



# Model 4200A-SCS

## KXCI Remote Control

### PROGRAMMING



Model 4200A-SCS  
KXCI Remote Control  
Programming

© 2025, Keithley Instruments

Cleveland, Ohio, U.S.A.

All rights reserved.

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

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

Actuate™

Copyright © 1993-2003 Actuate Corporation.

All Rights Reserved.

Microsoft, Visual C++, Excel, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Document number: 4200A-KXCI-907-01 Rev. E March 2025

---

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 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 products are designed for use with electrical signals that are measurement, control, and data I/O connections, with low transient overvoltages, and must not be directly connected to mains voltage or to voltage sources with high transient overvoltages. Measurement Category II (as referenced in IEC 60664) connections require protection for high transient overvoltages often associated with local AC mains connections. Certain Keithley measuring instruments may be connected to mains. These instruments will be marked as category II or higher.

Unless explicitly allowed in the specifications, operating manual, and instrument labels, do not connect any instrument to mains.

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 1000 V, 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.


For safety, instruments and accessories must be used in accordance with the operating instructions. If the instruments or accessories are used in a manner not specified in the operating instructions, the protection provided by the equipment may be impaired.


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

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 protective earth (safety 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 protective earth (safety ground) using the wire recommended in the user documentation.

The  symbol on an instrument means caution, risk of hazard. The user must 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 warning, 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 hazards 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.

The **CAUTION** heading with the  symbol in the user documentation explains hazards that could result in moderate or minor injury or damage the instrument. Always read the associated information very carefully before performing the indicated procedure. Damage to the instrument 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. Standard fuses with applicable national safety approvals may be used if the rating and type are the same. The detachable mains power cord provided with the instrument may only be replaced with a similarly rated power cord. 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 to maintain accuracy and functionality of the product). If you are unsure about the applicability of a replacement component, call a Keithley office for information.

Unless otherwise noted in product-specific literature, Keithley instruments are designed to operate indoors only, in the following environment: Altitude at or below 2,000 m (6,562 ft); temperature 0 °C to 50 °C (32 °F to 122 °F); and pollution degree 1 or 2.

To clean an instrument, use a cloth dampened with deionized water 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.

Safety precaution revision as of June 2018.

# Table of contents

---

<b>Introduction.....</b>	<b>1-1</b>
Introduction .....	1-1
4200A-SCS power supply limitations .....	1-2
<b>KXCI communications connections .....</b>	<b>2-1</b>
KXCI communications connections .....	2-1
GPIB connections .....	2-2
Ethernet connections .....	2-2
Using KCon to configure KXCI.....	2-3
Set up KXCI for GPIB control.....	2-3
Set up KXCI for ethernet control .....	2-5
Setting up KXCI as a 4145B emulator.....	2-6
GPIB communications.....	2-8
GPIB status indicators (GPIB communications only) .....	2-9
Ethernet communications.....	2-10
KXCI ethernet client driver .....	2-10
Driver functions .....	2-11
Example write and read with ethernet .....	2-11
Error messages .....	2-12
<b>Using KXCI.....</b>	<b>3-1</b>
Using KXCI.....	3-1
Output data formats for SMUs .....	3-1
Data format for system mode readings .....	3-1
Data format for user mode readings.....	3-2
Status byte and serial polling .....	3-3
Status byte .....	3-3
Bit B0, Data Ready.....	3-4
Bit B1, Syntax Error.....	3-4
Bit B4, Busy.....	3-4
Bit B6, RQS (request for service) .....	3-4
Serial polling.....	3-5
Waiting for SRQ .....	3-5
Logging commands, errors, and test results .....	3-5
Logging commands.....	3-6
Logging errors .....	3-7
Logging test results .....	3-7
Understanding the log file .....	3-7
Graphing the test results .....	3-8
<b>KXCI common commands.....</b>	<b>4-1</b>
Common commands .....	4-1
DR .....	4-1
:ERROR:LAST:CLEAR .....	4-2
:ERROR:LAST:GET .....	4-3

ID .....	4-3
*IDN? .....	4-4
*OPT? .....	4-4
*RST .....	4-5
SP .....	4-6
Calling KULT user libraries remotely .....	4-6
Summary of commands for remote calls .....	4-7
AB .....	4-7
EX .....	4-7
GD .....	4-9
GN .....	4-10
GP .....	4-11
GR .....	4-12
UL .....	4-13
SystemUtil User Library .....	4-13
instrumentinfo .....	4-14
remotemoduleinfo .....	4-15
systeminfo .....	4-16
<b>KXCI SMU commands .....</b>	<b>5-1</b>
KXCI SMU commands .....	5-1
Summary of KXCI SMU commands .....	5-2
Summary of page commands .....	5-4
SMU default settings .....	5-4
System mode SMU default settings .....	5-5
Command reference .....	5-6
System mode commands .....	5-6
CH .....	5-7
VM .....	5-8
VS .....	5-9
DT .....	5-10
FS .....	5-11
HT .....	5-12
RT .....	5-13
SC .....	5-14
ST .....	5-14
VC and IC .....	5-15
VL and IL .....	5-16
VP and IP .....	5-17
VR and IR .....	5-19
DM .....	5-22
IN .....	5-22
LI .....	5-23
NR .....	5-23
WT .....	5-24
XN .....	5-25
XT .....	5-25
YA .....	5-26
YB .....	5-27
ME .....	5-28
GT .....	5-29
SV .....	5-29
MP .....	5-30
SR .....	5-31
User mode commands (US) .....	5-31

DI .....	5-32
DS .....	5-33
DV .....	5-33
TI .....	5-34
TV .....	5-35
Modeless commands .....	5-36
AC .....	5-37
BC .....	5-37
DO .....	5-38
EC .....	5-39
EM .....	5-39
IT .....	5-40
RD .....	5-41
RG .....	5-42
RI .....	5-42
RS .....	5-43
RV .....	5-43
Code examples .....	5-44
Example 1: VAR1 and VAR2 sweep (system mode) .....	5-44
Example 2: Basic source-measure (user mode) .....	5-45
Example 3: Retrieving saved data (system mode) .....	5-46
Example 4: VAR1 sweep with real-time data retrieval .....	5-46

## **KXCI CVU commands..... 6-1**

KXCI CVU commands.....	6-1
Summary of KXCI CVU commands .....	6-2
User mode commands .....	6-3
:CVU:CABLE:COMP:LOAD .....	6-4
:CVU:CABLE:COMP:MEASCUSTOM .....	6-5
:CVU:CABLE:COMP:OPEN .....	6-5
:CVU:CABLE:COMP:SHORT .....	6-6
:CVU:MEASZ? .....	6-7
System mode commands.....	6-7
:CVU:BIAS:DCV:SAMPLE .....	6-7
:CVU:DATA:FREQ? .....	6-8
:CVU:DATA:STATUS? .....	6-8
:CVU:DATA:TSTAMP? .....	6-8
:CVU:DATA:VOLT? .....	6-8
:CVU:DATA:Z? .....	6-9
:CVU:DELAY:STEP .....	6-9
:CVU:DELAY:SWEEP .....	6-9
:CVU:FSTEPSIZE .....	6-10
:CVU:SAMPLE:HOLDT .....	6-10
:CVU:SAMPLE:INTERVAL .....	6-10
:CVU:SOAK:DCV .....	6-11
:CVU:STANDBY .....	6-11
:CVU:STEP:FREQ .....	6-11
:CVU:SWEEP:ACV .....	6-12
:CVU:SWEEP:DCV .....	6-12
:CVU:SWEEP:FREQ .....	6-13
:CVU:SWEEP:LISTDCV .....	6-14
:CVU:TEST:ABORT .....	6-14
:CVU:TEST:RUN .....	6-15
Modeless commands .....	6-15
:CVU:ACV .....	6-15
:CVU:ACZ:RANGE .....	6-16



:CVU:CHANNEL .....	6-16
:CVU:CONFIG:ACVHI .....	6-16
:CVU:CONFIG:DCVHI .....	6-17
:CVU:CORRECT .....	6-17
:CVU:DCV .....	6-19
:CVU:DCV:OFFSET .....	6-19
:CVU:FREQ .....	6-20
:CVU:LENGTH .....	6-20
:CVU:MODE .....	6-21
:CVU:MODEL .....	6-21
:CVU:OUTPUT .....	6-21
:CVU:RESET .....	6-22
:CVU:SPEED .....	6-22
:CVU:TEST:COMPLETE? .....	6-25
Code examples .....	6-25
Example 1 .....	6-25
Example 2 .....	6-27

## **KXCI PGU and PMU commands..... 7-1**

Introduction .....	7-2
Summary of KXCI PGU and PMU commands.....	7-3
:PMU:ABORT .....	7-4
:PMU:CONNECTION:COMP .....	7-5
:PMU:DATA:COUNT? .....	7-6
:PMU:DATA:GET .....	7-7
:PMU:EXECUTE .....	7-9
:PMU:INIT .....	7-9
:PMU:LLEC:CONFIGURE .....	7-10
:PMU:LOAD .....	7-11
:PMU:MEASURE:CONNECTION:COMP .....	7-12
:PMU:MEASURE:MODE .....	7-13
:PMU:MEASURE:PIV .....	7-14
:PMU:MEASURE:RANGE .....	7-15
:PMU:OUTPUT:STATE .....	7-16
:PMU:PULSE:BURST:COUNT .....	7-17
:PMU:PULSE:TIMES .....	7-18
:PMU:PULSE:TRAIN .....	7-21
:PMU:RETAIN:CONFIG .....	7-21
:PMU:RPM:CONFIGURE .....	7-22
:PMU:SAMPLE:RATE .....	7-23
:PMU:SOURCE:RANGE .....	7-24
:PMU:STEP:DC .....	7-25
:PMU:STEP:PULSE:AMPLITUDE .....	7-26
:PMU:STEP:PULSE:BASE .....	7-27
:PMU:SWEEP:DC .....	7-28
:PMU:SWEEP:PULSE:AMPLITUDE .....	7-29
:PMU:SWEEP:PULSE:BASE .....	7-31
:PMU:TEST:STATUS? .....	7-32
:PMU:TIMES:PIV .....	7-32
:PMU:TIMES:WAVEFORM .....	7-34
Programming examples .....	7-35
Example: Use KXCI to generate a pulse I-V drain family of curves.....	7-35
Example: Using KXCI to generate a pulse train .....	7-37
Summary of Segment Arb commands .....	7-39
:PMU:SARB:SEQ:MEAS:START .....	7-40
:PMU:SARB:SEQ:MEAS:START:ADD .....	7-41
:PMU:SARB:SEQ:MEAS:STOP .....	7-42

:PMU:SARB:SEQ:MEAS:STOP:ADD .....	7-43
:PMU:SARB:SEQ:MEAS:TYPE .....	7-44
:PMU:SARB:SEQ:MEAS:TYPE:ADD .....	7-45
:PMU:SARB:SEQ:SSR .....	7-46
:PMU:SARB:SEQ:SSR:ADD .....	7-47
:PMU:SARB:SEQ:STARTV .....	7-48
:PMU:SARB:SEQ:STARTV:ADD .....	7-49
:PMU:SARB:SEQ:STOPV .....	7-50
:PMU:SARB:SEQ:STOPV:ADD .....	7-51
:PMU:SARB:SEQ:TIME .....	7-52
:PMU:SARB:SEQ:TIME:ADD .....	7-53
:PMU:SARB:SEQ:TRIG .....	7-54
:PMU:SARB:SEQ:TRIG:ADD .....	7-55
:PMU:SARB:WFM:SEQ:LIST .....	7-56
:PMU:SARB:WFM:SEQ:LIST:ADD .....	7-56
Programming example for Segment Arb .....	7-57
KXCI pulse commands for Clarius versions v1.12 and earlier .....	7-59
Summary of KXCI PGU commands .....	7-60
PA .....	7-60
PD .....	7-61
PE .....	7-61
PG .....	7-62
PH .....	7-63
PN .....	7-63
PO .....	7-64
PS .....	7-65
PT .....	7-65
PV .....	7-67
RP .....	7-68
TO .....	7-69
TS .....	7-70
VF .....	7-71

# Introduction

### In this section:

Introduction .....	1-1
4200A-SCS power supply limitations .....	1-2

## Introduction

With Keithley External Control Interface (KXCI), you can use a remote computer to control the source-measure units (SMUs), capacitance-voltage units (CVUs), pulse generator units (PGUs, and pulse generator units (PMU) in the 4200A-SCS directly.

For the SMUs, the KXCI command set includes an HP 4145 compatibility mode, allowing many programs already developed for the HP4145 to be used on the 4200A-SCS instead.

This document contains:

- **KXCI communications connections:** Describes the hardware and software set up for GPIB and ethernet communications.
- **KXCI user interface:** Describes how to use the KXCI user interface, which is used to control GPIB operation.
- **KXCI common commands:** Descriptions of common commands, such as ID commands, reading commands, and reset commands.
- **KXCI SMU commands:** Descriptions of the commands used to set up and control SMUs in user mode and system mode.
- **KXCI CVU commands:** Descriptions of the commands used to control the CVU using KXCI in both user mode and system mode.
- **KXCI PGU and PMU commands:** Descriptions of the modeless commands used to control PGUs and PMUs.

For information on using Python 3 to control the 4200A-SCS, refer to Application Note *Controlling the Keithley 4200A-SCS Parameter Analyzer Using the External Control Interface (KXCI) and Python 3* (1KW-74006-0).

## 4200A-SCS power supply limitations

In some system configurations, the 4200A-SCS power supply cannot supply enough current if a test has too many high-power instruments enabled. Some system configurations may have enough instruments installed to exceed the power supply limit if the selected test has too many channels enabled.

Clarius tracks the instruments used in a test and calculates the maximum power required and compares it to the maximum available power. If the test requires too much power, a message is displayed and the test will not run.

The following table and the equations below it show how the power is calculated. The maximum power available for each instrument in the test module is used in the calculation. The 4210-SMU High Power SMU, 4211-SMU High Power SMU, 4225-PMU Ultra-Fast Pulse Measure Unit, and 4220-PGU High Voltage Pulse Generator draw the majority of the power in the 4200A-SCS chassis.

There are two parts to the total power supply draw. The first part is the power required for the instruments while the 4200A-SCS is idle (on, but not testing). The second part is the power required by the instruments taking part in the test. Note that medium power SMUs (4200-SMUs and 4201-SMUs), 4200 preamplifiers (4200-PAs), and 4210-CVU and 4215-CVU modules are not included in the equations, as their power draw is not significant.

### 4200A-SCS power requirements

Instrument	Idle power (watts)	Test power (watts)
High Power SMU (4210-SMU or 4211-SMU)	Not significant	45
4225-PGU	29.4	Not available
4225-PMU	50.4	Not available
4225-RPM	4.2	Not available
10 V PGU or PMU channel*	Not available	8.4
40 V PGU or PMU channel*	Not available	54.6
Medium Power SMU (4200-SMU or 4201-SMU)	Not significant	Not significant
4210-CVU or 4215-CVU	Not significant	Not significant

\* There are two channels for each PGU and PMU instrument card.

**Equations to calculate power:**

$$\text{PowerIDLE} = [(29.4 \times nPGU) + (50.4 \times nPMU) + (4.2 \times nRPM)]$$

$$\text{PowerTEST} = [(45 \times nHPSMU) + (8.4 \times nC10) + (54.6 \times nC40)]$$

$$\text{PowerTOTAL} = \text{PowerIDLE} + \text{PowerTEST}$$

PowerTOTAL = 500 W maximum. If PowerTOTAL is less than or equal to 500 W, the test proceeds.

Where:

- $nPGU$  = number of 4220-PGU cards in the 4200A-SCS chassis (idle power draw)
- $nPMU$  = number of 4225-PMU cards in the 4200A-SCS chassis (idle power draw)
- $nRPM$  = number of 4225-RPM modules connected to PMUs (idle power draw)
- $nHPSMU$  = number of High-Power SMUs (4210-SMUs or 4211-SMUs) in the test
- $nC10$  = number of 10 V PGU or PMU channels in the test
- $nC40$  = number of 40 V PGU or PMU channels in the test

**Example 1:** 4200A-SCS with two 4210-SMUs or 4211-SMUs, four 4225-PMUs, and eight 4225-RPMs. The test uses all eight PMU+RPM channels set to the 10 V range (no SMUs in test).

$$\text{PowerIDLE} = [(29.4 \times nPGU) + (50.4 \times nPMU) + (4.2 \times nRPM)] = [(29.4 \times 0) + (50.4 \times 4) + (4.2 \times 8)] = 201.6 + 33.6 = 235.2$$

$$\text{PowerTEST} = [(45 \times nHPSMU) + (8.4 \times nC10) + (54.6 \times nC40)] = [(45 \times 0) + (8.4 \times 8) + (54.6 \times 0)] = 67.2$$

$$\text{PowerTOTAL} = \text{PowerIDLE} + \text{PowerTEST} = 235.2 + 67.2 = 302.4$$

This test has PowerTOTAL ≤ 500 W, so this test will proceed.

**Example 2:** 4200A-SCS with two 4210-SMUs or 4211-SMUs, four 4225-PMUs, and eight 4225-RPMs. The test uses five PMU+RPM channels set to the 40 V range (no SMUs in test).

$$\text{PowerIDLE} = [(29.4 \times nPGU) + (50.4 \times nPMU) + (4.2 \times nRPM)] = [(29.4 \times 0) + (50.4 \times 4) + (4.2 \times 8)] = 201.6 + 33.6 = 235.2$$

$$\text{PowerTEST} = [(45 \times nHPSMU) + (8.4 \times nC10) + (54.6 \times nC40)] = [(45 \times 0) + (8.4 \times 0) + (54.6 \times 5)] = 273$$

$$\text{PowerTOTAL} = \text{PowerIDLE} + \text{PowerTEST} = 235.2 + 273 = 508.2$$

This test has PowerTOTAL > 500 W, so this test will not proceed. Reduce the number of PMU+RPM channels that are set to the 40 V range to less than five.

The following table shows the 4200A-SCS power requirements for valid combinations for the 4225-PMU, 4225-RPM, and high-power SMU.

**4200A-SCS power requirements for valid combinations of internal system instruments**

Idle power (watts)		Power used in test (watts)					
<i>nPMU</i>	<i>nRPM</i>	<i>nC10</i>	<i>nC40</i>	<i>nHPSMU</i>	PowerIDLE	PowerTEST	PowerTOTAL
2	4	0	4	0	117.6	218.4	336
2	4	0	4	1	117.6	263.4	381
2	4	0	4	2	117.6	308.4	426
2	4	4	0	2	117.6	123.6	241.2
3	6	6	0	0	176.4	50.4	226.8
3	6	6	0	1	176.4	95.4	271.8
3	6	6	0	2	176.4	140.4	316.8
3	6	0	5	0	176.4	273	449.4
3	6	0	5	1	176.4	318	494.4
4	8	8	0	0	235.2	67.2	302.4
4	8	8	0	1	235.2	112.2	347.4
4	8	8	0	2	235.2	157.2	392.4
4	8	0	4	0	235.2	218.4	453.6
4	8	0	4	1	235.2	263.4	498.6
5	10	10	0	0	294	84	378
5	10	10	0	1	294	129	423
5	10	10	0	2	294	174	468
5	10	0	3	0	294	163.8	457.8
5	10	0	2	1	294	154.2	448.2
5	10	0	2	2	294	199.2	493.2
6	12	12	0	0	352.8	218.4	336
6	12	12	0	1	352.8	263.4	381
6	12	0	2	0	352.8	308.4	426
6	12	0	1	1	352.8	123.6	241.2

## KXCI communications connections

### In this section:

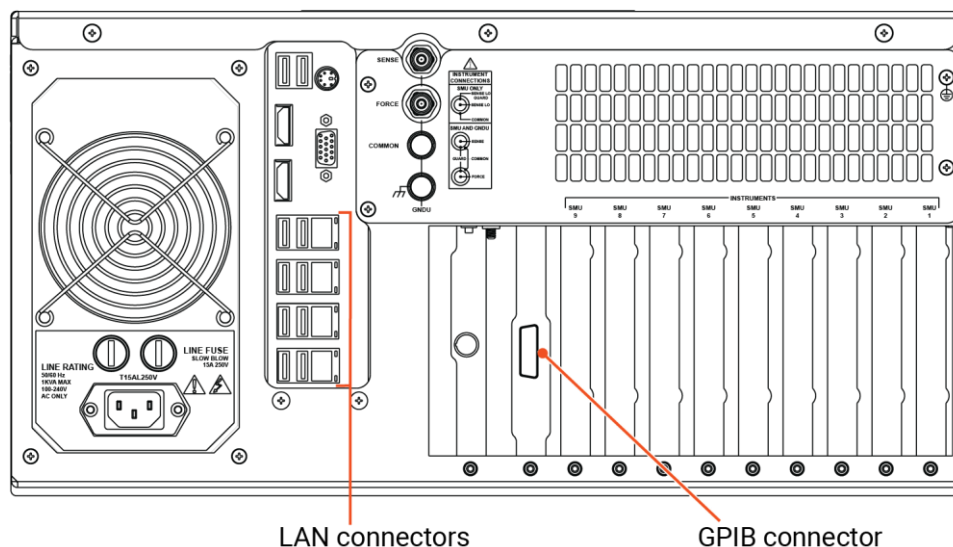
KXCI communications connections.....	2-1
GPIB connections .....	2-2
Ethernet connections .....	2-2
Using KCon to configure KXCI.....	2-3
GPIB communications.....	2-8
Ethernet communications.....	2-10
Error messages.....	2-12

## KXCI communications connections

You can communicate with the 4200A-SCS using GPIB or ethernet connections. The connections for each are described in the following topics.

The locations of the connections are shown in the following figure.

**Figure 1: GPIB and ethernet connectors on the 4200A-SCS**



## GPIO connections

The GPIO cable is the IEEE-488 instrumentation data bus with hardware and programming standards originally adopted by the Institute of Electrical and Electronic Engineers (IEEE) in 1975. The 4200A-SCS conforms to these standards:

- IEEE-488.1-1987
- IEEE-488.2-1992

To connect the 4200A-SCS, use a GPIO cable equipped with standard GPIO connectors, as shown in the following figure. Either end of this cable mates to the GPIO connector on the rear panel of the 4200A-SCS. Connect the other end of the cable to the GPIO connector on the computer.

The connectors on the cable are stackable to allow GPIO connection to other instruments. However, to avoid damage, do not stack more than three connectors on any one unit.

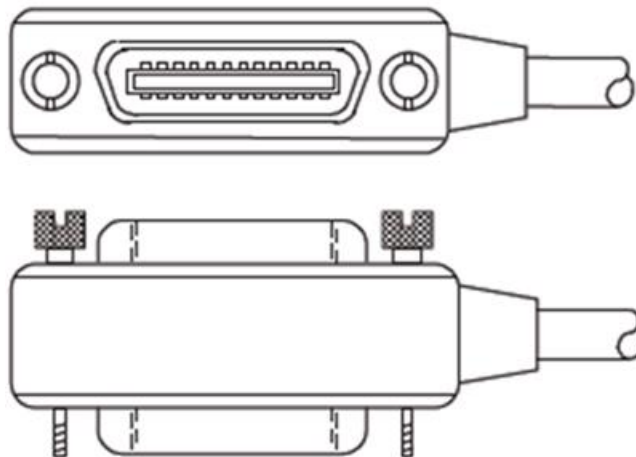
---

### NOTE

To minimize interference caused by electromagnetic radiation, use shielded GPIO cables.

---

**Figure 2: Standard IEEE-488 connectors**



## Ethernet connections

Use a standard cable (CAT-5, RJ-45 terminated) to connect to the 4200A-SCS, as shown in the figure in [Communications connections](#) (on page 2-1).

The ethernet cable must be connected to the network, not directly to a computer.



## Using KCon to configure KXCI

You use the Keithley Configuration Utility (KCon) to configure KXCI communications.

If you need to set up the 4200A-SCS as a subordinate on a GPIB or ethernet system, you do the initial setup through the KCon KXCI Settings. This allows you to use an external computer to remotely control the 4200A-SCS over GPIB or ethernet.

You can also use KXCI to set up emulation for Keysight 4145B Semiconductor Parameter Analyzers. In many cases, test programs developed for use with a Keysight 4145B run without modification when they are used with a 4200A-SCS running KXCI.

For more information on KCon, refer to “Keithley Configuration Utility (KCon)” in *Model 4200A-SCS Setup and Maintenance*.

---

### NOTE

Before opening KCon to change the present KXCI configuration, you must close KXCI.

---

The presently selected communications interface (GPIB or ethernet) and its settings are displayed in the KXCI console. By default, the 4200A-SCS is set up for GPIB remote control. The command and message area below the KXCI settings displays sent commands, KXCI error messages, and numerical test results (refer to [Using KXCI](#) (on page 3-1)).

## Set up KXCI for GPIB control

**To set up GPIB control:**

1. Open KCon.
2. Select **KXCI Settings**. The **KXCI Settings** dialog is displayed, as shown in the following figure.

**Figure 3: KXCI Settings dialog**

Instrument	Function	Number
SMU1:	SMU	1
SMU2:	SMU	2
Empty		
Empty		
Empty		
Empty		
Empty		
CVU1:		
Empty		

Communications

☒ GPIB ☐ Ethernet

GPIB Address: 17 Port Number: 1225

Reading Delimiter: ☐ String Terminator ☒ Comma

String Terminator: ☒ EOI On ☐ EOI Off Character: CR/LF

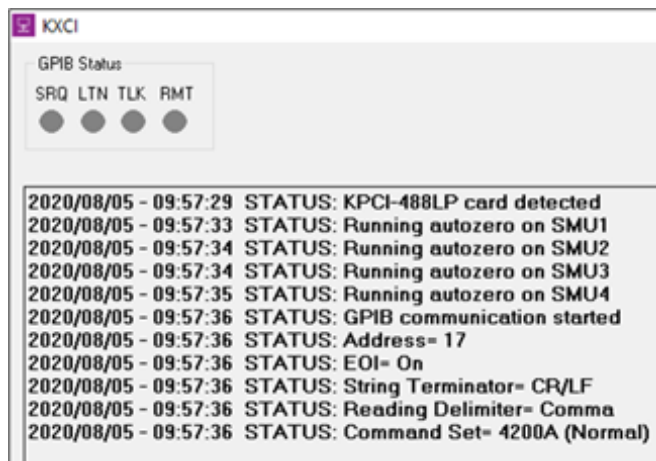
Command Set: ☒ 4200A (Normal) ☐ 4145 Emulation

OK Cancel

3. Set Communications to **GPIB**.
4. Set the **GPIB Address**. This is the primary address of the 4200A-SCS when operating under KXCI control. If the selected GPIB address conflicts with the GPIB address of another system component, a red exclamation-point symbol (!) is displayed next to the selected address.
5. Set the **Reading Delimiter** to determine the output data delimiter characters that are added to the end of each KXCI output message:
  - Select **String Terminator** to use a character.
  - Select **Comma** to terminate output data with a comma (,).
6. If you selected String Terminator, select the type of **Character**:
  - **None** to use no character.
  - **CR** to use a carriage return.
  - **LF** to use a line feed.
  - **CR/LF** to use a carriage return and line feed character sequence.
7. If **String Terminator** is selected, select **EOI ON** or **EOI OFF**. The EOI setting determines if the 4200A-SCS asserts the GPIB End Or Identify (EOI) signal with the last byte of each output data message.
8. Select **OK**.

When KXCI starts, the status messages look similar to the following figure when GPIB is set up.

**Figure 4: KXCI status when GPIB selected**



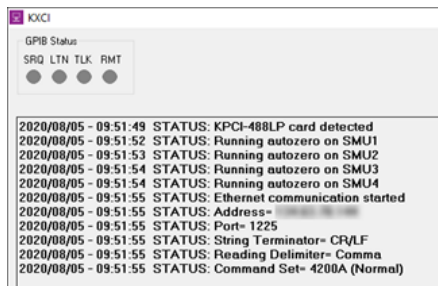
## Set up KXCI for ethernet control

### *To set up ethernet control:*

1. Open KCon.
2. Select **KXCI Settings**. The KXCI Settings dialog is displayed.
3. Set Communications to **Ethernet**.
4. Set the **Port Number** (the default is 1225, which is acceptable for most installations). Make note of the port number for use in the code.
5. Select a **Reading Delimiter**. The delimiter determines if the delimiter for output data is a string terminator or a comma.
6. If you selected String Terminator, select the **Character**. The options are:
  - **None** to use no character. Recommended to avoid extra termination characters. The 4200A-SCS automatically adds a null character as a terminator.
  - **CR** to use a carriage return.
  - **LF** to use a line feed.
  - **CR/LF** to use a carriage return and line feed character sequence.
7. Select **OK**.

When KXCI starts, the status messages look similar to the following figure when ethernet is set up.

**Figure 5: KXCI status when ethernet is selected**



## Setting up KXCI as a 4145B emulator

Although the KXCI command set is similar to the Keysight 4145B command set, the 4200A-SCS and Keysight 4145B hardware are different. To use existing 4145B code with the 4200A-SCS, you need to set up the 4200A-SCS SMUs to map to the 4145B instrument numbers. In many cases, test programs developed for use with a Keysight 4145B run without modification when they are used with a 4200A-SCS running KXCI. When a mode is selected, KXCI always starts in that mode.

The following table summarizes some differences and similarities between the modes.

### Mode comparison

Characteristic	Mode	
	4200A (Normal)	4145 Emulation
String reported in response to ID query	KI4200A Vx.x.x (where x.x.x is the version number)	ID HP4145B 1.1,1.0
GPIB data resolution	7 digits	5 digits
Maximum number of sweep points	4096	1024
Possible instrument configurations	8 source-measure units (SMU), voltmeters (VM), or voltage-sources (VS)	
Configuration query	*OPT? command	Not supported
Instrument self test	SMUs only	Not supported
Custom A/D control	IT4 command options	Not supported
200 V, 1 A capability	Supported	
1.0 pA source/measure-range capability (with preamplifier on SMU)	Supported	

The main difference between the two instruments is that the 4200A-SCS hardware is modular and the Keysight 4145B hardware is fixed, as shown in the following table.

#### Hardware comparisons

Instrument type	Keithley 4200A-SCS	Keysight 4145B
Source measure units (SMUs)	2 to 9	4 (fixed)
Voltage meter (VM)	You can configure up to nine SMUs to function as voltage meters.	2 (fixed)
Voltage source (VS)	You can configure up to nine SMUs to function as voltage sources.	2 (fixed)

KCon manages these hardware differences by allowing you to assign source-measure unit (SMU), voltage meter, or voltage source functions to any 4200A-SCS SMU, as shown in the following table.

#### KXCI SMU function assignment

Function selection	Description
SMU	Instructs the 4200-SMU, 4201-SMU, 4210-SMU, or 4211-SMU to emulate a Keysight 4145B Source Measure Unit.
VM1...VM8 (voltage meter)	Instructs the 4200-SMU, 4201-SMU, 4210-SMU, or 4211-SMU to emulate a Keysight 4145B VM1 or VM2 and additional voltage meters (VMs). You can map up to eight VMs to SMUs. You can assign a VM any number from 1 to 8, regardless of the number of SMUs in the system. Each VM number must be unique.
VS1...VS8 (voltage source)	Instructs the 4200-SMU, 4201-SMU, 4210-SMU, or 4211-SMU to emulate the capabilities of a Keysight 4145B VS1 or VS2 and additional voltage sources (VSs). You can map up to eight VSs to SMUs. You can assign a VS any number from 1 to 8, regardless of the number of SMUs in the system. Each VS number must be unique.

#### *To set up 4145B emulation:*

1. Open **KCon**.
2. Select **KXCI Settings**. The **KXCI Settings** dialog is displayed, as shown in the following figure.
3. In the **Function** column, select the 4145 function that the SMU will emulate.
4. In the **Number** column, select the number that is used for the device in your 4145B program.

**Figure 6: KXCI Settings for 4145B emulation**

The screenshot shows the 'KXCI Settings' dialog box. It has a title bar with a green icon and a close button. The main area is divided into several sections:

- Instrument List:** A table with columns 'Instrument', 'Function', and 'Number'.
 

Instrument	Function	Number
SMU1:	SMU	1
SMU2:	VS	2
Empty		
Empty		
Empty		
Empty		
Empty		
CVU1:		
PMU1:		
- Communication:** Radio buttons for 'GPIB' (selected) and 'Ethernet'.
- GPIB Address:** A dropdown menu showing '17'.
- Port Number:** A text box containing '1225'.
- Reading Delimiter:** Radio buttons for 'String Terminator' (selected) and 'Comma'.
- String Terminator:** Radio buttons for 'EOI On' (selected) and 'EOI Off'. A 'Character' dropdown menu shows 'CR/LF'.
- Command Set:** Radio buttons for '4200A (Normal)' and '4145 Emulation' (selected).
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

## GPIB communications

### NOTE

You cannot run Clarius and KXCI at the same time. When Clarius is running, the 4200A-SCS is the controller and controls all internal and external instruments. When KXCI is running, the 4200A-SCS is a subordinate to a controlling computer.

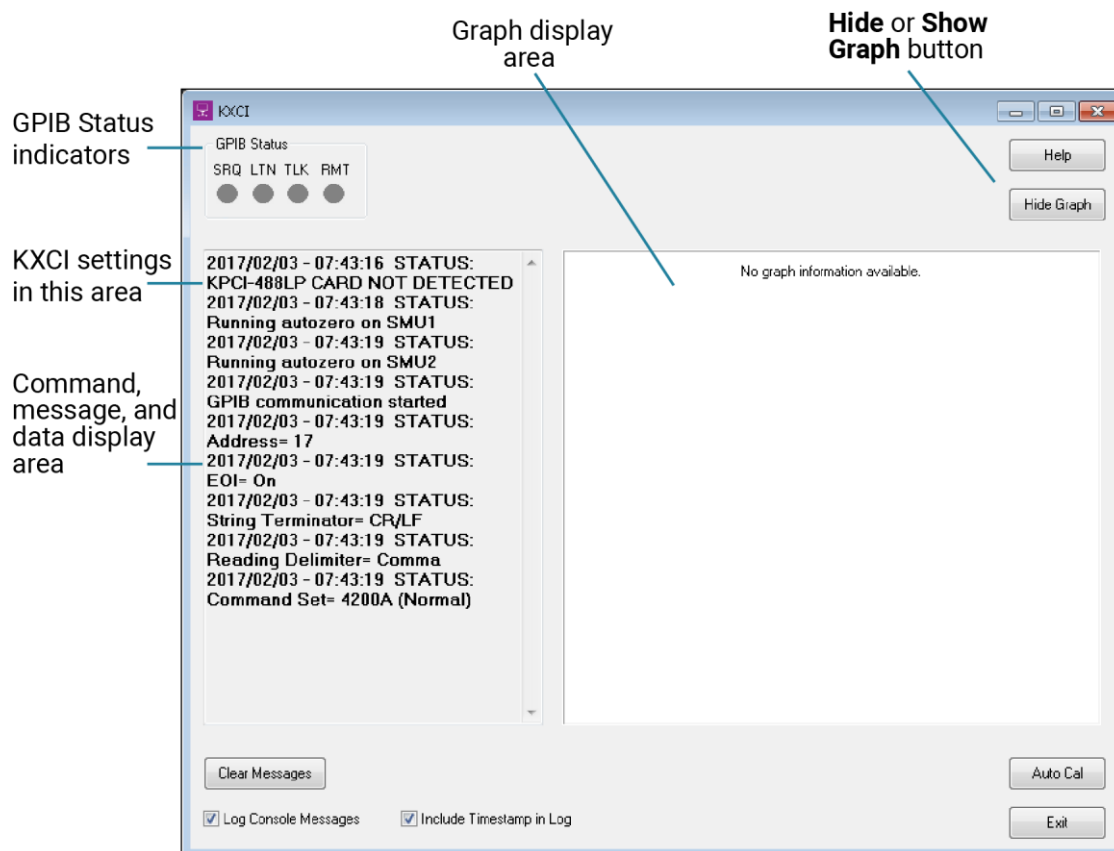
#### **To start KXCI:**

1. Close Clarius.
2. Select the KXCI icon on the desktop.

The KXCI user interface opens.

### NOTE

The GPIB Status indicators only apply if communications are set to GPIB.

**Figure 7: KXCI user interface**

## GPIB status indicators (GPIB communications only)

After KXCI is started, indicators in the GPIB Status box report interface status. A green indicator signals the present status:

- **SRQ:** Turns on when an error or operating condition occurs.
- **LTN:** When on, instrument is in the listener active state.
- **TLK:** When on, instrument is in the talker active state.
- **RMT:** When on, instrument is in the remote state.

## NOTE

If KXCI is running as a subordinate in a GPIB system, you may see the warning, "Access to all GPIB instruments from the 4200 as controller are illegal." This error occurs if you try to control other devices through user library modules. A subordinate node cannot control devices in the system.

## Ethernet communications

The 4200A-SCS is not LXI compliant, so be aware of the following communications considerations when using ethernet communications:

- Commands must always be terminated with null. If not, KXCI will not read and echo them.
- Ethernet commands always return a response. Commands that do not normally return a response respond with an acknowledge message and a null terminator (`ACK\0`).
- Commands that normally return a string or numerical data return that data terminated with the character that was set in KCon and a null terminator.

If you do not receive data as expected, there may have been an error with the command. Check the KXCI console for error messages. The command may have been sent too soon or had a syntax error. Another potential problem is that there are responses waiting in the buffer that must be cleared before the data can be returned.

### KXCI ethernet client driver

A KXCI client driver (32-bit or 64-bit) controls KXCI through the ethernet. You can copy this driver to your controlling computer. The DLL is stand-alone. It does not depend on any other DLLs, so it can be easily moved or copied.

