User Manual

Tektronix

VX4101A
MultiPaq™ Instrument

071-0049-01

This document supports firmware version 2.0

Warning
The servicing instructions are for use by qualified personnel only. To avoid personal injury, do not perform any servicing unless you are qualified to do so. Refer to the Safety Summary prior to performing service.

Printed in the U.S.A.

Tektronix, Inc., P.O. Box 1000, Wilsonville, OR 97070–1000

WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of three (3) years from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents

Getting Started

Operating Basics

Syntax and Commands

Status and Events

Appendices

Glossary and Index

List of Figures

List of Tables

General Safety Summary

Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it. To avoid potential hazards, use this product only as specified.

Only qualified personnel should perform service procedures.

While using this product, you may need to access other parts of the system. Read the *General Safety Summary* in other system manuals for warnings and cautions related to operating the system.

To Avoid Fire or Personal Injury

Connect and Disconnect Properly. Do not connect or disconnect probes or test leads while they are connected to a voltage source.

Ground the Product. This product is indirectly grounded through the grounding conductor of the mainframe power cord. To avoid electric shock, the grounding conductor must be connected to earth ground. Before making connections to the input or output terminals of the product, ensure that the product is properly grounded.

Observe All Terminal Ratings. To avoid fire or shock hazard, observe all ratings and marking on the product. Consult the product manual for further ratings information before making connections to the product.

Do not apply a potential to any terminal, including the common terminal, that exceeds the maximum rating of that terminal.

Do Not Operate Without Covers. Do not operate this product with covers or panels removed.

Use Proper Fuse. Use only the fuse type and rating specified for this product.

Avoid Exposed Circuitry. Do not touch exposed connections and components when power is present.

Do Not Operate With Suspected Failures. If you suspect there is damage to this product, have it inspected by qualified service personnel.

Do Not Operate in Wet/Damp Conditions.

Do Not Operate in an Explosive Atmosphere.

Keep Product Surfaces Clean and Dry.

Provide Proper Ventilation. Refer to the manual’s installation instructions for details on installing the product so it has proper ventilation.

Symbols and Terms

Terms in this Manual. These terms may appear in this manual:

WARNING. *Warning statements identify conditions or practices that could result in injury or loss of life.*

CAUTION. *Caution statements identify conditions or practices that could result in damage to this product or other property.*

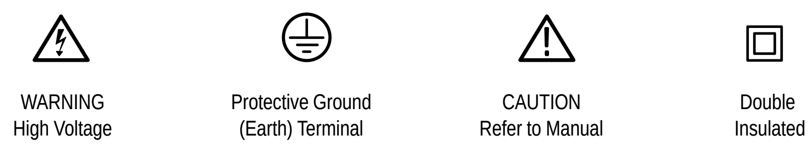
Terms on the Product. These terms may appear on the product:

DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

CAUTION indicates a hazard to property including the product.

Symbols on the Product. The following symbols may appear on the product:



Service Safety Summary

Only qualified personnel should perform service procedures. Read this *Service Safety Summary* and the *General Safety Summary* before performing any service procedures.

Do Not Service Alone. Do not perform internal service or adjustments of this product unless another person capable of rendering first aid and resuscitation is present.

Disconnect Power. To avoid electric shock, disconnect the main power by means of the power cord or, if provided, the power switch.

Use Care When Servicing With Power On. Dangerous voltages or currents may exist in this product. Disconnect power, remove battery (if applicable), and disconnect test leads before removing protective panels, soldering, or replacing components.

To avoid electric shock, do not touch exposed connections.

Preface

This manual assumes you are familiar with the operation of VXIbus instruments and with the purpose and function of this instrument.

Please read and follow all instructions for installation and configuration. Use the Installation Checklist to ensure proper installation and to record your initial settings.

The *Operating Basics* section gives a summary of VXIbus operation and presents an overview of the operation of this instrument.

The *Syntax and Commands* section provides a summary of all the commands followed by detailed descriptions of each command. *Appendix E: Examples* contains example programs that demonstrate the programmable features of this instrument.

The *Status and Events* section contains an explanation of the Status and Event Reporting System and lists the system messages.

The *Reference Guide* contains a summary of all the SCPI instrument commands.

Conventions

The names of all switches, controls, and indicators appear in this manual exactly as they appear on the instrument.

Specific conventions for programming are given in the sections *Syntax and Commands* and *Appendix E: Examples*.

Getting Started

Product Description

This section introduces the VX4101A MultiPaq™ Instrument, and includes the following information:

- The VX4101A description explains the key features, functionality, and instruments included with the VX4101A
- The physical description shows the locations of the fuses and indicators
- The list of accessories describes the standard and optional accessories
- The VX4101A self-test outlines the self-diagnostic routines run on each instrument
- Information about the logical IEEE-488 address of the VX4101A
- A description of the VXI*plug&play* software, including the soft front panels and device drivers that you can use to control the VX4101A

VX4101A Description

TheVX4101A MultiPaq™ Instrument is a C-size single slot VXI module for use in a mainframe conforming to the VXIbus specification. The VX4101A provides powerful functionality in a small package, and includes the following instruments needed in a typical VXI system:

- Full function Universal Counter/Timer
- Digital Input and Output
- Full function Digital Multimeter (DMM)
- Digital to Analog Converter (DAC)
- Relay Drivers
- Scanner master function for the Tektronix SurePath™ family of VXI relay modules

Features

All instruments included in the VX4101A are VXI message-based. Each function is located at the same logical address, but can be accessed and used separately. You can access the instruments sequentially in the same manner as multiple instruments in a typical test system. They can be set up sequentially, then triggered and operated concurrently, with local on-instrument processing and memory for each function. To understand the principles of operation of the VX4101A, see *Operating Basics*.

The VX4101A is programmed by issuing ASCII characters from the system controller via the VXIbus commander and the VXIbus mainframe backplane. Refer to the manual for the VXIbus device that will be the commander for details on the operation of that device.

Instrument Control. You can control the instrument through either SCPI commands or through VXI*plug&play* instrument drivers. The SCPI command sets for each instrument conform to SCPI 1995 standards. The VXI*plug&play* driver functions conform to 1997 standards as determined by the VXI*plug&play* Alliance.

