing.
<code>scan.mode=scan.MODE_FIXED_ABR</code>	Set the mode to fixed ABR so that the backplane relays are closed at the start of the scan and maintained closed throughout the scan without being opened or closed.
<code>scan.create("3001:3060", "DCVreadings")</code>	Create a scan list of channels 1 to 60 on slot 3.
<code>scan.background(DCvoltBuffer)</code>	Start the scan to execute in the background and save readings to a buffer called "DCvoltBuffer."

## Example: Optimize scan for speed

The following examples assume a Keithley Instruments Model 3723 card is installed in slot 1 of a Series 3700A.

Some cards, such as the Model 3723, use relays that are optimized for switching speed and reliability. However, these cards still use backplane relays, which are slow. To achieve full speed and reliability of the card, you can avoid scan modes that intelligently open and close backplane relays, such as `scan.MODE_OPEN_SELECTIVE`. Instead, you can set the scan mode (`scan.mode`) to `scan.MODE_FIXED_ABR`, which closes all required backplane relays before the start of the scan and keeps them closed until you program them to open.

The following is an example of a Series 3700A configured for fast scanning with the Model 3723 card. Sixty channels are scanned ten times on 200 V DC.

<b>Example program code for the Series 3700A</b>	
<b>Code</b>	<b>Notes and comments</b>
<code>dmm.reset()</code>	Reset the Series 3700A to factory defaults.
<code>dmm.func = "dcvolts"</code>	Set the DMM function to DC volt measurements.
<code>dmm.autorange=dmm.OFF</code>	Turn auto range off.
<code>dmm.range=200</code>	Select the range of 200 volts.
<code>dmm.autozero=dmm.OFF</code>	Turn autozero off.
<code>dmm.nplc=.006</code>	Set the NPLC to .006.
<code>dmm.autodelay=dmm.OFF</code>	Turn auto delay off.
<code>reading_buffer=dmm.makebuffer(600)</code>	Create a reading buffer that can hold up to 600 readings.
<code>dmm.configure.set("Chan1to60dcvolts")</code>	Save the current DMM settings as the configuration "Chan1to60dcvolts".
<code>dmm.setconfig( "1001:1060", "Chan1to60dcvolts")</code>	Assign the configuration "Chan1to60dcvolts" to channels 1 to 60 on slot 1.
<code>scan.mode=scan.MODE_FIXED_ABR</code>	Set the scan mode to fixed ABR.
<code>scan.create("1001:1060")</code>	Create a scan list of channels 1 to 60 on slot 1.
<code>scan.scancount=10</code>	Set the scan count to 10.
<code>scan.execute(reading_buffer)</code>	Scan in the foreground.
<code>dmm.savebuffer("reading_buffer", "/usb1/mydata.csv")</code>	Write the data out to a file on a USB flash drive.

## NOTE

The NPLC setting is at .006 in the example. The fastest NPLC setting supported in a Series 3700A is .0005. Another speed improvement option is to set the channel connect rule to OFF (channel.connectrule = channel.OFF). Using this setting allows channels to open and close at the same time, provided the application supports this operation.

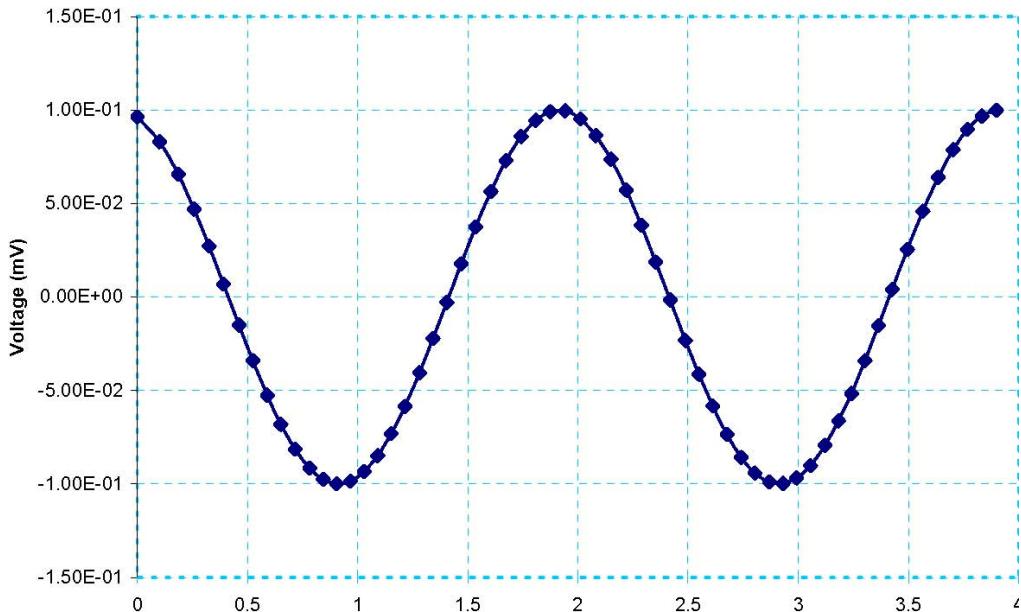
## Example: Fast DC volt measurement

The following example script samples a 500 Hz 70.7 mV sine wave into the 100 mVDC range at 0.0005 plc with autozero and autodelay disabled.

You can cut and paste the output data from the script into an editor like WordPad. From WordPad, you can save it as a .csv (comma separated value) file:

1. From the menu, select File > Save As.
2. For the file name extension, type **.csv**.
3. Open the .csv file in Microsoft Excel or other spreadsheet application.

<b>Example program code for sampling a 500 Hz, 70.7 mV sine wave</b>	
<b>Code</b>	<b>Notes and comments</b>
<pre>loadscript test_dcv_time dmm.func="dcvolts" dmm.range=100e-3 dmm.linesync=0 dmm.nplc=0.0005 dmm.autodelay=0 dmm.autozero=0 dmm.measurecount=1 print (dmm.measure()) print ("wait nplc delay")</pre>	Assigns the script name of "test_dcv_time". Set the DMM function to DC volts, with a range of 100 mV, line sync of 0, NPLC of 0.0005, no autodelay, no autozero, and set to measure once. Output the last measurement taken.
<pre>buf=dmm.makebuffer(30) buf.clear() buf.appendmode=1</pre>	Set the buffer size to 30 readings, clear the buffer, and set the readings to be appended to the existing buffer content.
<pre>dmm.measurecount=30 dmm.measure(buf)</pre>	Set the DMM to take 30 measurements and send the measurements to the buffer.
<pre>for x=1,buf.n do printbuffer(x,x,buf,     buf.relativetimes) end endscript</pre>	