The driver DLLs are named `KXCIclient.dll` (32-bit) and `KXCIclient64.dll` (64-bit) and are at the command path: `C:\s4200\sys\bin`

The `KXCIclient.lib` (32-bit) and `KXCIclient64.lib` (64-bit) files are at the command path:

`C:\s4200\sys\lib`

For convenience, a C-language header file (`KXCIclient.h`) is included with the above paths. It is at the command path:

`C:\s4200\sys\include`



## Driver functions

The `KXCIclient.dll` driver has the `OpenKXCICConnection_C`, `SendKXCICCommand_C`, `GetKXCISpollByte_C`, and `CloseKXCICConnection_C` functions, as described in the following.

```
int OpenKXCICConnection_C(char *IPAddrStr, int PortNum, int *err);
```

where:

- *IPAddrStr*: IP address in string format *nnn.nn.nn.nn* (for example, 198.00.02.00).
- *PortNum*: IP Port assigned in the KXCI tab of KCon.
- *err*: Socket error code returned by `WSAGetLastError()`.

```
int SendKXCICCommand_C(char *cmdstr, char *ReturnString, int *err);
```

where:

- *cmdstr*: KXCI command string, for example, "DE;CH1;CH2".
- *ReturnString*: Data returned by command, if any. If input command results in data to be returned, it is placed here.
- *err*: Socket error code returned by `WSAGetLastError()`

```
int GetKXCISpollByte_C(unsigned short *spbyte, int *err);
```

where:

- *spbyte*: KXCI serial poll byte (same as GPIB byte).
- *err*: Socket error code returned by `WSAGetLastError()`.

```
void CloseKXCICConnection_C(void);
```

## Example write and read with ethernet

The following commands and their responses were made with communications set to comma-separated data with the character set to carriage return (CR). These settings are made in KCon in the KXCI Settings option.

Written command	Response
*IDN?\0	KEITHLEY INSTRUMENTS,KI4200A,1442736,V1.8.1\r\0
CH1, 'VS', 'IS', 1, 3\0	ACK\0
ME\0	ACK\0
DO 'ID'\0	N 4.3555E-15,N 54.978E-15,...N 449.83E-15\r\0

## Error messages

KXCI error messages and numbers are shown in the following table.

**KXCI error messages**

Error number	Error message
-999	IEEE32.DLL GPIB driver is not loaded.
-998	Unable to initialize shared memory.
-997	Could not establish communications with console.
-996	GPIB address not sent as argv[1].
-995	GPIB address not in 0<= addr <= 31.
-994	Could not find configuration file.
-993	KXCI argument error.
-992	KXCI command error.
-991	Illegal setup error.
-990	Trigger Master card not found.
-989	Command not valid on this page.
-988	Instrument not mapped.
-987	Skipping instrument.
-986	Unsupported command received.
-985	Unsupported file format error.
-984	Could not open specified file.
-980	Command not valid during test execution.
-979	SMU not present in system.
-978	VPU not present in system.
-975	Command not valid in User Mode
-974	Command not valid in System Mode
-973	Command not valid when CVU test is running.
-972	Could not write log file.
-971	Custom cable compensation failed. HPOT/HCUR magnitude too small.
-970	Custom cable compensation failed. LPOT/LCUR magnitude too small.
-969	Invalid PMU argument.
-968	Pulse Instrument has reached thermal limit.
-967	Invalid pulse parameter configuration.
-966	Execution Aborted.
-965	Command not valid in the present pulse mode.
-964	Maximum Source Voltage Reached: Requested voltage across DUT resistance exceeds max voltage available.
-963	Maximum number of samples per channel exceeded.
-962	Invalid segment
-961	Unknown sequence
-960	Max sequences
-959	Start and stop values for defined segmented arb violate minimum slew rate.
-958	At least 3 segments must be defined per sequence
-957	Sequence time is too short for a measurement
-956	Segment start value does not match previous segment stop value

**KXCI error messages**

Error number	Error message
-955	SSR cannot be opened when using RPM ranges. Change SSR configuration to close relay or select PMU measure range.
-954	Maximum number of segments per PMU channel exceeded
-953	PMU Stepper cannot be used without Sweeper.
-952	Cannot adjust PMU measure sample rate for pulse configuration.
-951	Cannot adjust PMU measure sample rate for SARB configuration.
-950	This test has exceeded the system power limit.
-949	Maximum number of measure points exceeded for Pulse IV configuration.
-948	Sweep step count mismatch for the sweeping channels. All sweeping channels must have same # of steps.
-947	At least one Acquire High or Acquire Low parameter must be set to Enable for Pulse IV measurements.

## Using KXCI

### In this section:

Using KXCI.....	3-1
Output data formats for SMUs.....	3-1
Status byte and serial polling .....	3-3
Logging commands, errors, and test results .....	3-5

## Using KXCI

### *To start using KXCI:*

1. Close Clarius and KCon.
2. Double-click the KXCI icon on the desktop.

The 4200A-SCS is ready to accept commands immediately after you start KXCI.

For command information, refer to [KXCI SMU commands](#) (on page 5-1), [KXCI CVU commands](#) (on page 6-1), and [KXCI pulse generator commands](#) (on page 7-1).

### *To stop using KXCI:*

Select **Exit**. This stops operation and closes KXCI.

## Output data formats for SMUs

The following topics describe the data formats for system mode and user mode readings.

The hierarchy for data status is C, N:

- C: This channel is in compliance
- N: Normal

## Data format for system mode readings

For system mode operation, use the **DO** command to get one or more triggered readings. After sending the **DO** command string and addressing the 4200A-SCS to talk, the output data string is sent to the computer in the following format:

```
X±N.NNNNE±NN, X±N.NNNNE±NN, . . . X±N.NNNNE±NN
```

Where *X* is the data status and *N* is the reading.

## NOTE

Scientific notation for the reading exponent:

E+00 = 0

E-03 =  $10^{-3}$  (m)

E-06 =  $10^{-6}$  ( $\mu$ )

E-09 =  $10^{-9}$  (n)

E-12 =  $10^{-12}$  (p)

## Data format for user mode readings

For user mode operation, use the **TI** or **TV** command string to trigger and make a reading. After sending the **TI** or **TV** command string and addressing the 4200A-SCS to talk, the output data string is sent to the computer in the following format:

```
XYZ ±N.NNNN E±NN
```

Where *X* is the data status, *Y* is the instrument, *Z* is the measure mode, and *N* is the measurement. *Y* and *Z* are defined in the following tables.

Voltage measure mode specified (*Z* = V; see following tables).

Y =	A	SMU1
	B	SMU2
	C	SMU3
	D	SMU4
	E	VM1
	F	VM2
	G	SMU5
	H	SMU6
	I	SMU7
	J	SMU8
	K	VM3
	L	VM4
	M	VM5
	N	VM6
	O	VM7
	P	VM8

Current measure mode specified ( $Z = I$ ; see below)

Y =	A	SMU1
	B	SMU2
	C	SMU3
	D	SMU4
	E	SMU5
	F	SMU6
	G	SMU7
	H	SMU8

Measure mode

Z =	V	Voltage
	I	Current

Reading (mantissa and exponent)

$\pm N.NNNN E \pm NN$

## Status byte and serial polling

The status byte is an 8-bit register that provides status information on instrument operation. When a particular operating condition occurs, one or more bits of the status byte sets.

You can use serial polling to read the status byte.

The following sections describe the status byte and serial polling.

### Status byte

The status byte is an 8-bit register that provides status information on instrument operation. When an operating condition occurs, one or more bits of the status byte are set. The status byte register is shown in the following table.

**Status byte register**

Bit	B7	B6	B5	B4	B3	B2	B1	B0
Condition	—	RQS	—	Busy	—	—	Syntax error	Data ready
Binary value	—	0/1	—	0/1	—	—	0/1	0/1
Decimal weight	—	64	—	16	—	—	2	1

The bits of the status byte register that are used are described in the following topics.

## Bit B0, Data Ready

This bit sets (1) when all measurements are completed and data is ready to be output on the GPIB. Any of the following actions clear (0) bit B0:

- Clears (0) when the data transfer starts.
- Clears (0) when the BC (buffer clear) command is sent to the 4200A-SCS.
- Clears (0) when the 4200A-SCS is serial polled.

## Bit B1, Syntax Error

This bit sets (1) when an invalid command string is sent to the 4200A-SCS. Any of the following actions clear (0) bit B1:

- When the 4200A-SCS is serial polled.
- When a device clear command (`DCL` or `SDC`) is sent to the 4200A-SCS.

## Bit B4, Busy

This bit is set (1) while a measurement is being performed. It will clear (0) when the measurement is completed.

## Bit B6, RQS (request for service)

This bit sets (1) whenever bit B1 (syntax error) sets. If service request for data ready is enabled (`DR1` asserted), bit B6 will set whenever bit B0 (data ready) sets. If service request for data ready is disabled (`DR0` asserted), bit B6 will not be affected by bit B0.

Bit B6 remains set until one of the following actions occur:

- Clears (0) when the 4200A-SCS is serial polled.
- Clears (0) when a device clear command (`DCL` or `SDC`) is sent to the 4200A-SCS.

---

## NOTE

When bit B6 sets, the SRQ (service request) indicator on the KXCI user interface turns on. It turns off when B6 is cleared.

---

## Serial polling

The serial poll enable (SPE) and serial poll disable (SPD) general bus command sequence is used to serial poll the 4200A-SCS. Serial polling reads the status byte. Generally, the controller uses the serial polling sequence to determine which of several instruments has requested service with the SRQ line. However, the status byte of the 4200A-SCS may be read to determine when an operating condition has occurred.

If you try to get data before all the measurements in a test are completed, an error occurs. To prevent this, you can use serial polling in a program fragment to monitor the data ready bit (B0) of the status byte. When B0 sets, which indicates that data is ready, the program obtains the measurement data.

After the source-measure testing process is triggered to start (ME1 is sent to start a sweep), the following C-language programming statement serial polls the instrument:

```
spoll(addr, &poll, &status);
```

## Waiting for SRQ

Instead of serial polling the 4200A-SCS to detect an SRQ, you can monitor the service request line. When an SRQ occurs, the SRQ line is asserted.

If you are using GPIB connections, you can use the following C-language programming routine stop program execution until an SRQ occurs.

```
send(addr, "DR1", &status);  
while(!srq());
```

The first statement enables service request for data ready. The second command holds up program execution until the SRQ (data ready) occurs.

For ethernet connections, you can use the SP command to monitor the service request line.

## Logging commands, errors, and test results

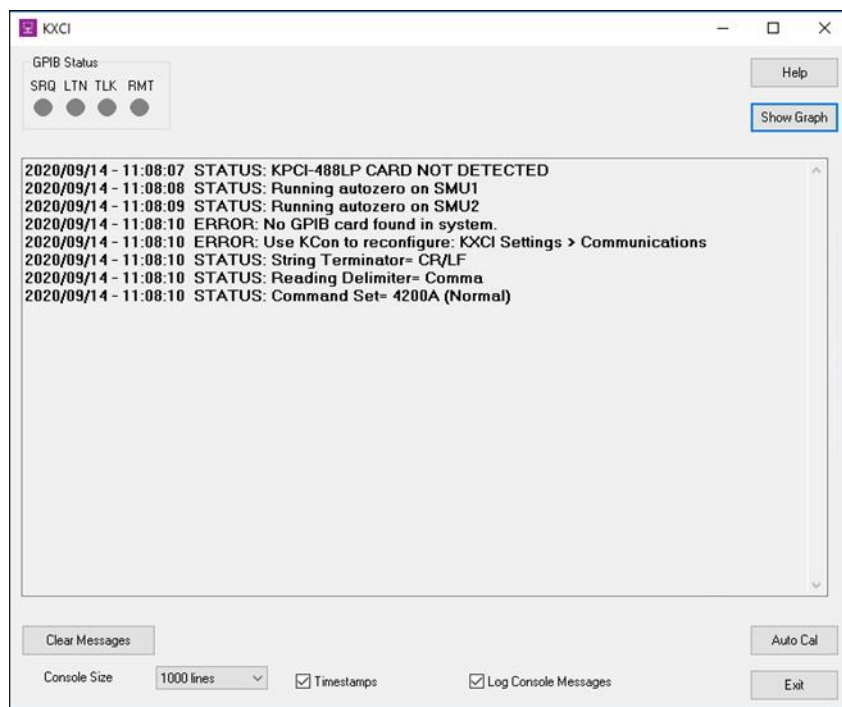
When you send commands, KXCI logs the commands, error messages, and test results as described in the following topics.



## Logging commands

When you send a command, the left side of the user interface (the command and message display area) displays each command as it is received, as shown in the following figure.

**Figure 8: KXCI command and message display area**



To remove existing messages, select **Clear Messages**.

You can use Console Size to set the number of messages that are displayed. You can select 1, 100, or 1000 lines.

If Log Console Messages is selected, KXCI also logs each command into the KXCI log file (C:\s4200\sys\KXCI\KXCILogfile.txt).

If Timestamps is selected, KXCI logs a timestamp for each message.

## Logging errors

The command and message display area displays error messages as they occur.

## Logging test results

The command and message display area also displays the numerical test results for both the 4200A and 4145 Emulation commands sets (refer to “Command Set” in *Model 4200A-SCS Setup and Maintenance* (4200A-908-01)).

You can turn off the graph display to better display a long sequence of test results. Select **Hide Graph** in the upper right to hide the graph.

---

### NOTE

The test results are 0.0000 if the interlock is disconnected.

---

If the sent commands include the graphics commands, the graph display area graphs the test data. Refer to [Graphing the test results](#) (on page 3-8).

## Understanding the log file

If Log Console Messages is selected, KXCI logs all commands to a file named `KXCILogfile.txt`. You can open the text file after KXCI is closed. Note that whenever you open KXCI, the log file clears.

The log file is in the directory:

`C:\s4200\sys\KXCI\`

You can use any text editor to open the file.

SMUs may return a line such as:

`DATA:NAI 22.329E-09`

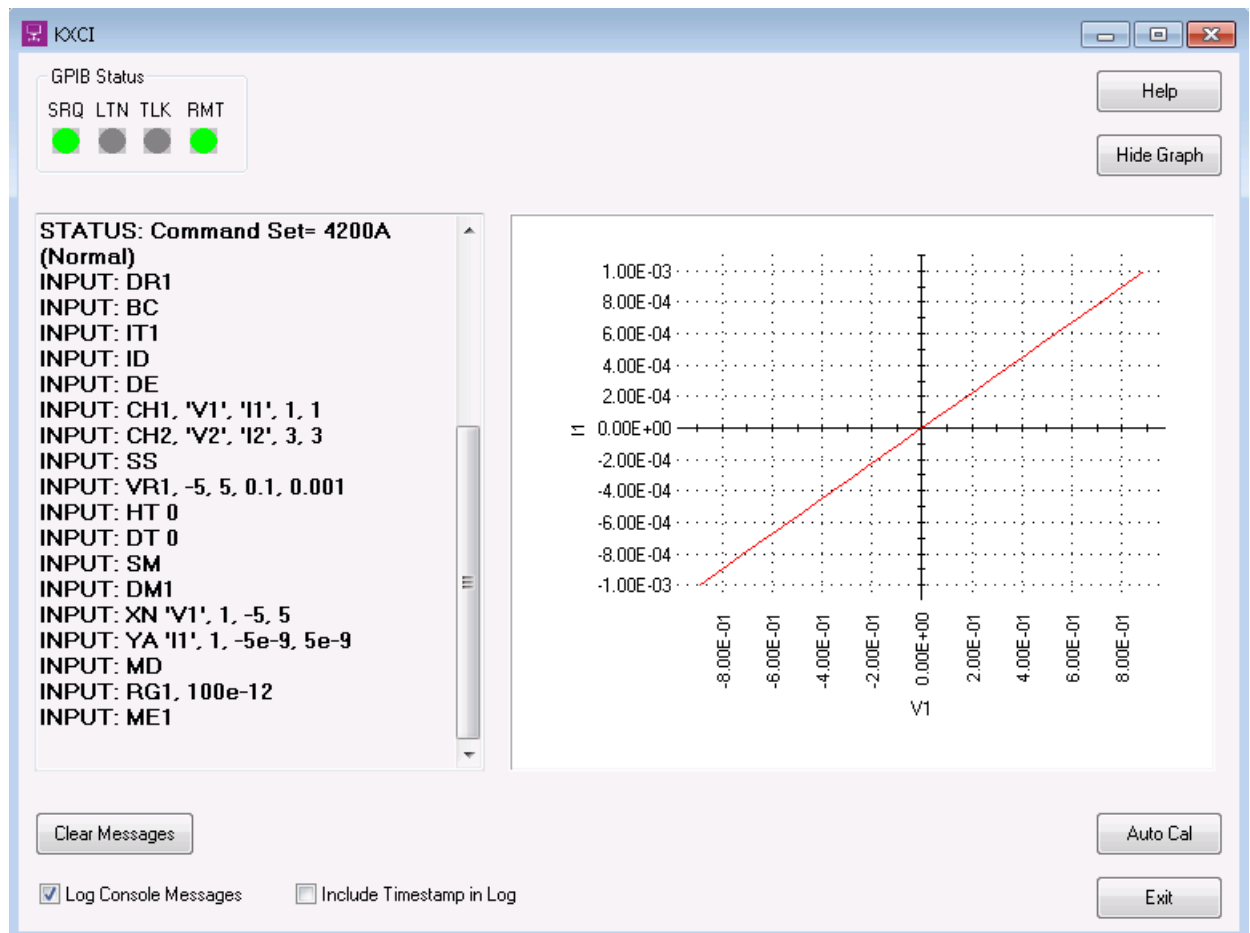
The characters after `DATA:` represent:

- The operating condition: `N` for normal or `C` if the SMU is in compliance.
- The instrument number, converted to a letter: `A` for SMU1, `B` for SMU2 and so on to letter `I`.
- The measurement function: `V` for voltage or `I` for current.

## Graphing the test results

If you have sent the graphics commands (the `DM1` command followed by the X-axis and Y-axis configuration commands), KXCI displays a graph of the generated data. See the example graph and graphics commands in the following figure.

**Figure 9: KXCI data graph**



## NOTE

KXCI plots all Y1 axis curves in red and all Y2 axis curves in blue. To hide the graph, select **Hide Graph**.

## KXCI common commands

### In this section:

Common commands .....	4-1
Calling KULT user libraries remotely .....	4-6

## Common commands

The command strings described in this section are valid for all instruments and for all operating modes unless otherwise noted. They are used for the operations listed in the following table.

Command	Brief description
<a href="#">DR</a> (on page 4-1)	Enables or disables service request for data ready when communications is set to GPIB.
<a href="#">:ERROR:LAST:CLEAR</a> (on page 4-2)	Clears the last error generated by a KXCI command.
<a href="#">:ERROR:LAST:GET</a> (on page 4-3)	Retrieves error messages from KXCI for any instrument.
<a href="#">ID</a> (on page 4-3)	Places the instrument ID of the 4200A-SCS in a buffer.
<a href="#">*IDN?</a> (on page 4-4)	Generates an identification query and reads identification information.
<a href="#">*OPT?</a> (on page 4-4)	Returns a result string that contains the 4200A-SCS slot configuration for KXCI.
<a href="#">*RST</a> (on page 4-5)	Resets the instrument settings to default settings.
<a href="#">SP</a> (on page 4-6)	Allows you to acquire the GPIB serial poll byte when ethernet communications are selected.

## DR

This command enables or disables service request for data ready when communications is set to GPIB.

### Usage

DRA

A	Set service request for data ready: <ul style="list-style-type: none"> <li>■ Disable service request for data ready: 0</li> <li>■ Enable service request for data ready: 1</li> </ul>
---	---

---

**Details**

---

Available for all instruments.

This command is not available for ethernet communications. Use `SP` for ethernet communications.

Use a service request to wait until operations are complete before downloading data.

GPIB provides a status byte register to monitor instrument operation. Two bits of this register are bit B0 (Data Ready) and bit B6 (RQS). After all programmed measurements are completed, the data becomes ready for output and bit B0 (Data Ready) sets.

If the service request for data ready is enabled, bit B6 (RQS) is also set when data is ready for output. If the service request for data ready is disabled, bit B6 is not set when data is ready.

---

**Also see**

---

[SP](#) (on page 4-6)

---

## **:ERROR:LAST:CLEAR**

This command clears the last error generated by a KXCI command.

---

**Usage**

---

```
:ERROR:LAST:CLEAR
```

---

**Details**

---

Call this command at the beginning or end of the test to clear the KXCI error queue.

---

**Example**

---

```
:ERROR:LAST:CLEAR
```

Errors are removed from the buffer and `:ERROR:LAST:GET` will no longer return errors.

---

**Also see**

---

[Error messages](#) (on page 2-12)

[:ERROR:LAST:GET](#) (on page 4-3)

---

## :ERROR:LAST:GET

This command retrieves error messages from KXCI for any instrument.

### Usage

---

```
:ERROR:LAST:GET
```

### Details

---

This command returns the last stored error in KXCI in the format:

```
<ErrorMessage><ErrorCode><StringTerminators>
```

The read is non-destructive. If you send `ERROR:LAST:GET` multiple times without clearing, the same error message is returned.

In a test script, send `:ERROR:LAST:CLEAR` at the beginning of a script to clear any earlier errors so that `:ERROR:LAST:GET` errors are specific to the present test.

### Example

---

```
:ERROR:LAST:GET
```

Returns the last error. For example:  
KXCI command error. (-992)

### Also see

---

[Error messages](#) (on page 2-12)

[:ERROR:LAST:CLEAR](#) (on page 4-2)

---

## ID

This command places the instrument ID of the 4200A-SCS in a buffer.

### Usage

---

```
ID
```

### Details

---

Available for all instruments.

The instrument ID depends on whether you are in 4200A mode or whether you are in the 4145 Emulation mode.

If you are in 4200A mode, the return is `KI4200A Vx.x.x` (where `x.x.x` is the version number).

If you are in 4145 Emulation mode, the return is `ID HP4145B 1.1,1.0`.

For additional details about the differences between the 4200A mode and 4145 emulation mode, refer to [Setting up KXCI as a 4145B emulator](#) (on page 2-6).

### Also see

---

None

---

## \*IDN?

This command generates an identification query and reads identification information.

### Usage

---

\*IDN?

### Details

---

Available for all instruments.

The identification code includes the manufacturer, model number, serial number, and firmware revision of the instrument. The string is formatted as:

```
KEITHLEY INSTRUMENTS,MODEL nnnn,xxxxxxx,yyyyy
```

Where:

- *nnnn* = the model number
- *xxxxxxx* = the serial number
- *yyyyy* = the firmware revision level

---

## NOTE

This command must include \* before the command.

---

### Also see

---

None

---

## \*OPT?

This command returns a result string that contains the 4200A-SCS slot configuration for KXCI.

### Usage

---

\*OPT?

### Details

---

When the 4200A-SCS receives this command, it returns a configuration string with up to eight sets of characters:

```
xxxn,xxxn,xxxn,xxxn,xxxn,xxxn,xxxn,xxxn
```

Each character set represents the configuration of a slot in the 4200A-SCS. If the corresponding slot contains a channel, *xxx* indicates the instrument in slot. *n* is the channel number (1 to 9).

The possible returns for `xxx` are described in the following table.

<b>SMUs</b>	
SMU	Medium-power SMU without a preamplifier
HPSMU	High-power SMU without a preamplifier
SMUPA	Medium-power SMU with a preamplifier
HPSMUPA	High-power SMU with a preamplifier
<b>CVUs</b>	
CVU	Capacitance Voltage Unit
<b>PMUs and PGUs</b>	
VPU	PGU (source only)
PMU	PMU without an RPM
PMURPM	PMU with an RPM
PMUnRPM1	PMU with RPM tied to channel 1
PMUnRPM2	PMU with RPM tied to channel 2
PMUnRPM1-2	PMU with RPM tied to channel 1 and channel 2
<b>Voltage sources</b>	
VS	Voltage source only (20 V is presently supported)
<b>Voltage meters</b>	
VM	Voltage meter
<b>Other</b>	
""	No channel in slot

This command is not available if you are using the 4145 Emulation mode.

#### Also see

None

## \*RST

This command resets the instrument settings to default settings.

#### Usage

\*RST

#### Details

Available for all instruments.

This command must include `*` before the command.

This command:

- Resets SMUs, PGUs, PMUs, and CVUs.
- Executes `:CVU:RESET`.
- Clears data from previous PMU pulses and deletes all Segment Arb sequences.



This command does not reset the 4200A-CVIV.

---

**Example**

```
*RST
```

Returns the instrument to default settings and cancels all pending commands.

---

**Also see**

[:CVU:RESET](#) (on page 6-22)

[:PMU:INIT](#) (on page 7-9)

---

## SP

The `SP` command allows you to acquire the GPIB serial poll byte when ethernet communications are selected.

---

**Usage**

`SP`

---

**Details**

Available for all instruments.

This command is not available for GPIB communications. Use `DR` for GPIB communications.

---

**Also see**

[DR](#) (on page 4-1)

## Calling KULT user libraries remotely

KXCI contains a set of commands to call user libraries built by the Keithley User Library (KULT) on the 4200A-SCS from a remote interface. Refer to *Model 4200A-SCS KULT and KULT Extension Programming* (4200A-KULT-907-01) for details on using KULT.

---

### NOTE

The LPT commands `pulse_measrt` and `PostDataDoubleBuffer` are not compatible with KXCI and cannot be used in user libraries called by KXCI. Use `PostDataDouble` for user test modules (UTMs) that are called using KXCI. Refer to *Model 4200A-SCS LPT Library Programming* for additional information on these commands.

---

## Summary of commands for remote calls

Command	Brief description
<a href="#">AB</a> (on page 4-7)	Aborts the execution of a module of the user library.
<a href="#">EX</a> (on page 4-7)	Executes a user module using specified parameter values.
<a href="#">GD</a> (on page 4-9)	Queries and returns the description of a KULT module.
<a href="#">GN</a> (on page 4-10)	Queries input or output parameter values, or both, by name for the last user module run in KXCI.
<a href="#">GP</a> (on page 4-11)	Queries input or output parameter values, or both, by number for the last user module run in KXCI.
<a href="#">GR</a> (on page 4-12)	Recalls the return value of a UTM and prints it in the KXCI console.
<a href="#">instrumentinfo</a> (on page 4-14)	Retrieves all information on the instrument cards in the 4200A-SCS and the 4200A-CVIV Multi-Switch, if attached.
<a href="#">remotemoduleinfo</a> (on page 4-15)	Retrieves all information on the preamplifiers and remote preamplifier/switch modules (RPMs) that are attached to the 4200A-SCS.
<a href="#">systeminfo</a> (on page 4-16)	Retrieves 4200A-SCS system information.
<a href="#">UL</a> (on page 4-13)	Switches KXCI operation to the user library ( <code>usrlib</code> ) mode.

---

### AB

This command aborts the execution of a module of the user library.

#### Usage

---

AB

#### Also see

---

[EX](#) (on page 4-7)

---

### EX

This command executes a user module using specified parameter values.

#### Usage

---

`EX UserLibrary UserModule(param1, param2, param3...)`

<i>UserLibrary</i>	The name of the user library that contains the module to be run
<i>UserModule</i>	The name of the user module to be run
<i>param1, param2, param3...</i>	The specified input or output parameters, or both, for the user module (function); see Details

---

**Details**

---

Use the following rules for the input and output parameters:

- Separate parameter values by commas. Put only the string or name between commas, without a leading space, to prevent parsing errors.
- Separate multiple values in an array by semicolons.
- Input and output array parameters must have fewer than 16,000 elements.
- Do not use braces or brackets to enclose array parameters.
- Do not use quotation marks to enclose strings or names.

For an input parameter, type in the value of the parameter. If the position for an input parameter is left empty, the default value for the parameter is used.

For an output parameter, leave the space empty, as shown in the following example.

---

## NOTE

When used before the `GN` or `GP` commands, you may need to add a delay to allow the execution of the module to finish.

---

---

**Example 1**

---

```
EX my_2nd_lib VSweep(0, 5, , 11, , 11)
```

Assume that the following user module (built in KULT) performs a voltage sweep and stores the test voltages and measured current readings in arrays:

- User Library: `my_2nd_lib`
- User Model: `VSweep`

The parameter sequence for the `VSweep` function is as follows:

```
Vstart (input), Vstop (input), Imeas (output), NumIPoints (input), Vforce  
(output), NumVPoints (input)
```

This example shows how a user function can be run from the KXCI console using parameters that perform an 11-point sweep that starts at 0 V and stops at 5 V.

When the KXCI is in the `usrlib` mode, the example command runs the KULT function.

After execution of the module completes, you can query input or output parameters or both to return the values. Use `GN` or `GP` to query parameters.

Example 2

```
EX sweep_lib list_sweep(SMU1,SMU2, 0.0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0, 7, , 7,
,7)
```

Assume that the following module (built in KULT) performs a voltage list sweep and stores the forced voltages and measured currents in the following arrays:

- User Library: `sweep_lib`
- User Module: `list_sweep`

The parameter sequence for the `list_sweep` function is as follows:

```
forceSMU (input), measureSMU (input), VoltageList (input), numVoltages
(input), ForceV (output), numForceV (input), measI (output), numMeasI
(input)
```

This example shows how a user module that has string and array input parameters can be run from the KXCI console. The two SMUs are specified by their string names, separated by commas, without spaces. The voltage list is an input array of doubles that represent the list of voltages to sweep. The values in the list are separated by semicolons so that the forcing SMU sweeps from 0 V to 3 V in 0.5 V steps. When KXCI is in the `usrlib` mode, this example command runs the user module. After execution is complete, you can query the input or output parameters to return values. Use `GN` or `GP` to query parameters.

Also see

- [AB](#) (on page 4-7)
- [GN](#) (on page 4-10)
- [GP](#) (on page 4-11)

GD

This command queries and returns the description of a KULT module.

Usage

GD *UserLibrary* *UserModule*

<i>UserLibrary</i>	The name of the user library that contains the KULT module to be run
<i>UserModule</i>	The name of the user module to be run

Example

```
GD my_2nd_lib VSweep
```

This command queries the description of the user module in the example in the `EX` command description. The description is displayed in the KXCI console.

Also see

- [EX](#) (on page 4-7)

# GN

This command queries input or output parameter values, or both, by name for the last user module run in KXCI.

## Usage

```
GN ParameterName
GN ParameterName NumValues
```

<i>ParameterName</i>	The name of the parameter in the KULT module
<i>NumValues</i>	The number of values in an output array to be returned; see <b>Details</b>

## Details

The *NumValues* parameter is only used for an output parameter that is an array. If *NumValues* is not used, one value is returned. *NumValues* must be less than 16,000.

Arrays are returned as a list of values separated by semicolons.

The value returned for an input parameter is the given value. The values returned for an output parameter is the outcome of the test (for example, measured readings).

The output is truncated after 1024 characters on the KXCI console, but the full data string is returned to the caller.

## NOTE

You can also query a parameter by specifying the corresponding number for the parameter in the KULT module using [GP](#) (on page 4-11).

## Example

```
GN Vforce 11
This command queries all 11 test voltages (Vforce parameter) for the function that was run in the
command description for EX.
The following array of test voltages are returned and displayed in the KXCI console:
0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0; 3.5; 4.0; 4.5; 5.0
```

## Also see

- [EX](#) (on page 4-7)
- [GP](#) (on page 4-11)

# GP

This command queries input or output parameter values, or both, by number for the last user module run in KXCI.

## Usage

```
GP ParameterPosition
GP ParameterPosition NumValues
```

<i>ParameterPosition</i>	The position of the parameter in the KULT module
<i>NumValues</i>	The number of values in an output array to be returned; see <b>Details</b>

## Details

*NumValues* is only used for an output parameter that is an array. If *NumValues* is not used, one value is returned. *NumValues* must be less than 16,000.

Arrays are returned as a list of values separated by semicolons.

The value returned for an input parameter is the given value. The values returned for an output parameter is the outcome of the test (for example, measured readings).

The output is truncated after 1024 characters on the KXCI console, but the full data string is returned to the caller.

## NOTE

You can also query a parameter by specifying the name of the parameter in the user module. See [GN](#) (on page 4-10).

## Example

```
GP 5 11
This command queries all test voltages for the Vforce parameter for the user module that was run in the EX example. Vforce is the fifth parameter in the function.
The following array of test voltages is returned and displayed in the KXCI console:
0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0; 3.5; 4.0; 4.5; 5.0
```

## Also see

None

---

## GR

This command recalls the return value of a UTM and prints it in the KXCI console.

### Usage

---

GR

### Details

---

Send the GR command after a UTM has finished to recall the return value of the UTM. The return value is printed in the KXCI console.

To use GR, you must call UL to set KXCI to UTM mode and then call a UTM using EX.

If the UTM encounters errors, GR returns the error message. If the error message is more than 32 characters, it is truncated.

GR does not return the result to the sender. For example, if you query the GR command (send and receive), the return is ACK. The actual return value is in the KXCI console. To retrieve the value, use the EX command.

If the return value of the UTM is void and there are no errors, GR returns void.

### Example 1

---

```
my4200a.write("UL")
value = my4200a.query("EX BeepLib BeepCharge()")
time.sleep(5) #Wait for test to finish
my4200a.write("GR")
```

This Python example requests that the return value of the BeepCharge user module of the BeepLib user library be written to the KXCI console. The return is:

```
2024/11/13 - 11:16:03 INPUT: UL
2024/11/13 - 11:16:03 INPUT: EX BeepLib BeepCharge()
2024/11/13 - 11:16:08 Return Value = void
2024/11/13 - 11:16:13 INPUT: GR
2024/11/13 - 11:16:13 Get Return Value = void
```

### Example 2

---

```
my4200a.write("UL")
value = my4200a.query("EX PMU_examples_ulib PMU_10ns_Pulse_Example()")
time.sleep(10)
my4200a.write("GR")
```

This Python example shown an example of an error return. It requests that the return value of the PMU\_10ns\_Pulse\_Example user module of the PMU\_examples\_ulib user library be written to the KXCI console. The return is:

```
2024/11/13 - 11:16:03 INPUT: UL
2024/11/13 - 11:16:03 INPUT: EX PMU_examples_ulib PMU_10ns_Pulse_Example()
2024/11/13 - 11:16:08 Return Value = EX ERROR: invalid number of UTM
    parameters:number of parameters in a call 0 number of parameters required:
    2
2024/11/13 - 11:16:13 INPUT: GR
2024/11/13 - 11:16:13 Get Return Value = EX ERROR: invalid number of U
```

### Also see

---

[EX](#) (on page 4-7)

[UL](#) (on page 4-13)

---

## UL

This command switches KXCI operation to the user library (`usrlib`) mode.

### Usage

---

UL

### Details

---

This command allows for the use of direct user library module calls.

This command needs to be sent only once before any of the other commands to call user modules. To switch back to normal KXCI command modes, send `DE` or `US`.

Send the `UL` command through GPIB or ethernet to change to the remote `usrlib` command set.

### Example

---

```
UL
EX cvivulib cviv_configure (CVIV1, 1, 2, 3, 0, 0, CVHI, CVLO, NC, NC, CV Meas)
```

Select the user library mode.  
Execute the user module `cviv_configure` in the user library `cvivulib`. Refer to the user module source code or the Clarius Help pane for descriptions of the user module parameters and return values.

### Also see

---

[Channel definition page \(DE\)](#) (on page 5-7)

[User mode commands](#) (on page 5-31)

## SystemUtil User Library

This user library permits KXCI using the `UL` mode (see [Calling KULT user libraries remotely](#) (on page 4-6)) to retrieve information about the 4200A-SCS instrument and the system.

---

## NOTE

This user library is not compatible with Clarius user test modules (UTMs). It only works with the `UL` mode of KXCI.

---



# instrumentinfo

This command retrieves all information on the instrument cards in the 4200A-SCS and the 4200A-CVIV Multi-Switch, if attached.

## Usage

```
int instrumentinfo(char *result, int maxlen, int *len);
```

<i>result</i>	String of results for the instruments
<i>maxlen</i>	The maximum number of bytes that <i>result</i> parameter can store in the buffer
<i>len</i>	Number of bytes returned by the function

## Details

This function returns all of the instrument-level information for the cards in the 4200A-SCS and the 4200A-CVIV, if attached. The results contain the following information for each instrument card in the 4200A-SCS chassis:

- Slot number (for the 4200A-CVIV, 1 is returned)
- Instrument card ID
- Model number
- Serial number
- Hardware version
- Firmware version
- Calibration date
- Calibration due date

All the information is comma-separated.

## Example

```
Output for a 4200A-SCS with three instrument cards installed in slot 3 (4220-PGU), slot 5 (4225-PMU),
and slot 7 (4210-CVU or 4215-CVU):
slotno:3,name:VPU1,model:KIVPU4220,serialno:1254281,hwver:1.0,fwver:1.50,calda
te:Dec 19, 2009,caldue:Dec 19, 2010,
slotno:5,name:PMU1,model:KIPMU4225,serialno:1276563,hwver:1.0:544911,fwver:1.5
0,caldate:Nov 28, 2011,caldue:Nov 27, 2012,
slotno:7,name:CVU1,model:KICVU4210,serialno:Z005712,hwver:3.0:493083,fwver:2.0
3,caldate:Aug 09, 2010,caldue:Aug 09, 2011
```

## Also see

[remotemoduleinfo](#) (on page 4-15)

---

## remotemoduleinfo

This command retrieves all information on the preamplifiers and remote preamplifier/switch modules (RPMs) that are attached to the 4200A-SCS.