NOTE*. SCPI permits a great deal of flexibility in the form in which you can enter commands. Examples throughout this manual use various forms of the command syntax to further illustrate these concepts.*

Fast Data Channel. To maximize throughput, some of the the VX4101A instruments support Fast Data Channel, a VXI Consortium standard protocol for high speed block transfers of data. The VX4101A architecture implements FDC V2.0 with a maximum of eight FDC channels per VXIbus module.

NOTE*. The Digital Multimeter (DMM) and Digital to Analog Converter (DAC) each use one FDC channel, which the VX4101A assigns at power-on.*

Triggers. Each instrument in the VX4101A that supports triggering can trigger any other instrument or the backplane TTL triggers.

Performance Options

You can enhance the performance of the VX4101A with the following options:

Table 1–1: VX4101A Performance Options

Description	Option #
500 MHz Ch1 and Ch2	1C
3 GHz prescaler Ch3	2C
8 Ch DAC	1A
32 bit Digital I/O	1D
TCXO	1T

About the Universal Counter

The Universal Counter in the VX4101A provides two input channels to make frequency, period, rise and fall time, positive and negative pulse width, positive and negative duty cycle, frequency ratio, totalizer, time interval, time interval with delay by time or events, phase, and AC/DC voltage measurements.

Measurement gating comes from one of several sources, including VXI TTL triggers, counter front panel arm, software triggers, periodic trigger, and another VX4101A instrument, such as SurePath™ relay switched and settled. Other key features of the Universal Counter are as follows:

- Frequency measurements with ten-digit resolution at 1 second aperture, with a range of 1 μHz - 250 MHz
- The option 1C extends the maximum frequency beyond 500 MHz
- 250 ns resolution (1 ps with averaging)
- Input signal conditioning: x1, x10, x100 attenuation with 0.4 to 10.0 variable gain and up to ±100 V offset depending on attenuation, DC or AC coupling, and 50 Ω or 1 MΩ input (with automatic over-current protection for the 50 Ω mode)

  For more information on Counter input operations, see *Appendix D: Counter Architecture*.
- Optional 3 GHz channel 3 prescaler input with option 2C
- Optional temperature controlled, oscillated or crystal Clock Source (TCXO) with Option 1T

***NOTE**. You can also use the VXI backplane 10 MHz clock (CLK10) as the Counter clock source.*

About the Digital Input and Output Modules

Both the Digital Input and Digital Output instruments provide low-speed data exchange with 32 bits of data. The Digital Input and Output share a single 32-bit interface on the 160-pin connector. The frequency range for both input and output is from 3.662 Hz to 48 kHz. The Digital Input and Output instruments include the following features:

- External handshaking
- 4K × 32 bits memory
- Pattern matching available on the Digital Input
- Programmable input threshold and selectable output voltage
- Outputs can be repeated continuously or counted a specified number of times

The module self-tests perform full read, write, and verify tests on both shared memory and read/write control registers. The SCPI TEST subsystem provides programming capabilities for the self-test functions.

Digital Input. The Digital Input has a single programmable voltage threshold for all 32 bits. The range of the input threshold voltage is from 0 volts to 20 volts.

The Digital Input has 8 K samples of digital input memory, 4 K samples each for the pre-match pattern buffer and the post-match memory. You can query all Digital Input settings.

Digital Output. Each of the 32 bits of digital output is programmable. Output voltages can be set for 5 V, 12 V, or 24 V nominal operation. You can also use an externally supplied voltage, if desired. You must set all digital output bits to the same voltage setting. The output sample frequency range is from 3.662 Hz to 48 kHz, and applies to all 32 bits. The Digital Output has 4 K × 32 samples of memory. You can repeat sample segments as either continuous, or with a programmable count and repetition frequency, You can query all Digital Output settings.

About the Digital Multimeter (DMM)

The DMM has full function autoranging with 5 1/2 digit resolution. Measurements can be returned as single measurements, or as an array. Power cycle averaging modes are available for 60 Hz and 50 Hz.

The device has autozero capability which minimize offset drift errors without removing the customer input connections. The DMM has floating input isolation of 300 VDC or AC_{RMS}.

Ranges of operation are:

- Array measurements returned with up to 4096 measurements via the Fast Data Channel Protocol.
- Over 500 programmable aperture times from 833 μs to 2 seconds
- DC voltage at 30 mV, 300 mV, 3 V, 30 V, and 300 V
- AC True RMS, either AC or DC coupled, at 30 mV_{RMS}, 300 mV_{RMS}, 3 V_{RMS}, 30 V_{RMS} and 300 V_{RMS}
- Four-wire or two-wire Ω measurement at ranges of 30 Ω, 300 Ω, 3 kΩ, 30 kΩ, 300 kΩ, 3 MΩ, 30 MΩ, and 300 MΩ
- DC Current at 150 mA and 1A

Acquisition triggering is provided from one of several sources, including VXI TTL triggers, counter front panel arm, software triggers, periodic trigger, and other VX4101A instrument triggers, such as the SurePath™ relay switched and settled.

About the Digital to Analog Converter

The Digital to Analog Converter (DAC) includes an eight-channel arbitrary waveform generator. Waveform length can be generated with one to 1024 sample points at sample rates from 3.662 to 15 kHz. Signals are generated on all eight channels when the DAC is initiated. A default 0 VDC output will be set on unused channels.

The VX4101A DAC includes the following features and functionality:

- Eight synchronized, 12-bit digital to analog channels synchronized to a single programmable sample clock (3.662 Hz to 15 kHz)
- Amplitudes of –14 V to +14 V are supported with 8 mV resolution on each channel
- Each channel is separately programmable to generate a different waveform
- Each channel has 1024 points of trace buffer memory available for waveform synthesis
- Waveforms can be repeated either continuously or with a programmable count and repetition frequency.