**Figure 28: 3706A ACV Input on DCV at 0.0005 plc**

<b>Example program code for the Series 3700A</b>	
<b>Code</b>	<b>Notes and comments</b>
loadscript test_acv_autodelay	Assigns the script name of "test_acv_autodelay".
reset()	Reset the Series 3700A to the factory defaults.
channel.connectrule=channel.OFF	Set the connect rule off. The instrument closes relays as efficiently as possible to improve speed performance without applying a rule.
dmm.func="acvolts" dmm.range=1 dmm.detectorbandwidth=300	Sets the DMM function to AC volts, with a range of 10 and detector bandwidth of 300.
dmm.nplc=0.06 dmm.autozero=0	When bandwidth set to 300, NPLC can be programmed from 0.0005 plc to 12 plc at 60 Hz or 15 plc at 50 Hz.
dmm.autodelay=dmm.AUTODELAY_ONCE	Include a single 50 ms delay before each measurement after channel closure.
scan.measurecount=25	DMM takes 25 readings on the same channel.
dmm.configure.set("my-1Vac")	Define this group of DMM settings as "my-1Vac".
dmm.setconfig("4004, 4024", "my-1Vac")	Assign the configuration for channels 4 and 24 on slot 4 to "my-1Vac".
buf=dmm.makebuffer(200) buf.clear() buf.appendmode=1	Set the buffer size set to 200 readings, clear the buffer, and set the readings to be appended to the existing buffer content.
scan.create("4004, 4024")	Create a scan list that includes channels 4 and 24 on slot 4. Backplane channels 4911 and 4921 are automatically paired.
scan.scancount=4	Set the scan to loop 4 times.
scan.execute(buf) for x=1,buf.n do printbuffer(x,x,buf, buf.relativetimes) end	Start the scan. Note that x,x prints reading and time vertically so you can copy and paste the information into Microsoft Excel.
endscript	

This table illustrates how to optimize AC volt or AC current measurements for input signal frequency, reading rate, autodelay, and measure count.

Setting	Detector bandwidth	Freq band	Aperture time	Fixed no. samples per reading	Auto zero	Autodelay (ms)		Measure count	Total measure-count time (s)	Average Reading(s)
SLOW	3	3 Hz-300 KHz	1 ms, fixed	2000	N/A	200	Off	10	21.41167	0.467
							Once		21.61796	0.463
							On		23.41259	0.427
MED	30	30 Hz-300 KHz	1 ms, fixed	200	N/A	200	Off	20	6.00595	3.33
							Once		6.22548	3.213
							On		10.0174	1.997
FAST	300	300 Hz-300 KHz	8.33 us (1 0 us) min	1	Off	50	Off	100	0.00765	13,070
							Once		0.15072	663.5
							On		5.08854	19.65

Setting	Detector bandwidth	Freq band	Aperture time	Fixed no. samples per reading	Auto zero	Autodelay (ms)		Measure count	Total measure-count time (s)	Average Reading(s)
FAST	300 ▲	300 Hz-300 KHz	16.67 ms (20 ms)	1	On ▲	50	Off	100	2.35427	42.48
							Once ▲		2.50335	39.95
							On		7.08160	14.12

Default setting ▲ at 50 Hz power line frequency

## Example: Fast alternating scan, DC volts, and two-wire ohms

The following example illustrates how to configure a Series 3700A and a Model 3723 switch card for fast alternating function scans. The example shows channel 1 measuring +7.5 VDC and channel 2 measuring a 1 KΩ resistor. Scanning, which includes a relay close, DMM configuration, DMM measure, and channel open, is achieved at rates of less than 1.6 ms per channel.

Example program code for the Series 3700A	
Code	Notes and comments
loadscript test_func_chg	Name the script "test_func_chg"
reset()	Reset the Series 3700A to factory defaults.
channel.connectrule=channel.OFF	Set the connect rule off. The instrument closes relays as efficiently as possible to improve speed performance without applying a rule.
dmm.func="dcvolts" dmm.range=10 dmm.nplc=0.0005 dmm.autozero=0	Set the DMM function to DC volts, with a range of 10, NPLC of 0.0005 and autozero set to off.
dmm.configure.set("fastDCV")	DC volts DMM settings defined as "fastDCV".
dmm.func="twowireohms" dmm.range=1000 dmm.nplc=0.0005 dmm.autozero=0 dmm.autodelay=dmm.OFF	Set the DMM function to 2-wire ohm measurement with a range of 1000, NPLC of 0.0005, autozero of 0 and no autodelay.
dmm.configure.set("my-2w")	Save the 2-wire ohm DMM settings in a configuration called "my-2w".
scan.measurecount=1	DMM takes one reading on each channel.
dmm.setconfig("4004", "fastDCV")	Use the "fastDCV" configuration for channel 4 on slot 4.
dmm.setconfig("4024", "my-2w")	Use the "my-2w" configuration for channel 24 on slot 4.
buf=dmm.makebuffer(20) buf.clear() buf.appendmode=1	Set the buffer size to 20 readings, clear the buffer, and set the readings to be appended to the existing buffer content.
scan.create("4004, 4024")	Scan list created with channels 4 and 24 on slot 4. Backplane channel 4911 and 4921 will be automatically paired.
scan.scancount=10	Set the scan to loop 10 times.

<pre> scan.execute(buf) for x=1,buf.n do printbuffer(x,x,buf,     buf.relativetimes) end </pre>	<p>Start the scan. Print the reading and relative time from the start of the scan. buf.relativetimes include the date and real time. Note that x,x prints reading and time vertically so you can copy and paste the information into Microsoft Excel.</p>
<pre> channel.open("allslots") endscript </pre>	Open all slots.

## Example: Thermocouple temperature measurement from the front panel

### To take temperature measurements from the front panel:

1. To select the temperature measurement function, press the **CONFIG** key, then press the **DMM** key. FUNC flashes on, then off. Press the **ENTER** key or wheel. **Function?** is displayed on the first line of the display and the second line displays available functions. Use the left or right arrow keys or the knob to select TEMP.
2. Set thermocouple device attributes:
  - Turn the navigation wheel to scroll to the "THERMO" menu item and press the navigation wheel or the **ENTER** key.
  - Turn the navigation wheel to scroll to the "THERMOCOUPLE" temperature connection and press the **ENTER** key.
  - Turn the navigation wheel to select the desired thermocouple type (J, K, T, E, R, S, B, or N) and press the navigation wheel or the **ENTER** key.
3. Set thermocouple device reference junction type:
  - Turn the navigation wheel to scroll to the "REFJUNCT" menu item and press the navigation wheel or the **ENTER** key.
  - Select the desired Reference Junction: SIMULATED, INTERNAL, or EXTERNAL. See Reference junctions for more information.
  - Press the navigation wheel or the **ENTER** key to set the selection.
4. If a SIMULATED reference junction was selected in step 5:
  - Turn the navigation wheel to scroll to the "SIMREF" menu item and press the navigation wheel or the **ENTER** key.
  - Using the navigation wheel, dial in the desired reference temperature (default values are units dependent: 023.00 °C, 296.15 °K, and 073.40 °F).
  - Press the navigation wheel or the **ENTER** key to set the selection.
5. Press the **EXIT** key to leave the "THERMO MENU."
6. Change temperature attributes as needed.
7. Press the **EXIT** key to leave the "TEMP ATTR MENU."
8. Press **TRIG** to take measurements.