### Usage

---

```
int remotemoduleinfo(char *result, int maxlen, int *len);
```

<i>result</i>	String of results for the remote modules
<i>maxlen</i>	The maximum number of bytes that the result can store in the buffer
<i>len</i>	Number of bytes returned by the function

### Details

---

This function returns all information on the preamplifiers and RPMs that are attached to the 4200A-SCS. The results contain the following information for each preamplifier and RPM:

- Slot number
- Instrument card ID
- Model number
- Serial number
- Hardware version
- Firmware version
- Calibration date
- Calibration due date

All the information is comma-separated.

### Example

---

```
Output for a 4200A-SCS with two preamplifiers (attached to SMUs in slot 1 and slot 2) and one RPM
attached to channel 1 of the PMU in slot 6:
slotno:1,name:PA1,model:KI4200,serialno:0734120,hwver:D7481,fwver:E02.1,caldate:Feb 29, 2016,caldue:Mar 01, 2017,
slotno:2,name:PA2,model:KI4200,serialno:0901553,hwver:D08288,fwver:E03,caldate:Feb 28, 2016,caldue:Feb 28, 2017,
slotno:6,name:RPM1-1,model:KIRPM4225,serialno:1314347,hwver:1.4,fwver:2.00,caldate:Jan 31, 2017,caldue:Jan 31, 2018
```

### Also see

---

[instrumentinfo](#) (on page 4-14)

---

# systeminfo

This command retrieves 4200A-SCS system information.

## Usage

---

```
int systeminfo(char *result, int maxlen, int *len);
```

<i>result</i>	String of results for the system
<i>maxlen</i>	The maximum number of bytes that the result can be stored in the buffer
<i>len</i>	Number of bytes returned by the function

## Details

---

This function returns system level information for the 4200A-SCS. The *result* string contains the following information:

- 4200A-SCS serial number
- System software version
- System platform version
- Clarius+ application version

The results are comma-separated.

## Example

---

Example system info output:  
serialno:1209478,swver:4200A-852-1.0,platformver:4200A-300-1, clariusver:V1.2

## Also see

---

None

---

## KXCI SMU commands

### In this section:

KXCI SMU commands .....	5-1
Summary of KXCI SMU commands .....	5-2
Summary of page commands .....	5-4
SMU default settings .....	5-4
System mode SMU default settings .....	5-5
Command reference .....	5-6
Code examples .....	5-44

## KXCI SMU commands

The SMU command set includes the following command types:

- Page commands
- System mode commands
- User mode commands
- 4200A only commands
- Modeless commands

KXCI locates the instrument control routines in pages that are similar to the Keysight Model 4145B command pages. On earlier instruments, page controls allowed you to switch between screens on the instrument. You could page between configuration, data returns, and graphing. This paradigm is used in KXCI, so before sending an instrument control command string, you must send one of the appropriate page commands, which are summarized in [Page commands](#) (on page 5-4).

The system mode commands configure the SMUs and make readings. They are a comprehensive set of commands that use all the source-measure capabilities of up to eight SMUs installed in the 4200A-SCS. These commands are described in [System mode commands](#) (on page 5-6).

The user mode commands are used for basic source-measure operation, including SMU configuration and data retrieval. These commands are described in [User mode commands](#) (on page 5-31).

The modeless commands can be used in any operating mode. These commands are described in [Modeless commands](#) (on page 5-36).

To retrieve the slot configuration for KXCI, refer to [\\*OPT?](#) (on page 4-4).

The KXCI command set for the SMUs is similar to the command set used by the Keysight Model 4145B. This similarity allows many programs already developed for the Keysight Model 4145B to run on the 4200A-SCS.

## Summary of KXCI SMU commands

Mode Page	Command	Brief description
System Channel Definition (DE)	<a href="#">CH</a> (on page 5-7)	Defines a SMU channel.
System Measurement Setup (SM)	<a href="#">DM</a> (on page 5-22)	Selects the Keysight 4145B display mode.
System Source Setup (SS)	<a href="#">DT</a> (on page 5-10)	Sets the time to wait between when the output voltage is set and when the measurement is made in a sweep.
System Source Setup (SS)	<a href="#">FS</a> (on page 5-11)	Sets the offset value when VAR1 ' is a selected source function.
System Any system mode page	<a href="#">GT</a> (on page 5-29)	Acquires (loads) the saved data file or program file.
System Source Setup (SS)	<a href="#">HT</a> (on page 5-12)	Sets a hold time that delays the start of a sweep.
System Measurement Setup (SM)	<a href="#">IN</a> (on page 5-22)	Sets the time between sample measurements.
System Measurement Setup (SM)	<a href="#">LI</a> (on page 5-23)	Enables voltage and current functions to be measured when the 4200A-SCS is in list display mode.
System Measurement Control (MD)	<a href="#">ME</a> (on page 5-28)	Controls measurements.
System Any system mode page	<a href="#">MP</a> (on page 5-30)	Maps channel <i>n</i> to a given VS, SMU, or VM function.
System Measurement Setup (SM)	<a href="#">NR</a> (on page 5-23)	Sets the number of readings that can be made for time domain measurements.
System Source Setup (SS)	<a href="#">RT</a> (on page 5-13)	Sets the ratio value when VAR1 ' is a selected source function.
System Source Setup (SS)	<a href="#">SC</a> (on page 5-14)	Configures the source to output a fixed voltage level for channels that are configured to be voltage source only.
System Any system mode page	<a href="#">SR</a> (on page 5-31)	Sets a fixed source range on channel <i>n</i> .
System Source Setup (SS)	<a href="#">ST</a> (on page 5-14)	Enables or disables automatic standby.
System Any system mode page	<a href="#">SV</a> (on page 5-29)	Saves a program file or data file.

Mode Page	Command	Brief description
System Source Setup (SS)	<a href="#">VC and IC</a> (on page 5-15)	Configure the SMU to output a fixed (constant) voltage or current level.
System Source Setup (SS)	<a href="#">VL and IL</a> (on page 5-16)	Set up a list sweep.
System Channel Definition (DE)	<a href="#">VM</a> (on page 5-8)	Defines channels that are used as voltmeters.
System Source Setup (SS)	<a href="#">VP and IP</a> (on page 5-17)	Set up the VAR2 step sweep.
System Source Setup (SS)	<a href="#">VR and IR</a> (on page 5-19)	Set up the VAR1 source function.
System Channel Definition (DE)	<a href="#">VS</a> (on page 5-9)	Specifies the name and selects the source function for each voltage-source channel.
System Measurement Setup (SM)	<a href="#">WT</a> (on page 5-24)	Delays the start of a test sequence for time domain measurements.
System Measurement Setup (SM)	<a href="#">XN</a> (on page 5-25)	Configures the X-axis of the graph to plot an electrical parameter.
System	<a href="#">XT</a> (on page 5-25)	Configures the X-axis of the graph to plot time domain values in seconds.
System Measurement Setup (SM)	<a href="#">YA</a> (on page 5-26)	Configures the Y1-axis of the graph.
System Measurement Setup (SM)	<a href="#">YB</a> (on page 5-27)	Configures the Y2-axis of the graph.
User	<a href="#">DI</a> (on page 5-32)	Sets up a SMU channel as a current source.
User	<a href="#">DS</a> (on page 5-33)	Specifies the channel number and the voltage output value for each voltage source.
User	<a href="#">DV</a> (on page 5-33)	Sets up a SMU channel as a voltage source.
User	<a href="#">TI</a> (on page 5-34)	Triggers a current measurement.
User	<a href="#">TV</a> (on page 5-35)	Triggers a voltage measurement.
Modeless	<a href="#">AC</a> (on page 5-37)	Autocalibrates a SMU channel.
Modeless	<a href="#">BC</a> (on page 5-37)	Clears all readings from the buffer.
Modeless	<a href="#">DO</a> (on page 5-38)	Returns the timestamp data that was acquired with the voltage or current measurement, or both.
Modeless	<a href="#">EC</a> (on page 5-39)	Sets the condition to exit the test if compliance is reached.
Modeless	<a href="#">EM</a> (on page 5-39)	Switches between 4145 Emulation and 4200A command sets.
Modeless	<a href="#">IT</a> (on page 5-40)	Sets the integration time of a SMU.
Modeless	<a href="#">RD</a> (on page 5-41)	Requests real-time readings.
Modeless	<a href="#">RG</a> (on page 5-42)	Sets the lowest current range of the SMU to be used when measuring.
Modeless	<a href="#">RI</a> (on page 5-42)	Instructs a SMU to go to a specified current range immediately without waiting until the initiation of a test.
Modeless	<a href="#">RS</a> (on page 5-43)	Sets the measurement resolution for all channels of a SMU.
Modeless	<a href="#">RV</a> (on page 5-43)	Instructs a SMU to go to a specified voltage range immediately without waiting until the initiation of a test.

## Summary of page commands

Page command	Function
<a href="#">DE</a> (on page 5-7)	Accesses SMU channel definition page.
<a href="#">SS</a> (on page 5-10)	Accesses source setup page.
<a href="#">SM</a> (on page 5-21)	Accesses measurement setup page.
<a href="#">MD</a> (on page 5-27)	Accesses measurement control page.
<a href="#">US</a> (on page 5-31)	Accesses user-mode page.
<a href="#">UL</a> (on page 4-13)	Allows for use of direct user library module calls. See <a href="#">Calling KULT user libraries remotely</a> (on page 4-6).

## SMU default settings

You can return SMUs to power-on default settings by transmitting the `DCL` (device clear) or `SDC` (selected device clear) general bus command to the 4200A-SCS. The power-on default settings for the user mode are listed in the following table.

The `DCL` command returns all instruments (including the SMUs) that are connected to the bus to their default settings. The `SDC` only affects the SMUs. Note that the device clear commands do not affect the KXCI configuration settings.

Use either of the following C programming language commands (GPIB address 17) to return the SMUs to their power-on default settings:

```
transmit ("UNL LISTEN 17 SDC", &status); // Reset 4200A only.
transmit ("DCL", &status); // Reset all instruments.
```

### NOTE

The `SDC` and `DCL` commands described above are not text-string commands, nor are any of the other commands in this manual that are sent using the `transmit` function. They are low-level commands that must be sent differently than text-string commands. Do not try to use the `transmit` function to output text-string commands across the GPIB; use the `send` function for text-string commands.

#### SMU power-on user mode default settings

Setting	Default
Range	100 $\mu$ A
Compliance	105 $\mu$ A
NRdgs	1
Delay Time	0
Hold Time	0
Wait Time	0
Interval	0
Mode	User mode

## System mode SMU default settings

When KXCI mode is started, the SMUs are in a default state that mimics the Keysight Model 4145B, as shown in the channel definition and source setup tables below. This means that each SMU is active and part of the test, whether or not the test used all the SMUs in the 4200A-SCS chassis.

This may be undesirable, as many tests use a different number of SMUs or SMUs in a different state from the Keysight 4145B default. To avoid the complication of building a KXCI test starting with the default setup, use the following commands to remove the SMUs from the setup:

```
DE CH1 CH2 CH3 CH4 VM1 VM2 VS1 VS2
```

The above shows the method for a system with eight SMUs. A system with only four SMUs would not require the last four definitions:

```
DE CH1 CH2 CH3 CH4
```

### Channel Definition (Page DE) KXCI defaults

Instrument	Name		Source		Command
	V	I	Mode	Function	
SMU1	V1	I1	COM	CONSTant	CH
SMU2	V2	I2	I	VAR2	CH
SMU3*	V3	I3	V	VAR1	CH
SMU4*	V4	I4	V	CONSTant	CH
SMU5*	VS1	N/A	V	CONSTant	VS
SMU6*	VS2	N/A	V	CONSTant	VS
SMU7*	VM1	N/A	N/A	N/A	VM
SMU8*	VM2	N/A	N/A	N/A	VM

\* If there are fewer than eight SMUs, only the SMUs in the chassis are programmed to the defaults shown.

### Source Setup (Page SS) KXCI defaults

	VAR1 (command VR)	VAR2 (command IP)
<b>Name</b>	V3	I2
<b>Sweep mode</b>	Linear	Linear
<b>Start</b>	0.00 V	0.00002 A (20.00 $\mu$ A)
<b>Stop</b>	1.00 V	N/A
<b>Step</b>	0.010 V	0.00002 A (20.00 $\mu$ A)
<b>Number of steps</b>	101	5
<b>Compliance</b>	0.1 A	2 V



Constant	Source	Compliance	Command
V1 COM	0.00 V	0.105 A	N/A
V4 V	0.00 V	0.1 A	VC
VS1 V	0.00 V	N/A	SC
VS2 V	0.00 V	N/A	SC

## Command reference

Commands to control instrument operation are grouped as follows:

- **System mode commands:** This comprehensive set of commands allows you to use all the source-measure capabilities of the SMUs installed in 4200A-SCS.
- **User mode commands:** This limited set of commands allows you to perform basic source-measure operation.
- **Modeless commands:** These commands are valid in any operating mode.

These commands include 4200A only commands, which are valid only while using the 4200A command set instead of the 4145 Emulation command set.

---

### NOTE

Numeric values can be entered in fixed decimal format (for example, 0.1234) or floating decimal format (for example, 123.4e-3). The maximum number of characters for the value is 12. The maximum number of digits for an exponent is 2.

---

## System mode commands

Most system mode commands are divided into groups, known as pages. In order to use these commands, the appropriate page has to be selected. System mode commands are grouped as follows. For the pages, the command to select the page is shown in parentheses.

- Channel definition page (DE)
- Source setup page (SS)
- Measurement setup page (SM)
- Measurement control page (MD)
- Data output and file commands (valid in any system mode page)
- Channel mapping command
- Fixed source ranging command

## Channel definition page (DE)

Use the command strings for the DE page for the following operations:

- SMU channel definition
- VS1...VS $n$  channel definition
- VM1...VM $n$  channel definition

To send the channel definition command strings to the 4200A-SCS, you must select the channel definition page by sending the command:

DE

---

## CH

This command defines a SMU channel.

### Usage

---

CHA  
CHA, 'BBBBBB', 'CCCCC', D, E

A	1, 2, ... or $n$ SMU channel number; the largest value of $n$ is the number of channels in the system (9 maximum); if no parameters are specified after the channel number, the channel is turned off
BBBBBB	Voltage name
CCCCC	Current name
D	Source mode or common: <ul style="list-style-type: none"><li>▪ 1: Voltage source mode</li><li>▪ 2: Current source mode</li><li>▪ 3: Common (output high connected to common)</li></ul>
E	Source function: <ul style="list-style-type: none"><li>▪ 1: VAR1 sweep source function</li><li>▪ 2: VAR2 step source function</li><li>▪ 3: Constant (fixed) source function</li><li>▪ 4: VAR1' source function</li></ul>

### Details

---

For every used channel that is configured as a SMU, you must specify names for voltage and current, select the source mode, and select the source function.

When the source mode ( $D$ ) is set to common (3), the source function ( $E$ ) must be set to constant (3).

The VAR1 source function performs a linear or logarithmic sweep that is synchronized to the steps of VAR2. The VAR1 sweep is performed whenever VAR2 goes to a new step value. The constant source function outputs a fixed (constant) source value.

The VAR1' source function is similar to the VAR1 function, except that each sweep step is scaled by the ratio value (RT) and an offset (FS) as follows:

$$\text{VAR1' sweep step} = (\text{VAR1 sweep step} \times \text{RT}) + \text{FS}$$

For example, assume VAR1 is set to sweep from +1 V to +3 V using 1 V steps. If RT is set to 2, and FS is set to 1, each step of VAR1' is calculated as follows:

$$\text{VAR1' step 1} = (1 \text{ V} \times 2) + 1 = 3 \text{ V}$$

$$\text{VAR1' step 2} = (2 \text{ V} \times 2) + 1 = 5 \text{ V}$$

$$\text{VAR1' step 3} = (3 \text{ V} \times 2) + 1 = 7 \text{ V}$$

If you send CHA with no other parameters, the channel specified by A is disabled.

Example 1

```
DE
CH3, 'V1', 'I1', 1, 3
```

This command string sets up the SMU assigned to channel 3 to source a fixed voltage (1 V). The specified names for voltage and current are V1 and I1, respectively.

Example 2

```
DE
CH1; CH2; CH3; CH4
```

This command string disables channels 1 through 4.

Also see

None

VM

This command defines channels that are used as voltmeters.

Usage

VMA, 'BBBBBB'

A	Voltmeter channel: 1 to 9; the assigned value for a voltmeter depends on how instruments are mapped in KCon
BBBBBB	User-specified name (up to 6 characters)

Details

KXCI allows up to eight source-measure units (SMUs) to function solely as voltmeters. You can use any channel for any voltmeter function between VM1 and VM9. For example, in a system containing four SMUs, you can use SMU3 as VM7.

If you do not define one of the 4200A-SCS SMUs to emulate a VM, attempts to measure voltages through the nonexistent VM result in data values of 9.000e+37.

If nothing is specified after the prefix and channel number, the channel is not used.

To measure voltage using the 4145B  $VM1 \dots VMn$  function, define one of the 4200A-SCS SMUs to emulate  $VM$ .

Example

VM1, 'VM1'

This command string defines the channel used for VM1 (specifies the name  $VM1$ ).

Also see

None

VS

This command specifies the name and selects the source function for each voltage-source channel.

Usage

VSA, 'BBBBBB', C

A	Voltage source (1 to 9); the assigned value depends on how instruments are mapped in KCon
BBBBBB	User-specified name (up to 6 characters)
C	Source function: <ul style="list-style-type: none"><li>1: VAR1</li><li>2: VAR2</li><li>3: Constant</li><li>4: VAR1'</li></ul>

Details

KXCI allows up to eight source-measure units to function solely as voltage sources. You can use any channel for any voltage-source function between VS1 and VS9. For example, in a system containing four SMUs, you can use SMU2 as VS5.

For each voltage source that is used, you must specify a name and select the source function. The VAR1 function performs a voltage sweep that is synchronized to the steps of VAR2. The VAR1 source function performs a linear or logarithmic voltage sweep. The VAR1 sweep is done whenever VAR2 goes to a new step value. The VAR1' function is the same as VAR1 except that each step of the sweep is scaled by a specified ratio (RT) and offset (FS).

The constant source function outputs a fixed (constant) voltage source value. More information on source functions is available throughout this section.

If nothing is specified after the prefix and channel number, the channel is not used.

To source voltage using the 4145B  $VS1 \dots VSn$  function, define one of the 4200A-SCS SMUs to emulate the VS.

Example

```
VS1, 'VS1', 1
```

This command string sets up the channel used by VS1 to perform a voltage sweep.

Also see

None

Source setup page (SS)

The Source Setup (SS) commands set up the source functions of the SMUs, including the different types of sweeping.

Use SS command for the following operations:

- Sweep source setup (CH set to VAR1)
- Step source setup (CH set to VAR2)
- Source function setup (CH set to VAR1 ' )
- Constant bias
- List sweep setup
- Interval or sweep delay time
- Sweep hold time
- RPM switching
- Compliance

To send the following command strings to the 4200A-SCS, you must first select the source setup page by sending the command:

```
SS
```

DT

This command sets the time to wait between when the output voltage is set and when the measurement is made in a sweep.

Usage

DT A.AAA	
A.AAA	Delay time in seconds: 0 to 6.553

Details

Sets the time to wait between when the output voltage is set and when the measurement is made. The total time spent on each step in the sweep is a combination of this user-specified delay time and the time it takes to make a measurement.

You typically use the delay time to allow the source to settle before making a measurement. For example, assume a delay time of 1 s. At each step of the sweep, the source is allowed to settle for 1 s before the measurement is made.

Example

DT 1

This command string sets delay time to 1 s.

Also see

None

FS

This command sets the offset value when VAR1 ' is a selected source function.

Usage

FS ±AAA.A  
FS ±AAA.A, B

AAA.A	Offset value: -210 to +210
B	Available SMU channel (1 to 9); if this parameter is not included, offset applies to all channels that are configured to VAR1 '.

Details

When VAR1 ' is a selected source function, it does the VAR1 sweep with each step scaled by the ratio (RT) value and offset (FS) value as follows:

VAR1 ' sweep step = (VAR1 sweep step × RT) + FS

The VAR1 sweep shown in the example in [VR and IR](#) (on page 5-19) has five steps: 1 V, 2 V, 3 V, 4 V, and 5 V. The corresponding VAR ' sweep has the following steps when RT=3 and FS = 2:

VAR1 ' step 1 = (1 V × 3) + 2 = 5 V

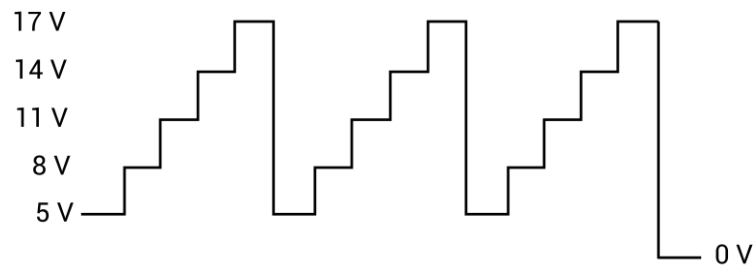
VAR1 ' step 2 = (2 V × 3) + 2 = 8 V

VAR1 ' step 3 = (3 V × 3) + 2 = 11 V

VAR1 ' step 4 = (4 V × 3) + 2 = 14 V

VAR1 ' step 5 = (5 V × 3) + 2 = 17 V

This VAR1 ' sweep (not drawn to scale) is shown in the following figure.

**Figure 10: VAR1 sweep when ratio = 3 and full scale = 2**

You can assign a unique offset value to any available SMU channel in the system.

### Example

```
RT +3,2
FS +2,2
```

These command strings set up the VAR1 ' sweep shown in the figure in the Details (ratio = 3, offset = 2). The above commands set up VAR1 ' for SMU channel 2. When channel 2 is defined for a VAR1 ' sweep, ratio is set to 3 and offset is set to 2.

### Also see

None

## HT

This command sets a hold time that delays the start of a sweep.

### Usage

```
HT AAA.A
```

AAA.A	Hold time in seconds: 0 to 655.3
-------	----------------------------------

### Details

You can delay the start of a sweep by setting a hold time. When the sweep is triggered, it starts after the hold time expires.

For example, you can use the hold time to allow for the device to charge up and settle to the first bias point in the sweep before starting the sweep.

### Example

```
HT 1
```

This command string sets the hold time to one second.

### Also see

None

# RT

This command sets the ratio value when VAR1 ' is a selected source function.

## Usage

```
RT ±AA.A
RT ±AA.A, B
```

AAA.A	Ratio value: -10 to +10
B	Available SMU channel (1 to 9); if this parameter is not included, ratio applies to all channels that are configured to VAR1 '

## Details

When VAR1 ' is a selected source function, it does the VAR1 sweep with each step scaled by the ratio (RT) value and offset (FS) value as follows:

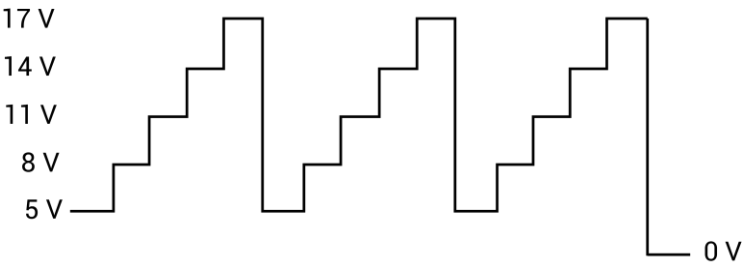
$$\text{VAR1 ' sweep step} = (\text{VAR1 sweep step} \times \text{RT}) + \text{FS}$$

The VAR1 sweep shown in the example in [VR and IR](#) (on page 5-19) has five steps: 1 V, 2 V, 3 V, 4 V, and 5 V. The corresponding VAR ' sweep has the following steps when RT=3 and FS = 2:

$$\begin{aligned} \text{VAR1 ' step 1} &= (1 \text{ V} \times 3) + 2 = 5 \text{ V} \\ \text{VAR1 ' step 2} &= (2 \text{ V} \times 3) + 2 = 8 \text{ V} \\ \text{VAR1 ' step 3} &= (3 \text{ V} \times 3) + 2 = 11 \text{ V} \\ \text{VAR1 ' step 4} &= (4 \text{ V} \times 3) + 2 = 14 \text{ V} \\ \text{VAR1 ' step 5} &= (5 \text{ V} \times 3) + 2 = 17 \text{ V} \end{aligned}$$

This VAR1 ' sweep (not drawn to scale) is shown in the following figure.

**Figure 11: VAR' sweep when ratio = 3 and full scale = 2**



You can assign a unique ratio value to any available SMU channel in the system.



**Example**

RT +3,2	Ratio
FS +2,2	Offset

These command strings set up the VAR1 ' sweep shown in the figure in the **Details** (ratio = 3, offset = 2). The above commands set up VAR1 ' for SMU Channel 2. When Channel 2 is defined for a VAR1 ' sweep, ratio is set to 3 and offset is set to 2.

**Also see**

None

## SC

This command configures the source to output a fixed voltage level for channels that are configured to be voltage source only.

**Usage**

SCA, ±BBB.BBBB

A	Voltage source channel: 1 to 9; see <b>Details</b>
BBB.BBBB	Output voltage value: -210.00 to +210.00

**Details**

This command is for use with any channel configured solely as a VS1 . . . VS $n$  voltage source. It configures the source to output a fixed (constant) voltage level.

If nothing is specified after the channel number, the channel is turned off.

The range of possible values for A depends on how instruments are mapped in KCon. The parameter A represents the  $n$  in voltage source VS $n$ .

**Example**

SC1, 20

This command string sets up VS1 to output a constant 20 V level.

**Also see**

"Keithley Configuration Utility (KCon)" in *Model 4200A-SCS Setup and Maintenance*

## ST

This command enables or disables automatic standby.

**Usage**

ST A, B

A	SMU channel number (1 to 9)
B	Enable or disable automatic standby: <ul style="list-style-type: none"> <li>■ 0: Disable automatic standby</li> <li>■ 1: Enable automatic standby</li> </ul>

Details

When automatic standby is enabled, the SMU automatically goes into standby when the test completes. When disabled, the output stays on when the test completes.

Also see

None

VC and IC

These commands configure the SMU to output a fixed (constant) voltage or current level.

Usage

*AAB, ±CCC.CCCC, ±DDD.DDDD*

<i>AA</i>	Source mode: <ul style="list-style-type: none"><li>■ Voltage: <i>VC</i></li><li>■ Current: <i>IC</i></li></ul>
<i>B</i>	SMU channel number: 1 to 9
<i>CCC.CCCC</i>	Output value: <ul style="list-style-type: none"><li>■ Voltage source: -210.00 to +210.00</li><li>■ Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li><li>■ Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li></ul>
<i>DDD.DDDD</i>	Compliance value: <ul style="list-style-type: none"><li>■ Voltage source: -210.00 to +210.00</li><li>■ Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li><li>■ Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li></ul>

Details

For any channel configured as a SMU, this command configures the SMU to output a fixed (constant) voltage or current level.

With the voltage source mode (*VC*) specified in the command string, you set the current compliance. For the current source mode (*IC*), you set the voltage compliance.

Also see

None

# VL and IL

These commands set up a list sweep.

## Usage

*AAB, C, ±DDD.DDDD, ±EE.EEEE, ... ±EE.EEEE*

<i>AA</i>	The source mode: <ul style="list-style-type: none"><li>■ Voltage source (SMU or VS1...VS9): <i>VL</i></li><li>■ Current source (SMU only): <i>IL</i></li></ul>
<i>B</i>	Channel number: 1 to 9
<i>C</i>	Master or subordinate mode: <ul style="list-style-type: none"><li>■ Subordinate mode: 0</li><li>■ Master mode: 1</li></ul>
<i>DDD.DDDD</i>	Compliance value: <ul style="list-style-type: none"><li>■ Voltage source: -210.00 to +210.00</li><li>■ Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li><li>■ Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li></ul>
<i>EE.EEEE</i>	List values: <ul style="list-style-type: none"><li>■ Voltage source: -210.00 to +210.00</li><li>■ Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li><li>■ Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li></ul>

## Details

A list sweep occurs in the order of the defined sweep parameters. If the source is a SMU, the source mode for the sweep can be voltage or current. If a voltage source (VS1 . . . Sn) is used, the source mode must be voltage.

With the voltage source mode (*VL*) specified in the command string, you are setting the current compliance. For the current source mode (*IL*), you are setting the voltage compliance. If you are sourcing voltage, if you specify a compliance current that is below the minimum allowable value (100 pA with a preamplifier installed and 100 nA without a preamplifier), KXCI sets it to the minimum allowable value.

The maximum number of points for the list is 4096. Sweep points must be delimited by commas.

## Example

*VL1,1, 0.01, 1, 5, 2*

This command string sets up a channel 1 list sweep (1 V, 5 V, 2 V arbitrary steps, and compliance = 10 mA).

## Also see

None

## VP and IP

These commands set up the VAR2 step sweep.

### Usage

*AA ±BBB.BBBB, ±CCC.CCCC, DD, ±EEE.EEEE*

*AA ±BBB.BBBB, ±CCC.CCCC, DD, ±EEE.EEEE, FF*

<i>AA</i>	The source mode: <ul style="list-style-type: none"> <li>VP: Voltage source (SMU or VS1 . . . VS8)</li> <li>IP: Current source (SMU only)</li> </ul>
<i>BBB.BBBB</i>	Start value: <ul style="list-style-type: none"> <li>Voltage source: -210.00 to +210.00</li> <li>Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li> <li>Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li> </ul>
<i>CCC.CCCC</i>	Step value: <ul style="list-style-type: none"> <li>Voltage source: -210.00 to +210.00</li> <li>Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li> <li>Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li> </ul>
<i>DD</i>	Number of steps: 1 to 32
<i>EEE.EEEE</i>	Compliance value (see <b>Details</b> ): <ul style="list-style-type: none"> <li>Voltage source: -210.00 to +210.00</li> <li>Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li> <li>Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li> </ul>
<i>FF</i>	VAR2 source stepper index: 1 to 4 (see <b>Details</b> )

### Details

With the voltage source mode (VP), the output value is in volts. For the current source mode (IP), the output value is in amperes.

If you specify a voltage start or step value below 0.001 V, KXCI automatically sets the value to zero.

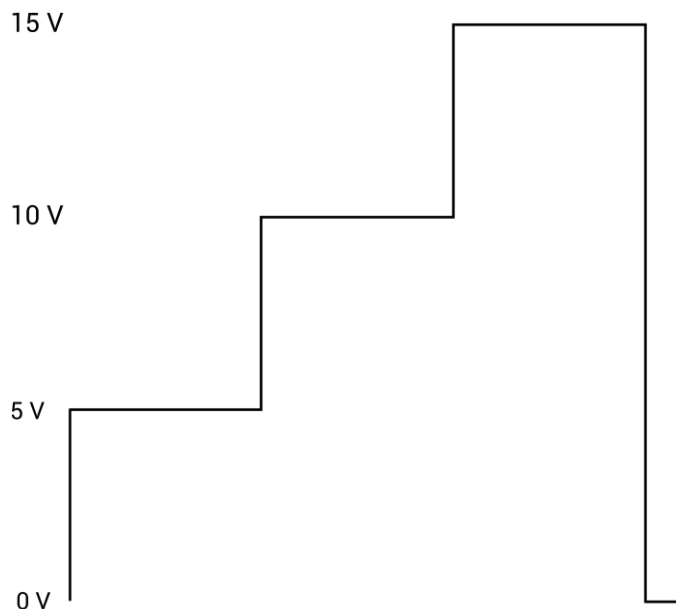
With the voltage source mode (VP), you are setting the current compliance. For the current source mode (IP), you are setting the voltage compliance. When sourcing voltage (VP), if you specify a compliance current that is below the minimum allowable value (100 pA with a preamplifier installed and 100 nA without a preamplifier), KXCI sets it to the minimum allowable value.

If you are using a SMU as the source, the source mode for the VAR2 steps can be voltage or current. If, however, a voltage source (VS1 . . . VS9) is used, the source mode must be voltage. For configuration details, refer to “Keithley Configuration Utility (KCon)” in *Model 4200A-SCS Setup and Maintenance*.

If the source stepper index (*FF*) is omitted, the default is 1 (first stepper). Use the *CH* command to define one or more VAR2 sources. The first defined VAR2 is index = 1. The second channel is defined as VAR2 is index = 2, and so on up to a maximum of four VAR2 sources. Note that this is an extension to the traditional VAR2 capability.

Parameters for the VAR2 step function include the start value, step value, and the number of steps. In the following figure, the VAR2 step function starts at 5 V, steps in 5 V increments and has three steps.

**Figure 12: Steps resulting from the VP 5, 5, 3, 0.01 command string**



---

### Example

```
VP 5, 5, 3, 0.01
```

This command string sets up a VAR2 voltage sweep with start = 5 V, step = 5 V, number of steps = 3, and compliance = 10 mA. This command string performs the steps shown in the figure in the **Details**. These are the same steps that are shown synchronized with sweeps in [VR and IR: VAR1 setup](#) (on page 5-19).

---

### Also see

[CH: SMU channel definition](#) (on page 5-7)

## VR and IR

These commands set up the VAR1 source function.

### Usage

*AAB, ±CCC.CCCC, ±DDD.DDDD, ±EEE.EEEE, ±FFF.FFFF*

<i>AA</i>	The source mode: <ul style="list-style-type: none"> <li>VR: Voltage source (SMU or VS1 . . . VS9)</li> <li>IR: Current source (SMU only)</li> </ul>
<i>B</i>	The sweep type: <ul style="list-style-type: none"> <li>1: Linear sweep</li> <li>2: Log10 sweep</li> <li>3: Log25 sweep</li> <li>4: Log50 sweep</li> </ul>
<i>CCC.CCCC</i>	Start value: <ul style="list-style-type: none"> <li>Voltage source: -210.00 to +210.00</li> <li>Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li> <li>Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li> </ul>
<i>DDD.DDDD</i>	Stop value: <ul style="list-style-type: none"> <li>Voltage source: -210.00 to +210.00</li> <li>Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li> <li>Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li> </ul>
<i>EEE.EEEE</i>	Step value (linear sweep only): <ul style="list-style-type: none"> <li>Voltage source: -210.00 to +210.00</li> <li>Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li> <li>Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li> </ul> For a log sweep, do not set a step value (steps are determined by the setting for <i>B</i> )
<i>FFF.FFFF</i>	Compliance value (also see <b>Details</b> ): <ul style="list-style-type: none"> <li>Voltage source: -210.00 to +210.00</li> <li>Current source, 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li> <li>Current source, 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li> </ul> For a log sweep, do not set a compliance value

### Details

If you specify a voltage start or step value below 0.001 V, KXCI automatically sets the value to zero.

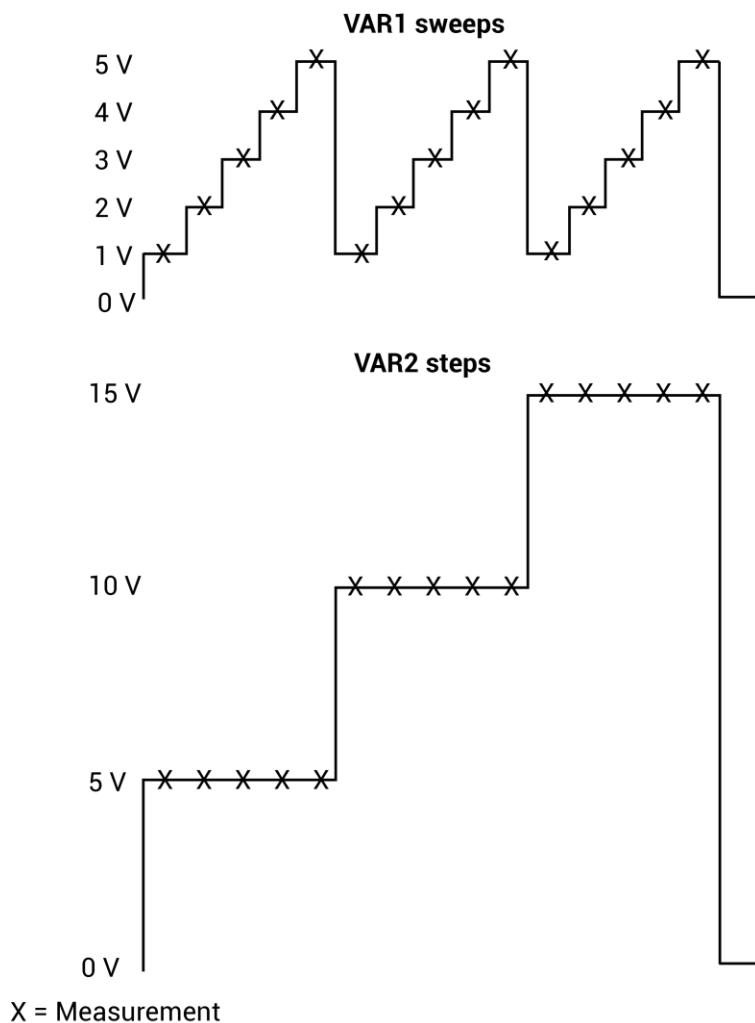
When setting the step value, be aware that the maximum number of points for VAR1 is 1024:

$$\text{Number of points} = (\text{int})(\text{Abs}((\text{Stop Value} - \text{Start Value}) / \text{Step Value}) + 1.5)$$

When you are setting a voltage source mode ( $\text{VR}$ ), you are setting a current compliance value. If you are setting a current source mode ( $\text{IR}$ ), you are setting voltage compliance. When sourcing voltage, note that if you specify a compliance current that is below the minimum allowable value (100 pA with a preamplifier installed and 100 nA without a preamplifier), KXCI sets it to the minimum allowable value.