***NOTE**. Setting the sample rate, segment length, and repetition frequency the same for one channel sets them for all channels.*

About the Relay Drivers

The high-current relay drivers include the following:

- Eight lines of high-current outputs
- Open collector, 100 mA sink per output

About the SurePath™ Modules

The scanner master control enables you to use the VX4101A to control the Tektronix SurePath™ family of VXI relay modules, including the VX4320, VX4330, VX4350, VX4351, and VX4380. Some of the features of the SurePath™ family are as follows:

VX4320 1.3 GHz RF Multiplexor. This module is a single-wide VXIbus module intended to switch RF and high-frequency digital signals. It is arranged as eight 1×4 coaxial switches with a 50 Ω characteristic impedance. Each 1×4 section has five type SMB male snap-on subminiature coaxial connectors.

VX4330 120-Channel Scanner/Multiplexor. This module provides six 1×10 4-wire multiplexor sections. Each of these six sections can be independently configured under software control as two 1×10 2-wire multiplexor, a 1×20 2-wire multiplexor, or as a 1×40 1-wire multiplexor. In addition, each section can be programmed to connect it to the section above or below to produce a 1 × 60 4-wire multiplexor, a 1 × 120 2-wire multiplexor, or a 1 × 240 1-wire multiplexor.

VX4350 64-Channel SPST/SPDT Switching Module. This module provides 64 independent single-pole double-throw relays. A 0 Ω resistor is placed in series with the common contact of each relay. This resistor may be replaced by a resistor with a larger value to limit the current that flows through the relay in order to protect the relay contacts. Pads are provided for optional metal oxide varistors (MOVs). These varistors are connected from the common contact to the normally closed contact, and from the common contact to the normally open contact of each relay. Using the optional MOVs protects the relay contacts from over voltage conditions encountered when switching electrical power to inductive loads. The optional MOVs and current limiting resistors are user installed, and are not available as factory options.

VX4351 40-Channel, 10 Amp, SPST Switch Module. This module contains 40 SPST (form A) relays. Each relay may be controlled independently. The contacts of each relay are connected to one of three 30-pin high current connectors which are mounted on the module's front panel. Circuitry is included on the VX4351 to verify the control signals that are applied to each relay coil driver.

VX4380 256-Crosspoint Relay Matrix Module. This module provides four 4 × 16 2-wire matrix sections. Each section can be configured to connect either the four rows or the sixteen columns to the section above or below it to produce up to a 16 × 16 2-wire matrix or a 4 × 64 2-wire matrix.

The VX4101A SurePath™ Master provides serial I/O interface for control of SurePath™ relay modules. It also monitors the power fuses of all SurePath™ Relay modules, and provides a serial input interface to identify each module that it controls.

***NOTE**. There is a query for the VX4101A which will return a list of the SurePath™ family relay modules that the VX4101A SurePath™ Master detects in the system.*

Power-On Sequence

The power-on sequence of the VX4101A meets the timing requirements of the VXIbus specification that communications may begin even if the instrument has not completely initialized. At either power-on or a VXIbus reset, the VX4101A initializes the VXIbus interface and all hardware and firmware necessary to begin communication. For more information, see *Powering On the VX4101A* in *Installation*.

Physical Description

Figure 1–1 shows the VX4101A switches and fuses, and Figure 1–2 shows the front panel.

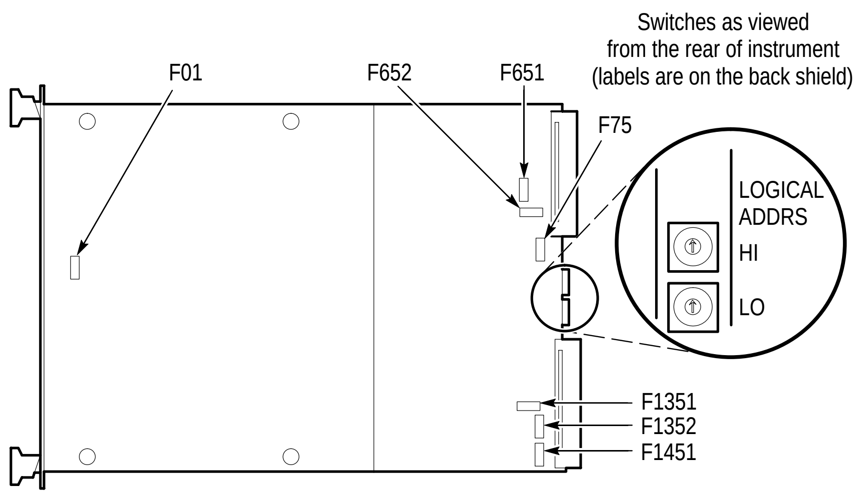


Figure 1–1: VX4101A VXIbus Connectors, Fuses, and Switch Locations

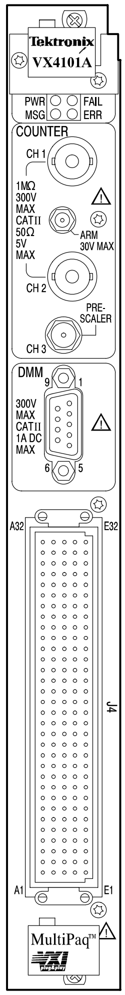


Figure 1–2: VX4101A Front Panel

Controls and Indicators

The following logical address switches must be correctly set to ensure proper operation. Refer to Figure 1–1 for their physical locations.

Logical Address Switches. The VX4101A supports VXI dynamic addressing. It is shipped with the switches set to FF so that the Slot 0 will automatically assign an address to the module.

Each functional module in a VXIbus System must be assigned a unique logical address, from 1 to decimal 255 (hexadecimal FF).