## Example: Thermocouple measurement through the remote interface

To take thermocouple measurements through the remote interface, send the following code:

Example program code to take thermocouple measurements through the remote interface	
Code	Notes and comments
loadscript test_temp	Create a script "test_temp".
<pre>reset() dmm.func=dmm.TEMPERATURE dmm.transducer=dmm.TEMP_THERMOCOUPLE dmm.thermocouple=dmm.THERMOCOUPLE_J dmm.opendetector=dmm.ON dmm.units=dmm.UNITS_FAHRENHEIT dmm.refjunction=dmm.REF_JUNCTION_INTERNAL dmm.configure.set("my_temp_j") dmm.setconfig("4001:4010", "my_temp_j") scan.measurecount=1</pre>	Set the DMM function to temperature with a thermocouple type J. Define this group of settings as the configuration "my_temp_j". Set the scan to measure once.
<pre>buf=dmm.makebuffer(20) buf.clear() buf.appendmode=1</pre>	Set the buffer size to 20 readings, clear the buffer, and set the readings to be appended to the existing buffer content.
<pre>scan.create("4001:4010") scan.scancount=2 scan.execute(buf)</pre>	Create a scan of channels 1 to 10 on slot 4. Run through the scan twice and store the readings in the buffer.
<pre>for x=1, buf.n do printbuffer (x,x,buf) end channel.open("allslots") endscript</pre>	Note that x, x prints reading and time vertically so you can copy and paste the information into Microsoft Excel.

## Example: TSP-Link and interactive triggers

This example uses a TSP script to configure and run the test. The script includes two separate functions for configuring the Series 3700A and System SourceMeter Instrument and a third function for running the test. TSP-Link trigger lines are used to coordinate the actions between the two instruments.

When the test is run, the instruments assert trigger signals after completing their respective actions. After connecting the diode, the Series 3700A sends a trigger signal to the System SourceMeter instrument. The SMU then performs a single I-V sweep on the diode and sends a trigger signal back to the Series 3700A to indicate that the test is completed. Hardware handshaking continues in this fashion until all six diodes have been tested. Data from the SMU is returned to the controlling computer at the end of all the tests.

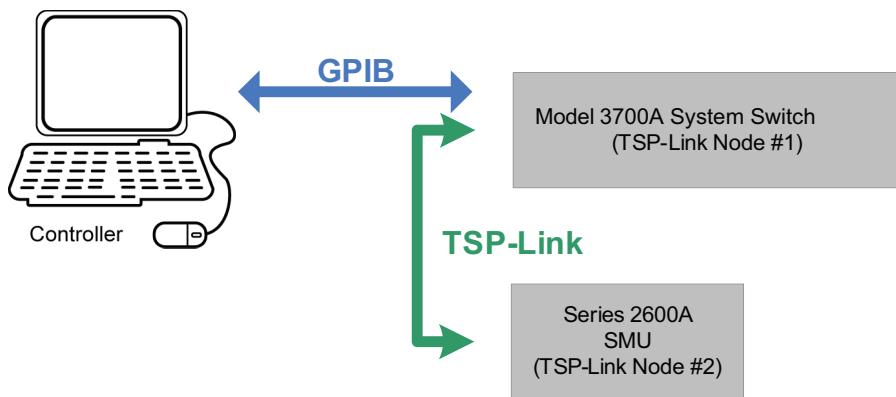
### Quick Tip

A TSP script is a sequence of instrument commands and programming statements. The Series 3700A can store the TSP script for later use. Using scripts allows the Series 3700A to perform simple and complex tasks with less interaction with the remote interface controller (the controlling computer). For more information on using scripts, please see "Instrument Programming" in the Series 3700A Reference Manual.

## Set up communication

The communication setup is illustrated in the diagram below. GPIB is used as an example, but this application can be run using any of the supported communication interfaces for the instruments. A TSP-Link connection enables communication between the two instruments and commands for the SourceMeter (on TSP-Link node 2) are sent over the TSP-Link interface.

**Figure 29: GPIB communication interface with TSP-Link**



To configure the TSP-Link communication interface, each instrument must have a unique TSP-Link node number. Configure the node number for the Series 3700A to 1 and the SourceMeter Instrument to 2.

***To set the TSP-Link node number using the front panel interface of either instrument:***

1. Press **MENU**.
2. Select **TSPLink**.
3. Select **NODE**.
4. Use the navigation wheel to adjust the node number.
5. Press **ENTER** to save the TSP-Link node number.

***On the Series 3700A, perform a TSP-Link reset to alert the Series 3700A to the presence of the SMU:***

1. Press **MENU**.
2. Select **TSPLink**.
3. Select **RESET**.

### Quick Tip

You can also perform a TSP-Link reset from the remote command interface by sending `tsplink.reset()` to the Series 3700A. This command is also included in the example program code below for completion.

## NOTE

If error 1205 is generated during the TSP-link reset, ensure that the System SourceMeter instrument has a unique TSP-Link node number.

In this example, in addition to the TSP-Link interface serving as a communication bus, it will also serve as triggering bus to enable hardware handshaking between the System SourceMeter instrument and Series 3700A.

There are three digital lines in the TSP-Link cable that may be used for triggering. This example uses trigger lines 1 and 2. The Series 3700A outputs a trigger signal on TSP-Link line 1 and waits for a trigger signal on TSP-Link line 2 before proceeding with the next set of crosspoint closures. The System SourceMeter instrument outputs a trigger signal on TSP-Link line 2 after completing its test and waits for a trigger signal on TSP-Link line #1 before testing.

## Program code

In the following table, all commands are sent from the computer to the Series 3700A. This example can be run using TSB Embedded. The Series 3700A sends all commands in the shaded rows to the System SourceMeter instrument through the TSP-Link interface. Commands in unshaded rows are executed by the Series 3700A.

<b>Example program code for the Series 3700A</b>	
<b>Code</b>	<b>Notes and comments</b>
<code>loadscript diodeTest</code>	Create a script "diodeTest" to contain test configuration and execution instructions.
<code>function configSwitch()</code>	Create a function to configure the Series 3700A.
<code>    tsplink.reset()     localnode.reset()     errorqueue.clear()</code>	Reset TSP-Link. Reset Series 3700A. Clear errors.
<code>    muxChannels = {}</code>	Define table that details crosspoints to close to connect each diode to the SMU.
<code>    muxChannels[1] = "1001, 1202"     muxChannels[2] = "1003, 1204"     muxChannels[3] = "1005, 1206"     muxChannels[4] = "1007, 1208"     muxChannels[5] = "1009, 1210"     muxChannels[6] = "1011, 1212"</code>	Add entries to the table.
<code>    tsplink.trigger[1].mode = tsplink.TRIG_FALLING     tsplink.trigger[2].mode = tsplink.TRIG_FALLING     tsplink.trigger[1].clear()     tsplink.trigger[2].clear()</code>	Set TSP-Link lines 1 and 2 to send and receive falling-edge triggers . Clear any latched triggers on TSP-Link trigger lines 1 and 2.
<code>    return muxChannels end</code>	Return the table as an output of the function.