When VAR1 is a selected source function, it does a sweep that is synchronized to the steps of the VAR2 step function. The VAR1 sweep is repeated whenever VAR2 goes to a new step value, as shown in the following figure.

**Figure 13: Synchronized VAR1 sweeps and VAR2 steps**



If the source is a SMU, the source mode for the sweep can be voltage or current. If, however, a voltage source ( $\text{VS1} \dots \text{Sn}$ ) is used, the source mode must be voltage.

You can do the sweep on a linear or logarithmic scale. With the linear sweep mode selected, the start, stop, and step value parameters define the sweep. Each VAR1 sweep in the figure above sweeps from 1 V (start) to 5 V (stop) in 1 V steps.

With a logarithmic sweep mode selected (log base 10, 25, or 50), only specify the start and stop values. Step size is automatically set to provide a symmetrical sweep on the logarithmic scale.

## NOTE

The time spent on each sweep step depends on the user-set delay time and the time it takes to perform the measurement.

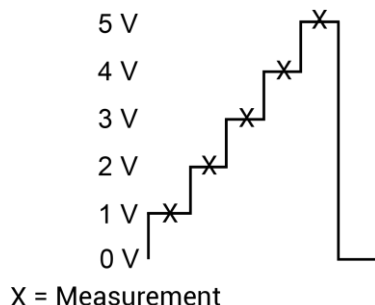
You can delay the start of the sweep by setting a hold time.

### Example

```
VR1, 1, 5, 1, 0.01
```

This command string sets up a VAR1 linear sweep with a start = 1 V, stop = 5 V, step = 1 V, and compliance = 10 mA.

The following figure shows the sweep that results from this setup. The figure in the **Details** shows the results of the same setup when used with a VAR2 step command.



### Also see

None

## Measurement setup page (SM)

Use the command strings for the SM page for the following operations:

- Set wait time
- Set interval
- Select number of readings
- Select list display mode
- Configure the X and Y data

To send the following command strings to the 4200A-SCS, select the **SM** page by sending the command:

```
SM
```



---

# DM

This command selects the Keysight 4145B display mode.

## Usage

---

DMA

A

- Display modes:
- Graphics display mode: 1
  - List display mode: 2

## Details

---

The 4200A-SCS supports the Keysight 4145B graphics display mode and the Keysight 4145B list display mode command.

The 4200A-SCS does not accept the matrix mode and SMU mode commands (DM3 and DM4).

## Example

---

DM1

This command string prepares the 4200A-SCS to receive graphics commands.

## Also see

---

None

---

# IN

This command sets the time between sample measurements.

## Usage

---

IN AA.AA

AA.AA

Interval in seconds: 0.01 to 10

## Details

---

For time domain measurements, you can set the time between sample measurements. After a sample measurement is made, the next measurement starts after the time interval expires.

For constant bias measurements, this is the time between readings.

## Example

---

IN 0.1

This command string sets the interval to 100 ms.

## Also see

---

None

# LI

This command enables voltage and current functions to be measured when the 4200A-SCS is in list display mode.

## Usage

```
LI 'AAAAAA'
LI 'AAAAAA', 'AAAAAA'
LI 'AAAAAA', 'AAAAAA', 'AAAAAA'
LI 'AAAAAA', 'AAAAAA', 'AAAAAA', 'AAAAAA'
LI 'AAAAAA', 'AAAAAA', 'AAAAAA', 'AAAAAA', 'AAAAAA'
LI 'AAAAAA', 'AAAAAA', 'AAAAAA', 'AAAAAA', 'AAAAAA', 'AAAAAA'
```

AAAAAA	A name assigned for CH, VS, or VM; see channel definition page (DE)
--------	---

## Details

When the 4200A-SCS is in the list display mode (DM2 asserted), the LI command enables functions to be measured in a test sequence. To enable a function, include the SMU channel name (as assigned by the CH command), voltage source name (as assigned by the VS command), or the voltmeter name (as assigned by the VM command) in the command string. The DE page is used to assign names to voltage and current functions.

Only functions that are specified (enabled) are measured. Data sheet columns for disabled functions are not shown.

## Example

L1 'V1', 'I1', 'VS1'

Assume the following names have been assigned using the DE page for SMU channel 1 (CH1):

- Voltage is named V1
- Current is named I1
- Voltage source is named VS1

The command string enables the above functions for a test sequence.

## Also see

[Channel definition page \(DE\)](#) (on page 5-7)

[DM](#) (on page 5-22)

# NR

This command sets the number of readings that can be made for time domain measurements.

## Usage

```
NR AAAA
```

AAAA	Number of measurements to make: <ul style="list-style-type: none"><li>■ 4200A command set: 1 to 4096</li><li>■ 4145 Emulation command set: 1 to 1024</li></ul>
------	--

---

**Details**

For time domain measurements, you can set the number of sample measurements that can be made. The readings are stored in the buffer.

---

**Example**

```
NR 200
```

This command string sets up the 4200A-SCS to make 200 sample measurements.

---

**Also see**

None

---

## WT

This command delays the start of a test sequence for time domain measurements.

---

**Usage**

```
WT AAA.AAA
```

```
AAA.AAA
```

Wait time in seconds: 0 to 100

---

**Details**

For time domain measurements, you can delay the start of a test sequence by setting a wait time. The test sequence starts after the wait time period expires.

For the sampling mode or constant bias, this command is a delay after the source output level is set and before the readings begin.

---

**Example**

```
WT 0.1
```

This command string sets the wait time to 100 ms.

---

**Also see**

None

# XN

This command configures the X-axis of the graph to plot an electrical parameter.

## Usage

```
XN 'AAAAAA', B, ±CCCC.CCC, ±DDDD.DDD
```

AAAAAA	The SMU channel name for the X-axis; up to 6 characters long; must be one of the SMU channel names that you specify on the channel definition (DE) page
B	X-axis scale type <ul style="list-style-type: none"><li>Linear scale: 1</li><li>Log scale: 2</li></ul>
CCCC.CCC	X-axis minimum value <ul style="list-style-type: none"><li>Volts: ±9999</li><li>Amperes: ±999</li></ul>
DDDD.DDD	X-axis maximum value <ul style="list-style-type: none"><li>Volts: ±9999</li><li>Amperes: ±999</li></ul>

## Example

```
XN 'V1', 1, -5, 5
```

This command string:

- Specifies that values from SMU channel V1 are to be plotted on the X-axis.
- Sets up the X-axis to be scaled linearly between -5 V and +5 V.

## Also see

[Channel definition page \(DE\)](#) (on page 5-7)

# XT

This command configures the X-axis of the graph to plot time domain values in seconds.

## Usage

```
XT AAAA.AA, BBBB.BB
```

AAAA.AA	X-axis minimum time value (in seconds): 0.01 to 9999
BBBB.BB	X-axis maximum time value (in seconds): 0.01 to 9999

## Example

```
XT 0, 10
```

This command string:

- Specifies that time domain values are to be plotted on the X-axis.
- Sets up the X-axis to be scaled between 0 s and 10 s.

## Also see

None

# YA

This command configures the Y1-axis of the graph.

## Usage

`YA 'AAAAAA', B, ±CCCC.CCC, ±DDDD.DDD`

AAAAAA	The SMU channel name for the Y1-axis, up to 6 characters; must be one of the SMU channel names that you specify on the channel definition (DE) page
B	Y1-axis scale type: <ul style="list-style-type: none"><li>Linear scale: 1</li><li>Log scale: 2</li><li>Log scale absolute value: 3</li></ul>
CCCC.CCC	Y1-axis minimum value: <ul style="list-style-type: none"><li>Volts: ±9999</li><li>Amps: ±999</li></ul>
DDDD.DDD	Y1-axis maximum value: <ul style="list-style-type: none"><li>Volts: ±9999</li><li>Amps: ±999</li></ul>

## Example

`YA 'I1', 1, -5E-9, 5E-9`

This command string:

- Specifies that values from SMU channel I1 are to be plotted on the Y1-axis.
- Sets up the Y1-axis to be scaled linearly between -5 nA and +5 nA.

## Also see

[Channel definition page \(DE\)](#) (on page 5-7)

# YB

This command configures the Y2-axis of the graph.

## Usage

```
YB 'AAAAAA', B, ±CCCC.CCC, ±DDDD.DDD
```

AAAAAA	The SMU channel name for the Y2-axis, up to 6 characters; must be one of the SMU channel names that you specify on the channel definition (DE) page
B	Y2 axis scale type: <ul style="list-style-type: none"><li>Linear scale: 1</li><li>Log scale: 2</li><li>Log scale absolute value: 3</li></ul>
CCCC.CCC	Y2-axis minimum value <ul style="list-style-type: none"><li>Volts: ±9999</li><li>Amperes: ±999</li></ul>
DDDD.DDD	Y2-axis maximum value <ul style="list-style-type: none"><li>Volts: ±9999</li><li>Amperes: ±999</li></ul>

## Example

YB 'I2', 2, 100E-9, 1E-3

The command string:

- Specifies that values from SMU channel I2 are to be plotted on the Y2-axis.
- Sets up the Y2-axis to be scaled logarithmically between 100 nA and 1 mA.

## Also see

[Channel definition page \(DE\)](#) (on page 5-7)

## Measurement control page (MD)

The measurement control page contains the command to trigger a test.

To send the command string to control measurements, select the measurement control page by sending the command:

```
MD
```

# ME

This command controls measurements.

## Usage

ME A

A	Measurement action: <ul style="list-style-type: none"><li>Run a single trigger test and store readings in a cleared buffer: 1</li><li>Run a repeat trigger test and store readings in a cleared buffer: 2</li><li>Run a single trigger test and append readings to the buffer: 3</li><li>Abort the test: 4</li></ul>
---	--

## Details

The ME1 or ME2 command triggers the start of the test and makes the programmed number of measurements. The measured readings are stored in the buffer. Note that the buffer is cleared before readings are stored.

The ME3 command also triggers the test but does not clear the buffer before storing the measured readings. The readings are appended to the readings already stored in the buffer. The buffer can hold up to 4096 readings.

The ME4 command aborts the test.

## Example

ME 1
This command string triggers the start of the test and stores the readings in the cleared buffer.

## Also see

None

## Data output and file commands

- The command strings for the following operations are valid in any system mode page:
- Obtain output data
  - Save file
  - Get file

# GT

This command acquires (loads) the saved data file or program file.

## Usage

```
GT 'A BBBBBB'
```

A	File type: <ul style="list-style-type: none"><li>■ Program file: P</li><li>■ Data/program file: D</li></ul>
BBBBBB	Name of file (up to 6 characters)

## Details

The file type and file name must be separated by a space.

For a program file, this command launches the program. For a data file, it opens the files. When the saved program file is recalled, the instrument returns to the settings stored in that file.

## NOTE

The save (SV) command string is used to save instrument settings or to store data acquired in a test.

## Example

```
GT 'P Setup1'
```

This command string gets the program file named Setup1.

## Also see

[SV command: Save file](#) (on page 5-29)

# SV

This command saves a program file or data file.

## Usage

```
SV 'A BBBBBB'
```

```
SV 'A BBBBBB CCCCCCCC'
```

A	File type: <ul style="list-style-type: none"><li>■ Program file: P</li><li>■ Data file: D</li></ul>
BBBBBB	Name of file (up to 6 characters)
CCCCCCCC	Comment (up to 8 characters)



Details

You must separate the file name and the comment from each other by a space.

When saving a program file, the present instrument settings are stored in a file at the directory path:

```
C:\s4200\sys\KXCI
```

You specify the name for the file.

NOTE

The `get` command string is used to acquire the saved file.

Example

```
SV 'P Setup1'
```

This command string saves the command sequence as a program file named `Setup1`.

Also see

[GT command: Get file](#) (on page 5-29)

Channel mapping command

You can use the following command with any system mode page. It maps channel *n* to a given *VS*, *SMU*, or *VM* function.

MP

This command maps channel *n* to a given *VS*, *SMU*, or *VM* function.

Usage

MP *A*, *BBBC*

<i>A</i>	The channel to be mapped: A value between 1 and the number of channels in the system (9 maximum)
<i>BBB</i>	SMU, VS, or VM
<i>C</i>	The number of the SMU, VS, or VM

Details

If *BBB* and *C* are not included in the command, the function defaults to `SMU<A>`, where `<A>` is the number of the channel to be mapped.

Example

```
MP 3, VM5
```

This command string maps channel 3 to `VM5`.

Also see

None

## Fixed source ranging command

You can use the following command with any system mode page.

### SR

This command sets a fixed source range on channel *n*.

#### Usage

SR <i>A</i> , <i>B</i>	
<i>A</i>	The channel to be controlled: A value between 1 and the number of channels in the system (9 maximum)
<i>B</i>	Range: <ul style="list-style-type: none"><li>▪ Auto: 0</li><li>▪ Best fixed range (determined by maximum sweep parameters): 2</li><li>▪ Fixed range: &gt; 0 to 1.0</li></ul>

#### Details

The default setting is autorange for backward compatibility. If you specify a range that is below the bias or sweep parameters that follow, the range is adjusted to accommodate the sweep.

#### Example

SR 1, 2
This command string selects best fixed range on channel 1.

#### Also see

None

## User mode commands (US)

You can use the user mode (US) command strings for the following operations:

- SMU setup
- VS1...VS9 setup
- Triggering

To send these command strings to the 4200A-SCS, select the user mode by sending the command:

US
----

# DI

This command sets up a SMU channel as a current source.

## Usage

```
DIB
DIB, CC, ±DDD.DDDD, ±EEE.EEEE
```

<i>B</i>	SMU channel: 1 to 9
<i>CC</i>	Current source range: <ul style="list-style-type: none"><li>■ Autorange: 0</li><li>■ 1 nA range (only with a preamplifier): 1</li><li>■ 10 nA range (only with a preamplifier): 2</li><li>■ 100 nA range: 3</li><li>■ 1 µA range: 4</li><li>■ 10 µA range: 5</li><li>■ 100 µA range: 6</li><li>■ 1 mA range: 7</li><li>■ 10 mA range: 8</li><li>■ 100 mA range: 9</li><li>■ 1 A range (only with a 4210-SMU or 4211-SMU): 10</li><li>■ 1 pA range (only with a preamplifier): 11</li><li>■ 10 pA range (only with a preamplifier): 12</li><li>■ 100 pA range (only with a preamplifier): 13</li></ul>
<i>DDD.DDDD</i>	Output value (amperes): <ul style="list-style-type: none"><li>■ 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li><li>■ 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li></ul>
<i>EEE.EEEE</i>	Compliance value (voltage): -210.00 to 210.00

## Details

For every channel that is configured as a SMU, you must select the source mode (voltage or current), source output range, output value, and compliance value.

### NOTE

To select the voltage source mode, see [DV](#) (on page 5-33).

If you send `DIB` with no other parameters, the output of the specified channel is disabled.

## Example

```
DI1, 1, 10, 1, 20
```

This command string configures SMU1 to source 1 nA on the 1 A source range and sets voltage compliance to 20 V.

## Also see

[DV](#) (on page 5-33)

# DS

This command specifies the channel number and the voltage output value for each voltage source.

## Usage

*DSA, ±BBB.BBBB*

<i>A</i>	<i>n</i> , for voltage source <i>VS<sub>n</sub></i> ; range 1 to 9
<i>BBB.BBBB</i>	Output voltage value: -200.00 to +200.00

## Details

KXCI allows up to nine source-measure units to function solely as voltage sources. You can use any channel for any voltage-source function between *VS1* and *VS9*. For example, in a system containing four SMUs, you can use SMU2 as *VS5*.

The assigned *n* value for a voltage source (*VS<sub>n</sub>*) or voltmeter (*VM<sub>n</sub>*) depends on how instruments are mapped in KCon.

## Example

*DS1, 20*

This command string sets *VS1* to output 20 V.

## Also see

None

# DV

This command sets up a SMU channel as a voltage source.

## Usage

*DVB*

*DVB, CC, ±DDD.DDDD, ±EEE.EEEE*

<i>B</i>	SMU channel: 1 to 9
<i>CC</i>	Voltage source range: <ul style="list-style-type: none"><li>■ Autorange: 0</li><li>■ 20 V range: 1</li><li>■ 200 V range: 2 or 3</li><li>■ 200 mV range: 4 (only with a preamplifier)</li><li>■ 2 V range: 5 (only with a preamplifier)</li></ul>
<i>DDD.DDDD</i>	Output value (voltage): -210.00 to +210.00
<i>EEE.EEEE</i>	Current compliance value (amperes): <ul style="list-style-type: none"><li>■ 4200-SMU or 4201-SMU: -0.1050 to +0.1050</li><li>■ 4210-SMU or 4211-SMU: -1.0500 to +1.0500</li></ul>

Details

For every channel that is configured as a SMU, you must select the source mode (voltage or current), source output range, output value, and compliance value.

NOTE

To select the current source mode, see [DI](#) (on page 5-32).

If you specify a compliance current that is below the minimum allowable value, KXCI sets it to the minimum allowable value.

If you send `DVB` with no other parameters, the output of the specified channel is disabled.

Example

```
DV1, 1, 10, 10E-3
```

This command string configures SMU1 to source 10 V on the 20 V source range and sets current compliance to 10 mA.

Also see

[DI](#) (on page 5-32)

TI

This command triggers a current measurement.

Usage

TI ABB	
BB	Measure channel for current measurements: <ul style="list-style-type: none"><li>SMU1: 1</li><li>SMU2: 2</li><li>SMU3: 3</li><li>SMU4: 4</li><li>SMU5: 5</li><li>SMU6: 6</li><li>SMU7: 7</li><li>SMU8: 8</li></ul>

Details

After sending the command string to trigger a measurement and addressing the 4200A-SCS to talk, the output data string is sent to the computer in the following format:

*X Y Z ±N.NNNN E±NN*

Where:

- *X*: The status of the data (where *X* = *N* for a normal reading)
- *Y*: The measure channel (*Y* = *A* through *F*)
- *Z*: The measure mode (*Z* = *V* or *I*)
- *±N.NNNN E±NN* is the reading (mantissa and exponent)
- When channels are mapped to different functions (*VM* or *VS*), KXCI tries to trigger measurements on the specified channels. However, if the mapped function for a channel does not match the requested measurement, KXCI reports an error. For example, if the mapped function for a channel is *VS2*, but the requested measurement is *TI2*, KXCI reports an error, because a *VS* cannot measure current.

Also see

- [TV](#) (on page 5-35)
- [VM](#) (on page 5-8)
- [VS](#) (on page 5-9)

TV

This command triggers a voltage measurement.

Usage

*TV BB*

<i>BB</i>	Measure channel for voltage measurements:
	<div><div>■ SMU1: 1</div><div>■ SMU2: 2</div><div>■ SMU3: 3</div><div>■ SMU4: 4</div><div>■ VM1: 5</div><div>■ VM2: 6</div><div>■ SMU5: 7</div><div>■ SMU6: 8</div></div> <div><div>■ SMU7: 9</div><div>■ SMU8: 10</div><div>■ VM3: 11</div><div>■ VM4: 12</div><div>■ VM5: 13</div><div>■ VM6: 14</div><div>■ VM7: 15</div><div>■ VM8: 16</div></div>

Details

You can use a SMU to measure voltage either directly (as mapped *SMU1* to *SMU9*) or as a mapped voltmeter (*VM1* to *VM9*), so the SMU is specified in the trigger command string by a unique identifier. For example, a physical SMU that has been mapped as *SMU5* (using *KCon*) is specified by the unique identifier *7*.

After sending the command string to trigger a measurement and addressing the 4200A-SCS to talk, the output data string is sent to the computer in the following format:

$X Y Z \pm N.NNNN E\pm NN$

Where:

- $X$ : The status of the data (where  $X = N$  for a normal reading)
- $Y$ : The measure channel ( $Y = A$  through  $F$ )
- $Z$ : The measure mode ( $Z = V$  or  $I$ )
- $\pm N.NNNN E\pm NN$  is the reading (mantissa and exponent)

When channels are mapped to different functions ( $VM$  or  $VS$ ), KXCI tries to trigger measurements on the specified channels. However, if the mapped function for a channel does not match the requested measurement, KXCI reports an error. For example, if the mapped function for a channel is  $VS2$ , but the requested measurement is  $TI2$ , KXCI reports an error, because a  $VS$  cannot measure current.

#### Also see

[TI](#) (on page 5-34)

[VM](#) (on page 5-8)

[VS](#) (on page 5-9)

## Modeless commands

The command strings listed in the following table are valid in both the system and user operating modes. Detailed descriptions are provided following the table.

Command	Brief description
<a href="#">AC</a> (on page 5-37)	Autocalibrates a SMU channel.
<a href="#">BC</a> (on page 5-37)	Clears all readings from the buffer.
<a href="#">DO</a> (on page 5-38)	Returns the timestamp data that was acquired with the voltage or current measurement, or both.
<a href="#">EC</a> (on page 5-39)	Sets the condition to exit the test if compliance is reached.
<a href="#">EM</a> (on page 5-39)	Switches between 4145 Emulation and 4200A command sets.
<a href="#">IT</a> (on page 5-40)	Sets the integration time of a SMU.
<a href="#">RD</a> (on page 5-41)	Requests real-time readings.
<a href="#">RG</a> (on page 5-42)	Sets the lowest current range of the SMU to be used when measuring.
<a href="#">RI</a> (on page 5-42)	Instructs a SMU to go to a specified current range immediately without waiting until the initiation of a test.
<a href="#">RS</a> (on page 5-43)	Sets the measurement resolution for all channels of a SMU.
<a href="#">RV</a> (on page 5-43)	Instructs a SMU to go to a specified voltage range immediately without waiting until the initiation of a test.

---

# AC

This command autocalibrates a SMU channel.

**Usage**

---

AC A

A	SMU number (1 to 9)
---	---------------------

**Details**

---

SMU only.

When this command is sent, the selected SMU channel is calibrated automatically. The busy bit in the status register is set so that you can detect when autocalibration is finished. The 4200A-SCS will not respond to any commands while autocalibration is executing.

**Example**

---

AC 1
This command string performs autocalibration on SMU channel number 1.

**Also see**

---

None

---

# BC

This command clears all readings from the buffer.

**Usage**

---

BC

**Details**

---

SMU only.

The GPIB data buffer can hold up to 4096 readings.

This command string clears all readings from the buffer. It also clears bit B0 (Data Ready) of the status byte.

**Also see**

---

None



# DO

This command returns the timestamp data that was acquired with the voltage or current measurement, or both.

## Usage

```
DO 'AAAAAA'  
DO 'AAAAAAB'
```

AAAAAA	User-specified name of the channel that made the measurement, up to 6 characters; cannot be changed through the CH or VS commands
B	T: Return timestamp data S: Return the SMU status

## Details

SMU only.

After making measurements, use this command string to request the readings. After the 4200A-SCS is addressed to talk, the readings are sent.

Output data is sent to the computer in the format  $\pm N.NNNN\ E\pm NN$ .

Output data of 0 indicates this point is not measured yet.

NOTE

When you run a voltage sweep without measure commands, you can use DO to retrieve the voltage measurement. However, you cannot use DO to retrieve the current measurement from a voltage sweep. To retrieve the current measurement, use LI, XN, YA, YB, or DV.

## Example

```
DO 'CH1'
```

This command string requests the reading string for SMU channel 1.

## Also see

- [DV and DI](#) (on page 5-32)
- [LI](#) (on page 5-23)
- [XN](#) (on page 5-25)
- [YA](#) (on page 5-26)
- [YB](#) (on page 5-27)

---

# EC

This command sets the condition to exit the test if compliance is reached.

## Usage

---

EC *A*

<i>A</i>	Action on compliance: <ul style="list-style-type: none"><li>Off (do not exit if compliance is reached): 0</li><li>On (exit if compliance is reached): 1</li></ul>
----------	---

## Details

---

SMU only.

## Example

---

EC 1
This command enables exit on compliance.

## Also see

---

None

---

# EM

This command switches between 4145 Emulation and 4200A command sets.

## Usage

---

EM *A, B*

<i>A</i>	Mode: <ul style="list-style-type: none"><li>4145 Emulation: 0</li><li>4200A: 1</li></ul>
<i>B</i>	Which sessions: <ul style="list-style-type: none"><li>This session only: 0</li><li>This and all subsequent sessions (writes to KCon file): 1</li></ul>

## Details

---

SMU only.

## Example

---

EM 0,1
This selects the 4145 Emulation command set permanently.

## Also see

---

None

# IT

This command sets the integration time of a SMU.

## Usage

```
ITA
IT4, X, Y, Z
```

A	Integration time: <ul style="list-style-type: none"><li>▪ Short (0.1 PLC): 1</li><li>▪ Medium (1.0 PLC): 2</li><li>▪ Long (10 PLC): 3</li><li>▪ Custom (4200A command set only): 4</li></ul>
X	Delay factor for custom setting: 0.0 to 100
Y	Filter factor for custom setting: 0.0 to 100
Z	A/D converter integration time in number of PLCs for custom setting: 0.01 to 10.0

## Details

SMU only.

The integration time is the time to convert a measurement. In general, a short integration time provides the fastest measurement speed at the expense of noise. Conversely, a long integration time provides stable readings at the expense of speed. Integration time is based on power line cycles (PLC). Assuming 60 Hz line power, the integration time for a 1 PLC setting is 16.67 ms (1/60).

The preconfigured integration time settings are equivalent to the Fast, Normal, and Quiet settings available in Clarius:

- **Short:** Equivalent to the Clarius Fast setting. Optimizes the SMU for speed at the expense of noise performance. It is a good choice for measurements where noise and settling time are not concerns.
- **Normal:** Equivalent to the Clarius Medium setting. The default and most commonly used setting. It provides a good combination of speed and low noise and is the best setting for most cases.
- **Long:** Equivalent to the Clarius Quiet setting. Optimizes the SMU for low-noise measurements at the expense of speed. If speed is not a critical consideration, it is a good choice when you need the lowest noise and most accurate measurements.

The custom setting combines delay factor, filter factor, and A/D integration time, which is comparable to the individual Clarius Delay Factor, Filter Factor, and A/D converter integration time settings.

Example 1

IT2

This command string sets the integration time to 1.0 PLC.

Example 2

IT4,2.5,0.6,1.3

This command string sets the delay factor to 2.5, the filter factor to 0.6, and the A/D converter integration time to 1.3 PLCs.

Also see

None

RD

This command requests real-time readings.

Usage

RD 'AAAAAA', N  
RD 'AAAAAAB', N

AAAAAA	User-specified name of the channel that made the measurement, up to 6 characters
B	T: Return timestamp data S: Return the SMU status
N	Index of the point to retrieve; valid range from 1 to the total number of readings

Details

SMU only.

While making measurements, use this command string to request the real-time reading. This can be used in a loop while measuring.

Output data is sent to the computer in the format  $\pm N.NNNN \ E \pm NN$ .

Output data of 0 indicates this point is not measured yet.

For an example that demonstrates how to use this command, refer to [Program 4: VAR1 sweep with real-time data retrieval](#) (on page 5-46).

Example

RD 'Volt', 1

This command string requests the first reading for a SMU channel that is named Volt.

Also see

[DO](#) (on page 5-38)

## RG

This command sets the lowest current range of the SMU to be used when measuring.

### Usage

RG *A, B*

<i>A</i>	SMU number (1 to 9)
<i>B</i>	<p>The lowest autoranged range in amps:</p> <ul style="list-style-type: none"> <li>▪ 4200-SMU or 4201-SMU without a preamplifier: <math>-100\text{e-}9</math> to <math>100\text{e-}3</math></li> <li>▪ 4200-SMU or 4201-SMU with a preamplifier: <math>-1\text{e-}12</math> to <math>100\text{e-}3</math></li> <li>▪ 4210-SMU or 4211-SMU without a preamplifier: <math>-100\text{e-}9</math> to 1</li> <li>▪ 4210-SMU or 4211-SMU with a preamplifier: <math>-1\text{e-}12</math> to 1</li> </ul>

### Details

SMU only.

The default autoranged ranges are 100 nA without a preamplifier and 1 nA with a preamplifier.

### Example

```
RG 2, 10E-12
```

This command string sets the lowest range of SMU2 with a preamplifier to 10 pA.

### Also see

None

## RI

This command instructs a SMU to go to a specified current range immediately without waiting until the initiation of a test.

### Usage

RI *channel, range, compliance*

<i>channel</i>	SMU number (1 to 9)
<i>range</i>	<p>Range value (amperes):</p> <ul style="list-style-type: none"> <li>▪ 4200-SMU or 4201-SMU without a preamplifier: <math>100\text{e-}9</math> to <math>100\text{e-}3</math></li> <li>▪ 4200-SMU or 4201-SMU with a preamplifier: <math>1\text{e-}12</math> to <math>100\text{e-}3</math></li> </ul>
<i>compliance</i>	<p>Current compliance value (amperes):</p> <ul style="list-style-type: none"> <li>▪ 4200-SMU or 4201-SMU: <math>-0.1050</math> to <math>+0.1050</math></li> <li>▪ 4210-SMU or 4211-SMU: <math>-1.0500</math> to <math>+1.0500</math></li> </ul>

### Details

SMU only.

Example

```
RI 1, 1E-3, 1E-3
```

This command string instructs SMU1 to go to the 1 mA range and set compliance to 1 mA.

Also see

None

RS

This command sets the measurement resolution for all channels of a SMU.

Usage

RS *A*

<i>A</i>	Resolution, in number of digits: <ul style="list-style-type: none"><li>■ 4200A command set: 3 to 7</li><li>■ 4145 Emulation command set: 3 to 5</li></ul>
----------	---

Details

SMU only.

Example

```
RS 7
```

This command string provides full SMU resolution when the 4200A command set is selected.

Also see

None

RV

This command instructs a SMU to go to a specified voltage range immediately without waiting until the initiation of a test.

Usage

RV *channel, range, compliance*

<i>channel</i>	SMU number (1 to 9)
<i>range</i>	Range value (voltage): -210.00 to +210.00
<i>compliance</i>	Voltage compliance value (voltage): -210.00 to +210.00

Details

SMU only.

The range is set to the range that is closest to the entered value. For example, if you set *range* to 10, the 20 V range is selected.

---

**Example**

---

```
RV 1, 2, 1
```

This command string instructs SMU1 to go to the 2 V range and to set compliance to 1 V.

---

**Also see**

---

[R1](#) (on page 5-42)

## Code examples

The following code examples use the C language and are provided to demonstrate system control using an external computer communicating through GPIB. For these programs, configure KXCI as follows:

- **GPIB address:** 17
- **Delimiter:** Comma
- **EOI:** Off

### Example 1: VAR1 and VAR2 sweep (system mode)

The following program demonstrates how to program the 4200A-SCS to perform a VAR1 and VAR2 sweep. It assumes that channels 1 through 4 of KXCI are configured for the SMU function.

```
MAXLEN = 2048;
addr = 17;

// Initialize card:
initialize(10, 0);

// Set speed to 0.01 PLC, clear buffer, and
// enable service request for data ready:
send(addr, "IT1 BC DR1",&status);

// Channel definition for SMU1; constant common:
send(addr, "DE CH1,'VE','IE',3,3", &status);

// Define SMU2 for VAR2 I sweep:
send(addr, "CH2,'VB','IB',2,2", &status);

// Define SMU3 for VAR1 V sweep:
send(addr, "CH3,'VC','IC',1,1", &status);

// Define SMU 4 to be off:
send(addr, "CH4", &status);

// Define V-sources and V-meters to be off:
send(addr, "VS1;VS2;VM1;VM2", &status);
```

```
// Source setup; VAR1 linear sweep from 0 V to 1 V in 50 mV
// steps, with I-compliance set to 50 mA:
send(addr, "SS VR1,0,1,0.05,50E-3", &status);

// Source setup; VAR2 sweep from 0 uA to 40 uA in 10 uA steps:
send(addr, "IP 10E-6,10E-6,4,3", &status);

// Select list display mode:
send(addr, "SM DM2", &status);

// Trigger start of test:
send(addr, "MD ME1", &status);

// Wait for data ready:
while(!srq());

// Save readings in a file named "PROG1":
send(addr, "SV 'D PROG1'", &status);
```

## Example 2: Basic source-measure (user mode)

The following program demonstrates how to program the 4200A-SCS to perform a basic source-measure operation. It assumes that channels 1 and 2 of KXCI are configured for the SMU function. The measured current reading performed by SMU1 (channel 1) is output to the computer.

```
MAXLEN = 2048;
addr = 17;

// Initialize card:
initialize(10, 0);

// Select user mode:
send(addr, "US", &status);

// Set speed to 0.01 PLC, clear buffer;
// enable service request for data ready:
send(addr, "IT1 BC DR1",&status);

// Set SMU1 to source 1.5 V on 20 V range; set compliance
// to 1 mA:
send(addr, "DV1,1, 1.5, 1E-3", &status);

// Set SMU2 to source 2 V on 20 V range; set compliance to 1 mA:
send(addr, "DV2,1,2,1E-3", &status);

// Trigger test; measure current using SMU1:
send(addr, "TI1", &status);

// Get reading:
enter(recv, MAXLEN, &len, addr, &status);

// Stop SMU outputs:
send(addr, "DV1;DV2", &status);
```



### Example 3: Retrieving saved data (system mode)

The following program demonstrates how to retrieve readings that are saved in a data file. In Example 1, SMU3 performed 80 measurements. The 80 current readings were then saved in a data file named 'PROG1'.

---

#### NOTE

The following program assumes that data file PROG1 already exists. This data file is created by Example 1.

---

```
MAXLEN = 2048;
addr = 17;

// Initialize card:
initialize(10, 0);

// Load data saved in file named "PROG1":
send(addr, "GT 'D PROG1'", &status);

// Output data; 'IC' is the measure channel (SMU3) used by
// Program 1:
send(addr, "DO 'IC'", &status);

// Obtain data:
enter(recv, MAXLEN, &len, addr, &status);
```

### Example 4: VAR1 sweep with real-time data retrieval

The following program demonstrates how to program the 4200A-SCS to perform a VAR1 sweep and retrieve data in real time. It assumes that channels 1 and 2 of KXCI are configured for the SMU function.

```
MAXLEN = 2048;
MAXTIMEOUT = 10000;

addr = 17;

// Initialize card:
initialize(10, 0);

// Set speed to 0.01 PLC, clear buffer, and
// enable service request for data ready:
send(addr, "IT1 BC DR1",&status);

// Channel definition for SMU1; VAR1 V sweep:
send(addr, "DE CH1,'V1','I1',1,1", &status);
```

```
// Channel definition for SMU2; constant common:
send(addr, "DE CH1,'V2','I2',3,3", &status);

// Source setup
// VAR1 linear sweep from -5V to 5V in 10 mV with 1mA compliance
numPoints = 1001;
send(addr, "SS VR1,-5,5,0.01,0.001", &status);

// Select list display mode (no graph):
send(addr, "SM DM2", &status);

// Trigger start of test:
send(addr, "MD ME1", &status);

// Get data at real time:
curPoint = 1;
timeoutCount = 0;
while (curPoint <= numPoints && timeoutCount < MAXTIMEOUT)
{
    Sleep(100);
    //Get timestamp for the present data point
    //It should be none zero for already completed reading
    sprintf(command, "RD'CH1T',%d", curPoint);
    send(addr, command, &status);

    // Get reading:
    enter(recv, MAXLEN, &len, addr, &status);
    if (len > 0)
    {
        //0 character return means no readings are ready for this data point
        if (!(len == 1 && recv[0] == '0'))
        {
            //Get voltage reading for curPoint
            sprintf(command, "RD'V1',%d", curPoint);
            send(addr, command, &status);
            enter(recv, MAXLEN, &len, addr, &status);

            //Get current reading for curPoint
            sprintf(command, "RD'I1',%d", curPoint);
            send(addr, command, &status);
            enter(recv, MAXLEN, &len, addr, &status);

            //Switch to next point
            curPoint++;
        }
    }
    timeoutCount++;
}
```

---

## KXCI CVU commands

### In this section:

KXCI CVU commands.....	6-1
Summary of KXCI CVU commands .....	6-2
User mode commands .....	6-3
System mode commands.....	6-7
Modeless commands .....	6-15
Code examples .....	6-25

## KXCI CVU commands

The commands in the capacitance-voltage unit (CVU) user and system command modes control the CVU using KXCI. This mimics the paradigm used to control the SMUs in I-V tests.

In user mode, all commands are processed sequentially and you have full control over the execution sequence and flow control. When a measurement is complete, only one point is generated (such as R + Jx, Cp, Gp, Cs, or Rs). User mode is similar to using the LPT library when running locally.

In system mode, when you send the run command, the system buffers and executes all CVU test parameters automatically. The CVU card executes full sweeps. The output sequencing, settling, and sweeping are all handled internally. The 4200A-SCS returns the entire buffer of test data. System mode is similar to running Clarius tests.

Many of the CVU commands are valid in both user and system mode, so they are considered to be modeless commands. Processing of these commands depends on the selected mode. If user mode is selected, the command is processed immediately and sent directly to the card. If system mode is selected, the parameter is buffered in a test object, and does not take effect until the execute command is sent. At that time, the CVU firmware acts on the settings and executes the test sequence.

To select the mode, use the [:CVU:MODE](#) (on page 6-21) command.

To set the mode to user, send:

```
:CVU:MODE 0
```

To set the mode to system, send:

```
:CVU:MODE 1
```

## NOTE

All the commands act on the selected card, which is set using the `:CVU:CHANNEL` command.