***NOTE**. If you do not want to use dynamic addressing, align the desired switch position with the arrow on the module shield.*

The physical address of the instrument is on a 64 byte boundary. If the Logical Address switch representing the most significant digit (LA-HI) of the logical address is set to position X and the switch representing the least significant digit (LA-LO) of the logical address is set to position Y, then the base physical address of the module will be $[(40_{16} \times XY_{16}) + C000_{16}]$. For example:

			Base Physical Address	
L.A.	**HI**	**LO**	**decimal**	**hexadecimal**
A_{16}	0_{16}	A_{16}	(64 * 10) + 49152 = 49792	$(40_{16} * A_{16}) + C000_{16} = C280_{16}$
15_{16}	1_{16}	5_{16}	(64 * 21) + 49152 = 50496	$(40_{16} * 15_{16}) + C000_{16} = C540_{16}$

L.A. is the Logical Address

LEDs

The VX4101A has four LEDs visible on its front panel. These LEDs are labeled as follows:

- Power LED – this LED is On if all six fuses for the six power buses are intact. Any single fuse being blown results in the LED turning OFF
- Fail LED – this LED is normally OFF. During power-on or reset self-test, the LED will be ON for the duration of the test. If the VX4101A detects a failure during normal operation, the LED will come ON and the SYSFAIL line on the backplane will be true
- Message LED – this LED flickers ON when the VX4101A is being addressed on the VME backplane by its commander
- ERR LED – this LED is normally OFF. However, it may blink on and off to indicate error conditions. The most common reason is a command syntax error has been detected. Other error conditions that will cause the LED to blink are discussed elsewhere in this manual. Sending the “SYSTem:ERRor?” query to the instrument will return the cause of the error. When all errors in the queue have been retrieved, the error LED will return to the OFF state

Front Panel Connectors

Refer to *Appendix B:Input/Output Connections* for more information.

Fuses

The VX4101A has 6 fuses that limit the amount of current that each module can draw from the VXI backplane +5, -5.2, +24, -24, +12, and -12 V power pins. These fuses protect the module in case of an accidental shorting of the power bus or any other situation where excessive current might be drawn.

If any fuse opens, the module will assert SYSFAIL* on the VXIbus.

If the +5 V fuse opens, the VXIbus Resource Manager will be unable to assert SYSFAIL INHIBIT to disable SYSFAIL*.

If any fuse opens, remove the fault before replacing the fuse. Replacement fuse information is given in *Appendix H: Replaceable Parts.* Refer to the Separator page, following page C-10, before performing any service to this product.

IEEE-488 Address

In order to use and program the VX4101A in an IEEE-488.2 environment, you must know the IEEE-488 address of the module. Different manufacturers of IEEE-488 interface devices have different algorithms for equating a logical address with an IEEE-488 address. Consult the operating manual of the IEEE-488 module being used.

Self-Test

The VX4101A performs an interface self-test at power-on. This test ensures that the VXIbus interface is fully functional and ready for communication with the controller and that each instrument is initialized and ready for operation.

Built-In Test is provided by extensive self-tests which can be invoked via IEEE 488.2 and SCPI commands and queries.

The following are highlights of each test performed:

VX4101A MultiPaq™ Instrument Self-Test

The self-test for the VX4101A MultiPaq™ Instrument tests the following components:

- CPU RAM
- CPU timers, including periodic trigger
- Internal CPU interrupts
- Software triggers

Counter Self-Test

The self-test for the Counter tests the following components:

- The two 4 Kb Counter measurement buffers.
- Logic registers
- The analog front end pre-amp offset, pre-amp inverter, and pre-amp gain digital to analog converters (DACs).
- A 2.5 MHz signal is routed in through a test source and checked for accuracy.

Digital Input and Output Self-Test

The self-test for the Digital Input and Digital Output includes a test of the Digital Output hardware, as well as read/write/verify tests on shared memory and read/write control registers.

DMM Self-Test

The DMM self-test includes RAM testing, power supply and DMM reference testing, amplifier and analog to digital converter testing, and control circuitry testing. The self-test for the DMM tests the following components:

- The two 4 Kb DMM measurement buffers
- Logic registers
- Measurements of Ground are taken at the 30 mV DC range, 3 VDC range, and 300 mV AC/DCV range
- The 2.75 V reference is tested in the 3 V range

DAC Self-Test

The DAC performs the following self-test procedure:

- Read/write/verify test on shared memory
- Read/write/verify on all read/write control registers

Relay Drivers Self-Test

There is no self-test for the Relay Drivers.

SurePath™ Module Self-Test

There is no automatic self-test for SurePath™ modules. You use the TEST:ALL? query to test the control logic and data path for both the VX4101A and the SurePath™ card in use. For more information on the TEST:ALL? query, see *SCPI Commands for the SurePath Modules* in this manual

Accessories

The following tables list the standard and optional accessories for the VX4101A:

Table 1–2: Standard Accessories

Tektronix part number	Qty	Name and description	Mfr. Code	Mfr. part number
VX1784S	1	CONN HOODED; DE–9 FEMALE SOCKET	TK2548	
071–0049–XX	1	MANUAL,TECH:USERS,VX4101A	TK2548	071–0049–XX
071–0051–XX	1	MANUAL,TECH:REFERENCE,VX4101A	TK2548	071–0050–XX
063–2598–00	1	VXI*plug&play* 16-bit driver		063–2598–XX
063–2822–00	1	VXI*plug&play* 32-bit driver		063–2822–XX

Table 1–3: Optional Accessories

Tektronix part number	Qty	Name and description	Mfr. Code	Mfr. part number
VX1630	1	CABLE, ANALOG; 160 PIN CONN W/160 COND CBL		
VX1630M	1	MOD KIT; VX4101 DMM MOD KIT FOR VX1630 CABLE, 3 TWISTED PAIRS, DE–9P MALE SUBMINIATURE, 24L		
VX1630S	1	160 PIN CON KIT; BAK SHL, PIN HOUSNG, 200 PINS		
VX1729	1	CABLE,INTCON; COAX,VX1729;RFD,50 OHM,RG188, 10 FT,BNC,MALE X SMB,STR		
VX1730	1	CABLE, ADAPTER; SMB SNAP ON PLUG TO PLUG		
VX1760	1	SMA–BNC MALE; LE COAX CABLE, RFD50–OHM, 1784S RG188, 10FT, BNC MALE–SMA, STR		
174–1428–00	1	CABLE ASSY:COAX,RFD,50 OHM,60.0L,SMA,STR,BOTH ENDS,MALE	060D9	174–1428–00
012–0057–01	1	CA ASSY,RF:COAXIAL,RFD,50 OHM,43 L,BNC,MALE	060D9	012–0057–01
003–1493–00	1	HAND TOOL:DISASSEMBLY TOOL FOR CRIMP & POKE CONTACTS	6V439	471 555
003–1494–00	1	HAND TOOL:HAND TOOL FOR LOOSE CRIMP & POKE CONTACTS	6V439	014 374