function configSMU()	Create a function to configure the System SourceMeter instrument.
<pre>node[2].reset() node[2].errorqueue.clear() node[2].smua.nvbuffer1.clear()</pre>	Reset the instrument. Clear all errors and reset status bits. Clear existing data from data buffer.
<pre>node[2].smua.source.func = node[2].smua.OUTPUT_DCAMPS node[2].smua.source.rangei = 0.01</pre>	Set source function to current source. Set source current range to 10 mA.
<pre>node[2].smua.measure.rangev = 6 node[2].smua.measure.nplc = 1 node[2].smua.nvbuffer1.appendmode = 1</pre>	Set voltage measurement range to 6 V. Set integration rate. Enable buffer append mode.
<pre>node[2].smua.trigger.source.limitv = 10 node[2].smua.trigger.source.lineari(0, 0.01, 11) node[2].smua.trigger.source.action = node[2].smua.ENABLE</pre>	Set voltage compliance during sweep. Configure linear staircase sweep. Enable the source action during the sweep.
<pre>node[2].smua.trigger.measure.v(node[2].smua.nvbuffer1) node[2].smua.trigger.measure.action = node[2].smua.ENABLE</pre>	Configure sweep voltage measurements. Enable voltage measurements during the sweep.
<pre>node[2].smua.trigger.count = 11 node[2].smua.trigger.arm.count = 6</pre>	Set trigger count to the number of steps in the sweep. Set arm count to six, the total number of sweeps that will be run for all diodes.
<pre>node[2].tsplink.trigger[1].mode = node[2].tsplink.TRIG_FALLING node[2].tsplink.trigger[2].mode = node[2].tsplink.TRIG_FALLING</pre>	Set TSP-Link trigger lines 1 and 2 to send and receive falling edge trigger signals.
<pre>node[2].tsplink.trigger[1].clear() node[2].tsplink.trigger[2].clear()</pre>	Clear any latched triggers on TSP-Link trigger lines 1 and 2.
<pre>node[2].smua.trigger.arm.stimulus = node[2].tsplink.trigger[1].EVENT_ID</pre>	Configure sweep to start when signal received on TSP-Link line 1.
<pre>node[2].tsplink.trigger[2].stimulus = node[2].smua.trigger.SWEEP_COMPLETE_EVENT_ID</pre>	Configure System SourceMeter to output trigger signal on TSP-Link line 2 when sweep is complete.
end	

function runDiodeTest()	This function sequences the crosspoint closures and triggering.
xptTable = configSwitch()	Call the function to configure the switch storing the result in the variable xptTable.
configSMU() node[2].smua.source.output = 1	Call function to configure the SMU and turn on its output.
node[2].smua.trigger.initiate()	Initiate the System SourceMeter so that it is ready to sweep once a trigger signal is received (the output turns on but no power is applied).
for i = 1, table.getn(xptTable) do  if i > 1 then triggered = tsplink.trigger[2].wait(3) if triggered == true then channel.exclusiveclose(xptTable[i]) tsplink.trigger[1].assert() else print("Trigger not received") end else channel.exclusiveclose(xptTable[i]) tsplink.trigger[1].assert()  end end	The "for" loop iterates through the channel list, waiting three seconds for a trigger on TSP-Link line 2 before closing a channel. The first channel of the list does not wait for a trigger. After channel closure is complete, an output trigger is asserted on TSP-Link line 1.
waitcomplete() node[2].smua.source.output = 0 channel.open("allslots") print("done")	After all switch and measure actions have completed, turn off the SMU output, open all crosspoints, and print "done"; the computer controller can query for "done" to determine when the test is complete.
end	Ends the function "runDiodeTest."
endscript	Ends the script "diodeTest."

## Program code to run the test

The example program code stores the program in the instrument for later use. To run the program, send the following commands to the Series 3700A.

Example program code to run the test	
Code	Notes and comments
diodeTest() runDiodeTest()	Run the script to define the functions. Execute the test by calling the function that configures the instruments and sequences their actions.

To retrieve data from the System SourceMeter instrument through TSP-Link, use the `printbuffer()` command to print the data to the computer communication interface. An example is shown below.

Example: Retrieve buffer data	
Code	Notes and comments
<code>printbuffer(1, node[2].smua.nvbuffer1.n,               node[2].smua.nvbuffer1.readings)</code>	Print all readings from the SMU data buffer. Note that "readings" is a buffer attribute.

## Example: Using the scanning and triggering model

This example demonstrates how to use the scanning and triggering model of the Series 3700A to maximize the speed of the test by preconfiguring the sequence of crosspoints used.

This example uses a TSP script to configure the instruments for testing. Once the instruments are configured, the Series 3700A internally manages all close and open actions, as well as any triggering signals, to interface with the System SourceMeter instrument. No separate script is required to sequence these actions.

As with the previous examples, a full I-V sweep is performed on each of the six diodes. Data from these tests is available from the System SourceMeter instrument after all tests have been completed.

### Set up communication

In this example, in addition to the TSP-Link interface serving as a communication bus, it will also serve as triggering bus to enable hardware handshaking between the System SourceMeter instrument and Series 3700A.

The communication setup is identical to the setup in the previous example; see [Example: TSP-Link and interactive triggers \(on page 4-10\)](#).

There are three digital lines in the TSP-Link cable that may be used for triggering. This example uses trigger lines 1 and 2. The Series 3700A outputs a trigger signal on TSP-Link line 1 and waits for a trigger signal on TSP-Link line 2 before proceeding with the next set of crosspoint closures. The System SourceMeter instrument outputs a trigger signal on TSP-Link line 2 after completing its test and waits for a trigger signal on TSP-Link line #1 before testing. The triggering setup is illustrated in the figure named "TSP-Link interface" in "TSP-Link and interactive triggers."

## Program code

In the following table, all commands are sent from the computer to the Series 3700A. This example can be run using TSB Embedded. The Series 3700A sends all commands in the shaded rows to the System SourceMeter instrument through the TSP-Link interface. Commands in unshaded rows are executed by the Series 3700A.

Example program code for the Series 3700A	
Code	Notes and comments
<pre>loadscript diodeTest</pre>	Create a script "diodeTest" to contain test configuration and execution instructions.
<pre>function configSwitch()</pre>	Create a function to configure the Series 3700A.
<pre>    tsplink.reset()     localnode.reset()     errorqueue.clear()</pre>	Reset TSP-Link. Reset Series 3700A. Clear errors.
<pre>    muxChannels = {}</pre>	Define table that details crosspoints to close to connect each diode to the SMU.
<pre>    muxChannels[1] = "1001, 1202"     muxChannels[2] = "1003, 1204"     muxChannels[3] = "1005, 1206"     muxChannels[4] = "1007, 1208"     muxChannels[5] = "1009, 1210"     muxChannels[6] = "1011, 1212"      for i = 1, 6 do         scan.addimagestep(muxChannels[i])     end</pre>	Add entries to the table.  Use the values in the table to create a scan list.
<pre>    tsplink.trigger[1].mode = tsplink.TRIG_FALLING     tsplink.trigger[2].mode = tsplink.TRIG_FALLING      tsplink.trigger[1].clear()     tsplink.trigger[2].clear()</pre>	Set TSP-Link lines 1 and 2 to send and receive falling-edge triggers.  Clear any latched triggers on TSP-Link trigger lines 1 and 2.
<pre>    scan.bypass = scan.ON      scan.trigger.arm.stimulus = 0     scan.trigger.channel.stimulus =         tsplink.trigger[2].EVENT_ID     tsplink.trigger[1].stimulus =         scan.trigger.EVENT_CHANNEL_READY     scan.scancount = 1</pre>	Configure the scanning and triggering mode of Series 3700A. Bypass needed for trigger signal to start scan.  Arm scan immediately. Trigger channel close action with receipt of trigger on TSP-Link line 2. Output trigger on TSP-link line 1 after channel close action occurs.  Loop through scan list once.
<pre>end</pre>	End of the configSwitch function.