## Summary of KXCI CVU commands

Mode	Command	Brief description
User	<a href="#">:CVU:CABLE:COMP:LOAD</a> (on page 6-4)	Performs load compensation and collects the load compensation cable data for the CVU.
User	<a href="#">:CVU:CABLE:COMP:MEASCUSTOM</a> (on page 6-5)	Performs custom cable-length compensation and collects the compensation cable data for the CVU.
User	<a href="#">:CVU:CABLE:COMP:OPEN</a> (on page 6-5)	Performs open compensation and collects the open compensation cable data for the CVU.
User	<a href="#">:CVU:CABLE:COMP:SHORT</a> (on page 6-6)	Performs short compensation and collects the short compensation cable data for the CVU.
User	<a href="#">:CVU:MEASZ?</a> (on page 6-7)	Triggers and returns a single Z measurement using the present CVU settings.
System	<a href="#">:CVU:BIAS:DCV:SAMPLE</a> (on page 6-7)	Configures the CVU card to bias a DC voltage and sample $n$ Z measurements.
System	<a href="#">:CVU:DATA:FREQ?</a> (on page 6-8)	Queries the frequency measurement of the CVU when a test is complete.
System	<a href="#">:CVU:DATA:STATUS?</a> (on page 6-8)	Queries the status of the CVU when a test is complete.
System	<a href="#">:CVU:DATA:TSTAMP?</a> (on page 6-8)	Queries the timestamp of the measurements of the CVU when a test is complete.
System	<a href="#">:CVU:DATA:VOLT?</a> (on page 6-8)	Queries the voltage measurement of the CVU when a test is complete.
System	<a href="#">:CVU:DATA:Z?</a> (on page 6-9)	Queries the Z measurement of the CVU when a test is complete.
System	<a href="#">:CVU:DELAY:STEP</a> (on page 6-9)	Sets the hold time for the CVU test on the selected card for the <code>:CVU:SWEEEP:FREQ</code> and <code>:CVU:SWEEEP:DCV</code> operations.
System	<a href="#">:CVU:DELAY:SWEEEP</a> (on page 6-9)	Sets the sweep delay for the CVU test on the selected card.
System	<a href="#">:CVU:SAMPLE:HOLDT</a> (on page 6-10)	Sets the hold time for a sampling mode test on the selected card.
System	<a href="#">:CVU:SAMPLE:INTERVAL</a> (on page 6-10)	Sets the delay between samples for the selected card.
System	<a href="#">:CVU:SOAK:DCV</a> (on page 6-11)	Sets the presoak DC voltage for all sweeps for the selected CVU card.
System	<a href="#">:CVU:STANDBY</a> (on page 6-11)	Configures the selected CVU card to disable DC bias at the end of a test or to leave the DC bias active.
System	<a href="#">:CVU:STEP:FREQ</a> (on page 6-11)	Configures the selected CVU card to step frequency and sample Z measurements.

System	<a href="#">:CVU:SWEEP:ACV</a> (on page 6-12)	Configures the selected CVU card to sweep AC voltage and sample Z measurements.
System	<a href="#">:CVU:SWEEP:DCV</a> (on page 6-12)	Configures the selected CVU card to sweep DC voltage and sample Z measurements.
System	<a href="#">:CVU:SWEEP:FREQ</a> (on page 6-13)	Configures the selected CVU card to sweep frequency and sample Z measurements.
System	<a href="#">:CVU:SWEEP:LISTDCV</a> (on page 6-14)	Configures the selected CVU card to sweep arbitrary DC voltage points and sample Z measurements.
System	<a href="#">:CVU:TEST:ABORT</a> (on page 6-14)	Stops a running KXCI CVU test.
System	<a href="#">:CVU:TEST:RUN</a> (on page 6-15)	Starts a CVU test on the specified card.
Modeless	<a href="#">:CVU:ACV</a> (on page 6-15)	Sets the AC drive level for specified CVU card.
Modeless	<a href="#">:CVU:ACZ:RANGE</a> (on page 6-16)	Sets the AC measurement range for the specified CVU card.
Modeless	<a href="#">:CVU:CHANNEL</a> (on page 6-16)	Selects the CVU card on which subsequent CVU commands will act.
Modeless	<a href="#">:CVU:CONFIG:ACVHI</a> (on page 6-16)	Defines the source terminal (AC only) for the CVU test.
Modeless	<a href="#">:CVU:CONFIG:DCVHI</a> (on page 6-17)	Defines the source terminal (DC only) for the CVU test.
Modeless	<a href="#">:CVU:CORRECT</a> (on page 6-17)	Enables open, short, and load correction for the specified CVU card.
Modeless	<a href="#">:CVU:DCV</a> (on page 6-19)	Sets the DC bias voltage for the specified CVU card.
Modeless	<a href="#">:CVU:DCV:OFFSET</a> (on page 6-19)	Applies an offset value to the DC low terminal.
Modeless	<a href="#">:CVU:FREQ</a> (on page 6-20)	Sets the frequency for the AC source for the specified CVU card.
Modeless	<a href="#">:CVU:FSTEPSIZE</a> (on page 6-10)	Configures the frequency step size for the selected 4215-CVU card.
Modeless	<a href="#">:CVU:LENGTH</a> (on page 6-20)	Selects the cable length for the specified CVU card.
Modeless	<a href="#">:CVU:MODE</a> (on page 6-21)	Sets the mode to User or System.
Modeless	<a href="#">:CVU:MODEL</a> (on page 6-21)	Sets the measurement model for the selected CVU card.
Modeless	<a href="#">:CVU:OUTPUT</a> (on page 6-21)	Enables or disables the CVU output.
Modeless	<a href="#">:CVU:RESET</a> (on page 6-22)	Sends a soft reset to the specified card.
Modeless	<a href="#">:CVU:SPEED</a> (on page 6-22)	Sets the measurement speed for the selected CVU card.
Modeless	<a href="#">:CVU:TEST:COMPLETE?</a> (on page 6-25)	Queries the status of the test.

## User mode commands

In user mode, all commands are processed sequentially and you have full control over the execution sequence and flow control. When a measurement is complete, only one point is generated (such as R + Jx, Cp, Gp, Cs, or Rs). User mode is similar to using the LPT library when running locally.

# **:CVU:CABLE:COMP:LOAD**

This command performs load compensation and collects the load compensation cable data for the CVU.

## **Usage**

:CVU:CABLE:COMP:LOAD *length*, *load*

<i>length</i>	The length of the cable: <ul style="list-style-type: none"><li>0: ≤ 0.5 m</li><li>1.5: &gt;0.5 m to &lt; 2.5 m</li><li>3.0: ≥ 2.5 m to 5 m</li><li>4.0: Custom</li><li>5.0: CVIV 2W</li><li>6.0: CVIV 4W black 0.75 m cable</li><li>7.0: CVIV 4W blue 0.61 m cable</li></ul>
<i>load</i>	The load value

## **Details**

If you use the custom setting, run :CVU:CABLE:COMP:MEASCUSTOM before sending the :CVU:CABLE:COMP:LOAD command.

Custom cable compensation supports only one set of cable constants for a CVU. For example, if you run custom cable compensation for CVU2 and CVU1, the constants for CVU2 are lost.

To enable or disable load compensation, send :CVU:CORRECT.

For additional information on cable compensation, refer to “Connection compensation” in the *Model 4200A-SCS Capacitance-Voltage Unit (CVU) User’s Manual*.

## **Also see**

- [:CVU:CABLE:COMP:MEASCUSTOM](#) (on page 6-5)
- [:CVU:CORRECT](#) (on page 6-17)

---

# :CVU:CABLE:COMP:MEASCUSTOM

This command performs custom cable-length compensation and collects the compensation cable data for the CVU.

## Usage

:CVU:CABLE:COMP:MEASCUSTOM

## Details

If you are using the custom parameter for the open, short, or load commands, send the :CVU:CABLE:COMP:MEASCUSTOM command before sending those commands.

For additional information on cable compensation, refer to “Connection compensation” in the *Model 4200A-SCS Capacitance-Voltage Unit (CVU) User’s Manual*.

## Also see

- [:CVU:CABLE:COMP:LOAD](#) (on page 6-4)
- [:CVU:CABLE:COMP:OPEN](#) (on page 6-5)
- [:CVU:CABLE:COMP:SHORT](#) (on page 6-6)

---

# :CVU:CABLE:COMP:OPEN

This command performs open compensation and collects the open compensation cable data for the CVU.

## Usage

:CVU:CABLE:COMP:OPEN *length*

<i>length</i>	<div>The length of the cable:<ul style="list-style-type: none"><li>0: ≤ 0.5 m</li><li>1.5: &gt;0.5 m to &lt; 2.5 m</li><li>3.0: ≥ 2.5 m to 5 m</li><li>4.0: Custom</li><li>5.0: CVIV 2W</li><li>6.0: CVIV 4W black 0.75 m cable</li><li>7.0: CVIV 4W blue 0.61 m cable</li></ul></div>
---------------	--

## Details

If you use the custom setting, send the :CVU:CABLE:COMP:MEASCUSTOM command before sending the :CVU:CABLE:COMP:OPEN command.

Custom cable compensation supports only one set of cable constants for a CVU. For example, if you run custom cable compensation for CVU2 and CVU1, the constants for CVU2 are lost.

To enable or disable load compensation, send the `CVU:CORRECT` command.

For additional information on cable compensation, refer to “Connection compensation” in the *Model 4200A-SCS Capacitance-Voltage Unit (CVU) User’s Manual*.

Also see

- [:CVU:CABLE:COMP:MEASCUSTOM](#) (on page 6-5)
- [:CVU:CORRECT](#) (on page 6-17)

:CVU:CABLE:COMP:SHORT

This command performs short compensation and collects the short compensation cable data for the CVU.

Usage

:CVU:CABLE:COMP:SHORT *length*

<i>length</i>	The length of the cable: <ul style="list-style-type: none"><li>0: ≤ 0.5 m</li><li>1.5: &gt;0.5 m to &lt; 2.5 m</li><li>3.0: ≥ 2.5 m to 5 m</li><li>4.0: Custom</li><li>5.0: CVIV 2W</li><li>6.0: CVIV 4W black 0.75 m cable</li><li>7.0: CVIV 4W blue 0.61 m cable</li></ul>
---------------	--

Details

If you use the custom parameter, send the `:CVU:CABLE:COMP:MEASCUSTOM` command before sending the `:CVU:CABLE:COMP:SHORT` command.

Custom cable compensation supports only one set of cable constants for a CVU. For example, if you run custom cable compensation for CVU2 and CVU1, the constants for CVU2 are lost.

To enable or disable load compensation, send the `CVU:CORRECT` command.

For additional information on cable compensation, refer to “Connection compensation” in the *Model 4200A-SCS Capacitance-Voltage Unit (CVU) User’s Manual*.

Also see

- [:CVU:CABLE:COMP:MEASCUSTOM](#) (on page 6-5)
- [:CVU:CORRECT](#) (on page 6-17)



---

# :CVU:MEASZ?

This command triggers and returns a single Z measurement using the present CVU settings.

**Usage**

---

:CVU:MEASZ?

**Details**

---

When the command is complete, the single reading is available.

**Also see**

---

None

## System mode commands

The CVU system mode commands are similar to Clarius tests. When you send the run command, the system buffers and executes all CVU test parameters automatically. The CVU executes full sweeps. The output sequencing, settling, and sweeping are all handled internally on the CVU.

---

# :CVU:BIAS:DCV:SAMPLE

This command configures the CVU card to bias a DC voltage and sample *n* Z measurements.

**Usage**

---

:CVU:BIAS:DCV:SAMPLE *biasv*, *samples*

<i>biasv</i>	Voltage to source when sampling Z-measurements: -30 to +30
<i>samples</i>	The number of Z-measurements the CVU makes for the test operation: 1 to 4096

**Details**

---

Configures the CVU to bias a DC voltage and sample *n* Z measurements for the CVU card. The other parameters are set by their respective commands.

**Also see**

---

None

---

## **:CVU:DATA:FREQ?**

This command queries the frequency measurement of the CVU when a test is complete.

### **Usage**

---

:CVU:DATA:FREQ?

### **Also see**

---

None

---

## **:CVU:DATA:STATUS?**

This command queries the status of the CVU when a test is complete.

### **Usage**

---

:CVU:DATA:STATUS?

### **Also see**

---

None

---

## **:CVU:DATA:TSTAMP?**

This command queries the timestamp of the measurements of the CVU when a test is complete.

### **Usage**

---

:CVU:DATA:TSTAMP?

### **Also see**

---

None

---

## **:CVU:DATA:VOLT?**

This command queries the voltage measurement of the CVU when a test is complete.

### **Usage**

---

:CVU:DATA:VOLT?

### **Also see**

---

None

---

## :CVU:DATA:Z?

This command queries the Z measurement of the CVU when a test is complete.

### Usage

:CVU:DATA:Z?

### Details

Z measurements are returned as semicolon delimiter pairs. Each reading is then delimited with commas.

### Also see

None

---

## :CVU:DELAY:STEP

This command sets the hold time for the CVU test on the selected card for the :CVU:SWEEP:FREQ and :CVU:SWEEP:DCV operations.

### Usage

:CVU:DELAY:STEP *stepd*

*stepd*

Hold time to apply the presoak value in seconds: 0 to 999

### Details

The presoak value is set by the :CVU:SOAK:DCV command

### Also see

[:CVU:SOAK:DCV](#) (on page 6-11)

[:CVU:SWEEP:DCV](#) (on page 6-12)

[:CVU:SWEEP:FREQ](#) (on page 6-13)

---

## :CVU:DELAY:SWEEP

This command sets the sweep delay for the CVU test on the selected card.

### Usage

:CVU:DELAY:SWEEP *sweepd*

*sweepd*

Delay in seconds: 0 to 999

### Details

This command is used in sweeping mode for all sweep types except :CVU:BIAS:DCV:SAMPLE.

### Also see

None

## :CVU:FSTEPSIZE

This command configures the frequency step size for the selected 4215-CVU card.

### Usage

```
:CVU:FSTEPSIZE fstep
```

<i>fstep</i>	The frequency size for the sweep (1e3 to 10e6); see <b>Details</b>
--------------	--

### Details

The resolution of *fstep* is 1 kHz.

The lower and upper bounds shown are the maximum range. They are also limited based on the start and stop values set for the sweep.

Set *fstep* to 0 if you are using the 4210-CVU.

### Also see

[:CVU:STEP:FREQ](#) (on page 6-11) (4210-CVU only)

[:CVU:SWEEP:FREQ](#) (on page 6-13)

## :CVU:SAMPLE:HOLDT

This command sets the hold time for a sampling mode test on the selected card.

### Usage

```
:CVU:SAMPLE:HOLDT holdt
```

<i>holdt</i>	Hold time in seconds: 0 to 999
--------------	--------------------------------

### Details

This command is only used when executing the `:CVU:BIAS:DCV:SAMPLE` command.

### Also see

[:CVU:BIAS:DCV:SAMPLE](#) (on page 6-7)

## :CVU:SAMPLE:INTERVAL

This command sets the delay between samples for the selected card.

### Usage

```
:CVU:SAMPLE:INTERVAL interval
```

<i>interval</i>	Delay in seconds: 0 to 999
-----------------	----------------------------

### Details

This command is only used when executing the `:CVU:BIAS:DCV:SAMPLE` command.

### Also see

[:CVU:BIAS:DCV:SAMPLE](#) (on page 6-7)

---

# **:CVU:SOAK:DCV**

This command sets the presoak DC voltage for all sweeps for the selected CVU card.

**Usage**

---

:CVU:SOAK:DCV *voltage*

<i>voltage</i>	The voltage to bias before test sequence begins: -30 to +30
----------------	---

**Also see**

---

None

---

# **:CVU:STANDBY**

This command configures the selected CVU card to disable DC bias at the end of a test or to leave the DC bias active.

**Usage**

---

:CVU:STANDBY *state*

<i>state</i>	The state of the CVU card at the end of the test: <ul style="list-style-type: none"><li>■ Disable the DC voltage output: 1</li><li>■ Leave the DC bias active: 0</li></ul>
--------------	--

**Also see**

---

None

---

# **:CVU:STEP:FREQ**

This command configures the selected CVU card to step frequency and sample Z measurements.

**Usage**

---

:CVU:STEP:FREQ *fstart*, *fstop*

<i>fstart</i>	The frequency that is used for capturing the initial sample in the sweep
<i>fstop</i>	The frequency that is used for capturing the final sample in the sweep

**Details**

---

Only the start frequency and stop frequency are set by this command for the step order. The other parameters are set by their respective commands.

---

## **NOTE**

Values are coerced to one of the 37 discrete frequencies for the 4210-CVU.

---

NOTE

To perform a linear sweep using the 4215-CVU, use the `:CVU:FSTEPsize` command.

Also see

[:CVU:FSTEPsize](#) (on page 6-10)

:CVU:SWEEP:ACV

This command configures the selected CVU card to sweep AC voltage and sample Z measurements.

Usage

`:CVU:SWEEP:ACV acvstart, acvstop, acvstep`

<i>acvstart</i>	Start AC voltage, in volts: <ul style="list-style-type: none"><li>■ 4210-CVU: 0.01 to 0.1</li><li>■ 4215-CVU: 0.01 to 1.0</li></ul>
<i>acvstop</i>	Stop AC voltage, in volts: <ul style="list-style-type: none"><li>■ 4210-CVU: 0.01 to 0.1</li><li>■ 4215-CVU: 0.01 to 1.0</li></ul>
<i>acvstep</i>	The AC voltage step size

Details

The other parameters (such as DC voltage level and frequency) are set by their respective commands.

Also see

None

:CVU:SWEEP:DCV

This command configures the selected CVU card to sweep DC voltage and sample Z measurements.

Usage

`:CVU:SWEEP:DCV dcvstart, dcvstop, dcvstep`

<i>dcvstart</i>	Start voltage, in volts: -30 to +30
<i>dcvstop</i>	Stop voltage, in volts: -30 to +30
<i>dcvstep</i>	Voltage step size

Details

The other parameters (AC voltage level, frequency, and so on) are set by their respective commands.

Example

```
:CVU:MODEL 2
:CVU:SPEED 2
:CVU:ACZ:RANGE 0
:CVU:FREQ 1E6
:CVU:SWEEP:DCV 5, -5, -0.2
:CVU:DELAY:SWEEP 0.1 - sweep delay
:CVU:TEST:RUN
```

Set the measurement model to Cp-GP.

Set the measurement speed to quiet.

Set the AC measurement range to autorange.

Set the frequency for the AC source to 1E6.

Set sweep to start at 5 V, stop at -5 V, and step in -0.2 V increments.

Set the sweep delay to 0.1 s.

Run the test.

Also see

None

:CVU:SWEEP:FREQ

This command configures the selected CVU card to sweep frequency and sample Z measurements.

Usage

```
:CVU:SWEEP:FREQ fstart, fstop
:CVU:SWEEP:FREQ fstart, fstop, order
```

<i>fstart</i>	The frequency that is used for capturing the initial sample in the sweep
<i>fstop</i>	The frequency that is used for capturing the final sample in the sweep
<i>order</i>	Determines whether the test biases DC voltage and sweeps frequency or steps voltage and sweeps frequency: <ul style="list-style-type: none"><li>1: The voltage bias is the voltage set by the <code>:CVU:DCV</code> command (default)</li><li>2: The voltage steps based on the <i>vstart</i>, <i>vstop</i>, <i>vstep</i> parameters set by the <code>:CVU:SWEEP:DCV</code> command.</li></ul>

Details

This command sets the start frequency, stop frequency, and sweep order. The other parameters are set by their respective commands.

NOTE

Values are coerced to one of the 37 discrete frequencies for the 4210-CVU.

NOTE

To perform a linear sweep using the 4215-CVU, use the `:CVU:FSTEPSIZE` command.

Also see

- [:CVU:DCV](#) (on page 6-19)
- [:CVU:FSTEPSIZE](#) (on page 6-10)
- [:CVU:SWEEP:DCV](#) (on page 6-12)

---

## :CVU:SWEEP:LISTDCV

This command configures the selected CVU card to sweep arbitrary DC voltage points and sample Z measurements.

### Usage

---

```
:CVU:SWEEP:LISTDCV pt1, pt2, pt3 ... ptn
```

<i>pt1, pt2, pt3 ... ptn</i>	Sweep points: 1 to 4096
------------------------------	-------------------------

### Details

---

Only starting voltage sweep points are set with this command.

The other parameters (AC voltage level, frequency, and so on) are set by their respective commands.

### Also see

---

[:CVU:SWEEP:ACV](#) (on page 6-12)

[:CVU:SWEEP:DCV](#) (on page 6-12)

[:CVU:SWEEP:FREQ](#) (on page 6-13)

---

## :CVU:TEST:ABORT

This command stops a running KXCI CVU test.

### Usage

---

```
:CVU:TEST:ABORT
```

### Details

---

This command terminates all ongoing processes and returns the 4200A-SCS to the idle state. Data that results from the ongoing processes may be corrupt.

This command is not appropriate to stop a CVU KXCI test running from a remote UTM. For more information, refer to [Calling KULT user libraries remotely](#) (on page 4-6).

This command is not valid in user mode.

### Also see

---

None



---

# :CVU:TEST:RUN

This command starts a CVU test on the specified card.

**Usage**

---

:CVU:TEST:RUN

**Details**

---

Use the serial poll byte to determine when not busy or data ready.

**Also see**

---

None

---

## Modeless commands

---

# :CVU:ACV

This command sets the AC drive level for specified CVU card.

**Usage**

---

:CVU:ACV *aclevel*

*aclevel*

AC voltage level (default 0.03):

- 4210-CVU: 0.01 to 0.1
- 4215-CVU: 0.01 to 1

**Details**

---

If the instrument is in user mode, this takes immediate effect.

If the instrument is in system mode, the value is buffered for use in all sweeps except the AC voltage sweep. The voltage level for the AC voltage sweep is set using the :CVU:SWEEP:ACV command.

**Also see**

---

[:CVU:SWEEP:ACV](#) (on page 6-12)

---

# :CVU:ACZ:RANGE

This command sets the AC measurement range for the specified CVU card.

**Usage**

---

:CVU:ACZ:RANGE *range*

<i>range</i>	Range in amps: <ul style="list-style-type: none"><li>■ Auto: 0</li><li>■ 1 <math>\mu</math>A: 1e-6</li><li>■ 30 <math>\mu</math>A: 30e-6</li><li>■ 1 mA: 1e-3 (default)</li></ul>
--------------	---

**Also see**

---

None

---

# :CVU:CHANNEL

This command selects the CVU card on which subsequent CVU commands will act.

**Usage**

---

:CVU:CHANNEL *chan*

<i>chan</i>	CVU card on which KXCI CVU commands act (default 1)
-------------	---

**Details**

---

The majority of systems only have one CVU card. You can use this command to configure multiple cards.

**Also see**

---

None

---

# :CVU:CONFIG:ACVHI

This command defines the source terminal (AC only) for the CVU test.

**Usage**

---

:CVU:CONFIG:ACVHI *source*

<i>source</i>	The source terminal to be used: <ul style="list-style-type: none"><li>■ HCUR/HPOT (default): 1</li><li>■ LCUR/LPOT: 2</li></ul>
---------------	---

**Also see**

---

None

---

# :CVU:CONFIG:DCVHI

This command defines the source terminal (DC only) for the CVU test.

## Usage

---

:CVU:CONFIG:DCVHI *source*

<i>source</i>	The source terminal to be used: <ul style="list-style-type: none"><li>■ HCUR/HPOT (default): 1</li><li>■ LCUR/LPOT: 2</li></ul>
---------------	---

## Also see

---

None

---

# :CVU:CORRECT

This command enables open, short, and load correction for the specified CVU card.

## Usage

---

:CVU:CORRECT *open, short, load*

<i>open</i>	Off: 0 On: 1
<i>short</i>	Off: 0 On: 1
<i>load</i>	Off: 0 On: 1

## Details

---

Each of the parameters is a Boolean (0 = off, 1 = on).

This command enables the values set by the :CVU:CABLE:COMP:LOAD, :CVU:CABLE:COMP:OPEN, and :CVU:CABLE:COMP:SHORT commands.

To disable the correction setting values, before sending :CVU:CORRECT, you must send :CVU:RESET to set the CVU to its default settings.

---

**Example 1**

```
:CVU:RESET  
:CVU:MEASZ?  
:CVU:CORRECT 1,0,0  
:CVU:MEASZ?  
:CVU:RESET  
:CVU:MODEL 2  
:CVU:CORRECT 0,1,0  
:CVU:MEASZ?
```

Reset the CVU to its default state.  
Trigger and return a single measurement.  
Enable open compensation.  
Trigger and return a single measurement.  
Reset the CVU to its default state.  
Set the CVU model to Cp, Gp.  
Enable short compensation and disable open compensation.  
Trigger and return a single measurement.

---

**Example 2**

```
:CVU:RESET  
:CVU:MEASZ?  
:CVU:CORRECT 1,0,0  
:CVU:MEASZ?  
:CVU:CORRECT 1,1,0  
:CVU:MEASZ?
```

Reset the CVU to its default state.  
Trigger and return a single measurement.  
Enable open compensation.  
Trigger and return a single measurement.  
Enable short compensation while keeping open compensation enabled.  
Trigger and return a single measurement.

---

**Also see**

[:CVU:CABLE:COMP:LOAD](#) (on page 6-4)  
[:CVU:CABLE:COMP:OPEN](#) (on page 6-5)  
[:CVU:CABLE:COMP:SHORT](#) (on page 6-6)  
[:CVU:RESET](#) (on page 6-22)

---

## :CVU:DCV

This command sets the DC bias voltage for the specified CVU card.

### Usage

---

```
:CVU:DCV dclevel
```

<i>dclevel</i>	DC bias voltage: -30 to +30 (default is 0)
----------------	--

### Details

---

In user mode, this takes immediate effect.

In system mode, the value is buffered for use in frequency and AC voltage sweeps and DC bias/sample.

### Also see

---

None

---

## :CVU:DCV:OFFSET

This command applies an offset value to the DC low terminal.

### Usage

---

```
:CVU:DCV:OFFSET offsetv
```

<i>offsetv</i>	Offset voltage: -30 to +30
----------------	----------------------------

### Details

---

Voltage offset to apply while sampling Z measurements.

### Also see

---

None

# :CVU:FREQ

This command sets the frequency for the AC source for the specified CVU card.

## Usage

CVU:FREQ <i>freq</i>	
<i>freq</i>	Range (Hz): 1000 to 10000000; default is 100000

## Details

Values that fall between supported frequencies are coerced to the nearest supported frequency.

For the CVU-4210, values that fall between the supported discrete frequencies are coerced to the closest discrete frequency.

For the CVU-4215, the values are adjusted to 1 kHz resolution.

## Also see

[:CVU:DATA:FREQ?](#) (on page 6-8)

[:CVU:SWEEP:ACV](#) (on page 6-12)

# :CVU:LENGTH

This command selects the cable length for the specified CVU card.

## Usage

:CVU:LENGTH <i>len</i>	
<i>len</i>	<div>The cable length:<ul style="list-style-type: none"><li>0: 0 m (no cable compensation)</li><li>1.5: 1.5 m CVU cable</li><li>3.0: 3.0 m CVU cable</li><li>4.0: Custom</li><li>5.0: 1.5 m CVIV cable; 2-wire mode</li><li>6.0: 1.5 m CVU to CVIV cable with 0.75 m CVIV to DUT cable; 4-wire mode</li><li>7.0: Blue 1.5 m CVU to CVIV cable with 0.61 m CVIV to DUT cable; 4-wire mode</li></ul></div>

## Also see

None

---

# :CVU:MODE

This command sets the mode to User or System.

**Usage**

---

:CVU:MODE *mode*

<i>mode</i>	The mode: <ul style="list-style-type: none"><li>▪ User mode: 0</li><li>▪ System mode: 1</li></ul>
-------------	---

**Also see**

---

None

---

# :CVU:MODEL

This command sets the measurement model for the selected CVU card.

**Usage**

---

:CVU:MODEL *model*

<i>model</i>	The model: <ul style="list-style-type: none"><li>▪ 0: Z, theta</li><li>▪ 1: R + jx (default)</li><li>▪ 2: Cp, Gp</li><li>▪ 3: Cs, Rs</li><li>▪ 4: Cp, D</li><li>▪ 5: Cs, D</li><li>▪ 7: Y, theta</li></ul>
--------------	--

**Also see**

---

None

---

# :CVU:OUTPUT

This command enables or disables the CVU output.

**Usage**

---

:CVU:OUTPUT *state*

<i>state</i>	The state of the CVU output: <ul style="list-style-type: none"><li>▪ Enable CVU output: 1</li><li>▪ Disable the CVU output: 0</li></ul>
--------------	---

**Also see**

---

None

---

# :CVU:RESET

This command sends a soft reset to the specified card.

## Usage

---

:CVU:RESET

## Details

---

This command places the card in its default state.

The following parameters are reset to the following default values:

- DC voltage: 0 V
- AC voltage: 0.03 V (30 mV)
- Frequency: 100000 Hz (100 kHz)
- Model: R + jx
- Range: 0.001 A (1 mA)

\*RST executes the :CVU:RESET command.

## Also see

---

- [\\*RST](#) (on page 4-5)
- [:CVU:ACV](#) (on page 6-15)
- [:CVU:ACZ:RANGE](#) (on page 6-16)
- [:CVU:DCV](#) (on page 6-19)
- [:CVU:FREQ](#) (on page 6-20)
- [:CVU:MODEL](#) (on page 6-21)

---

# :CVU:SPEED

This command sets the measurement speed for the selected CVU card.

## Usage

---

:CVU:SPEED speed  
:CVU:SPEED 3, delay\_factor, filter\_factor, aperture

speed	Applies the speed selection: <ul style="list-style-type: none"><li>■ Fast: 0</li><li>■ Normal: 1</li><li>■ Quiet: 2</li><li>■ Custom: 3</li></ul>
delay_factor	Delay factor (Custom only): 0 to 100; see <b>Details</b>
filter_factor	Filter factor (Custom only): 0 to 707; see <b>Details</b>
aperture	Aperture (Custom only) in PLCs: 0.006 to 10.002



---

**Details**

---

**delay\_factor parameter**

After applying a voltage or current, the instrument waits for a delay time before making a measurement. The delay time allows for source settling. The delay factor setting adjusts the delay time for the selected speed. It is fixed for the fast, normal, and quiet speeds. If you select custom speed, you can enter a factor for the delay.

The applied delay time is a multiple of the default delay time. The delay factor setting specifies this multiple. That is:

$$\text{Applied delay time} = (\text{Default delay time}) \times (\text{Delay factor})$$

For example, if the default delay time is 1 ms and the delay factor is 0.7, the actual applied delay time is 0.7 ms (1 ms  $\times$  0.7).

The following table summarizes the delay factor settings.

Speed mode	Delay factor
Fast	0.7
Normal	1.0
Quiet	1.3
Custom	0 to 100

When entering a custom delay factor setting, consider the following:

- A delay factor of 1 allows for settling before the analog-to-digital (A/D) converter is triggered to make a measurement.
- Each doubling of the delay factor doubles the time allowed for settling.
- A delay factor of 0 multiplies the default delay by zero, resulting in no delay.

---

**NOTE**

If you set the filter factor and delay factor to 0, the internal preprogrammed values are ignored.

In general, cables and matrices increase the settling time. You may need to experiment to find the ideal time. However, for a good quality switch, such as the Keithley 7174A High-Speed Low Leakage Current Matrix, you should not need to increase the delay factor by more than two times.

---

### filter\_factor parameter

To reduce measurement noise, each 4200A-SCS applies filtering, which averages multiple readings to make one measurement. The filter factor is fixed for fast, normal, and quiet speeds. If you select the custom speed mode, you can enter a filter factor of 0 to 100.

The default filter factor settings are listed in the following table.

Speed mode	Filter factor
Fast	0.2
Normal	1
Quiet	3
Custom	0 to 100

When typing a custom filter factor, consider the following:

- A filter factor of 1 specifies a normal level of filtering.
- As a rule of thumb, doubling the filter factor halves the measurement noise.
- A filter factor of 0 nullifies the SMU internal filtering.

### aperture parameter

You can specify the A/D converter integration time that is used to measure a signal. Each measured reading is the result of one or more A/D conversions. A short integration time for each A/D conversion results in a relatively fast measurement speed with increased noise. A long integration time results in a relatively low-noise reading at a relatively low speed.

The integration time setting is based on the number of power line cycles (NPLCs). For 60 Hz line power, 1.0 PLC = 16.67 ms (1/60). For 50 Hz line power, 1.0 PLC = 20 ms (1/50).

The applied A/D conversion time also depends on the filter factor setting:

- If the filter factor is not zero, the CVU applies an optimum A/D converter time that is based on the filter factor setting. The applied A/D converter time value is never less than the specified A/D aperture time.
- If the filter factor setting is zero, the CVU applies a fixed A/D converter time that equals the specified A/D aperture time.

The automatic A/D aperture and A/D aperture time settings for each speed mode are shown in the following table.

Speed mode	A/D aperture time (PLCs)
Fast	Auto
Normal	Auto
Quiet	Auto
Custom	0.01 to 10

### Also see

None

---

## :CVU:TEST:COMPLETE?

This command queries the status of the test.

### Usage

---

:CVU:TEST:COMPLETE?

### Also see

---

None

## Code examples

---

### NOTE

The GPIB commands in the following examples are specific to the Keithley KUSB-488B USB to GPIB Converter. For information on using the KUSB-488B for GPIB communications, refer to the KUSB-488B documentation, available on [tek.com](http://tek.com).

---

### Example 1

The following code segment sets CVU1 to perform a system mode sweep of DC voltage from 5 V to 10 V in 1 V steps. After the test completes, the Z, DCV, F, timestamps, and status values are queried. The operation mode is then changed to perform a list sweep of DC voltage (2 V, 4 V, 3 V, 5 V, 7 V) and the test is run again with all other test conditions as before. There are no compensation corrections made to the reading.

```
// Soft reset the card.
send(addr, ":CVU:RESET", &status);

// Set CVU to system mode.
send(addr, ":CVU:MODE 1", &status);

// Set measurement model to Z, theta.
send(addr, ":CVU:MODEL 0", &status);

// Set speed to normal.
send(addr, ":CVU:SPEED 1", &status);

// Set the AC drive level to 45 mV at 7 MHz.
send(addr, ":CVU:ACV 0.045", &status);
send(addr, ":CVU:FREQ 7E+6", &status);

// Set the DC bias level to 10 V.
send(addr, ":CVU:DCV 10", &status);

// Select the 1 mA measurement range.
send(addr, ":CVU:ACZ:RANGE 1E-3", &status);
```

```
//Set cable length to 1.5 m.
send(addr, ":CVU:LENGTH 1.5", &status);

// Set the test function to
// DC voltage sweep from 5 V to 10 V.
send(addr, ":CVU:SWEEP:DCV 5,10,1", &status);

// Set 1 s delay between points.
send(addr, ":CVU:DELAY:SWEEP 1.0", &status);

// Start the test.
send(addr, ":CVU:TEST:RUN", &status);

// Monitor the spill byte for test completion.
WaitForTestCompletion();

// Query all data.
send(addr, ":CVU:DATA:Z?", &status);
enter(recvstr, MAXLEN, &len, addr, &status);

send(addr, ":CVU:DATA:VOLT?", &status);
enter(recvstr, MAXLEN, &len, addr, &status);

send(addr, ":CVU:DATA:FREQ?", &status);
enter(recvstr, MAXLEN, &len, addr, &status);

send(addr, ":CVU:DATA:TSTAMP?", &status);
enter(recvstr, MAXLEN, &len, addr, &status);

send(addr, ":CVU:DATA:STATUS?", &status);
enter(recvstr, MAXLEN, &len, addr, &status);

// Change the sweep mode to list sweep.
send(addr, ":CVU:SWEEP:LISTDCV 2,4,3,5,7", &status);

// Rerun the test.
send(addr, ":CVU:TEST:RUN", &status);
WaitForTestCompletion();

send(addr, ":CVU:DATA:VOLT?", &status);
enter(recvstr, MAXLEN, &len, addr, &status);
```

## Example 2

This example makes a single measurement in user mode. The program sets up CVU1 to output 30 mV at 10 MHz AC. The DC voltage is set to 5 V. Once a single measurement is retrieved, the AC source is set to 60 mV at 5 MHz and a new reading is acquired.

```
// Set the CVU to user mode.
send(addr, ":CVU:MODE 0", &status);

// Soft reset the card.
send(addr, ":CVU:RESET", &status);

// Set measurement model to Cs,Rs.
send(addr, ":CVU:MODEL 3", &status);

// Set speed to quiet.
send(addr, ":CVU:SPEED 2", &status);

// Set AC drive level to 30 mV at 10 MHz.
send(addr, ":CVU:ACV 0.030", &status);
send(addr, ":CVU:FREQ 10E+6", &status);

// Set DC bias level to 5 V.
send(addr, ":CVU:DCV 5", &status);

// Select 30 uA measurement range.
send(addr, ":CVU:ACZ:RANGE 30E-6", &status);

// Trigger a new measurement and retrieve it.
send(addr, ":CVU:MEASZ", &status);
enter(recvstr, MAXLEN, &len, addr, &status);

// Change the AC source to 60 mV at 5 MHz.
send(addr, ":CVU:ACV 0.060", &status);
send(addr, ":CVU:FREQ 5E+6", &status);

// Get a new Z measurement.
send(addr, ":CVU:MEASZ", &status);
enter(recvstr, MAXLEN, &len, addr, &status);
```

---

## KXCI PGU and PMU commands

### In this section:

Introduction .....	7-2
Summary of KXCI PGU and PMU commands .....	7-3
:PMU:ABORT .....	7-4
:PMU:CONNECTION:COMP .....	7-5
:PMU:DATA:COUNT? .....	7-6
:PMU:DATA:GET .....	7-7
:PMU:EXECUTE .....	7-9
:PMU:INIT .....	7-9
:PMU:LLEC:CONFIGURE .....	7-10
:PMU:LOAD .....	7-11
:PMU:MEASURE:CONNECTION:COMP .....	7-12
:PMU:MEASURE:MODE .....	7-13
:PMU:MEASURE:PIV .....	7-14
:PMU:MEASURE:RANGE .....	7-15
:PMU:OUTPUT:STATE .....	7-16
:PMU:PULSE:BURST:COUNT .....	7-17
:PMU:PULSE:TIMES .....	7-18
:PMU:PULSE:TRAIN .....	7-21
:PMU:RETAIN:CONFIG .....	7-21
:PMU:RPM:CONFIGURE .....	7-22
:PMU:SAMPLE:RATE .....	7-23
:PMU:SOURCE:RANGE .....	7-24
:PMU:STEP:DC .....	7-25
:PMU:STEP:PULSE:AMPLITUDE .....	7-26
:PMU:STEP:PULSE:BASE .....	7-27
:PMU:SWEEP:DC .....	7-28
:PMU:SWEEP:PULSE:AMPLITUDE .....	7-29
:PMU:SWEEP:PULSE:BASE .....	7-31
:PMU:TEST:STATUS? .....	7-32
:PMU:TIMES:PIV .....	7-32
:PMU:TIMES:WAVEFORM .....	7-34
Programming examples .....	7-35
Summary of Segment Arb commands .....	7-39
:PMU:SARB:SEQ:MEAS:START .....	7-40
:PMU:SARB:SEQ:MEAS:START:ADD .....	7-41
:PMU:SARB:SEQ:MEAS:STOP .....	7-42
:PMU:SARB:SEQ:MEAS:STOP:ADD .....	7-43
:PMU:SARB:SEQ:MEAS:TYPE .....	7-44
:PMU:SARB:SEQ:MEAS:TYPE:ADD .....	7-45
:PMU:SARB:SEQ:SSR .....	7-46
:PMU:SARB:SEQ:SSR:ADD .....	7-47
:PMU:SARB:SEQ:STARTV .....	7-48
:PMU:SARB:SEQ:STARTV:ADD .....	7-49
:PMU:SARB:SEQ:STOPV .....	7-50
:PMU:SARB:SEQ:STOPV:ADD .....	7-51
:PMU:SARB:SEQ:TIME .....	7-52
:PMU:SARB:SEQ:TIME:ADD .....	7-53
:PMU:SARB:SEQ:TRIG .....	7-54
:PMU:SARB:SEQ:TRIG:ADD .....	7-55
:PMU:SARB:WFM:SEQ:LIST .....	7-56

:PMU:SARB:WFM:SEQ:LIST:ADD .....	7-56
Programming example for Segment Arb .....	7-57
KXCI pulse commands for Clarius versions v1.12 and earlier	7-59

## Introduction

The following command descriptions are for the KXCI PGU and PMU commands that are available in Clarius+ versions 1.13 and later. These system mode commands provide pulse control for PGUs and pulse and measurement control for PMUs. These commands support standard pulse mode and Segment Arb mode.