Performance Options

You can purchase the following options to enhance performance of the VX4101A:

Table 1–4: VX4101A Performance Options

Description	Option Number
500 MHz Counter Channel One and Channel Two	1C
3 GHz Counter prescaler Channel Three	2C
Eight Channel DAC	1A
32 Bit Digital I/O with Eight Relay Drivers	1D
TCXO	1T

About the VXI*plug&play* Software

The VXI*plug&play* software included with the VX4101A consists of two components:

- Device drivers
- Soft Front Panels (SFPs)

VXI*plug&play* device drivers enable you to operate the VX4101A under program control.

The graphical user interfaces of the SFPs emulate the physical controls and displays typically found on monolithic instruments. The instrument drivers call a common set of I/O control functions that are independent of instrument types, interface types, operating systems, programming languages, and networking mechanisms.

The installation program installs the VXI*plug&play* drivers for the framework appropriate for your processing environment. The frameworks were developed as established by the VXIplug&play Alliance. The possible frameworks are as follows:

- WIN
- WINNT
- WIN95

Using the *VXI plug&play* Software

Double clicking on the appropriate instrument icon launches the soft front panel (SFP). The SFP displays a representation of the traditional controls and indicators for an instrument. By selecting the appropriate controls on the SFP, you can verify that the instrument has been correctly installed and is functional, and perform almost all of the functions of the instrument. The SFP will:

- Control the instrument
- Display data
- Provide command line query and response (talk/listen)
- Provide error and event reporting

See *Installation* for instructions on installing and running the SFP and using the C driver for program control. Detailed descriptions of the VXI*plug&play* drivers are given in on-line Help and text files on the disk shipped with the instrument.

Installation

This section contains the information you will need to install the VX4101A MultiPaq™ Instrument and its associated software, and to verify that the instrument is functioning properly. This includes the following:

- Installing the VX4101A module in the mainframe
- Installing the VXI*plug&play* software
- Running a functional check
- Using the soft front panels (SFPs) included with the instrument

At the end of the section, you will find a checklist to summarize your installation choices.

Installing the Module in the Mainframe

Installing the VX4101A in a Tektronix mainframe meets all instrument cooling requirements.

Tools Required

A slotted screwdriver set is required for proper installation.

Requirements and Cautions

The VX4101A is a C-size VXIbus instrument module and therefore can be installed in any C- or D-size VXIbus mainframe slot other than Slot 0. To install the module in a D-size mainframe, consult the operating manual for the mainframe. Refer to *Controls and Indicators* for information on selecting and setting the Logical Address switch of the module. This switch defines the programming address of your module.

⚠ **CAUTION.** *Note that there are two printed ejector handles on the card. To avoid installing the card incorrectly, make sure the ejector marked "VX4101A" is at the top. Installing it incorrectly may damage the DIN connectors on the module.*

***NOTE**. If the VX4101A is inserted in a slot with any empty slots to the left of the module, the VME daisy-chain jumpers must be installed on the backplane in order for the VXI Module to operate properly. Check the manual of the mainframe being used for instructions on jumper settings. If the jumpers are not installed properly, there will be no interrupts and bus masters will not operate properly. Jumpers are not necessary for auto-configuring backplane designs such as those in Tektronix mainframes.*

Module Installation Procedure

Follow these steps to install the VX4101A:

***CAUTION.** The VX4101A Module is a piece of electronic equipment and therefore has some susceptibility to electrostatic damage (ESD). ESD precautions must be taken whenever the module is handled.*

1. Record the revision levels, serial numbers (located on the label on the top shield of the VX4101A), and switch settings on the *Installation Checklist*.

2. Verify that the switches are set to the correct values. Refer to *Controls and Indicators* for more information on setting switches.

3. Make sure that the mainframe power is Off.

4. Insert the module into one of the instrument slots of the mainframe (see Figure 1–3).

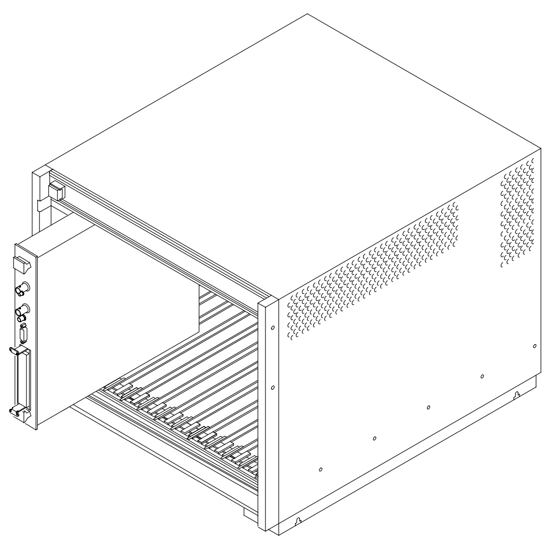


Figure 1–3: Module Installation

CAUTION. *Verify that the mainframe is able to provide adequate cooling and power with this module installed. Refer to the mainframe Operating Manual for instructions. If the mainframe cannot cool the unit adequately, the unit may not operate properly and may be damaged.*

Installing the VXI*plug&play* Software

Each VXI*plug&play* instrument includes either a 3 1/2 inch diskette or CD-ROM storage media containing the SFPs and device drivers with which you can control the instrument interactively. The SFPs are capable of controlling the instrument immediately following a successful installation, without requiring a specific application development environment. The soft front panels and the supporting software were developed in conformance with the guidelines of the VXI*plug&play* Systems Alliance.

All VXI*plug&play* products are classified within a particular framework, as developed by the VXI*plug&play* Systems Alliance to categorize operating systems, programming languages, and I/O software libraries. The framework supported by the driver distributed with this VXI module is printed on the label of the media.