function configSMU()	Create a function to configure the System SourceMeter instrument.
node[2].reset()  node[2].errorqueue.clear()  node[2].smua.nvbuffer1.clear()	Reset the instrument.  Clear all errors and reset status bits.  Clear existing data from data buffer.
node[2].smua.source.func = node[2].smua.OUTPUT_DCAMPS  node[2].smua.source.rangei = 0.01	Set source function to current source.  Set source current range to 10 mA.
node[2].smua.measure.rangev = 6  node[2].smua.measure.nplc = 1  node[2].smua.nvbuffer1.appendmode = 1	Set voltage measurement range to 6 V  Set integration rate.  Enable buffer append mode.
node[2].smua.trigger.source.limitv = 10  node[2].smua.trigger.source.lineari(0, 0.01, 11)  node[2].smua.trigger.source.action = node[2].smua.ENABLE	Set the voltage compliance during the sweep.  Configure linear staircase sweep.  Enable the source action during the sweep.
node[2].smua.trigger.measure.v(node[2].smua.nvbuffer 1)  node[2].smua.trigger.measure.action = node[2].smua.ENABLE	Configure sweep voltage measurements.  Enable voltage measurements during the sweep.
node[2].smua.trigger.count = 11  node[2].smua.trigger.arm.count = 6	Set trigger count to the number of steps in the sweep.  Set arm count to six, the total number of sweeps that will be run for all diodes.
node[2].tsplink.trigger[1].mode = node[2].tsplink.TRIG_FALLING node[2].tsplink.trigger[2].mode = node[2].tsplink.TRIG_FALLING	Set TSP-Link trigger lines 1 and 2 to send and receive falling edge trigger signals.
node[2].tsplink.trigger[1].clear() node[2].tsplink.trigger[2].clear()	Clear any latched triggers on TSP-Link trigger lines 1 and 2.
node[2].smua.trigger.arm.stimulus = node[2].tsplink.trigger[1].EVENT_ID	Configure sweep to start when signal received on TSP-Link line 1.
node[2].tsplink.trigger[2].stimulus = node[2].smua.trigger.SWEEP_COMPLETE_EVENT_ID	Configure System SourceMeter to output trigger signal on TSP-Link line 2 when sweep is complete.
end	End of the function "Config SMU."
function runDiodeTest()	This function sequences the crosspoint closures and triggering.
configSwitch()	Call function to configure the Series 3700A.

configSMU() node[2].smua.source.output = 1  node[2].smua.trigger.initiate()  scan.execute() waitcomplete()	Call function that configures the System SourceMeter instrument and turns on its output. Initiate the System SourceMeter so that it is ready to sweep once a trigger signal is received. Note that the output turns on but no power is applied. Initiate the System Switch to start a foreground scan. Waits until Model 2635A finishes all measurements. Remote operation pauses until all six tests are complete. For other options on running a scan, see <a href="#">Using background scans for longer scan lists</a> (on page 4-18).
node[2].smua.source.output = 0 channel.open("allslots")	Turn off the SMU output. Open all crosspoints.
end	Ends the runDiodeTest function.
endscript	Ends the diodeTest script.

## Program code to run the test

The example program code stores the program in the instrument for later use. To run the program, send the following commands to the Series 3700A.

Example program code to run the test	
Code	Notes and comments
diodeTest() runDiodeTest()	Run the script to define the functions. Execute the test by calling the function that configures the instruments and sequences their actions.

To retrieve data from the System SourceMeter instrument through TSP-Link, use the `printbuffer()` command to print the data to the computer communication interface. An example is shown below.

Example: Retrieve buffer data	
Code	Notes and comments
<code>printbuffer(1, node[2].smua.nvbuffer1.n, node[2].smua.nvbuffer1.readings)</code>	Print all readings from the SMU data buffer. Note that "readings" is a buffer attribute.

## Using background scans for longer scan lists

Example 4 illustrates use of the scanning and triggering model to create a foreground scan.

While running a foreground scan, you must wait for the scan to complete or you must abort the scan before you can query the instrument state or any reading buffers. For a scan with a few channels or crosspoints in the scan list, this might not be a problem. However, when there are many channels in the scan list, or when scans are run over long time periods, it can be useful to determine instrument state. In these situations, you can run a background scan. A background scan allows you to query settings during a scan.

To change this example from a foreground scan to a background scan, you need to change one command in the "runDiodeTest" function. Change `scan.execute()` to `scan.background()`. You can then use the `scan.state()` command to query the status of the scan, including the number of steps in the scan, the number of steps performed, and the number of entire scans performed.

The following table shows a revised example of the `runDiodeTest` function in this example.

<code>function runDiodeTest()</code>	This function sequences the crosspoint closures and triggering.
<code>configSwitch()</code>	Call function to configure the Series 3700A.
<code>configSMU() node[2].smua.source.output = 1  node[2].smua.trigger.initiate()</code>	Call function that configures the System SourceMeter instrument and turns on its output. Initiate the System SourceMeter so that it is ready to sweep once a trigger signal is received. Note that the output turns on but no power is applied. Initiate the System Switch to start a background scan.
<code>scan.background()  delay(2) scanState, scanCount, stepCount = scan.state()  print(scanState, " ", scanCount, " ", stepCount, " ")</code>	Delay 2 seconds to allow the scan to start. Use variables to hold the scan state response. Print the response to scan state, which you can use to determine when the scan is complete. Query the scan state every second to determine when the scan completes. When the scan is complete ( <code>scan.state = 6</code> ), exit the loop.
<code>while scanState ~=6 do     delay(1) end</code>	
<code>node[2].smua.source.output = 0 channel.open("allslots")</code>	Turn off the SMU output. Open all crosspoints.
<code>end</code>	Ends the <code>runDiodeTest</code> function.

## Example: IEEE-1588 in Series 3700A-based systems

This section discusses examples of a few applications that are possible using IEEE-1588.

### Scheduling alarms

You can schedule alarms to request the Series 3700A to perform actions at a specific time and date or at a specific time interval. You can schedule alarms in UTC or PTP time; however, it is important to be consistent in defining the alarms using the same time format. Otherwise, the alarms will fire on the networked devices at different times, with a time difference equal to the difference between PTP and UTC.

You can set a maximum of two alarms for each Series 3700A.

To schedule an alarm, first convert the desired alarm time to UTC seconds. You can perform this conversion using `os.time`. If you are specifying alarms in UTC time, you can use this value with `schedule.alarm[N].seconds` to schedule an alarm, where *N* represents the tag number of the alarm that you configure.

#### NOTE

`os.time` is a Lua function that can be used to return the current time or convert a local date and time to UTC-based seconds elapsed since January 1, 1970. When used without parameters, `os.time` returns the current date and time. When used with parameters, the syntax is `os.time{year = <n>, month = <n>, day = <n>, hour = <n>, sec = <n>, isdst = <b>}`. *<n>* is a number and *<b>* is a Boolean where true is Daylight Savings Time. It is not necessary to specify all parameters.

The following example demonstrates how to use `os.time`.

Example program code to use <code>os.time</code>	
Code	Notes and comments
<code>print(os.time)</code>	Retrieve current UTC time in seconds since 1/1/1970.
<code>local l_start_Time l_start_Time = os.time{year=2008, month=3, day=1, hour=15}</code>	Convert 3:00PM March 1, 2008 to UTC seconds since 1/1/1970.
<code>local l_start_Time l_start_Time = os.time() + 60</code>	Create start time to occur 60 seconds after current time.

To specify alarms in PTP format, convert UTC seconds to PTP seconds by adding the value returned by `ptp.utcoffset` to the UTC time. The Series 3700A alone does not differentiate between PTP and UTC time. Use the converted PTP time in setting values for `schedule.alarm[N].ptpseconds`, where *N* represents the number of the alarm you configure.