The PGU commands that were available in earlier versions are still available. Refer to [KXCI pulse commands for Clarius versions v1.12 and earlier](#) (on page 7-59) for command descriptions.

---

### NOTE

You cannot use the two-letter KXCI commands for PGUs with the latest commands.

---

Guidelines for using KXCI with PGUs and PMUs:

- If you need to remove or add an RPM, close KXCI, open KCon, and update the configuration. Save the configuration. After the save, you can reopen KXCI and continue programming.
- If you are using an RPM, use the command [:PMU:RPM:CONFIGURE](#) (on page 7-22) to connect the pulse source of the PMU to the RPM.
- If you are using Segment Arb, you must use fixed ranges.
- You can return data while a test is running using the [:PMU:DATA:GET](#) (on page 7-7) command.
- The maximum number of points that can be measured and returned is 65536.
- Check that you are using the current and voltage ranges based on the device you are testing.
- Always turn off the output of the channel at the end of a test for safety using [:PMU:OUTPUT:STATE](#) (on page 7-16).
- The source range and measure range are the same for voltage.

Example programs that demonstrate how to assemble the KXCI PGU and PMU commands for testing are provided in [Programming examples](#) (on page 7-35).

## Summary of KXCI PGU and PMU commands

Command	Brief description	Default value
<a href="#">:PMU:ABORT</a> (on page 7-4)	Aborts processes that are presently running on PGUs and PMUs and turns off the PGU and PMU outputs.	Not applicable
<a href="#">:PMU:CONNECTION:COMP</a> (on page 7-5)	Enables or disables PMU connection compensation and sets the type of compensation to be used for the specified channel.	0 (disabled)
<a href="#">:PMU:DATA:COUNT?</a> (on page 7-6)	Used to determine how many readings are stored in the data buffer for the specified channel.	Not applicable
<a href="#">:PMU:DATA:GET</a> (on page 7-7)	Retrieves data, block by block, for the specified channel.	Not applicable
<a href="#">:PMU:EXECUTE</a> (on page 7-9)	Starts a configured pulse test.	Not applicable
<a href="#">:PMU:INIT</a> (on page 7-9)	Initializes the pulse card to pulse mode or Segment Arb mode and its default conditions.	Not applicable
<a href="#">:PMU:LLEC:CONFIGURE</a> (on page 7-10)	Enables or disables load-line effect compensation for the specified channel.	0 (disabled)
<a href="#">:PMU:LOAD</a> (on page 7-11)	Sets the device under test (DUT) impedance (pulse load) for the specified channel.	1e6 ohms
<a href="#">:PMU:MEASURE:CONNECTION:COMP</a> (on page 7-12)	Measures and stores short or offset current connection compensation data for a PMU.	Not applicable
<a href="#">:PMU:MEASURE:MODE</a> (on page 7-13)	Sets the measure mode for the waveform capture and pulse I-V test modes.	1
<a href="#">:PMU:MEASURE:PIV</a> (on page 7-14)	Sets the readings to be returned in pulse I-V mode for the specified channel.	All parameters enabled
<a href="#">:PMU:MEASURE:RANGE</a> (on page 7-15)	Sets the current measure range for the specified pulse channel.	Fixed range of 10E-3
<a href="#">:PMU:OUTPUT:STATE</a> (on page 7-16)	Sets the output state for the specified channel.	0 (off)
<a href="#">:PMU:PULSE:BURST:COUNT</a> (on page 7-17)	Sets the total number of pulses to be output for all active pulse channels.	1
<a href="#">:PMU:PULSE:TIMES</a> (on page 7-18)	Sets the pulse period, pulse width, pulse rise time, pulse fall time, and pulse delay for the specified pulse channel.	Period: 1e-6 Width: 500e-9 Pulse rise time: 100e-9 Pulse fall time: 100e-9 Delay: 0
<a href="#">:PMU:PULSE:TRAIN</a> (on page 7-21)	Configures the pulse base and amplitude voltage levels of the specified pulse channel.	Not applicable
<a href="#">:PMU:RETAIN:CONFIG</a> (on page 7-21)	Retains the present configuration for the specified channel.	Not applicable
<a href="#">:PMU:RPM:CONFIGURE</a> (on page 7-22)	Switches the RPM output between the PGUs, PMUs, CVUs, and SMUs.	Not applicable
<a href="#">:PMU:SAMPLE:RATE</a> (on page 7-23)	Sets the number of samples taken in samples per second.	200 MSamples
<a href="#">:PMU:SOURCE:RANGE</a> (on page 7-24)	Sets the voltage range for measure and pulse for the specified channel.	10 V
<a href="#">:PMU:STEP:DC</a> (on page 7-25)	Configures the pulse step for stepping the DC voltage level for the specified channel.	Not applicable



Command	Brief description	Default value
<a href="#">:PMU:STEP:PULSE:AMPLITUDE</a> (on page 7-26)	Configures the pulse step for stepping the pulse voltage amplitude with a fixed base level for the specified channel.	Not applicable
<a href="#">:PMU:STEP:PULSE:BASE</a> (on page 7-27)	Configures the pulse step for stepping the pulse voltage base level with a fixed voltage amplitude for the specified channel. Only available if the mode is set to pulse I-V.	Not applicable
<a href="#">:PMU:SWEEP:DC</a> (on page 7-28)	Configures the pulse sweep for sweeping the DC voltage level for the specified channel	Not applicable
<a href="#">:PMU:SWEEP:PULSE:AMPLITUDE</a> (on page 7-29)	Configures the pulse sweep for sweeping the pulse voltage amplitude with a fixed base level for the specified channel.	Not applicable
<a href="#">:PMU:SWEEP:PULSE:BASE</a> (on page 7-31)	Configures the pulse sweep for sweeping the pulse voltage base level with a fixed voltage amplitude for the specified channel.	Not applicable
<a href="#">:PMU:TEST:STATUS?</a> (on page 7-32)	Used to determine if a test is running or idle.	Not applicable
<a href="#">:PMU:TIMES:PIV</a> (on page 7-32)	Sets the spot mean measurement window for pulse I-V capture.	Stop mean start: 0.75 (75%) Stop mean stop: 0.90 (90%)
<a href="#">:PMU:TIMES:WAVEFORM</a> (on page 7-34)	Defines the predata and postdata measurement windows for waveform capture for the specified channel.	Prepercent: 0.2 Postpercent: 0.2

## :PMU:ABORT

This command aborts processes that are presently running on PGUs and PMUs and turns off the PGU and PMU outputs.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

### Usage

```
:PMU:ABORT
```

### Details

If test data was generated before the abort command was sent, the data remains in the buffer and you can retrieve it.

### Example

```
:PMU:ABORT
```

This command string aborts processes that are running on PGU and PMU channels and turns off all PGU and PMU outputs.

### Also see

None

## :PMU:CONNECTION:COMP

This command enables or disables PMU connection compensation and sets the type of compensation to be used for the specified channel.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT	0 (Disabled)

### Usage

:PMU:CONNECTION:COMP *ch*, *type*, *key*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system; see <b>Details</b>
<i>type</i>	The type of compensation data to enable or disable: <ul style="list-style-type: none"> <li>Short compensation: 1</li> <li>Offset current compensation: 3</li> </ul>
<i>key</i>	Enable or disable short or offset current compensation values: <ul style="list-style-type: none"> <li>Disable compensation: 0</li> <li>Enable compensation: 1</li> </ul>

### Details

Errors caused by the connections and cable length between the 4225-PMU and the device under test (DUT) can be corrected by using connection compensation. When connection compensation is enabled, the compensation values are factored into each DUT measurement.

The *ch* parameter determines which PMU has compensation data enabled or disabled. Select the lowest number channel for the PMU. For example, to gather compensation data for PMU2, select channel 3.

Connection compensation is a two-part process:

1. Measure and store connection compensation data using  
:PMU:MEASURE:CONNECTION:COMP. The compensation values are stored in tables.
2. Use the :PMU:CONNECTION:COMP command to enable the custom connection compensation values.

If connection compensation is disabled, the compensation values are not used by the test.

To view the compensation data, in Clarius, select **Tools**, then select **PMU Connection Compensation**.

## NOTE

For details on performing connection compensation, refer to “Perform connection compensation” in the *Model 4200A-SCS Pulse Card (PGU and PMU) User’s Manual*.

Example

```
-- Reset the PMU to default settings
:PMU:INIT 0
-- Configure the RPM input for channel 1 of pulse card 1 to the PMU.
:PMU:RPM:CONFIGURE PMU1-1, 0
-- Configure the RPM input for channel 2 of pulse card 1 to the PMU.
:PMU:RPM:CONFIGURE PMU1-2, 0
-- Set the measure mode to spot mean discrete.
:PMU:MEASURE:MODE 1
-- Configure short connection compensation for PMU 1 of the pulse card.
:PMU:CONNECTION:COMP 1, 1, 1
-- Configure a pulse amplitude step for channel 1 of the PMU starting
-- at 2 V, stopping at 4 V, stepping in 0.5 V steps.
-- Setting the base to 0 V with dual sweep off.
:PMU:SWEEP:PULSE:AMPLITUDE 1, 2, 4, 0.5, 0, 0
-- Configure the current measure range to a fixed range of 10 mA.
:PMU:MEASURE:RANGE 1, 2, 0.01
-- Set the output state of channels 1 to on.
:PMU:OUTPUT:STATE 1, 1
-- Execute the commands.
:PMU:EXECUTE
```

This example executes a sweep that uses short connection compensation values.

Also see

[:PMU:MEASURE:CONNECTION:COMP](#) (on page 7-12)

:PMU:DATA:COUNT?

This command is used to determine how many readings are stored in the data buffer for the specified channel.

Instrument	Affected by	Default value
PMU	PMU INIT *RST	Not applicable

Usage

```
:PMU:DATA:COUNT? ch
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
-----------	--

Details

Use this command to return the number of readings presently stored in the data buffer. This command can be called while a test is in progress or after test completion.

For a short test, this command is typically called after test completion to determine the total number of readings stored in the buffer. For a long test, you can use this command during execution to track the progress of the test.

**Example**

```
:PMU:DATA:COUNT? 1
```

This command returns the number of readings stored in the buffer for channel 1.

**Also see**

[:PMU:DATA:GET](#) (on page 7-7)

[:PMU:TEST:STATUS?](#) (on page 7-32)

**:PMU:DATA:GET**

This command retrieves data, block by block, for the specified channel.

Instruments	Affected by	Default value
PMU	:PMU:MEASURE:PIV :PMU:SAMPLE:RATE :PMU:EXECUTE :PMU:INIT *RST	Not applicable

**Usage**

```
:PMU:DATA:GET ch
:PMU:DATA:GET ch, startIndex
:PMU:DATA:GET ch, startIndex, numPoints
:PMU:DATA:GET ch, startIndex, numPoints, requestedValues
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>startIndex</i>	The start index (0 based); any valid point as given by :PMU:DATA:COUNT?
<i>numPoints</i>	The number of points to return: 1 to 2048
<i>requestedValues</i>	Comma-delimited list to return specific data for all the measured values; see <b>Details</b>

**Details**

The PMU data buffer may contain up to 65,536 data points per channel. If no data is available from *startIndex*, the command returns an empty string.

Large data sets are broken into 2048 character blocks for the KXCI buffer and

:PMU:DATA:GET returns 2048 points at a time. To get all the data, send :PMU:DATA:GET multiple times with an appropriate *startIndex* for each call and then concatenate the data. You can use the :PMU:DATA:COUNT? command to retrieve the number of data points. For an example program that demonstrates retrieving more than 2048 points, refer to [Example: Using KXCI to generate a pulse train](#) (on page 7-37).

If the data buffer contains less data than the *startIndex* + *numPoints* amount of data, this query may return fewer points than requested.

If you only specify *ch*, all available points are returned (up to 2048 points).

If you specify the *startIndex* and omit *numPoints*, all available points from the specified index to the end of the data (up to 2048 points) are returned.

You can use *requestedValues* to specify which data values are returned. For waveform capture, the options are:

- Voltage: *V*
- Current: *I*
- Timestamp: *T*
- Status: *S*

For spot mean, the options are:

- Voltage high: *VH*
- Current high: *IH*
- Timestamp high: *TH*
- Status high: *SH*
- Voltage low: *VL*
- Current low: *IL*
- Timestamp low: *TL*
- Status low: *SL*

You can specify more than one value. The data points are returned in the order in which they are defined by the `:PMU:DATA:GET` command.

In the returned data, the data points are separated by semicolons. Values within data points are separated by commas, as shown here:

```
<vh1>,<ih1>,<th1>,<sh1>,<vl1>,<il1>,<tl1>,<sl1>;<vh2>,<ih2>,<th2>,<sh2>,<vl2>,<il2>,<tl2>,<sl2>;<vh3>,<ih3>,<th3>,<sh3>,<vl3>,<il3>,<tl3>,<sl3>
```

This is a non-destructive read. If you call this command multiple times with the same parameters, the same data is returned. New data is appended to the existing buffer. Data is purged when `:PMU:EXECUTE`, `:PMU:INIT`, or `*RST` is called.

#### Example 1

```
:PMU:DATA:GET 1, 0, 2048, V, S
```

Returns the voltage and status for waveform mode on channel 1.

#### Example 2

```
:PMU:DATA:GET 1, 0, 2048, VL, VH
```

Returns the voltage low and voltage high (in that order) for pulse I-V mode on channel 1.

#### Also see

[:PMU:DATA:COUNT?](#) (on page 7-6)

[:PMU:MEASURE:PIV](#) (on page 7-14)

## :PMU:EXECUTE

This command starts a configured pulse test.

Instruments	Affected by	Default value
PGU PMU	:PMU:ABORT	Not applicable

### Usage

```
:PMU:EXECUTE
```

### Details

When this command is run, it does a final verification, clears the data buffers, and executes the test.

This command returns immediately. Use `:PMU:TEST:STATUS?` to check the status of the test.

### Also see

[:PMU:ABORT](#) (on page 7-4)

[:PMU:RETAIN:CONFIG](#) (on page 7-21)

[:PMU:TEST:STATUS](#) (on page 7-32)

## :PMU:INIT

This command initializes the pulse card to pulse mode or Segment Arb mode and its default conditions.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

### Usage

```
:PMU:INIT mode
```

*mode*

Set the mode on reset:

- Standard pulse mode: 0
- Segment Arb mode: 1

### Details

This command resets both channels of the pulse card to the default settings of the specified pulse mode. It must be the first command sent before sending other commands.

This command also clears the data buffer and deletes all Segment Arb sequences.

You can also send `*RST` to reset all cards in the systems. When `*RST` is used, the pulse mode is set to standard pulse mode.

Default values are listed in [Summary of KXCI PGU and PMU commands](#) (on page 7-3) and [Summary of Segment Arb commands](#) (on page 7-39).

Example

<code>:PMU:INIT 0</code>
Performs a reset of the system and initializes the PMU into standard pulse mode.

Also see

[\\*RST](#) (on page 4-5)

**:PMU:LLEC:CONFIGURE**

This command enables or disables load-line effect compensation for the specified channel.

Instruments	Affected by	Default value
PMU	Not applicable	0

Usage

`:PMU:LLEC:CONFIGURE ch, LLEComp`

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>LLEComp</i>	Disable load-line effect compensation: 0 Enable load-line effect compensation: 1

Details

When load-line effect compensation is enabled, the PMU test will take more time to complete. Load-line effect compensation runs the pulse in iterations before the actual test to calculate the actual DUT load and compensate for it.

For a detailed example that uses this command, refer to [Example: Use KXCI to generate a pulse I-V drain family of curves](#) (on page 7-35).

Example

<code>:PMU:LLEC:CONFIGURE 1, 1</code>
Enable load-line effect compensation for channel 1.

Also see

None

# :PMU:LOAD

This command sets the device under test (DUT) impedance (pulse load) for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	1e6

## Usage

:PMU:LOAD *ch*, *load*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>impedance</i>	Pulse load, in ohms: 1.0 to 1e7

## Details

You may need to set the DUT load so the 4200A-SCS can correctly calculate the PMU output levels for a specific DUT. The calculations will adjust the pulse levels to the PMU output impedance of 50 Ω.

For example, if the DUT load used by default is 1 MΩ, but the actual DUT load has a low impedance of 50 Ω, setting a voltage level of 4 V results in a 2 V pulse at the DUT. Setting the DUT load to 50 Ω permits the set voltage to match the actual voltage, so setting a 4 V level results in a 4 V pulse on DUT, with the pulse card taking the DUT impedance into account.

## NOTE

Use the :PMU:LLEC:CONFIGURE command to enable automatic detection of the DUT load.

## Example

```
:PMU:LOAD 1, 1e3
```

This example sets the load impedance to 1000 Ω.

## Also see

[:PMU:LLEC:CONFIGURE](#) (on page 7-10)

[:PMU:MEASURE:PIV](#) (on page 7-14)



## :PMU:MEASURE:CONNECTION:COMP

This command measures and stores short or offset current connection compensation data for a PMU.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT	Not applicable

### Usage

```
:PMU:MEASURE:CONNECTION:COMP ch, compType
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system; see <b>Details</b>
<i>compType</i>	The type of compensation data to retrieve: <ul style="list-style-type: none"> <li>Short compensation: 1</li> <li>Offset current compensation: 3</li> </ul>

### Details

Errors caused by the connections and cable length between the 4225-PMU and the device under test (DUT) can be corrected by using connection compensation. When connection compensation is enabled, the compensation values are factored into each DUT measurement.

The *ch* parameter determines which PMU the compensation data is generated for. Select the lowest number channel for the PMU. For example, to gather compensation data for PMU2, select channel 3.

To measure short connection compensation values:

1. Connect the center conductors of channel 1 and 2 together by contacting probes to a common pad or THRU structure.
2. Send the `:PMU:MEASURE:CONNECTION:COMP` command with *compType* set to 1.
3. Check the KXCI Console to verify it was successfully completed.

To measure current offset compensation values:

1. Lift the probes for both PMU channels or remove the device from the test fixture.
2. Send the `:PMU:MEASURE:CONNECTION:COMP` command with *compType* set to 3.

To enable the compensation values, use the `:PMU:CONNECTION:COMP` command.

To view the compensation data, in Clarius, select **Tools** and select **PMU Connection Compensation**.

## NOTE

For detail on performing connection compensation, refer to “Perform connection compensation” in the *Model 4200A-SCS Pulse Card (PGU and PMU) User's Manual*.

Example 1

`:PMU:MEASURE:CONNECTION:COMP 1, 1`

This example measures and stores short compensation data for PMU channels 1 and 2.  
On the KXCI Console, a typical response for the short measurement is:  
Compensate: PMU1, ch1 R=0.56 Ohms  
Compensate: PMU1, ch2 R=0.58 Ohms

Example 2

`-- Measure and store offset current compensation data for PMU channels 1 and 2.  
:PMU:MEASURE:CONNECTION:COMP 1, 3`

This example measures and stores offset current compensation data for PMU channels 1 and 2.  
The first few lines of a typical response for the current offset measurement (with RPM) are:  
ForceRange=40V, MeasureRange=0.8A      Offset:    Ch1=-5.63e-04    CH2=-5.11e-04  
ForceRange=40V, MeasureRange=0.01A    Offset:    Ch1=-7.48e-06    CH2=-9.00e-06  
ForceRange=40V, MeasureRange=0.0001A   Offset:    Ch1=-4.13e-08    CH2=-6.39e-08  
ForceRange=10V, MeasureRange=0.2A      Offset:    Ch1=-1.96e-05    CH2=3.37e-05

Also see

[:PMU:CONNECTION:COMP](#) (on page 7-5)

:PMU:MEASURE:MODE

This command sets the measure mode for the waveform capture and pulse I-V test modes.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT	1

Usage

`:PMU:MEASURE:MODE mode`

<i>mode</i>	No measurements: 0 Spot mean discrete: 1 Waveform discrete: 2 Spot mean average: 3 Waveform average: 4
-------------	--

Details

- For pulse I-V measurements:
- When spot mean discrete is selected, the 4200A-SCS returns the averaged voltage and current readings for every pulse in the burst.
  - When the spot mean average is selected, the 4200A-SCS averages the spot mean readings for all pulses in the burst into a single data set.

For waveform measurements:

- When discrete is selected, the 4200A-SCS returns samples of voltage and current readings and timestamp readings for every pulse in the burst.
- When average is selected, the 4200A-SCS returns readings as a mean average of the corresponding samples for all the pulses in the burst.

For the average modes, the number of pulses is defined by the burst count set in :PMU:PULSE:BURST:COUNT.

### Example

```
:PMU:MEASURE:MODE 1
```

Sets the pulse measurement mode to the spot mean discrete.

### Also see

[:PMU:PULSE:BURST:COUNT](#) (on page 7-17)

[:PMU:MEASURE:PIV](#) (on page 7-14)

## :PMU:MEASURE:PIV

This command sets the readings to be returned in pulse I-V mode for the specified channel.

Instruments	Affected by	Default value
PMU	:PMU:MEASURE:MODE	All parameters enabled

### Usage

```
:PMU:MEASURE:PIV ch, AcquireHigh, AcquireLow
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>AcquireHigh</i>	Acquire high measurements: <ul style="list-style-type: none"> <li>■ Disable: 0</li> <li>■ Enable: 1</li> </ul>
<i>AcquireLow</i>	Acquire low measurements: <ul style="list-style-type: none"> <li>■ Disable: 0</li> <li>■ Enable: 1</li> </ul>

### Details

To acquire pulse I-V measurements:

- :PMU:MEASURE:MODE must be set to spot mean discrete or spot mean average.
- At least one of the :PMU:MEASURE:PIV parameters must be set to Enable (1). If both parameters are disabled, an error is generated and the PMU acquires all measurements.

To speed up your test execution, measure only high or low values. Measuring only high or low values is faster than measuring both values.

**Example**

```
:PMU:MEASURE:PIV 1, 1, 1
```

Acquire voltage high, current high, time high, status high, voltage low, current low, time low, and status low values for channel 1.

**Also see**

[:PMU:DATA:GET](#) (on page 7-7)

[:PMU:MEASURE:MODE](#) (on page 7-13)

**:PMU:MEASURE:RANGE**

This command sets the current measure range for the specified pulse channel.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT	Fixed range of 10E-3

**Usage**

```
:PMU:MEASURE:RANGE ch, IRangeType
```

```
:PMU:MEASURE:RANGE ch, IRangeType, IMeasRange
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>IRangeType</i>	Current range type: <ul style="list-style-type: none"> <li>■ Autorange: 0</li> <li>■ Limited autorange: 1</li> <li>■ Fixed range: 2</li> </ul>
<i>IMeasRange</i>	Current measure range in amps; see <b>Details</b>

**Details**

The current range depends on the PMUs and RPMs that are available in the system, as defined in the following table.

IMeasRange (A)	40 V PMU	40 V RPM	10 V PMU	10 V RPM
0.8	OK	OK	Not available	Not available
0.2	Not available	Not available	OK	OK
0.01	OK	OK	OK	OK
0.001	Not available	Not available	Not available	OK
0.0001	OK	OK	Not available	OK
0.00001	Not available	Not available	Not available	OK
0.000001	Not available	Not available	Not available	OK
0.0000001	Not available	Not available	Not available	OK

Limited autorange is a compromise between autorange and fixed range. It uses the *IMeasRange* value as the minimum range during autorange. This reduces test time if you do not need maximum resolution at minimum currents.

If you are using Segment Arb, you must use a fixed range.

The voltage range is set by `:PMU:SOURCE:RANGE`.

#### Example 1

```
:PMU:MEASURE:RANGE 1, 0, 0.01
```

Sets autorange as the current measure type for PMU channel 1. The value for the current measure range is ignored.

#### Example 2

```
:PMU:MEASURE:RANGE 1, 2, 0.02
```

Sets a fixed current measurement range of 0.2 A on PMU channel 1.

#### Also see

[:PMU:MEASURE:MODE](#) (on page 7-13)

[:PMU:SOURCE:RANGE](#) (on page 7-24)

## :PMU:OUTPUT:STATE

This command sets the output state for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	0

#### Usage

```
:PMU:OUTPUT:STATE ch, state
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>state</i>	Output state: <ul style="list-style-type: none"> <li>■ Off: 0</li> <li>■ On: 1</li> </ul>

#### Details

When off is sent, this command takes effect immediately.

When on is sent, this command takes effect when `:PMU:EXECUTE` is sent.

You must set the output state to on to execute a pulse. When the pulse is complete, set the output state back to off.

### NOTE

It is good practice to routinely turn off the outputs of the pulse card after a test has been completed.

Example

```
:PMU:INIT 0

:PMU:LOAD 1, 1e6
:PMU:CONNECTION:COMP 1, 1, 0
:PMU:LLEC:CONFIGURE 1, 0
:PMU:SOURCE:RANGE 1, 40
:PMU:MEASURE:RANGE 1, 2, 0.01
:PMU:PULSE:BURST:COUNT 2
:PMU:PULSE:TRAIN 1, 0, 5
:PMU:PULSE:TIMES 1, 1e-6, 0.5e-6, 1e-7, 1e-7, 0
:PMU:MEASURE:MODE 4
:PMU:OUTPUT:STATE 1, 1
:PMU:EXECUTE
-- Wait until test completes.
:PMU:OUTPUT:STATE 1, 0
```

Reset all PMU channels and select standard pulse mode.  
Configure PMU channel 1 to output a pulse train and use waveform capture mode with waveform average for measurements.  
Start the test.  
Wait for test completion.  
Turn off the output.

Also see

None

:PMU:PULSE:BURST:COUNT

This command sets the total number of pulses to be output for all active pulse channels.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	1

Usage

```
:PMU:PULSE:BURST:COUNT burstCount
```

<i>burstCount</i>	Pulse count in number of pulses: 1 to 10000
-------------------	---

Details

This command sets the number of pulses used in average mode or discrete mode with either spot mean or waveform measurements.

The burst count setting affects the channels of all active pulse card channels.

The number of data points that the PMU generates is determined by the measurement type:

- For spot mean discrete, *burstCount* data points are generated.
- For waveform discrete, *burstCount* \* *Waveform Points Count* data point are generated.
- For spot mean average, 1 data point is generated.
- For waveform average, *Waveform Points Count* data points are generated.

**Example**

```
:PMU:PULSE:BURST:COUNT 50
```

Sets a burst count of 50 for all active PMU channels.

**Also see**

[:PMU:PULSE:TIMES](#) (on page 7-18)

**:PMU:PULSE:TIMES**

This command sets the pulse period, pulse width, pulse rise time, pulse fall time, and pulse delay for the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Period: 1e-6 (1 $\mu$ s) Width: 500e-9 (500 ns) Pulse rise time: 100e-9 (100 ns) Pulse fall time: 100e-9 (100 ns) Delay: 0 (no delay)

**Usage**

```
:PMU:PULSE:TIMES ch, period, width, riset, fallt
```

```
:PMU:PULSE:TIMES ch, period, width, riset, fallt, delay
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>period</i>	Pulse period (in seconds): <ul style="list-style-type: none"> <li>■ <b>PGU 10 V range:</b> 20e-9 to 1 (20 ns to 1 s)</li> <li>■ <b>PMU 10 V range:</b> 60e-9 (60 ns to 1 s)</li> <li>■ <b>40 V range:</b> 500e-9 to 1 (500 ns to 1 s)</li> </ul> Must be the same for all channels
<i>width</i>	Pulse width in seconds: <ul style="list-style-type: none"> <li>■ <b>PGU 10 V:</b> 10e-9 to (Period – 10e-9)</li> <li>■ <b>PMU 10 V:</b> 40e-9 to (Period – 10e-9)</li> <li>■ <b>40 V:</b> 250e-9 to (Period – 10e-9)</li> </ul> Must be less than <i>period</i> and more than (0.5( <i>riset</i> + <i>fallt</i> ))
<i>riset</i>	Pulse rise time in seconds (floating-point number): <ul style="list-style-type: none"> <li>■ <b>PGU 10 V:</b> 10e-9 to 33e-3 (10 ns to 33 ms)</li> <li>■ <b>PMU 10 V:</b> 20e-9 to 33e-3 (20 ns to 33 ms)</li> <li>■ <b>40 V:</b> 50e-9 to 33e-3 (50 ns to 33 ms)</li> </ul>
<i>fallt</i>	Pulse fall time in seconds (floating-point number): <ul style="list-style-type: none"> <li>■ <b>PGU 10 V:</b> 10e-9 to 33e-3 (10 ns to 33 ms)</li> <li>■ <b>PMU 10 V:</b> 20e-9 to 33e-3 (20 ns to 33 ms)</li> <li>■ <b>40 V:</b> 50e-9 to 33e-3 (50 ns to 33 ms)</li> </ul>
<i>delay</i>	The pulse delay time; see <b>Details</b> : <ul style="list-style-type: none"> <li>■ <b>No delay:</b> 0</li> <li>■ <b>PGU 10 V:</b> 10e-9 (10 ns)</li> <li>■ <b>PMU 10 V:</b> 20e-9 (20 ns)</li> <li>■ <b>40 V:</b> 50e-9 (50 ns)</li> </ul>

---

**Details**

---

Use this command to set the timing parameters for the test. Pulse width, rise time, fall time, and delay are individually set for each channel. However, the period must be the same on all PMU channels. The period defined in the most recently sent `:PMU:PULSE:TIMES` command is used for all channels.

There are several restrictions when setting the pulse time:

- The *width* must be less than the *period* and more than  $0.5(\text{riset} + \text{fallt})$ .
- The rise time of a pulse cannot be longer than the pulse width.
- For the 40 V range, the minimum transition time for pulse source only (no measurement) on the rise-time high voltage is 50 ns for the 4225-PMU and 4220-PGU.
- The *delay* must be less than  $(\text{period} - \text{width} - 0.5(\text{riset} + \text{fallt}))$ . Set to 0 for no delay.
- The minimum off time, equal to  $(\text{period} - \text{delay} - \text{width} - 0.5 \times (\text{riset} + \text{fallt}))$ , must be more than 40 ns.

The values are checked when `:PMU:EXECUTE` is sent.

If you are setting the timing for a sample (waveform capture), setting the delay to a small value allows the PMU to better capture the rising edge of the pulse. This value is sample rate dependent, but for the 200 MSa/s rate, a pulse delay of 20 ns to 100 ns allows the rising edge of the pulse to be captured.

To see the whole pulse transition to high when capturing waveform data, use a small nonzero value like 10 ns for the delay.



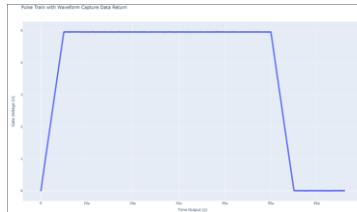
## Example

```
:PMU:INIT 0
:PMU:RPM:CONFIGURE PMU1-1, 0
:PMU:MEASURE:RANGE 1, 2, 10e-6
:PMU:MEASURE:MODE 2
:PMU:PULSE:TRAIN 1, 0.0, 5.0
:PMU:PULSE:TIMES 1, 100e-6, 50e-6, 5e-6, 5e-6
:PMU:OUTPUT:STATE 1, 1
:PMU:EXECUTE
:PMU:TEST:STATUS?
:PMU:OUTPUT:STATE 1, 0
```

This example outputs a pulse train and measures the voltage. This example:

- Resets and initializes the PMU and sets the mode to standard pulse mode.
- Configures the RPM to channel 1 of the PMU.
- Sets the current range to 10  $\mu\text{A}$  for 1  $\text{M}\Omega$  scope impedance.
- Sets the measure mode to 2, waveform capture discrete.
- Sets channel 1 of the pulse card for a pulse train with a 5 V amplitude.
- Sets a period of 100  $\mu\text{s}$ , 50  $\mu\text{s}$  width, and 5  $\mu\text{s}$  rise and fall times.
- Sets the output state of channel 1 to on.
- Executes the commands.
- Returns 1 if the test is running, 0 if it is complete.
- Turns off the output.

The graph of the voltage compared to time is shown in the following figure.



## Also see

[:PMU:EXECUTE](#) (on page 7-9)

## :PMU:PULSE:TRAIN

This command configures the pulse base and amplitude voltage levels of the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	Not applicable

### Usage

```
:PMU:PULSE:TRAIN ch, vbase, vamplitude
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>vbase</i>	Voltage level for pulse base level
<i>vamplitude</i>	Voltage level for pulse amplitude

### Details

*vbase* sets the starting voltage level and *vamplitude* sets the peak voltage level.

You can use this command to set up a pulse train to output on the specified pulse channel.

If the source range is set to 10 V, you can only span 10 volts at a time. If the source range is set to 40 V, you can only span 40 volts at a time. For example, if the 10 V range is selected, you can set *vbase* to -5 V and *vamplitude* to 5 V.

For a detailed example that uses this command, refer to [Example: Using KXCI to generate a pulse train](#) (on page 7-37).

### Example

```
:PMU:PULSE:TRAIN 1, 0, 2
```

This example is a pulse train that goes from 0 V (base) to 2 V (amplitude).