The source code as well as the dynamic link library (DLL) are distributed so that you have the flexibility of using either of them in the end application.

About the Device Drivers

The device driver distributed with the Tektronix VX4101A complies with all current VXI*plug&play* requirements. The device driver uses VISA calls that are portable across platforms and development environments. Tektronix uses only the ANSI C implementation in instrument driver source code. No platform-specific libraries are included in the driver. The driver source code will compile using MSVC, Borland, Symantec or Watcom compilers without having to use foreign libraries, other than the VISA Dynamic Link Library (DLL). All driver .DLL files are located in:

VXIPNP\<Framework>\BIN

Installation Procedure

Use the following procedure to install the VXI*plug&play* software:

1. Insert the media containing the driver files in the appropriate drive.
2. Locate the file *Setup.exe*, as follows:
   - On a 3 1/2 inch floppy disk, the file is on disk 1
   - On a CD-ROM, the file will be in the *tkvx4101* directory

3. Launch *setup.exe* as follows:

- In Windows or Windows NT/3.x, use the File menu and select Run. Then, browse to find setup.exe or type the drive letter and program name
- In Windows 95 or or Windows NT/4.x, use Start and select Run. Browse to locate setup.exe and click on OK

4. Follow the directions of the installation program.

Following installation, driver files (see list below) will be found in locations defined by the VXI*plug&play* Alliance. Where required, modifications to your *autoexec.bat* and *system.ini* files may be automatically completed.

Driver Files

The instrument driver for the VX4101A is distributed with a number of C source code files, header files, dynamic link libraries and other supporting files. A breakdown of the modules that the files control are as follows:

VX4101. This is the high level "controller" portion of the driver. This part of the driver does the actual communication with the instrument. The tkvx4101.c or tkvx4101.dll files have functions which perform VISA function calls. The other modules, making up the total driver, call functions in the tkvx4101 to get system configuration information and to communicate to each specific instrument.

Device-Specific Files. The files supporting the specific instruments are as follows:

Table 1–5: Instrument-Specific Files

Prefix	Controls instrument
tkmpdmm	DMM
tkmpctr	Counter
tkmpscan	SurePath™ scanners
tkmpdac	Digital to Analog Converter
tkmpdrv	Relay Drivers
tkmpdigi	Digital Input
tkmpdigo	Digital Output

File Locations. For the VX4101A driver, the following files will be in the VXIPNP\<Framework>\TKVX4101 directory:

tkvx4101.c
tkvx4101.fp
tkvx4101.mak
tkvx4101.def

tkvx4101.hlp
tksf4101.exe

Supporting driver files will be located in the following directories:

VXIPNP\<Framework>\support\tkmpdmm:

tkmpdmm.c
tkmpdmm.fp
tkmpdmm.mak
tkmpdmm.def
tkmpdmm.hlp

VXIPNP\<Framework>\support\tkmpctr:

tkmpctr.c
tkmpctr.fp
tkmpctr.mak
tkmpctr.def
tkmpctr.hlp

VXIPNP\<Framework>\support\tkmpscan:

tkmpscan.c
tkmpscan.fp
tkmpscan.mak
tkmpscan.def
tkmpscan.hlp

VXIPNP\<Framework>\support\tkmpdac:

tkmpdac.c
tkmpsdac.fp
tkmpdac.mak
tkmpdac.def
tkmpdac.hlp

VXIPNP\<Framework>\support\tkmpdigo:

tkmpdigo.c
tkmpdigo.fp
tkmpdigo.mak
tkmpdigo.def
tkmpdigo.hlp

VXIPNP\<Framework>\support\tkmpdigi:

tkmpdigi.c
tkmpdigi.fp
tkmpdigi.mak

tkmpdigi.def
tkmpdigi.hlp

VXIPNP\<Framework>\support\tkmprdrv:

tkmprdrv.c
tkmprdrv.fp
tkmprdrv.mak
tkmprdrv.def
tkmprdrv.hlp

VXIPNP\<Framework>\support\tkfdc:

tkfdc.c
tkfdc.fp
tkfdc.mak
tkfdc.def
tkfdc.hlp

The directories listed above contain several *.uir (User Interface Resource) files that support the GUI executables (CVI files).

The following files are installed in the VXIPNP\<Framework>\INCLUDE directory:

tkvx4101.h
tkmpdac.h
tkmpdigo.h
tkmpdigi.h
tkmprdrv.h
tkmpdmm.h
tkmpctr.h
tkmpscan.h
tkvx4101.bas
tkmpdac.bas
tkmpdigo.bas
tkmpdigi.bas
tkmprdrv.bas
tkmpdmm.bas
tkmpctr.bas
tkmpscan.bas
tkfdc.bas

The VXIPNP\<Framework>\BIN directory includes the following files:

tkvx4101.dll
tkmpdmm.dll
tkmpctr.dll
tkmpscan.dll
tkmpdac.dll

tkmprdrv.dll
tkmpdigo.dll
tkmpdigi.dll
tkfdc.dll

The VXIPNP\KB directory includes the following files:

tkvx4101.kb
tkmpdmm.kb
tkmpctr.kb
tkmpscan.kb
tkmpdac.kb
tkmprdrv.kb
tkmpdigo.kb
tkmpdigi.kb
tkfdc.lib

The VXIPNP\<Framework>\LIB\MSC\ directory includes the following files:

tkvx4101.lib
tkmpdmm.lib
tkmpctr.lib
tkmpscan.lib
tkmpdac.lib
tkmprdrv.lib
tkmpdigo.lib
tkmpdigi.lib
tkfdc.lib

The .DLL files are tested in LabWindows/CVI, LabView, HPVEE, Visual Basic, MSVC/C++ and Borland C/C++ before distribution. If you want to modify the driver algorithms, all files are distributed to facilitate rebuilding the .DLL file with the modifications.

***NOTE**. Tektronix recommends that you back up your original source files before modifying the driver files*

***CAUTION.** The Soft Front Panel (SFP) distributed with this driver is built using the distributed .DLL files. Any modification to the .DLL files used by the SFP could make it unusable. Re-installing the driver will write over any modified files with the original files and restore SFP operation.*

Installation parameters will vary depending on the mainframe being used. Be sure to consult the mainframe operating manual before installing and operating the module.