You can also schedule alarms to occur at a fractional second using either PTP or UTC format with `schedule.alarm[N].fractionalseconds`.

After defining the alarm, configure the number of times you would like to repeat this alarm using `schedule.alarm[N].repetition`.

To set the time (in seconds) between firings of the alarm, use `schedule.alarm[N].period`. To fire the alarm just once, set `schedule.alarm[N].period` to zero. If you want the alarm to repeat forever, set `schedule.alarm[N].period` to a non-zero value and set `schedule.alarm[N].repetition` to zero.

To enable an alarm, set `schedule.alarm[N].enable` to 1. To disable an alarm, set it to 0. To disable all alarms, send `schedule.disable()`.

For more detail on the schedule alarm commands, see the descriptions in the Command Reference of the Series 3700A Reference Manual.

## Scheduling alarms on a stand-alone Series 3700A

To configure a single Series 3700A to perform an event at a particular date and time, you must schedule alarms, but you do not need to enable IEEE-1588. Therefore, you can send these commands over any remote interface and not just LAN.

To initiate a specific action at the firing of the alarm, you must use the event identifier for the scheduled alarm, `schedule.alarm[N].EVENT_ID`, as the stimulus of one of the control sources defined in the trigger model.

The following example demonstrates how to configure a scan of five channels to run once every hour starting at 3 am on September 1, 2008.

<b>Example program code to configure a scan of five channels to run on a specific schedule</b>	
<b>Code</b>	<b>Notes and comments</b>
<code>Start_time = os.time(year=2008, month=9, day=1, hour=3)</code>	Convert to UTC time.
<code>Start_time = Start_time + ptp.utcoffset</code>	Convert to PTP time.
<code>schedule.alarm[1].ptpseconds = Start_time</code>	Configure the alarm.
<code>schedule.alarm[1].fractionalseconds = 0</code>	
<code>schedule.alarm[1].repetition = 5</code>	Configure the alarm repetition count.
<code>schedule.alarm[1].period = 60*60</code>	Set the alarm period to 1 hour = 60 seconds x 60 minutes.
<code>schedule.alarm[1].enable = 1</code>	Enable the alarm.
<code>dmm.setconfig("1001:1005", "dcvolts")</code>	Associate a DMM configuration and configure a scan.
<code>scan.create("1001:1005")</code>	
<code>buf = dmm.makebuffer(25)</code>	Set 5 scans of 5 channels.
<code>scan.trigger.arm.stimulus = schedule.alarm[1].EVENT_ID</code>	Command the scan to start when alarm 1 fires.
<code>scan.scancount = 5</code>	Set scan count and initiate execution of background scan.
<code>scan.background(buf)</code>	

## Example: External DMM and switch triggering

This example demonstrates how to perform DMM and switch triggering with legacy DMMs, such as the Model 2000, or SMUs, such as the Model 2400.

Example program code for the Series 3700A	
Code	Notes and comments
<pre>loadscript Ext_Trig_Dmm</pre>	Create a script called "Ext_Trig_Dmm" that will contain the following commands.
<pre>display.screen=display.MAIN</pre>	Display the main screen on the front panel of the Series 3700.
<pre>channel.open("allslots") channel.connectrule=0</pre>	Open all slots. Set the instrument closes relays as efficiently as possible to improve speed performance without applying a rule.
<pre>dmm.func="dcvolts" dmm.displaydigits = dmm.DIGITS_5_5 dmm.range=10 dmm.nplc=0.0005 dmm.rel.enable= dmm.OFF dmm.autozero=dmm.OFF dmm.autodelay=dmm.OFF dmm.configure.set("dc_10V") dmm.setconfig("1014", "dc_10V")</pre>	Set the DMM function to DC volts, with 5½ display digits, a range of 10, NPLC of 0.005, relative offset enable off, autozero off, and autodelay off. Name this DMM configuration dc_10V and set it as the DMM configuration for measurements on channel 14 on slot 1.
<pre>dmm.range=100e-3 dmm.nplc=0.0005 dmm.rel.enable= dmm.OFF dmm.autozero=0 dmm.autodelay=dmm.OFF dmm.configure.set("dc_100mV") dmm.setconfig("1012", "dc_100mV")</pre>	Change the DC volt function settings for range to be Set the DMM function to DC volts, with 5½ display digits, a range of 100 mV, NPLC of 0.005, relative offset enable off, autozero off, and autodelay off. Name this DMM configuration dc_100mV and set it as the DMM configuration for measurements on channel 12 on slot 1.
<pre>scan.create("1014, 1012") scan.mode=scan.MODE_FIXED_ABR scan.measurecount=100 scan.scancount=3</pre>	Create a scan that includes channels 14 and 12 of slot 1. Set the scan mode so that all required backplane relays close before the start of the scan and remain closed. Set the scan to take 100 measurements for each channel closure. Set the scan to loop 3 times.
<pre>scan.trigger.arm.clear() scan.trigger.channel.clear() scan.trigger.sequence.clear() scan.trigger.measure.clear()</pre>	Clear and remove any Series 3700 latched triggers.

<pre>digio.writebit(1, 1) digio.writebit(2, 1) digio.writebit(3, 1) digio.writebit(4, 1) digio.writebit(10, 1)</pre>	Define the Series 3700 digital I/O as inputs, then redefined by trigger model triggers.
<pre>scan.trigger.arm.clear() scan.trigger.arm.stimulus = 0 scan.trigger.channel.clear() scan.trigger.channel.stimulus = 0</pre>	Set the trigger event for each trigger layer. Set the arm and channel trigger events to pass through immediately.
<pre>scan.bypass = scan.ON digio.trigger[10].mode = digio.TRIG_RISING</pre>	Define the Series 3700 digital I/O input trigger. Set scan bypass on, which does not gate the channel trigger layer. <code>scan.execute</code> is the first trigger, closing the first channel in the channel list. Detects the rising edge input triggers and automatically latches and drives the trigger line low. Asserts a TTL-low pulse as an output trigger. Note that you must use digital I/O 10 through 14 with legacy DMMs. The pull-up of the Series 3700 is too strong on lines 1 to 9.
<pre>scan.trigger.sequence.stimulus = scan.trigger.EVENT_CHANNEL_READY</pre>	Allows the trigger model to drop to next layer immediately.
<pre>scan.trigger.measure.stimulus = digio.trigger[10].EVENT_ID</pre>	Receives VMC signal from an external DMM. The measure stimulus gates every DMM reading. Every DMM reading is triggered by external digital I/O.
<pre>digio.trigger[2].mode = digio.TRIG_FALLING digio.trigger[2].pulsewidth = 5e-6</pre>	Define the digital I/O output trigger. Send a trigger to an external DMM.
<pre>digio.trigger[3].stimulus = scan.trigger.EVENT_MEASURE_COMP digio.trigger[3].mode = digio.TRIG_FALLING digio.trigger[3].pulsewidth = 5e-6</pre>	Define the Series 3700 digital I/O output 3 trigger.
<pre>digio.trigger[4].stimulus= scan.trigger.EVENT_CHANNEL_READY digio.trigger[4].mode = digio.TRIG_FALLING digio.trigger[4].pulsewidth = 5e-6</pre>	Define the Series 3700 digital I/O output 4 trigger.
<pre>dmm_buffer_size = (scan.measurecount * scan.scancount * scan.stepcount) buf_size_max = dmm_buffer_size buf = dmm.makebuffer(buf_size_max) buf.clear() buf.appendmode=1</pre>	Initialize the buffers to a variable buffer size.