### Also see

[:PMU:SOURCE:RANGE](#) (on page 7-24)

## :PMU:RETAIN:CONFIG

This command retains the present configuration for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	0

### Usage

```
:PMU:EXECUTE:STATE ch, state
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>state</i>	Disable: 0 Enable: 1

## Details

When this command is set to enable (1), the programmed PGU and PMU settings for the specific channel are saved so that you can send the `:PMU:EXECUTE` command multiple times without sending all the PMU commands each time.

## Example

```
:PMU:RETAIN:CONFIG 1, 0
```

Sets channel 1 of the pulse card to not retain the configuration.

## Also see

[:PMU:EXECUTE](#) (on page 7-9)

# :PMU:RPM:CONFIGURE

This command switches the RPM output between the PGUs, PMUs, CVUs, and SMUs.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

## Usage

```
:PMU:RPM:CONFIGURE HRID, mode
```

<i>HRID</i>	<p>The PGU or PMU instrument and channel (<i>N</i> is the card index, 1 to 9):</p> <ul style="list-style-type: none"> <li>■ <code>PMUN-1</code>, <code>PMUN-2</code></li> <li>■ <code>SMUN</code></li> <li>■ <code>CVHN</code> or <code>CVLN</code></li> </ul>
<i>mode</i>	<p>The mode index number:</p> <ul style="list-style-type: none"> <li>■ PMU: 0</li> <li>■ CV_2W: 1</li> <li>■ SMU: 2</li> <li>■ CV_4W: 3</li> </ul>

## Details

In addition to the connection to the PMU, the 4225-RPM includes input connections for the following instruments:

- SMUs: 4200-SMU, 4201-SMU, 4210-SMU, or 4211-SMU
- CVUs: 4210-CVU or 4215-CVU

Use this command to control switching inside the RPM to connect the PMU, CVU, or SMU to the output.

When using the PMU with an RPM, `:PMU:RPM:CONFIGURE` must be called to connect the pulse source to the RPM output.

For detailed examples that use this command for two channels, refer to [Programming examples](#) (on page 7-35).

Example

```
:PMU:INIT 0
:PMU:RPM:CONFIGURE PMU1-1, 0
:PMU:LOAD 1, 1e3
:PMU:MEASURE:MODE 1
:PMU:SWEEP:PULSE:AMPLITUDE 1, -5, 5, 0.1, 0, 0
:PMU:PULSE:TIMES 1, 10e-6, 5e-6, 5e-6, 1e-7, 0
:PMU:MEASURE:RANGE 1, 2, 10e-3
:PMU:TIMES:PIV 1, 0.75, 0.9
:PMU:OUTPUT:STATE 1, 1
:PMU:EXECUTE
```

This example creates a pulse I-V sweep. This example:

- Initializes the PMUs and sets them to standard pulse mode.
- Connects PMU 1, channel 1, to the RPM.
- Sets load impedance to 1000 Ω.
- Sets the measure mode to spot mean discrete.
- Sets the amplitude sweep parameters to start V = -5 V, stop V = 5 V, step size = 0.1, and base voltage = 0 V.
- Sets the pulse timing parameters to period = 10e-6 s, pulse width = 5e-6 s, and rise and fall times = 1e-7 s.
- Sets the measure range to a fixed range of 10 mA.
- Sets the spot mean measurement window for pulse I-V capture to 75% for the start and 90% for the stop location.
- Turns the PMU output on.
- Executes the test.

In this example, for each pulse, one reading is derived.

Also see

None

:PMU:SAMPLE:RATE

This command sets the number of samples taken in samples per second.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT	200 MSamples

Usage

```
:PMU:SAMPLE:RATE rate
```

rate	The rate (samples per second, entered as an integer): 1e3 to 200e6
------	--

Details

The sample rate must be fast enough to sample the shortest measure interval in the waveform. If the rate is too slow, KXCI automatically adjusts the sample rate to the lowest value that can be used for the test, based on the formula:

$$slowestSampleRate = 1 / shortestMeasureInterval$$

The number of data points generated by the PMU with configured sample rate must be below 65536 (the maximum number of allowed points for KXCI). If the data points exceed 65536, KXCI will automatically adjust the sample rate to the fastest sample rate KXCI can use that does not exceed 65536, based on the formula:

$$fastestSampleRate = 65536 / TotalMeasureDuration$$

The *slowestSampleRate* cannot be more than the *fastestSampleRate*.

Example

```
:PMU:SAMPLE:RATE 2e3
```

Sets the number of samples to be 2000 per second.

Also see

None

:PMU:SOURCE:RANGE

This command sets the voltage range for measure and pulse for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	10 V

Usage

```
:PMU:SOURCE:RANGE ch, VSrcRange
```

ch	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
VSrcRange	Voltage source range in volts: 10 or 40

Details

Setting the pulse range of a pulse card channel to 10 V selects the low-voltage range.

Setting the pulse range of a pulse card channel to 40 V selects the high-voltage range.

This setting takes effect when :PMU:EXECUTE is sent. At that time, the pulse parameters are verified. If any of these parameters are out of bounds, an error message is generated.

**Example**

```
:PMU:SOURCE:RANGE 1, 40
```

Set channel 1 to the 40 V source range.

**Also see**

[:PMU:PULSE:TIMES](#) (on page 7-18)

**:PMU:STEP:DC**

This command configures the pulse step for stepping the DC voltage level for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	Not applicable

**Usage**

```
:PMU:STEP:DC ch, start, stop, step
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>start</i>	Initial step value in volts (limited by range)
<i>stop</i>	Final step value in volts (limited by range)
<i>step</i>	Step size in volts

**Details**

Every voltage step receives a full voltage sweep, which is configured on a different PMU channel.

You must configure a sweep for another PMU channel before setting this command.

**Example**

```
:PMU:STEP:DC 1, 0, 5, 0.5
:PMU:SWEEP:DC 2, 0, 5, 0.5, 0
```

Steps the DC voltage level on channel 1 starting at 1 V and going to 5 V in 0.5 V steps. Sweep the DC voltage level on channel 2 starting at 0 V and stopping at 5 V, stepping in 0.5 V steps. Dual sweep is disabled.

**Also see**

[:PMU:MEASURE:MODE](#) (on page 7-13)

[:PMU:SOURCE:RANGE](#) (on page 7-24)

[:PMU:STEP:PULSE:AMPLITUDE](#) (on page 7-26)

[:PMU:STEP:PULSE:BASE](#) (on page 7-27)

## :PMU:STEP:PULSE:AMPLITUDE

This command configures the pulse step for stepping the pulse voltage amplitude with a fixed base level for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	Not applicable

### Usage

```
:PMU:STEP:PULSE:AMPLITUDE ch, start, stop, step, vbase
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>start</i>	Initial step value in voltage (limited by range)
<i>stop</i>	Final step value in voltage (limited by range)
<i>step</i>	Step size in voltage; cannot be 0
<i>vbase</i>	Base voltage for the pulse (starting voltage level; limited by range)

### Details

You must configure a sweep for another PMU channel before setting this command.

If the source range is set to 10 V, you can only step a maximum of 10 volts in a test. If the source range is set to 40 V, you can only step 40 volts in a test. For example, if the 10 V range is selected, you can set the start to –5 V and the stop value to 5 V.

For a detailed example that uses this command, refer to [Example: Use KXCI to generate a pulse I-V drain family of curves](#) (on page 7-35).

### Example

```
:PMU:STEP:PULSE:AMPLITUDE 1, 1, 5, 0.5, 0
```

Steps the amplitude on channel 1 starting at 1 V and going to 5 V, stepping in 0.5 V steps, with a base voltage of 0.

### Also see

[:PMU:PULSE:BURST:COUNT](#) (on page 7-17)

[:PMU:PULSE:TIMES](#) (on page 7-18)

[:PMU:SOURCE:RANGE](#) (on page 7-24)

[:PMU:STEP:DC](#) (on page 7-25)

[:PMU:STEP:PULSE:BASE](#) (on page 7-27)

[:PMU:SWEEP:DC](#) (on page 7-28)

## :PMU:STEP:PULSE:BASE

This command configures the pulse step for stepping the pulse voltage base level with a fixed voltage amplitude for the specified channel. Only available if the mode is set to pulse I-V.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	Not applicable

### Usage

```
:PMU:STEP:PULSE:BASE ch, start, stop, step, vamplitude
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>start</i>	Initial step value in voltage (limited by range)
<i>stop</i>	Final step value in voltage (limited by range)
<i>step</i>	Step size in voltage; cannot be 0
<i>vamplitude</i>	Amplitude of the pulse (peak voltage level; limited by range)

### Details

You must configure a sweep for another PMU channel before setting this command.

If the source range is set to 10 V, you can only step a maximum of 10 volts in a test. If the source range is set to 40 V, you can only step 40 volts in a test. For example, if the 10 V range is selected, you can set the start to -5 V and the stop value to 5 V.

### Example

```
:PMU:STEP:PULSE:BASE 1, 1, 5, 0.5, 1
```

Steps the base voltage on channel 1, starting at 1 V and going to 5 V in 0.5 V steps with a 1 V amplitude.

### Also see

[:PMU:SOURCE:RANGE](#) (on page 7-24)

[:PMU:STEP:DC](#) (on page 7-25)

[:PMU:STEP:PULSE:AMPLITUDE](#) (on page 7-26)

[:PMU:SWEEP:DC](#) (on page 7-28)



## :PMU:SWEEP:DC

This command configures the pulse sweep for sweeping the DC voltage level for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	Not applicable

### Usage

```
:PMU:SWEEP:DC ch, start, stop, step, dualSweep
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>start</i>	Initial sweep value in voltage (limited by range)
<i>stop</i>	Final sweep value in voltage (limited by range)
<i>step</i>	Step size in voltage; cannot be 0
<i>dualSweep</i>	Enable or disable dual sweep: <ul style="list-style-type: none"> <li>Disable dual sweep: 0</li> <li>Enable dual sweep: 1</li> </ul>

### Details

When you enable Dual Sweep, the instrument sweeps from start to stop, then from stop to start. When you disable Dual Sweep, the instrument sweeps from start to stop only.

If you are sweeping more than one channel, the sweep must have the same number of steps for each channel.

### Example

```
:PMU:SWEEP:DC 1, 0, 5, 0.5, 0
```

Sweep the DC voltage level on channel 1 starting at 0 V and stopping at 5 V in 0.5 V steps. Dual sweep is disabled.

### Also see

“Dual Sweep” in the *Model 4200A-SCS Pulse Card (PGU and PMU) User's Manual*

[:PMU:SOURCE:RANGE](#) (on page 7-24)

[:PMU:SWEEP:PULSE:AMPLITUDE](#) (on page 7-29)

[:PMU:SWEEP:PULSE:BASE](#) (on page 7-31)

# :PMU:SWEEP:PULSE:AMPLITUDE

This command configures the pulse sweep for sweeping the pulse voltage amplitude with a fixed base level for the specified channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

## Usage

:PMU:SWEEP:PULSE:AMPLITUDE *ch, start, stop, step, vbase, dualSweep*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>start</i>	Initial sweep value in voltage (limited by range)
<i>stop</i>	Final sweep value in voltage (limited by range)
<i>step</i>	Step size in voltage; cannot be 0
<i>vbase</i>	Base voltage for the pulse (starting voltage level; limited by range)
<i>dualSweep</i>	Enable or disable dual sweep: <ul style="list-style-type: none"><li>■ Disable dual sweep: 0</li><li>■ Enable dual sweep: 1</li></ul>

## Details

When you enable Dual Sweep, the instrument sweeps from start to stop, then from stop to start. When you disable Dual Sweep, the instrument sweeps from start to stop only.

If you are sweeping more than one channel, the sweep must have the same number of steps for each channel.

If the source range is set to 10 V, the initial sweep value and final sweep value must be within 10 volts of each other. If the source range is set to 40 V, they must be within 40 volts. For example, if the 10 V range is selected, you can set *start* to -5 V and *stop* to 5 V.

For a detailed example that uses this command, refer to [Example: Use KXCI to generate a pulse I-V drain family of curves](#) (on page 7-35).

---

**Example**

```
:PMU:INIT 0
:PMU:RPM:CONFIGURE PMU1-1, 0
:PMU:LOAD 1, 1e3
:PMU:MEASURE:MODE 1
:PMU:SWEEP:PULSE:AMPLITUDE 1, -5, 5, 0.1, 0, 0
:PMU:PULSE:TIMES 1, 10e-6, 5e-6, 5e-6, 1e-7, 0
:PMU:MEASURE:RANGE 1, 2, 10e-3
:PMU:TIMES:PIV 1, 0.75, 0.9
:PMU:OUTPUT:STATE 1, 1
:PMU:EXECUTE
```

This example creates a pulse I-V sweep. This example:

- Initializes the PMUs and sets them to standard pulse mode.
- Connects PMU 1, channel 1, to the RPM.
- Sets load impedance to 1000  $\Omega$ .
- Sets the measure mode to spot mean discrete.
- Sets the amplitude sweep parameters to start V = -5 V, stop V = 5 V, step size = 0.1, and base voltage = 0 V.
- Sets the pulse timing parameters to period = 10e-6 s, pulse width = 5e-6 s, and rise and fall times = 1e-7 s.
- Sets the measure range to a fixed range of 10 mA.
- Sets the spot mean measurement window for pulse I-V capture to 75% for the start and 90% for the stop location.
- Turns the PMU output on.
- Executes the test.

In this example, for each pulse, one reading is derived.

---

**Also see**

“Dual Sweep” in the *Model 4200A-SCS Pulse Card (PGU and PMU) User's Manual*

[:PMU:SWEEP:DC](#) (on page 7-28)

[:PMU:SWEEP:PULSE:BASE](#) (on page 7-31)

## :PMU:SWEEP:PULSE:BASE

This command configures the pulse sweep for sweeping the pulse voltage base level with a fixed voltage amplitude for the specified channel.

Instruments	Affected by	Default value
PGU PMU	*RST :PMU:INIT	Not applicable

### Usage

```
:PMU:SWEEP:PULSE:BASE ch, start, stop, step, vamplitude, dualSweep
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>start</i>	Initial sweep value in voltage (limited by range)
<i>stop</i>	Final sweep value in voltage (limited by range)
<i>step</i>	Step size in voltage
<i>vamplitude</i>	Amplitude of the pulse (the peak voltage level; limited by range)
<i>dualSweep</i>	Enable or disable dual sweep: <ul style="list-style-type: none"> <li>Disable dual sweep: 0</li> <li>Enable dual sweep: 1</li> </ul>

### Details

When you enable Dual Sweep, the instrument sweeps from start to stop, then from stop to start. When you disable Dual Sweep, the instrument sweeps from start to stop only.

If you are sweeping more than one channel, the sweep must have the same number of steps for each channel.

If the source range is set to 10 V, the initial sweep value and final sweep value must be within 10 volts of each other. If the source range is set to 40 V, they must be within 40 volts. For example, if the 10 V range is selected, you can set *start* to -5 V and *stop* to 5 V.

### Example

```
:PMU:SWEEP:PULSE:BASE 1, 1, 5, 0.5, 1, 0
```

Sweep the base on channel 1 starting at 1 V and stopping at 5 V, stepping in 0.5 V steps with a 1 V amplitude.

### Also see

"Dual Sweep" in the *Model 4200A-SCS Pulse Card (PGU and PMU) User's Manual*

[:PMU:SOURCE:RANGE](#) (on page 7-24)

[:PMU:SWEEP:PULSE:AMPLITUDE](#) (on page 7-29)

## :PMU:TEST:STATUS?

This command is used to determine if a test is running or idle.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

### Usage

```
:PMU:TEST:STATUS?
```

### Details

This command returns the status information on all pulse cards. It indicates whether a test is complete or idle (return of 0) or running (return of 1).

The primary use of this command is to ensure that the test is completed before fetching PMU data or ending the test.

### Example

```
:PMU:TEST:STATUS?
```

This command returns 1 if the test is still running or 0 if the test is idle.

### Also see

None

## :PMU:TIMES:PIV

This command sets the spot mean measurement window for pulse I-V capture for the specified channel.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT	StartPercent: 0.75 (75%) StopPercent: 0.90 (90%)

### Usage

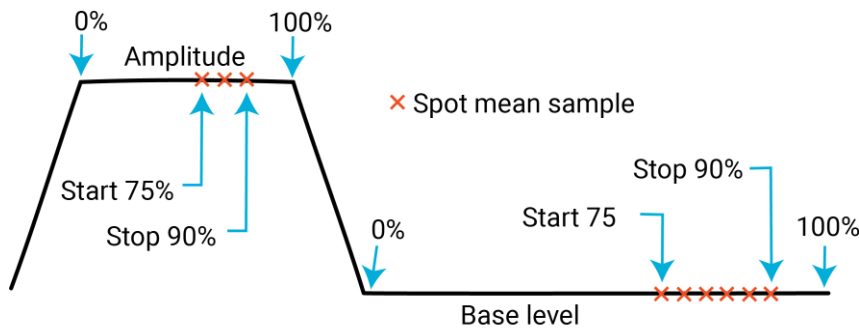
```
:PMU:PIV:TIMES ch, StartPercent, StopPercent
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>StartPercent</i>	The start location, specified as a percentage of the widths set for the amplitude and base level: 0 to 1
<i>StopPercent</i>	The stop location, specified as a percentage of the widths set for the amplitude and base level: 0 to 1

## Details

The following figure shows example start and stop locations when spot mean measurements are made. The start and stop percentage values indicate the portions of the pulse that are sampled.

**Figure 14: Spot mean measurements example**



## Example

```
:PMU:INIT 0
:PMU:RPM:CONFIGURE PMU1-1, 0
:PMU:LOAD 1, 1e3
:PMU:MEASURE:MODE 1
:PMU:SWEEP:PULSE:AMPLITUDE 1, -5, 5, 0.1, 0, 0
:PMU:PULSE:TIMES 1, 10e-6, 5e-6, 5e-6, 1e-7, 0
:PMU:MEASURE:RANGE 1, 2, 10e-3
:PMU:TIMES:PIV 1, 0.75, 0.9
:PMU:OUTPUT:STATE 1, 1
:PMU:EXECUTE
```

This example creates a pulse I-V sweep. This example:

- Initializes the PMUs and sets them to standard pulse mode.
- Connects PMU 1, channel 1, to the RPM.
- Sets load impedance to 1000  $\Omega$ .
- Sets the measure mode to spot mean discrete.
- Sets the amplitude sweep parameters to start V = -5 V, stop V = 5 V, step size = 0.1, and base voltage = 0 V.
- Sets the pulse timing parameters to period = 10e-6 s, pulse width = 5e-6 s, and rise and fall times = 1e-7 s.
- Sets the measure range to a fixed range of 10 mA.
- Sets the spot mean measurement window for pulse I-V capture to 75% for the start and 90% for the stop location.
- Turns the PMU output on.
- Executes the test.

In this example, for each pulse, one reading is derived.

## Also see

None

:PMU:TIMES:WAVEFORM

This command defines the predata and postdata measurement windows for waveform capture for the specified channel.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT	Prepercent: 0.2 Postpercent: 0.2

Usage

```
:PMU:TIMES:WAVEFORM ch, PrePercent, PostPercent
```

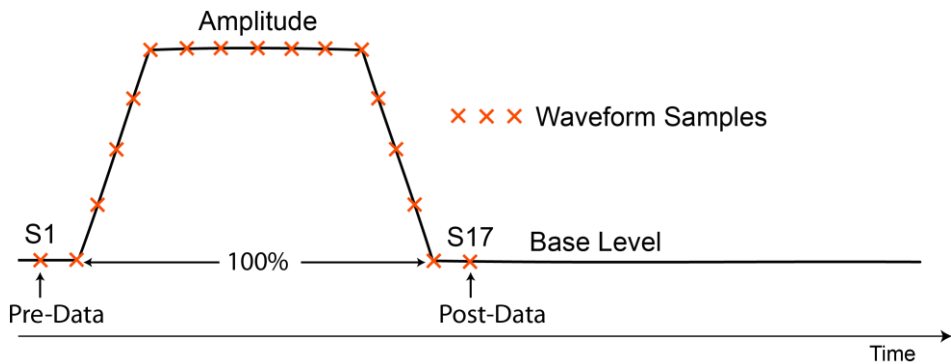
ch	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
PrePercent	Pre-data, specified as a percentage of the amplitude pulse duration: 0 to 1.0; refer to <b>Details</b>
PostPercent	Post-data, specified as a percentage of the amplitude pulse duration: 0 to 1.0

Details

This command defines the pre-data and post-data measurement windows for waveform capture mode. When these values are not set to 0, data is collected before and after the pulse. The pre-data interval can be less than or equal to the pulse delay (set by :PMU:PULSE:TIMES). If the pulse delay is 0, no pre-data is collected.

For waveform capture, measurement samples are acquired on pulse rise, pulse amplitude, pulse fall, and a small portion before the rise (pre-data) and after the fall (post-data) on the base level, as shown in the following figure.

Figure 15: Waveform capture



The amplitude pulse duration used for the percent calculation is  $width + 0.5 \times (rise + fall)$

This is the duration of the pulse when the amplitude is not 0.

Example

```
:PMU:TIMES:WAVEFORM 1, 0, 0.1
```

Sets the pre-data to get no data before the pulse and post-data to get 10% of data.

---

**Also see**

[:PMU:PULSE:TIMES](#) (on page 7-18)

## Programming examples

The following programming examples use the Python programming language to demonstrate how to assemble the KXCI PGU and PMU commands for testing.

To run these scripts, you need to download `Instrcomm.py` from the following link and store it in the same folder as your active Python file.

More detailed versions of these examples are available at [https://github.com/tektronix/keithley/tree/main/Instrument\\_Examples/Model\\_4200A/KXCI\\_Examples](https://github.com/tektronix/keithley/tree/main/Instrument_Examples/Model_4200A/KXCI_Examples).

### Example: Use KXCI to generate a pulse I-V drain family of curves

This example uses RPM1 and RPM2 of PMU1 to generate a pulse I-V drain family of curves. It creates a pulse amplitude step on channel 1 from 2 V to 4 V in 0.5 V steps and a pulse amplitude sweep on channel 2 from 0 V to 5 V in 0.2 V steps. This test is similar to the Clarius `pulse-vds-id` test.

Device used: 4-terminal nMOSFET in an 8101-PIV fixture with terminals connected to the top slot, using AC + DC IN 1 and AC + DC IN 2.

```
from instrcomms import Communications
import time

#Instrument resource string obtained from NI MAX.
INST_RESOURCE_STR = "TCPIP0::X.X.X.X::1225::SOCKET"

#Open the resource manager in PyVISA with the corresponding instrument resource
string.
my4200 = Communications(INST_RESOURCE_STR)

#Open connections to the 4200A-SCS.
my4200.connect()
#Set PyVISA write terminator.
my4200._instrument_object.write_termination = "\0"
#Set PyVISA read terminator.
my4200._instrument_object.read_termination = "\0"
#Initialize to standard pulse mode.
my4200.query(":PMU:INIT 0")

#RPM on, channel 1.
my4200.query(":PMU:RPM:CONFIGURE PMU1-1, 0")
#RPM on, channel 2.
my4200.query(":PMU:RPM:CONFIGURE PMU1-2, 0")
```



```
#Spot mean discrete measure mode.
my4200.query(":PMU:MEASURE:MODE 1")
#Channel 1, 10 mA fixed current range.
my4200.query(":PMU:MEASURE:RANGE 1, 2, 0.01")
#Channel 2, 200 mA fixed current range.
my4200.query(":PMU:MEASURE:RANGE 2, 2, 0.2")
#Load-line effect compensation on, channel 1.
my4200.query(":PMU:LLEC:CONFIGURE 1, 1")
#Amplitude step, 2 V to 4 V, 0.5 V steps. Dual sweep is optional and is disabled by
  default.
my4200.query(":PMU:STEP:PULSE:AMPLITUDE 1, 2, 4, 0.5, 0")
#Amplitude sweep, 0 to 5 V, 0.2 V steps. Dual sweep is disabled explicitly.
my4200.query(":PMU:SWEEP:PULSE:AMPLITUDE 2, 0, 5, 0.2, 0, 0")

#Set the output state of channel 1 to on.
my4200.query(":PMU:OUTPUT:STATE 1, 1")
#Set the output state of channel 2 to on.
my4200.query(":PMU:OUTPUT:STATE 2, 1")

#Execute the test.
my4200.query(":PMU:EXECUTE")

#Continue looping until the test is complete.
while True:
#Return 1 if it is running, 0 if it is complete.
    status = my4200.query(":PMU:TEST:STATUS?")
    if int(status) == 0:
        print("Measurement Complete.")
        break
#Print status to the terminal.
    print(f"Status: {status}")
#Sleep for 1 second between checks.
    time.sleep(1)

#Data points from channel 1.
print(my4200.query(":PMU:DATA:COUNT? 1"))
#Data points from channel 2.
print(my4200.query(":PMU:DATA:COUNT? 2"))
#All data from channel 1.
datachan1 = my4200.query(f":PMU:DATA:GET 1")
#Print the response to the console.
print("Response from 4200A channel 1: ", datachan1)
#All data from channel 2.
datachan2 = my4200.query(f":PMU:DATA:GET 2")
#Print the response to the console.
print("Response from 4200A channel 2: ", datachan2)

#Set output state of CH1 to off.
my4200.query(":PMU:OUTPUT:STATE 1, 0")
#Set output state of CH2 to off.
my4200.query(":PMU:OUTPUT:STATE 2, 0")

#Close communications with the 4200A-SCS.
my4200.disconnect()
```

## Example: Using KXCI to generate a pulse train

This example creates a 5 V amplitude pulse train on channel 1, output from a 4225-PMU with 4225-RPMs. It returns the current, voltage, time, and status measurements on channel 1 using waveform capture mode. This example also demonstrates how to retrieve more than 2048 points. Two channels are used, with channel 2 set to 0 V and the load set to 1 M $\Omega$ .

Device used: 1 M $\Omega$  oscilloscope impedance.

```
from instrcomms import Communications
import time

#Instrument resource string obtained from NI MAX.
INST_RESOURCE_STR = "TCPIP0::XXX.XX.XX.XXX::1225::SOCKET"
#Open the resource manager in PyVISA with the corresponding instrument resource
    string.
my4200 = Communications(INST_RESOURCE_STR)
#Open connections to the 4200A-SCS.
my4200.connect()
#Set PyVISA write terminator.
my4200._instrument_object.write_termination = "\0"
#Set PyVISA read terminator.
my4200._instrument_object.read_termination = "\0"
#Initialize to standard pulse mode.
my4200.query(":PMU:INIT 0")
#Configure the RPM.
my4200.query(":PMU:RPM:CONFIGURE PMU1-1, 0")
#Set to discrete timing, waveform capture.
my4200.query(":PMU:MEASURE:MODE 2")
#Pulse train with 0 V base, 5 V amplitude.
my4200.query(":PMU:PULSE:TRAIN 1, 0.0, 5.0")

#Set pulse timing to a period of 100 us, 50 us width, and 5 us rise and fall times.
my4200.query(":PMU:PULSE:TIMES 1, 100e-6, 50e-6, 5e-6, 5e-6")
#Set output state to on.
my4200.query(":PMU:OUTPUT:STATE 1, 1")
#Execute the test.
my4200.query(":PMU:EXECUTE")

#Continue looping until the test is complete.
while True:
#Return 1 if it is running; 0 if it is complete.
    status = my4200.query(":PMU:TEST:STATUS?")
    if int(status) == 0:
        print("Measurement Complete.")
        break
#Print status to the terminal.
    print(f"Status: {status}")
#Sleep for 1 second between checks.
    time.sleep(1)
```

```

#Ask for the number of data points.
count = my4200.query(":PMU:DATA:COUNT? 1")

#Print out the number of data points.
print(f"Data count for channel 1: {count}")

data_points = int(count)
#Loop through the data points in chunks.
coords = []
for start_point in range(0, data_points, 2048):
    #Get data for the current chunk.
    response = my4200.query(f":PMU:DATA:GET 1, {start_point}, 2048")

    #Split the response using semicolon as the delimiter for each point.
    coords += response.split(";")

#Split into 4 separate values for each point.
coords2d = [value.split(",") for value in coords]
print(f"Retrieved {len(coords)} data points")

#Print the first data point
first_point = coords[0]
print(f"First data point: {first_point}")
first_point_values = first_point.split(",")

#Also print the voltage and current measurements separately.
print(f"Voltage: {first_point_values[0]}, Current: {first_point_values[1]}")
print("Voltage values:")
#Print 2048 voltage values.
print(my4200.query(":PMU:DATA:GET 1, 0, 2048, VH"))
print("Current values:")
#Print 2048 current values.
print(my4200.query(":PMU:DATA:GET 1, 0, 2048, IH"))
#Set the output state to off.
my4200.query(":PMU:OUTPUT:STATE 1, 0")
#Close communications with the 4200A-SCS.
my4200.disconnect()

```

This example produces output similar to the following:

```

Status: 1
Measurement Complete.
Total data points for channel 1: 3300
Retrieved 3300 data points
First data point: 0.00788628,-21.7246e-9,0,01020021
Voltage: 0.00788628, Current: -21.7246e-9

```

## Summary of Segment Arb commands

The following command descriptions are for the KXCI PGU and PMU commands that are available in Clarius+ versions 1.13 and later. These system commands support the Segment Arb™ waveforms. They do not support full-arb pulse mode.

To use Segment Arb, at a minimum, you must set `:PMU:SARB:SEQ:STARTV`, `:PMU:SARB:SEQ:STOPV`, and `:PMU:SARB:SEQ:TIME` for at least three sequences. Every `:START` command must have a corresponding `:STOP` command.

### NOTE

For more information, refer to “Segment Arb waveform” in the *Model 4200A-SCS Pulse Card (PGU and PMU) User's Manual*.

Command	Brief description
<a href="#">:PMU:SARB:SEQ:MEAS:START</a> (on page 7-40)	Defines an array of start measurement times for a Segment Arb waveform pulse-measure sequence on the specified channel.
<a href="#">:PMU:SARB:SEQ:MEAS:START:ADD</a> (on page 7-41)	Adds an array of start measurement times for a Segment Arb waveform pulse-measure on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:MEAS:STOP</a> (on page 7-42)	Defines an array of stop measurement times for a Segment Arb waveform pulse-measure sequence on the specified channel.
<a href="#">:PMU:SARB:SEQ:MEAS:STOP:ADD</a> (on page 7-43)	Adds an additional array of stop measurement times for a Segment Arb waveform pulse-measure sequence on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:MEAS:TYPE</a> (on page 7-44)	Defines an array of measure types for a Segment Arb waveform pulse-measure sequence for the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:MEAS:TYPE:ADD</a> (on page 7-45)	Adds an array of measure types for a Segment Arb waveform pulse-measure sequence on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:SSR</a> (on page 7-46)	Defines an array of values to control the high endurance output relay (HEOR) for a Segment Arb sequence on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:SSR:ADD</a> (on page 7-47)	Adds an array of values to control the high endurance output relay (HEOR) for a Segment Arb sequence on the specified channel.
<a href="#">:PMU:SARB:SEQ:STARTV</a> (on page 7-48)	Defines an array of start voltage levels for a Segment Arb sequence on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:STARTV:ADD</a> (on page 7-49)	Adds an array of start voltage levels to a Segment Arb sequence on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:STOPV</a> (on page 7-50)	Defines an array of stop voltage levels for a Segment Arb sequence on the specified pulse channel.

Command	Brief description
<a href="#">:PMU:SARB:SEQ:STOPV:ADD</a> (on page 7-51)	Adds an array of stop voltage levels to a Segment Arb waveform sequence for the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:TIME</a> (on page 7-52)	Defines an array of segment times for a Segment Arb waveform sequence on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:TIME:ADD</a> (on page 7-53)	Adds an array of segment times for a Segment Arb sequence on the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:TRIG</a> (on page 7-54)	Defines an array of trigger values for the trigger output of a Segment Arb sequence for the specified pulse channel.
<a href="#">:PMU:SARB:SEQ:TRIG:ADD</a> (on page 7-55)	Array of trigger values for the trigger output of a Segment Arb for the specified pulse channel.
<a href="#">:PMU:SARB:WFM:SEQ:LIST</a> (on page 7-56)	Defines the Segment Arb waveform pulse-measure sequence list for the specified pulse channel.
<a href="#">:PMU:SARB:WFM:SEQ:LIST:ADD</a> (on page 7-56)	Adds values to the Segment Arb waveform pulse-measure sequence list for the specified pulse channel.

## :PMU:SARB:SEQ:MEAS:START

This command defines an array of start measurement times for a Segment Arb waveform pulse-measure sequence on the specified channel.

Instruments	Affected by	Default value
PMU	:PMU:SARB:SEQ:MEAS:STOP	0

### Usage

```
:PMU:SARB:SEQ:MEAS:START ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Start measurement time in seconds with 10 ns resolution
<i>val2</i>	Start measurement time in seconds with 10 ns resolution
<i>valN</i>	Start measurement time in seconds with 10 ns resolution; <i>N</i> is limited to 128

### Details

To start measuring at the beginning of a segment, set the start time to 0. If you do not send this command, the start time is set to 0 for all segments.

The segment value in this array cannot be higher than corresponding value in the :PMU:SARB:SEQ:MEAS:STOP array or the corresponding value in the :PMU:SARB:SEQ:TIME array.

If you call this command more than once, the previous values are overwritten.

If you need to send more than 128 values, you can append additional values by sending the :PMU:SARB:SEQ:MEAS:START:ADD command with the same sequence number.

If `:PMU:SARB:SEQ:MEAS:TYPE` is set to 0 (no measurements),  
`:PMU:SARB:SEQ:MEAS:START` has no effect.

For a detailed example that uses this command, refer to [Programming example for Segment Arb](#) (on page 7-57).

### Example

```
:PMU:SARB:SEQ:MEAS:START 1, 1, 0, 0, 0, 0, 0
```

Defines an array of start measurement times for sequence 1 on channel 1.

### Also see

[:PMU:SARB:SEQ:MEAS:START:ADD](#) (on page 7-41)

[:PMU:SARB:SEQ:MEAS:STOP](#) (on page 7-42)

[:PMU:SARB:SEQ:MEAS:TYPE](#) (on page 7-44)

[:PMU:SARB:SEQ:TIME](#) (on page 7-52)

## :PMU:SARB:SEQ:MEAS:START:ADD

This command adds an array of start measurement times for a Segment Arb waveform pulse-measure on the specified pulse channel.

Instruments	Affected by	Default value
PMU	:PMU:SARB:SEQ:MEAS:STOP	0

### Usage

```
:PMU:SARB:SEQ:MEAS:START:ADD ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Start measurement time in seconds with 10 ns resolution
<i>val2</i>	Start measurement time in seconds with 10 ns resolution
<i>valN</i>	Start measurement time in seconds with 10 ns resolution; <i>N</i> is limited to 128

### Details

This command allows you to send more than 128 start measurement times using the same sequence number.

To start measuring at the beginning of a segment, set the start time to 0.

This command appends values to the sequence list.

If `:PMU:SARB:SEQ:MEAS:TYPE` is set to 0 (no measurements),  
`:PMU:SARB:SEQ:MEAS:START` has no effect.

### Example

```
:PMU:SARB:SEQ:MEAS:START:ADD 1, 1, 0, 0, 0, 0, 0
```

Adds an array of start measurement times to sequence 1 on channel 1.

**Also see**

[:PMU:SARB:SEQ:MEAS:START](#) (on page 7-40)

[:PMU:SARB:SEQ:MEAS:STOP](#) (on page 7-42)

[:PMU:SARB:SEQ:MEAS:TYPE](#) (on page 7-44)

[:PMU:SARB:SEQ:TIME](#) (on page 7-52)

**:PMU:SARB:SEQ:MEAS:STOP**

This command defines an array of stop measurement times for a Segment Arb waveform pulse-measure sequence on the specified channel.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT :PMU:SARB:SEQ:MEAS:START	Segment length set by :PMU:SARB:SEQ:TIME

**Usage**

```
:PMU:SARB:SEQ:MEAS:STOP ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Stop measurement time in seconds with 10 ns resolution
<i>val2</i>	Stop measurement time in seconds with 10 ns resolution
<i>valN</i>	Stop measurement time in seconds with 10 ns resolution; <i>N</i> is limited to 128

**Details**

This is the elapsed time within the segment when the measurement stops.

The measurement stop time must be after the measurement start time unless the start and stop time are both 0. The measurement stop time must be less than or equal to the time of the segment specified by the `:PMU:SARB:SEQ:TIME` command.

If you call this command more than once, the previous values are overwritten.

For a detailed example that uses this command, refer to [Programming example for Segment Arb](#) (on page 7-57).

**Example**

```
:PMU:SARB:SEQ:MEAS:STOP 1, 1, 1e-5, 1e-6, 1e-5, 1e-6, 1e-6
```

Creates an array of stop measurement times for sequence 1 on channel 1.

**Also see**

[:PMU:SARB:SEQ:MEAS:START](#) (on page 7-40)

[:PMU:SARB:SEQ:MEAS:STOP:ADD](#) (on page 7-43)

[:PMU:SARB:SEQ:MEAS:TYPE](#) (on page 7-44)

[:PMU:SARB:SEQ:TIME](#) (on page 7-52)

# :PMU:SARB:SEQ:MEAS:STOP:ADD

This command adds an additional array of stop measurement times for a Segment Arb waveform pulse-measure sequence on the specified pulse channel.

Instruments	Affected by	Default value
PMU	*RST :PMU:INIT :PMU:SARB:SEQ:MEAS:START	Segment length set by :PMU:SARB:SEQ:TIME

## Usage

:PMU:SARB:SEQ:MEAS:STOP:ADD *ch*, *seq*, *val1*, *val2*, ..., *valN*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Stop measurement time in seconds with 10 ns resolution
<i>val2</i>	Stop measurement time in seconds with 10 ns resolution
<i>valN</i>	Stop measurement time in seconds with 10 ns resolution; <i>N</i> is limited to 128

## Details

This is the elapsed time within the segment when the measurement stops.

The measurement stop time must be after the measurement start time unless the start and stop time are both 0. The measurement stop time must also be less than or equal to the time of the segment specified by the :PMU:SARB:SEQ:TIME command.

This command appends values to the sequence list.

## Example

PMU:SARB:SEQ:MEAS:STOP 1, 1, 1e-5, 1e-6
Adds an array of stop measurement times to sequence 1 on channel 1.

## Also see

- [:PMU:SARB:SEQ:MEAS:START](#) (on page 7-40)
- [:PMU:SARB:SEQ:MEAS:STOP:ADD](#) (on page 7-43)
- [:PMU:SARB:SEQ:MEAS:TYPE](#) (on page 7-44)
- [:PMU:SARB:SEQ:TIME](#) (on page 7-52)



# :PMU:SARB:SEQ:MEAS:TYPE

This command defines an array of measure types for a Segment Arb waveform pulse-measure sequence for the specified pulse channel.