Powering-On the VX4101A

The VX4101A instrument runs its interface test and is ready for communication within five seconds. The VXIbus Resource Manager can add an additional delay. The Power LED will be on. The Failed LED will be off. The default condition of the module after power-on is listed in the *RST command description. For information on the *RST command, see IEE488.2 commands.

The power-on sequence of the VX4101A meets the timing requirements of the VXIbus specification and permits communications to begin even if the instrument has not completely initialized. At either power-on or a VXIbus reset, the VX4101A initializes the VXIbus interface and all hardware and firmware necessary to begin communication.

Preparing the Instrument to Receive Commands

At this point, the VX4101A can receive the word serial *Begin Normal Operations*. A subset of instrument commands are available at this point to enable you to query instrument status. For a full explanation of these commands, consult the Syntax and Commands section. The available commands are as follows:

Table 1–6: Commands Available at Power-On

Command Syntax	Command Type	Description
*CLS	IEEE 488.2	Clears all event status registers and queues.
*ESE	IEEE 488.2	Sets the contents of the IEEE 488.2 Standard Event Status Enable Register.
*ESE?	IEEE 488.2	Queries the contents of the IEEE 488.2 Standard Event Status Enable Register.
*ESR	IEEE 488.2	Sets the contents of the IEEE 488.2 Standard Event Status Register.
*ESR?	IEEE 488.2	Queries the contents of the IEEE 488.2 Standard Event Status Register.
*RST	IEEE 488.2	Resets VX4101A components to their reset values.
NOTE: *See the command summary for the *RST command in the section* IEEE 488.2 Commands *for the specific reset values.*		
*SRE	IEEE 488.2	Sets the contents of the IEEE 488.2 Service Request Enable Register.

Table 1–6: Commands Available at Power-On (Cont.)

Command Syntax	Command Type	Description
*SRE?	IEEE 488.2	Queries the contents of the IEEE 488.2 Service Request Enable Register.
*STB?	IEEE 488.2	Queries the contents of the IEEE 488.2 Status Byte Register.
STATus:OPERation:CONDition?	Status and Events Reporting System	Returns the current operational status of the VX4101A
STATus:OPERation:ENABle	Status and Events Reporting System	Sets the Operational Enable Register for the VX4101A
STATus:OPERation:ENABle?	Status and Events Reporting System	Queries the Operational Enable Register for the VX4101A
STATus:OPERation[:EVENt]?	Status and Events Reporting System	Returns contents of Operational Event Register for the VX4101A.
STATus:OPERation:NTRansition	Status and Events Reporting System	Sets the Operational Negative Transition Filter for the VX4101A.
STATus:OPERation:NTRansition?	Status and Events Reporting System	Queries the Operational Negative Transition Filter for the VX4101A.
STATus:OPERation:PTRansition	Status and Events Reporting System	Sets the Operational Positive Transition Filter for the VX4101A
STATus:OPERation:PTRansition?	Status and Events Reporting System	Queries the Operational Positive Transition Filter for the VX4101A
STATus:PRESet	Status and Events Reporting System	Clears the enable registers of all Operational Status Registers, sets all Positive Transition Filters, and clears all Negative Transition Filters
STATus:QUEue:ENABle	Status and Events Reporting System	Allows you to specify which errors and events, by error number, should be placed in the error/ event queue
STATus:QUEue[:NEXT]?	Status and Events Reporting System	Returns next item from error/event queue in FIFO order
STATus:QUEue:ENABle?	Status and Event Reporting System	Queries the Questionable Enable Register for the VX4101A
STATus:QUEStionable[EVENt]?	Status and Event Reporting System	Returns contents of Questionable Event Register for the VX4101A
STATus:QUEStionable:CONDi-tion?	Status and Event Reporting System	Returns contents of Questionable Condition Register for the VX4101A
STATus:QUEStionable:ENABle	Status and Event Reporting System	Sets the Questionable Enable Register for the VX4101A

Table 1–6: Commands Available at Power-On (Cont.)

Command Syntax	Command Type	Description
STATus:QUEStionable:ENABle?	Status and Event Reporting System	Queries the Questionable Enable Register for the VX4101A
STATus:QUEStionable:PTRansition	Status and Event Reporting System	Sets the Questionable Positive Transition Filter for the VX4101A
STATus:QUEStionable:PTRansition?	Status and Event Reporting System	Queries the Questionable Positive Transition Filter for the VX4101A
STATus:QUEStionable:NTRansition	Status and Event Reporting System	Sets the Questionable Negative Transition Filter for the VX4101A
STATus:QUEStionable:NTRansition?	Status and Event Reporting System	Queries the Questionable Negative Transition Filter for the VX4101A
SYSTem:ERRor?	VX4101A MultiPaq ™ Instrument Commands	Returns in first-in first-out order any error messages which have been queued for any instruments
SYSTem:OPTions?	VX4101A MultiPaq ™ Instrument Commands	Returns the options for which the VX4101A is currently configured
SYSTem:READy?	VX4101A MultiPaq ™ Instrument Commands	Queries for completion of the power-on initialization sequence for all instruments
***NOTE:** Polling the instrument, i.e. querying it as fast as possible, might slow the completion of the second initialization stage. You should have a delay of at least one second between SYSTem:READy? queries.*		
SYSTem:SNUMber?	VX4101A MultiPaq ™ Instrument Commands	This query returns the serial number of the VX4101A
SYSTem:VERSion?	VX4101A MultiPaq ™ Instrument Commands	Returns the firmware version of the VX4101A

***NOTE**. Until initialization is complete, the VX4101A will recognize only a limited set of commands listed above. Attempting to enter any other command will return a Command Not Found Error.*

Querying the Instrument. If you are using SCPI commands to control the instrument, use the following query to determine if the VX4101A has completed its initialization sequence:

```
SYSTem:READy?
```

The format and syntax for the command strings are described in the *Command Syntax* section. A complete description if each command in alphabetical order is in the *Command Descriptions* section.