<pre> chan_loop_cnt = scan.scancount * scan.stepcount int_vmc_cnt = scan.measurecount -1 digio.trigger[1].clear() digio.trigger[2].clear() digio.trigger[3].clear() digio.trigger[4].clear() digio.trigger[10].clear() scan.background(buf) for z=1, chan_loop_cnt do     print(scan.state()) digio.trigger[4].wait(5) digio.trigger[2].assert() digio.trigger[3].clear() for y=1, int_vmc_cnt do     digio.trigger[3].wait(5)     digio.trigger[2].assert() end while (scan_state() &lt; scan.SUCCESS) do end print(scan.state()) dmm.autozero=dmm.ON endscript </pre>	<p>Manual digital I/O triggering is required to ensure that the next open or close channel operation does not occur while the external DMM is measuring.</p>
<pre> for z=1, chan_loop_cnt do     print(scan.state()) </pre>	Sends the scan status to PC (Labview) to wait for the scan to finish.
<pre> digio.trigger[4].wait(5) </pre>	Waits up to 5 seconds for the last CHANNEL_READY (digital I/O 4) to be triggered.
<pre> digio.trigger[2].assert() </pre>	Triggers measurement on an external DMM.
<pre> digio.trigger[3].clear() </pre>	Ensures that the last VMC pulse cleared.
<pre> for y=1, int_vmc_cnt do     digio.trigger[3].wait(5)     digio.trigger[2].assert() end </pre>	Waits up to 5 seconds for last DMM VMC pulse (digital I/O 3) to be triggered.
<pre> while (scan_state() &lt; scan.SUCCESS) do end print(scan.state()) dmm.autozero=dmm.ON endscript </pre>	Software handshake to verify that Series 3700 scan completed properly. Waits for scan_state() to update to SUCCESS. Issues scan complete status to the computer.

## Example: DMM buffer statistics with interactive operation

Example program code for DMM buffer statistics with interactive front panel operation	
Code	Notes and comments
localnode.prompts=1 localnode.showerrors=1	Sets the instrument to generate prompts in response to command messages and to send generated errors.
loadscript Generic_Noise_loop	Assign the name "Generic_Noise_loop" to the script.
display.screen=display.MAIN	Display the main screen on the front panel.
dmm.func="fourwireohms" dmm.displaydigits=dmm.DIGITS_7_5 dmm.range=1 dmm.nplc=0.0005 dmm.rel.enable=dmm.OFF dmm.autozero=0 dmm.linesync=0 dmm.offsetcompensation=dmm.OFF dmm.autodelay=dmm.OFF dmm.measurecount=1000	Determines the DMM settings.
dmm.measurecount = display.prompt("00000", " Meas_Cnts", "Enter Meas_Count", dmm.measurecount, 1, 50000)	Prompts the user to enter the measurement count from the front panel within a range of 1 to 50 K.
num_str7 = string.format(dmm.measurecount) num_str9 = string.format(dmm.func) display.clear() display.settext("\$BCalculating \$N" .. num_str9.. " Noise "..num_str7.. " Rdg")	
buf=dmm.makebuffer(10000)	Create a buffer that can hold up to 10000 readings.
delay (2.5) display.screen=display.MAIN buf.clear() buf.appendmode=1 hi_last_rdg = -1.2e8 lo_last_rdg = 1.2e8 sum_sq = 0 sum = 0 ave = 0 diff = 0 count = 1 dsply_cnt = 8	stop = dmm.measurecount ()
time = 0 timer.reset() dmm.measure(buf) time=timer.measure.t()	

<pre> for x=1,dmm.measurecount do value = buf.readings[x]     if value &gt; hi_last_rdg then hi_last_rdg = value end     if value &lt; lo_last_rdg then lo_last_rdg = value end     sum = value + sum     ave = sum /count     diff = (value - ave) * (value - ave)     sum_sq = diff + sum_sq     rdg_per_sec = (time / count)*1e6     count = count + 1     stdev = ( (sum_sq) / (count - 1) )^0.5     peak_peak = (hi_last_rdg - lo_last_rdg) --print(peak_peak) </pre>	<pre> for x=1, buf.n do value = buf.readings[x]. Print "Total time" </pre>
<pre> num_str1 = string.format("%4.4e", stdev) num_str2 = string.format("%4.4e", peak_peak) num_str3 = string.format("%4.4e", ave) num_str4 = string.format("%4.4e", buf.readings[x]) num_str8 = string.format(buf.units[1]) </pre>	<p>Creates stdev numeric to string variable. "e" for exponential, "f" for floating point. String variable formatted with four leading digits and 4 digits after decimal point.</p>
<pre> -- display.clear() -- display.settext(num_str4.. num_str8) -- display.settext(num_str1.. "\018rms".. "\$Npp=..   num_str2.. "\018".. " Ave=.. num_str3.. "\018")  -- delay(2.5) -- display.clear() -- display.settext(num_str1.. "rms " ..num_str8..   "\$Npp=.. num_str2.. " Ave=.. num_str3) -- display.settext(num_str4.. " " ..num_str8.. "\$Npp=..   num_str2.. " Std=.. num_str1) dsply_cnt = dsply_cnt + 1     if dsply_cnt &gt; 8 then         display.clear()         display.settext(num_str4.. " " ..num_str8..   "\$Npp=.. num_str2.. " Std=.. num_str1)         delay (0.125)         dsply_cnt = 0     end -- delay(0.125) end -- dmm.measurecount=1 -- dmm.autozero=0 print("finished Generic_Noise_AZ_Off () ") -- print("collect data")  endscript </pre>	
<pre> for x=1,buf.n do printbuffer(x,x,buf,   buf.relativetimes) end </pre>	<p>Get buffer readings for use in Microsoft Excel.</p>

## Example: Commonsode ohm measurement with Model 3721

This example configures channels 38 and 39 in slot 1 to perform commonsode four-wire ohm measurements on the 3721 card. Please refer to the Series 3700A Switching and Control Cards Reference Manual for the physical connections for a commonsode ohms measurement.

Example program code for Model 3721 commonsode ohm measurement	
Code	Notes and comments
loadscript CommonSide4wOhms	Name the script "CommonSide4wOhms"
channel.open("allslots")	Open all slots.
dmm.func="commonsideohms" dmm.autorange=dmm.ON dmm.nplc=0.0005 dmm.autozero=dmm.OFF dmm.autodelay=dmm.OFF	Set the DMM function to commonsode ohms, with autorange on, an NPLC of 0.005, autozero off, and autodelay off.
dmm.configure.set("commonside4w")	Name this DMM configuration "commonside4w".
dmm.setconfig("1038,1039","commonside4w")	Set the configuration as the DMM configuration for measurements on channels 38 and 39 on slot 1.
scan.mode=scan.MODE_FIXED_ABR scan.create("1038,1039") scan.measurecount=10	Measures two resistors in commonsode ohms configuration. DMM takes 10 readings on each channel.
mybuffer = dmm.makebuffer(1000) mybuffer.clear() mybuffer.appendmode= 1	Make buffer named mybuffer and configure it to store up to 1000 readings.
scan.execute(mybuffer)	Initiate the scan to execute in the background and save readings to the buffer called mybuffer. Backplane channels 1927 and 1928 are automatically closed with either channel 1038 or 1039.
printbuffer(1, mybuffer.n, mybuffer,mybuffer.relativetimestamps) endscript	Prints the reading and relative time from start of scan. mybuffer.timestamp includes date and real time.