Instruments	Affected by	Default value
PMU	Not applicable	0 (no measurements)

## Usage

:PMU:SARB:SEQ:MEAS:TYPE *ch*, *seq*, *val1*, *val2*, ..., *valN*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	An array of measure modes: <ul style="list-style-type: none"><li>■ No measurements for this segment: 0</li><li>■ Spot mean discrete: 1</li><li>■ Waveform discrete: 2</li><li>■ Spot mean average: 3</li><li>■ Waveform average: 4</li></ul>
<i>val2</i>	An array of measure modes: <ul style="list-style-type: none"><li>■ No measurements for this segment: 0</li><li>■ Spot mean discrete: 1</li><li>■ Waveform discrete: 2</li><li>■ Spot mean average: 3</li><li>■ Waveform average: 4</li></ul>
<i>valN</i>	A comma-separated list of measure mode values; <i>N</i> is limited to 128

## Details

If you call this command more than once, the previous values are overwritten.

Each segment can use a different type of measurement.

For a detailed example that uses this command, refer to [Programming example for Segment Arb](#) (on page 7-57).

## Example

:PMU:SARB:SEQ:MEAS:TYPE 1, 1, 2, 0, 2, 0, 3
Defines the measure types for channel 1, sequence 1 to be waveform discrete, no measurements, waveform discrete, no measurements, then spot mean average.

## Also see

[:PMU:SARB:SEQ:MEAS:START](#) (on page 7-40)

[:PMU:SARB:SEQ:MEAS:STOP](#) (on page 7-42)

[:PMU:SARB:SEQ:MEAS:TYPE:ADD](#) (on page 7-45)

:PMU:SARB:SEQ:MEAS:TYPE:ADD

This command adds an array of measure types for a Segment Arb waveform pulse-measure sequence on the specified pulse channel.

Instruments	Affected by	Default value
PMU	Not applicable	Not applicable

Usage

```
:PMU:SARB:SEQ:MEAS:TYPE ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	An array of measure modes: <ul style="list-style-type: none"><li>■ No measurements for this segment: 0</li><li>■ Spot mean discrete: 1</li><li>■ Waveform discrete: 2</li><li>■ Spot mean average: 3</li><li>■ Waveform average: 4</li></ul>
<i>val2</i>	An array of measure modes: <ul style="list-style-type: none"><li>■ No measurements for this segment: 0</li><li>■ Spot mean discrete: 1</li><li>■ Waveform discrete: 2</li><li>■ Spot mean average: 3</li><li>■ Waveform average: 4</li></ul>
<i>valN</i>	A comma-separated list of measure mode values; <i>N</i> is limited to 128

Details

This command appends values to the sequence list. Each segment can use a different type of measurement.

Example

:PMU:SARB:SEQ:MEAS:TYPE 1, 1, 2, 0, 2, 0, 3

Adds the measure types waveform discrete, no measurements, waveform discrete, no measurements, then spot mean average for channel 1, sequence 1 .

Also see

[:PMU:SARB:SEQ:MEAS:TYPE](#) (on page 7-44)

:PMU:SARB:SEQ:SSR

This command defines an array of values to control the high endurance output relay (HEOR) for a Segment Arb sequence on the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	1 (closed)

Usage

```
:PMU:SARB:SEQ:SSR ch, seq, val1, val2, ..., valN
```

ch	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
seq	Segment Arb sequence number: 1 to 512
val1	Open: 0 Closed: 1
val2	Open): 0 Closed: 1
valN	Open: 0 Closed: 1 N is limited to 128

Details

If you call this command more than once, the previous values are overwritten.

Each output channel of a pulse card has a high-speed, solid-state output relay. When this relay is closed, the waveform segment is output. When opened, the channel output is electrically isolated (floating) from the DUT. The minimum time for a segment with a relay transition (open-to-close or close-to-open) is 25  $\mu$ s.

For more information on the output relay, refer to “PMU block diagram” in the *Model 4200A-SCS Pulse Card (PGU and PMU) User’s Manual*.

Example

```
:PMU:SARB:SEQ:SSR 1, 2, 1, 1, 1, 1, 1
```

Defines an array of relay settings that closes the relay for all segments of sequence 2 on channel 1.

Also see

[:PMU:SARB:SEQ:SSR:ADD](#) (on page 7-47)

**:PMU:SARB:SEQ:SSR:ADD**

This command adds an array of values to control the high endurance output relay (HEOR) for a Segment Arb sequence on the specified channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	1 (closed)

**Usage**

:PMU:SARB:SEQ:SSR:ADD *ch*, *seq*, *val1*, *val2*, ..., *valN*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Open: 0 Closed: 1
<i>val2</i>	Open: 0 Closed: 1
<i>valN</i>	Open: 0 Closed: 1 <i>N</i> is limited to 128

**Details**

This command appends values to the sequence list.

**Example**

:PMU:SARB:SEQ:SSR:ADD 1, 2, 1, 1, 1, 1, 1

Adds an array of relay settings that close the relay for all segments in sequence 1 on channel 1.

**Also see**

[:PMU:SARB:SEQ:SSR](#) (on page 7-46)

# :PMU:SARB:SEQ:STARTV

This command defines an array of start voltage levels for a Segment Arb sequence on the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	:PMU:SOURCE:RANGE	Not applicable

## Usage

:PMU:SARB:SEQ:STARTV *ch*, *seq*, *val1*, *val2*, ..., *valN*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>val2</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>valN</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range); <i>N</i> is limited to 128

## Details

The voltages of all segment transitions in a sequence must be seamless. Seamless means that there are no voltage differences: The stop voltage in a segment must equal the start voltage of the next segment.

If you call this command more than once, the previous values are overwritten.

## Example

```
:PMU:SARB:SEQ:STARTV 1, 1, 0, 0.1, 0.2, 0.3, 0.4
```

Defines an array of start voltage levels for sequence 1 of channel 1.

## Also see

- [:PMU:SARB:SEQ:STARTV:ADD](#) (on page 7-49)
- [:PMU:SARB:SEQ:STOPV](#) (on page 7-50)
- [:PMU:SOURCE:RANGE](#) (on page 7-24)

# :PMU:SARB:SEQ:STARTV:ADD

This command adds an array of start voltage levels to a Segment Arb sequence on the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	:PMU:SOURCE:RANGE	Not applicable

## Usage

:PMU:SARB:SEQ:STARTV:ADD *ch*, *seq*, *val1*, *val2*, ..., *valN*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>val2</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>valN</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range); <i>N</i> is limited to 128

## Details

The voltages of all segment transitions in a sequence must be seamless. Seamless means that there are no voltage differences: The stop voltage in a segment must equal the start voltage of the next segment.

This command appends values to the sequence list.

## Example

```
:PMU:SARB:SEQ:STARTV:ADD 1, 1, 0, 0.1, 0.2, 0.3, 0.4
```

Adds an array of start voltage levels for sequence 1 of channel 1.

## Also see

[:PMU:SARB:SEQ:STARTV](#) (on page 7-48)

[:PMU:SARB:SEQ:STOPV](#) (on page 7-50)

[:PMU:SOURCE:RANGE](#) (on page 7-24)

## :PMU:SARB:SEQ:STOPV

This command defines an array of stop voltage levels for a Segment Arb sequence on the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

### Usage

```
:PMU:SARB:SEQ:STOPV ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>val2</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>valN</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range); <i>N</i> is limited to 128

### Details

The voltages of all segment transitions in a sequence must be seamless. Seamless means that there are no voltage differences: The stop voltage in a segment must equal the start voltage of the next segment.

If you call this command more than once, the previous values are overwritten.

For a detailed example that uses this command, refer to [Programming example for Segment Arb](#) (on page 7-57).

### Example

```
:PMU:SARB:SEQ:STOPV 1, 1, 0.1, 0.2, 0.3, 0.4, 0
```

Defines an array of stop voltages for sequence 1 of channel 1.

### Also see

[:PMU:SARB:SEQ:STARTV](#) (on page 7-48)

[:PMU:SARB:SEQ:STOPV:ADD](#) (on page 7-51)

[:PMU:SOURCE:RANGE](#) (on page 7-24)

# :PMU:SARB:SEQ:STOPV:ADD

This command adds an array of stop voltage levels to a Segment Arb waveform sequence for the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

## Usage

:PMU:SARB:SEQ:STOPV:ADD *ch*, *seq*, *val1*, *val2*, ..., *valN*

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>val2</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range)
<i>valN</i>	Voltage level; an array of double values in voltage: -40 to 40 or -10 to 10 (limited by source range); <i>N</i> is limited to 128

## Details

The voltages of all segment transitions in a sequence must be seamless. Seamless means that there are no voltage differences: The stop voltage in a segment must equal the start voltage of the next segment.

This command appends values to the sequence list.

## Example

```
:PMU:SARB:SEQ:STOPV:ADD 1, 1, 0.1, 0.2, 0.3, 0.4, 0
```

Adds an array of stop voltages to sequence 1 of channel 1.

## Also see

[:PMU:SARB:SEQ:STARTV](#) (on page 7-48)

[:PMU:SARB:SEQ:STOPV](#) (on page 7-50)

[:PMU:SOURCE:RANGE](#) (on page 7-24)



## :PMU:SARB:SEQ:TIME

This command defines an array of segment times for a Segment Arb waveform sequence on the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

### Usage

```
:PMU:SARB:SEQ:TIME ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Segment time; an array of double values in seconds with 10 ns resolution: 0.00000002 to 1
<i>val2</i>	Segment time; an array of double values in seconds with 10 ns resolution; 20 ns minimum
<i>valN</i>	Segment time; an array of double values in seconds with 10 ns resolution; 20 ns minimum; <i>N</i> is limited to 128 Segment time limits in seconds: <ul style="list-style-type: none"> <li>PGU 10 V: 10e-9 to 1.0 (10 ns to 1 s)</li> <li>PMU 10 V: 20e-9 to 1.0 (20 ns to 1 s)</li> <li>40 V: 50e-9 to 1.0 (50 ns to 1 s)</li> </ul>

### Details

If you call this command more than once, the previous values are overwritten.

For a detailed example that uses this command, refer to [Programming example for Segment Arb](#) (on page 7-57).

### Example

```
:PMU:SARB:SEQ:TIME 1, 1, 50e-9, 100e-9, 20e-9, 150e-9, 50e-9, 500e-9, 130e-9
```

Defines an array of segment times for sequence 1 on channel 1.

### Also see

[:PMU:SARB:SEQ:MEAS:START](#) (on page 7-40)

[:PMU:SARB:SEQ:MEAS:STOP](#) (on page 7-42)

[:PMU:SARB:SEQ:STARTV](#) (on page 7-48)

[:PMU:SARB:SEQ:STOPV](#) (on page 7-50)

[:PMU:SARB:SEQ:TIME:ADD](#) (on page 7-53)

# :PMU:SARB:SEQ:TIME:ADD

This command adds an array of segment times for a Segment Arb sequence on the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

## Usage

```
:PMU:SARB:SEQ:TIME:ADD ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	Segment time; an array of double values in seconds with 10 ns resolution: 0.00000002 to 1
<i>val2</i>	Segment time; an array of double values in seconds with 10 ns resolution; 20 ns minimum
<i>valN</i>	Segment time; an array of double values in seconds with 10 ns resolution; 20 ns minimum; <i>N</i> is limited to 128

## Details

This command appends values to the sequence list.

## Example

:PMU:SARB:SEQ:TIME 1, 1, 50e-9, 100e-9, 20e-9, 150e-9, 50e-9, 500e-9, 130e-9
Adds an array of segment times to sequence 1 on channel 1.

## Also see

- [:PMU:SARB:SEQ:STARTV](#) (on page 7-48)
- [:PMU:SARB:SEQ:STOPV](#) (on page 7-50)
- [:PMU:SARB:SEQ:TIME](#) (on page 7-52)

# **:PMU:SARB:SEQ:TRIG**

This command defines an array of trigger values for the trigger output of a Segment Arb sequence for the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	1 (trigger high)

## **Usage**

```
:PMU:SARB:SEQ:TRIG ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	An array of trigger values (for trigger output only): <ul style="list-style-type: none"><li>▪ Trigger low: 0</li><li>▪ Trigger high: 1</li></ul>
<i>val2</i>	An array of trigger values (for trigger output only): <ul style="list-style-type: none"><li>▪ Trigger low: 0</li><li>▪ Trigger high: 1</li></ul>
<i>valN</i>	An array of trigger values (for trigger output only): <ul style="list-style-type: none"><li>▪ Trigger low: 0</li><li>▪ Trigger high: 1</li></ul> <i>N</i> is limited to 128

## **Details**

If you call this command more than once, the previous values are overwritten.

## **Example**

```
:PMU:SARB:SEQ:TRIG 1, 2, 1, 1, 1, 1, 1
```

Defines an array of trigger settings that sets the trigger high for all segments of sequence 2 on channel 1.

## **Also see**

[:PMU:SARB:SEQ:TRIG:ADD](#) (on page 7-55)

:PMU:SARB:SEQ:TRIG:ADD

This command adds an array of trigger values for the trigger output of a Segment Arb for the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

Usage

```
:PMU:SARB:SEQ:TRIG:ADD ch, seq, val1, val2, ..., valN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq</i>	Segment Arb sequence number: 1 to 512
<i>val1</i>	An array of trigger values (for trigger output only): <ul style="list-style-type: none"><li>Trigger low: 0</li><li>Trigger high: 1</li></ul>
<i>val2</i>	An array of trigger values (for trigger output only): <ul style="list-style-type: none"><li>Trigger low: 0</li><li>Trigger high: 1</li></ul>
<i>valN</i>	An array of trigger values (for trigger output only): <ul style="list-style-type: none"><li>Trigger low: 0</li><li>Trigger high: 1</li></ul> <i>N</i> is limited to 128

Details

This command appends values to the sequence list.

Example

```
:PMU:SARB:SEQ:TRIG:ADD 1, 2, 1, 1, 1, 1, 1
```

Adds an array of trigger settings that sets the trigger high for all segments of sequence 2 on channel 1.

Also see

[:PMU:SARB:SEQ:TRIG](#) (on page 7-54)

## :PMU:SARB:WFM:SEQ:LIST

This command defines the Segment Arb waveform pulse-measure sequence list for the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

### Usage

```
:PMU:SARB:WFM:SEQ:LIST ch, seq1, loop1, ..., seqN, loopN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq1</i>	Segment Arb sequence number: 1 to 512; must already be defined
<i>loop1</i>	Number of times to output a sequence; loop value range is 1 to 1E12
<i>seqN</i>	Segment Arb sequence number: 1 to 512; must already be defined
<i>loopN</i>	Number of times to output a sequence; loop value range is 1 to 1E12

### Details

:PMU:SARB:WFM:SEQ:LIST requires that the sequence number already exists. You must create the sequences before you can use them in a sequence list.

If you call this command more than once, the previous values are overwritten.

For a detailed example that uses this command, refer to [Programming example for Segment Arb](#) (on page 7-57).

### Example

```
:PMU:SARB:WFM:SEQ:LIST 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 3
```

Defines a waveform sequence list for sequences 1 and 2 of channel 1.

### Also see

[:PMU:SARB:WFM:SEQ:LIST:ADD](#) (on page 7-56)

## :PMU:SARB:WFM:SEQ:LIST:ADD

This command adds values to the Segment Arb waveform pulse-measure sequence list for the specified pulse channel.

Instruments	Affected by	Default value
PGU PMU	Not applicable	Not applicable

**Usage**

```
:PMU:SARB:WFM:SEQ:LIST:ADD ch, seq1, loop1, ... seqN, loopN
```

<i>ch</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>seq1</i>	Segment Arb sequence number: 1 to 512
<i>loop1</i>	Number of times to output a sequence; loop value range is 1 to 1E12
<i>seqN</i>	Segment Arb sequence number: 1 to 512
<i>loopN</i>	Number of times to output a sequence; loop value range is 1 to 1E12

**Details**

This command appends values to the sequence list.

**Example**

```
:PMU:SARB:WFM:SEQ:LIST 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 3
```

Adds a waveform sequence list for sequences 1 and 2 of channel 1.

**Also see**

[:PMU:SARB:WFM:SEQ:LIST](#) (on page 7-56)

## Programming example for Segment Arb

The following programming example uses the Python programming language to demonstrate how to assemble the KXCI PGU and PMU commands to generate Segment Arb waveforms using a 4225-PMU with 4225-RPMs. This test outputs a 1 V sequence with configured rise and fall times on channel 1.

To run this script, you need to download `Instrcomm.py` from the following link and store it in the same folder as your active Python file.

A more detailed version of this example code is available at

[https://github.com/tektronix/keithley/tree/main/Instrument\\_Examples/Model\\_4200A/KXCI\\_Examples](https://github.com/tektronix/keithley/tree/main/Instrument_Examples/Model_4200A/KXCI_Examples).

```
from instrcomms import Communications
import time

#Instrument resource string obtained from NI MAX.
INST_RESOURCE_STR = "TCPIP0::XXX.XX.XX.XXX::1225::SOCKET"

#Open the resource manager in PyVISA with the corresponding instrument resource
string.
my4200 = Communications(INST_RESOURCE_STR)
#Open connections to the 4200A-SCS.
my4200.connect()
#Set PyVISA write terminator.
my4200._instrument_object.write_termination = "\0"
#Set PyVISA read terminator.
my4200._instrument_object.read_termination = "\0"
```

```

#Initialize to Segment ARB mode.
my4200.query(":PMU:INIT 1")
#Configure the RPM input for channel 1.
my4200.query(":PMU:RPM:CONFIGURE PMU1-1, 0")
#Set to a 200 mA fixed current range.
my4200.query(":PMU:MEASURE:RANGE 1, 2, 0.2")
#Array of starting voltages.
my4200.query(":PMU:SARB:SEQ:STARTV 1, 1, 0, 1, 1, 1.5, 1.5, 0, 0")
#Array of stopping voltages.
my4200.query(":PMU:SARB:SEQ:STOPV 1, 1, 1, 1, 1.5, 1.5, 0, 0, 0")
#Array of sequence times.
my4200.query(":PMU:SARB:SEQ:TIME 1, 1, 50e-9, 100e-9, 20e-9, 150e-9, 50e-9, 500e-9, 130e-9")
#Array of measure types, 1 = pulse IV.
my4200.query(":PMU:SARB:SEQ:MEAS:TYPE 1, 1, 0, 1, 0, 1, 0, 0, 0")
#Array of start measure times.
my4200.query(":PMU:SARB:SEQ:MEAS:START 1, 1, 0, 25e-9, 0, 50e-9, 0, 0, 0")
#Array of stop measure times.
my4200.query(":PMU:SARB:SEQ:MEAS:STOP 1, 1, 0, 75e-9, 0, 100e-9, 0, 0, 0")
#Sequence list, loop 1 time.
my4200.query(":PMU:SARB:WFM:SEQ:LIST 1, 1, 1")
#Turn on the output.
my4200.query(":PMU:OUTPUT:STATE 1, 1")

#Execute the test.
my4200.query(":PMU:EXECUTE")
#Continue looping until the test is complete.
while True:
#Return 1 if it is running, 0 if it is complete.
    status = my4200.query(":PMU:TEST:STATUS?")
    if int(status) == 0:
        print("Measurement Complete.")
        break
#Print the status to the terminal.
    print(f"Status: {status}")
#Sleep for 1 second in between checks.
    time.sleep(1)

#Get the total number of points.
print(my4200.query(":PMU:DATA:COUNT? 1"))
#Get all data from channel 1.
response = my4200.query(":PMU:DATA:GET 1")
#Print the response to the console.
print("Response from 4200A: ", response)
#Turn off the output.
my4200.query(":PMU:OUTPUT:STATE 1, 0")
#Close communications with the 4200A-SCS.
my4200.disconnect()

```

## KXCI pulse commands for Clarius versions v1.12 and earlier

The following command descriptions are for the KXCI commands that were available in Clarius+ versions 1.12 and earlier. These commands provide pulse control for PGUs and PMUs, but they do not include measurement functionality.

---

### NOTE

These commands are available in versions 1.13 and later, but cannot be used with the updated commands. Links to the equivalent updated commands are provided in the description of each command to help you convert code for use with the updated commands.

---

There are no two-letter KXCI commands for pulse with measure. To control pulse with measure remotely, you can call Keithley User Library Tool (KULT) user libraries in KXCI. Refer to [Calling KULT user libraries remotely](#) (on page 4-6) for details. You can also use the PGU and PMU KXCI commands introduced in Clarius+ version 1.13, listed in [Summary of KXCI PGU and PMU commands](#) (on page 7-3).

The following documentation includes the corresponding Linear Parametric Test Library (LPT) functions for each KXCI command string. Refer to “LPT commands for PGUs and PMUs” in *Model 4200A-SCS LPT Library Programming* for details on pulsing functions.



## Summary of KXCI PGU commands

This is a summary of the KXCI pulse commands that available in versions before 1.12.

Command	Brief description
<a href="#">PA</a> (on page 7-60)	Aborts the process that is presently running on the specified pulse channel.
<a href="#">PD</a> (on page 7-61)	Sets the device under test (DUT) impedance (pulse load) for the specified channel.
<a href="#">PE</a> (on page 7-61)	Starts a configured pulse test.
<a href="#">PG</a> (on page 7-62)	Sets the trigger mode and either initiates the start of pulse output or arms the pulse card for the specified channel.
<a href="#">PH</a> (on page 7-63)	Stops all pulse output from the pulse card.
<a href="#">PN</a> (on page 7-63)	Configures the pulse of the pulse channel.
<a href="#">PO</a> (on page 7-64)	Sets the output state (on or off) and mode (normal or complement) independently for each channel.
<a href="#">PS</a> (on page 7-65)	Resets both channels of the specified pulse card to the default settings.
<a href="#">PT</a> (on page 7-65)	Sets the pulse period, pulse width, pulse rise time, pulse fall time, and pulse delay for the specified pulse channel.
<a href="#">PV</a> (on page 7-67)	Sets pulse high, pulse low, range, and current limit independently for each channel of the specified pulse card.
<a href="#">RP</a> (on page 7-68)	Switches the RPM output between the PGUs, PMUs, CVUs, and SMUs.
<a href="#">TO</a> (on page 7-69)	Sets the trigger output parameters for pulse delay and trigger polarity for the specified channel.
<a href="#">TS</a> (on page 7-70)	Sets the trigger source that is used to trigger the pulse card to start its output.
<a href="#">VF</a> (on page 7-71)	Sets the state of the floating relay for the specified pulse channel.

## PA

This command aborts the process that is presently running on the specified pulse channel.

### Usage

PA A

A	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
---	--

### Corresponding LPT library function

None

### Example

PA 2

This command string aborts the process running on channel 2 of pulse card 1.

### Also see

[:PMU:ABORT](#) (on page 7-4)

---

# PD

This command sets the device under test (DUT) impedance (pulse load) for the specified channel.

## Usage

---

PD A, BBB.BBBB

A	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
BBB.BBBB	Pulse load, in ohms: 1.0 to 10e6; default 50.0

## Corresponding LPT library function

---

pulse\_load

## Details

---

You can use this command to set the device under test (DUT) impedance of the load (DUT) independently for each channel from 1  $\Omega$  to 10 M $\Omega$ .

## Example

---

PD 1, 1e3

This command string sets channel 1 of pulse card 1 for an output impedance of 1 k $\Omega$ .

## Also see

---

[:PMU:LOAD](#) (on page 7-11)

---

# PE

This command starts a configured pulse test.

## Usage

---

PE

## Corresponding LPT library functions

---

pulse\_exec

## Details

---

This command is used with the **PN** command, which configures the magnitude and timing of the pulse.

This is a blocking command. No other commands will execute until the pulse output is complete.

To generate an output, a trigger source (**TS**) must be set.

Example

```
PN 1, 0.0, 5.0, 10, 100e-6, 20e-6, 200e-9, 200e-9
TS 1, 0
PO 1, 1, 0
PE
```

Set channel 1 of the pulse card for a pulse train with a 5 V amplitude, 100 μs period, 20 μs width, and 200 ns rise and fall times. This pulse outputs 10 times.

Set the trigger source of channel 1 to be software triggered.

Set the output state of channel 1 to on with an output mode of normal.

Start execution.

Also see

- [PN](#) (on page 7-63)
- [TS](#) (on page 7-70)
- [:PMU:EXECUTE](#) (on page 7-9)

PG

This command sets the trigger mode and either initiates the start of pulse output or arms the pulse card for the specified channel.

Usage

PG *A, B, C*

<i>A</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>B</i>	Trigger mode: <ul style="list-style-type: none"><li>■ Burst: 0</li><li>■ Continuous: 1 (default)</li><li>■ Trigger burst: 2</li></ul>
<i>C</i>	Burst or trigger burst only: Pulse count in number of pulses: 1 to 2 <sup>32</sup> -1; set to 1 for continuous (default)

Corresponding LPT library functions

pulse\_trig  
pulse\_burst\_count

Details

The trigger mode setting affects both channels of a pulse card. For example, setting channel 1 of pulse card 1 to continuous also sets channel 2 to continuous.

This command is used with the `PV` and `PT` commands, which configure the magnitude and timing of the pulse.

Example

```
PT 1, 100e-6, 20e-6, 200e-9, 200e-9
PV 1, 2, -2, 5, 200e-3
PG 1, 2, 50
```

For pulse card 1 channel 1, set the pulse period of pulse card 1 to 100 μs and set channel 1 for a pulse width of 20 μs and rise and fall times for 200 ns.

Set pulse high to +2 V, pulse low to -2 V, pulse range to 5 V, and current limit to 200 mA.

Set the channel for the trigger burst trigger mode and a burst count of 50.

Also see

[PT](#) (on page 7-65)  
[PV](#) (on page 7-67)  
[:PMU:PULSE:BURST:COUNT](#) (on page 7-17)

PH

This command stops all pulse output from the pulse card.

Usage

PH A	
A	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system

Corresponding LPT library functions

pulse\_halt

Example

PH 2
This command string stops pulse output from pulse card 2.

Also see

[:PMU:ABORT](#) (on page 7-4)

PN

This command configures the pulse of the pulse channel.

Usage

PN A, BBB.BBBB, CCC.CCCC, DDD.DDDD, EEE.EEEE, FFF.FFFF, GGG.GGGG, HHH.HHHH	
PN A, BBB.BBBB, CCC.CCCC, DDD.DDDD, EEE.EEEE, FFF.FFFF, GGG.GGGG, HHH.HHHH, JJJ.JJJJ	
A	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
B	Voltage level for pulse base level
C	Voltage level for pulse amplitude
D	Pulse trigger count
E	Pulse period in seconds: 20e-9 to 1
F	Pulse width in seconds: Minimum of 10e-9; see <b>Details</b>
G	Pulse rise time in seconds: 10e-9 to 3.3e-2
H	Pulse fall time in seconds: 10e-9 to 3.3e-2
J	Pulse delay time in seconds; default is 0; must be more than: $E - F - 0.5 (G + H)$

Corresponding LPT library function

pulse\_train  
pulse\_source\_timing

Details

You can use this command to set up a pulse train to output on the pulse channel.

The Pulse Width (parameter *F*) must be less than the pulse period and more than  $0.5(G + H)$ .

Example

```
PN 1, 0.0, 5.0, 10, 100e-6, 20e-6, 200e-9, 200e-9
TS 1, 0
PO 1, 1, 0
PE
```

Set channel 1 of the pulse card for a pulse train with a 5 V amplitude, 100 μs period, 20 μs width, and 200 ns rise and fall times. This pulse outputs 10 times.

Set the trigger source of channel 1 to be software triggered.

Set the output state of channel 1 to on with an output mode of normal.

Start execution.

Also see

- [:PMU:PULSE:BURST:COUNT](#) (on page 7-17)
- [:PMU:PULSE:TIMES](#) (on page 7-18)
- [:PMU:PULSE:TRAIN](#) (on page 7-21)

PO

This command sets the output state (on or off) and mode (normal or complement) independently for each channel.

Usage

```
PO A, B, C
```

A	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
B	Output state: <ul style="list-style-type: none"><li>Off: 0 (default)</li><li>On: 1</li></ul>
C	Output mode: <ul style="list-style-type: none"><li>Normal: 0 (default)</li><li>Complement: 1</li></ul>

Corresponding LPT library functions

```
pulse_output
pulse_output_mode
```

Example

```
PO 1, 1, 1
```

This command string turns on channel 1 of pulse card 1 and selects complement pulse output mode.

Also see

- [:PMU:OUTPUT:STATE](#) (on page 7-16)

# PS

This command resets both channels of the specified pulse card to the default settings.

## Usage

PS A

A	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
---	--

## Corresponding LPT library functions

pulse\_init

## Example

PS 2

This command string resets pulse card 2.

## Also see

[:PMU:INIT](#) (on page 7-9)

# PT

This command sets the pulse period, pulse width, pulse rise time, pulse fall time, and pulse delay for the specified pulse channel.

## Usage

PT A, BBB.BBBB, CCC.CCCC, DDD.DDDD, EEE.EEEE

A	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
BBB.BBBB	Pulse period in seconds (floating-point number): <ul style="list-style-type: none"><li>5 V range: 10e-9 to 1</li><li>20 V range: 500e-9 to 1</li><li>Default: 1e-6</li></ul>
CCC.CCCC	Pulse width in seconds (floating-point number): <ul style="list-style-type: none"><li>High speed: 10e-9 to (Period – 10e-9)</li><li>High voltage: 250e-9 to (Period – 10e-9)</li><li>Default: 500e-9</li></ul>
DDD.DDDD	Rise time in seconds (floating-point number): <ul style="list-style-type: none"><li>High speed: 10e-9 to 33e-3</li><li>High voltage: 50e-9 to 33e-3</li><li>Default: 100e-9</li></ul>
EEE.EEEE	Fall time in seconds (floating-point number): <ul style="list-style-type: none"><li>High speed: 10e-9 to 33e-3</li><li>High voltage: 50e-9 to 33e-3</li><li>Default: 100e-9</li></ul>

---

**Corresponding LPT library functions**

---

```
pulse_period  
pulse_width  
pulse_rise  
pulse_fall
```

---

**Details**

---

The pulse period setting affects both channels of a pulse card. For example, setting channel 1 of pulse card 1 to 100 ms also sets the pulse period of channel 2 to 100 ms. For the other timing parameters (pulse width and rise and fall times), each available channel can have its own setting.

The minimum transition time for pulse source only (no measurement) on the rise-time high voltage for the 40 V range is 50 ns for the 4225-PMU and 4220-PGU.

---

**Example**

---

```
PS 1  
PD 1, 1e6  
PT 1, 100e-6, 50e-6, 1e-6, 1e-6  
PV 1, 2, 0, 5, 0.01  
TS 1, 0  
PO 1, 1, 0  
PG 1, 0, 5  
PO 1, 0, 0
```

Reset the PMU.

Set the output impedance to 1,000,000.

Set the timing parameters to a pulse period of 100e-6, pulse width of 50e-6, rise time of 1e-6, and fall time to 1e-6.

Set the voltage source and current limit to pulse high of 2 V, pulse low of 0 V, pulse range of 5 V, and current limit of 0.01.

Select the software trigger source.

Set the output on in normal mode.

Set the trigger mode to burst for 5 pulses.

Set the output off in normal mode.

---

**Also see**

---

[:PMU:PULSE:TIMES](#) (on page 7-18)

# PV

This command sets pulse high, pulse low, range, and current limit independently for each channel of the specified pulse card.

## Usage

PV *A, BBB.BBBB, CCC.CCCC, DDD.DDDD, EEE.EEEE*

<i>A</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>BBB.BBBB</i>	Pulse high in volts: <ul style="list-style-type: none"><li>5 V range (high speed): -5.0 to +5.0</li><li>20 V range (high voltage): -20.0 to +20.0</li><li>Default: 0.0</li></ul>
<i>CCC.CCCC</i>	Pulse low in volts <ul style="list-style-type: none"><li>5 V range (high speed): -5.0 to +5.0</li><li>20 V range (high voltage): -20.0 to +20.0</li><li>Default: 0.0</li></ul>
<i>DDD.DDDD</i>	Pulse range in volts <ul style="list-style-type: none"><li>5 V range: 5</li><li>20 V range: 20</li><li>Default: 5</li></ul>
<i>EEE.EEEE</i>	Current limit in amperes (range and load dependent) <ul style="list-style-type: none"><li>5 V range (high speed): -0.2 to +0.2</li><li>20 V range (high voltage): -0.8 to +0.8</li><li>Default (5 V range): 0.105</li></ul>

## Corresponding LPT library functions

pulse\_vhigh  
pulse\_vlow  
pulse\_range  
pulse\_current\_limit

## Example

PV 1, 2, -2, 5, 200e-3

For pulse card 1 channel 1, this command string sets pulse high to +2 V, pulse low to -2 V, pulse range to 5 V, and current limit to 200 mA.

## Also see

None



# RP

This command switches the RPM output between the PGUs, PMUs, CVUs, and SMUs.

## Usage

<i>RP HRID, mode</i>	
<i>HRID</i>	The instrument and channel ( <i>N</i> is the card index, 1 to 9): <ul style="list-style-type: none"><li>■ PMUN-1, PMUN-2</li><li>■ SMUN</li><li>■ CVHN or CVLN</li></ul>
<i>mode</i>	The mode index number: <ul style="list-style-type: none"><li>■ Pulse: 0</li><li>■ CV 2W: 1</li><li>■ SMU: 2</li><li>■ CV_4W: 3</li></ul>

## Corresponding LPT library function

`rpm_config`

## Details

In addition to the connection to the PMU, the 4225-RPM includes input connections for the following instruments:

- SMUs: 4200-SMU, 4201-SMU, 4210-SMU, or 4211-SMU
- CVUs: 4210-CVU or 4215-CVU

Use this command to control switching inside the RPM to connect the PMU, CVU, or SMU to the output.

If there is no RPM connected to the PMU channel, the `RP` command will not cause an error.

## Example

```
SS
RP PMU1-1, 2
SS
RP SMU2, 2
RP CVH1, 1
RP CVL1, 1
```

Set the source setup page.  
Switch the RPM on channel 1 of PMU1 to SMU (blue LED).  
Set the source setup page.  
Switch the RPM connected to SMU2 to SMU (blue LED).  
Switch each RPM to CVU 2-wire (red LED) for the CVU1 high and CVU1 low connections.  
At the end of the KXCI test, you should turn the outputs to 0 V on the instrument to which the RPM is connected. This is to reduce the risk of damaging instruments and the DUT. You should also return the RPM setting to pulse.

## Also see

[:PMU:RPM:CONFIGURE](#) (on page 7-22)

# TO

This command sets the trigger output parameters for pulse delay and trigger polarity for the specified channel.

## Usage

TO *A*, *BBB.BBBB*, *C*

<i>A</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>BBB.BBBB</i>	Pulse delay in seconds (default 0.0): 0.0 to (Period – 10e-9)
<i>C</i>	Trigger polarity <ul style="list-style-type: none"><li>▪ Negative (falling edge): 0</li><li>▪ Positive (rising edge): 1 (default)</li></ul>

## Corresponding LPT library functions

pulse\_delay  
pulse\_trig\_polarity  
pulse\_output

## Details

This command sets pulse delay and trigger polarity. The polarity setting affects both channels of a pulse card. For example, setting channel 1 of pulse card 1 to positive trigger polarity to also sets the trigger polarity of channel 2 to positive. For the trigger delay parameter, each available channel can have a unique delay setting.

## Example

TO 1, 2e-6, 1

This command string sets the pulse delay for channel 1 of pulse card 1 to 2 μs and selects positive trigger polarity.

## Also see

None

# TS

This command sets the trigger source that is used to trigger the pulse card to start its output.

## Usage

TS *A, B*

<i>A</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>B</i>	Trigger source: <ul style="list-style-type: none"><li>■ Software triggered: 0</li><li>■ External, initial trigger only, rising edge: 1</li><li>■ External, initial trigger only, falling edge: 2</li><li>■ External, trigger every pulse, rising edge: 3</li><li>■ External, trigger every pulse, falling edge: 4</li></ul>

## Corresponding LPT library functions

`pulse_trig_source`

## Details

With the software trigger source selected, the `PG` command selects the trigger mode (continuous, burst, or trigger burst), and initiates the start of pulse output.

### NOTE

Since trigger source is a card-level setting and not a channel setting, using channel 1 or 2 sets the card to the specified source card 1. Similarly, channel 3 or 4 sets the source for card 2.

With an external trigger source selected, the `PG` command selects the trigger mode and arm pulse output.

The pulse output starts when the required external trigger pulse is applied to the Trigger In connector. There is a trigger-in delay of 560 ns. This is the delay from the trigger-in pulse to the time of the rising edge of the output pulse.

For an initial trigger-only setting, only the first rising or falling trigger pulse starts and controls the pulse output.

For a trigger every pulse setting, rising or falling edge trigger pulses start and control the pulse output. After the initial pulse, the pulse output, either continuous or burst, is output based on the internal pulse generator clock. If pulse-to-pulse synchronization is required over higher count pulse trains, use a trigger every pulse setting.

**Example**

TS 1, 1

This command string sets the trigger source to channel 1 and external, initial trigger only, rising edge.

**Also see**

[PG](#) (on page 7-62)

---

# VF

This command sets the state of the floating relay for the specified pulse channel.

**Usage**

VF <i>A, B</i>	
<i>A</i>	Pulse-card channel number: 1 to 8; the largest value is the number of channels in the system
<i>B</i>	Floating relay state: <ul style="list-style-type: none"><li>■ Disabled: 0</li><li>■ Enabled: 1</li></ul>

**Corresponding LPT library functions**

pulse\_float

**Details**

You can use this command to allow the pulse channel to float.

**Example**

VF 1, 1

This command string sets channel 1 of the pulse card to enable the floating relay.

**Also see**

None

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

Keithley Instruments • 28775 Aurora Road • Cleveland, Ohio 44139 • 1-800-833-9200 • [tek.com/keithley](http://tek.com/keithley)

---