## Section 5

---

# Troubleshooting FAQs

### In this section:

---

About this section.....	5-1
Need different line frequency or voltage.....	5-1
Switch matrix does not recognize my card .....	5-1
The internal web page of the instrument is not accessible.....	5-2
Error when I try to close a channel.....	5-2
Switch relays will not close.....	5-3
Save the present state of the instrument.....	5-3
LabVIEW driver for the Series 3700A .....	5-3

## About this section

The purpose of this section is to help you troubleshoot the most common problems encountered with the Series 3700A. For additional troubleshooting information, please see the "Troubleshooting" section in the Series 3700A Reference Manual.

## Need different line frequency or voltage

**I need to use a Series 3700A at a different line frequency or voltage. What do I need to do?**

The Series 3700A only supports a line frequency of 50 or 60 Hz. The line frequency is automatically sensed and no user action is needed.

## Switch matrix does not recognize my card

**The Series 3700A does not recognize my switch card. What do I do?**

***Card is not correctly seated in the slot***

Check the seating of the card:

1. Remove power from the instrument.
2. Pull the card out of the slot.
3. Carefully guide the card inside the rails while reinstalling.
4. Make sure the outer shield is not inserted into the card rails.
5. Make sure that no other portions of the card catch on any part of the instrument frame while installing.
6. Make sure the card is fully seated before restarting the instrument.

***Edge connector is dirty***

If the card was stored outside of the instrument, it is possible that the edge connector has gotten dirty.

1. Check the edge connector. The gold edge connector fingers should have a bright surface when properly cleaned.
2. If necessary, clean the edge connector.
3. Reinstall the card.

***Confirm that the card is supported in the Series 3700A***

See the Series 3700A specifications.

## The internal web page of the instrument is not accessible

**I cannot access the internal web page of the instrument. How do I fix this?**

Ensure that the instrument has been assigned a valid and unique IP address.

- To view the IP address of the Series 3700A, press **MENU**. Select **LAN > STATUS > IP-ADDRESS**. If a valid IP address is displayed, the instrument configuration is correct. If **0.0.0.0** is displayed, the IP address is not valid.
- If you are using the automatic configuration method over a corporate network, contact your network administrator for assistance.
- If you are using the automatic configuration method and you have a one-to-one connection between the instrument and the computer, verify that the computer is set to receive an IP address automatically.
- If you are using the manual configuration method over a corporate network, verify that you have entered the IP address, gateway, and subnet mask provided by your network administrator.
- If you are using the manual configuration method and a one-to-one connection with the computer, verify that the subnet mask agrees with the computer's settings.
- If the above suggestions do not work, see the Series 3700A Reference Manual, "Communication interfaces."

## Error when I try to close a channel

**The instrument generates an error when I try to close a channel. What do I do?**

If the instrument generates "Error 1115: Parameter error no forbidden channels accepted," make sure that the channel specifier is valid. For more information on channel specifiers, see "Specifying a channel" in the Series 3700A Reference Manual.

If the specifier is correct, the channel's forbidden attribute may be set to true. To close the specified channel, the forbidden attribute must be set to false.

***To set the forbidden attribute to false using the front panel:***

1. Press the **CONFIG** key, and then select **CHAN**.
2. Select **FORBID**.
3. Select **NO**.

**To set the forbidden attribute to false using the web interface:**

1. On the Cards page, right-click the channel.
2. Clear the Forbidden box.
3. Click **OK**.

**To set the forbidden attribute to false using remote commands:**

Use the command:

```
channel.clearforbidden()
```

## Switch relays will not close

**Why are the switch relays not closing?**

Verify that the system power is sufficient to close the switch relays.

Check the front panel or script output for error that might have been returned from the operation.

## Save the present state of the instrument

**How do I save the present state of the instrument?**

Use the Create Config Script option (or the command `createconfigscript`). See the Series 3700A Reference Manual, "Save the present configuration."

## LabVIEW driver for the Series 3700A

**Where do I find the LabVIEW driver for the Series 3700A?**

The latest NI<sup>TM</sup> LabVIEW<sup>TM</sup> driver is available from the [Keithley Instruments support website](http://www.keithley.com/support) (<http://www.keithley.com/support>).

## Section 6

---

### Next steps

#### In this section:

---

Next steps ..... 6-1

### Next steps

This manual has prepared you to start using your new Series 3700A for your real-world applications. For more detailed information about the Series 3700A, refer to the Keithley Instruments Series 3700A Reference Manual, part number 3700A-901-01.

For information on Series 3700A cards, refer to the Keithley Instrument Series 3700A Switching and Control Cards Reference Manual.

For support and up-to-date product information, refer to the Keithley Instruments website (<http://www.keithley.com>).

---

# Index

## A

ACV

    Autodelay Once script example • 4-6

alarms

    scheduling • 4-22

## B

backplane relay notation • 2-7

    analog backplane relay channel specifiers • 2-7

## C

channel

    patterns • 2-10

channel patterns • 2-10

contact information • 1-1

## D

DCV

    fast measurements • 4-5

Delete script • 3-14

Discovery Tool, LXI • 3-1

Documentation • 1-2

## E

example applications

    in Series 3700A-based systems • 4-21

Examples

    ACV Autodelay Once script • 4-6

    scanning • 4-1

Export script • 3-14

## F

fast alternating scan • 4-8

fast DCV measurements • 4-5

## I

IEEE-1588

    in Series 3700A-based systems • 4-21

Import script • 3-14

IP address • 5-2

## J

Java plug-in • 3-1

## K

Keys • 2-5

## L

LabVIEW drivers • 5-3

LXI Discovery Tool • 3-1

    Downloading from website • 3-1

## M

Manuals • 1-2

matrix card notation • 2-7

Measure

    and switching capabilities • 1-3

    capabilities • 1-3

MUX channel notation • 2-8

## P

Plug-in, Java • 3-1

pseudocards • 2-14

## S

scanning

    examples • 4-1

    fast alternating • 4-8

Scripts

    Clearing • 3-14

    Deleting • 3-14

    Exporting • 3-14

    Importing • 3-14

    Management • 3-14

    Stopping • 3-14

    Viewing • 3-14

switching capabilities • 1-3

## T

thermocouple • 4-10

Troubleshooting • 5-1

    IP address • 3-1, 5-2

    LabVIEW driver • 5-3

    Line frequency • 5-1

    Save present state • 5-3

    Voltage • 5-1

    Web page • 3-1, 3-2, 5-2

TSB Embedded • 3-11

    Script, create with • 3-11

## W

Warranty • 1-1

Web interface • 3-1

    Accessing • 3-1

    Browser requirements • 3-1

    Home page • 3-2

    TSB Embedded • 3-11

    Using • 3-1

Specifications are subject to change without notice.  
All Keithley trademarks and trade names are the property of Keithley Instruments, Inc.  
All other trademarks and trade names are the property of their respective companies.

---

**KEITHLEY**

A G R E A T E R M E A S U R E O F C O N F I D E N C E

**Keithley Instruments, Inc.**

Corporate Headquarters • 28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168 • 1-888-KEITHLEY • [www.keithley.com](http://www.keithley.com)