Using VXI*plug&play* Device Drivers. If you are using instrument drivers to control the VX4101A, instrument initialization occurs automatically. A SYStem:READy? query is sent upon execution of *tkvx4101.init*. For more information on
VXI *plug&play* instrument drivers, see the Installation section of this manual or the online help accompanying the instrument.

LEDs at Power-On. At power-on following a successful interface test, the front panel LEDs will be in the following states:

Power LED	On
Fail	Off
Message LED	Off
ERR LED	Off

SYSFAIL* Operation

SYSFAIL* will operate under the following circumstances:

If any fuse opens, the module will assert SYSFAIL* on the VXIbus.

If the +5 V fuse opens, the VXIbus Resource Manager will be unable to assert SYSFAIL INHIBIT to disable SYSFAIL*.

***NOTE**. If a +5 V fuse opens, remove the fault before replacing the fuse.*

Replacement fuse information is given in the Specifications *section of the manual.*

Functional Check

A VXIbus hard reset occurs when another device, such as the VXIbus Resource Manager, asserts the backplane SYSRESET* line. A VXIbus soft reset occurs when another device, such as the Slot 0 Controller, sets the Reset Bit in the VX4101A Control Register.

Self-Test

The VX4101A has a two level hierarchy of self-tests. At power-on or at a VXIbus hard or soft reset, the instrument runs an extensive interface test. IEEE

488.2 commands can be used to run more extensive self-tests of instrument specific functions. During the interface test, or during a hard or soft reset, the following actions take place:

1. The backplane SYSFAIL* line is asserted, indicating that the module is not ready for communication.
2. A test of the VXIbus interface logic is performed.
3. Each instrument is configured to its *RST initial state.
4. On successful completion of the interface test, the backplane SYSFAIL* line is negated and the VX4101A enters the VXIbus PASSED state (ready for normal operation). If the interface test is not successful, the backplane SYSFAIL* line remains asserted and the VX4101A enters the VXIbus FAILED state.

Instrument Self-Tests. Instrument specific self-tests can be run at any time during normal operations. To run self-test solely on the active instrument, send the TEST:ALL? query. To test all instruments on the VX4101A, send the *TST? query. See the instrument specific *Syntax and Commands* section for each instrument for more detailed information on each self-test.

During a commanded self-test:

1. The backplane SYSFAIL* line is not asserted.
2. At the end of a successful instrument self-test, the instrument is placed in its *RST (power-on) configuration.
3. At the end of a successful *TST?, the entire VX4101A will be placed in its *RST (power-on) state.

Operational Check

This section tells you how to do an operational check of the VX4101A instruments using the soft front panel (SFP). Depending on the configuration you have purchased, the operational check includes the following procedures:

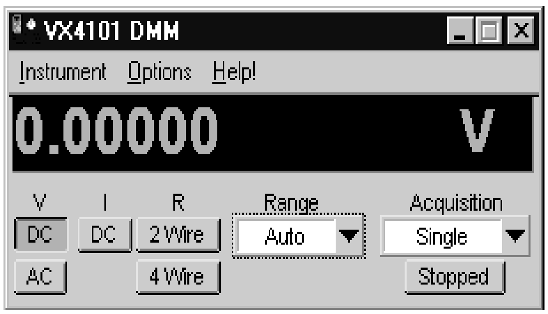
- Standard configuration
- Option 1D configuration
- Option 1A configuration

Operational Check for Standard Configuration

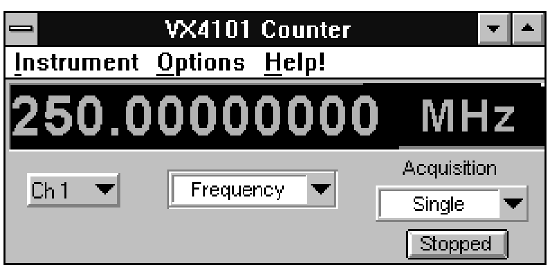
To perform the operational check, do the following:

1. Double click on the TKVX4101A icon to start the SFP. The Tektronix VX4101A MultiPaq™ SFP displays on the screen:

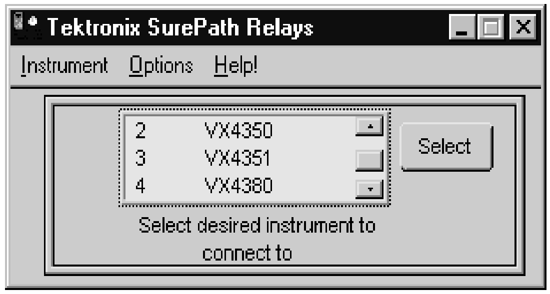
2. Launch the DMM by clicking on the DMM icon.



3. Click on *Stopped* and the DMM will take a DC Voltage measurement.
4. Close the DMM and launch the Counter by clicking on the icons.



5. Using the center pull-down ring control select DC Volts and click on the Stopped button. The Counter will take a DC Voltage measurement.
6. Close the Counter and launch SurePath™.



7. Any SurePath™ modules that are installed immediately to the right of the VX4101A in the card cage will appear in the list box. Select the module number that you want to use to close a relay.

***NOTE**. If there are no SurePath™ modules in the system, the list box will be empty and a Demo button will be available to the right of the list box. Use the Demo button to show how the VX4101A works with SurePath™ modules.*

---

8. Close SurePath™ and do one of the following:
   - Exit the Soft Front Panel
   - Perform the operational check(s) for the options 1D or 1A

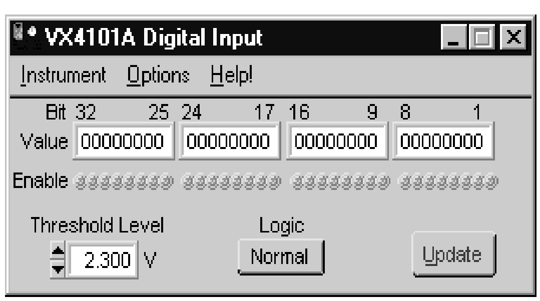
Option 1D Operational Check

If you have purchased option 1D, your VX4101A includes the following instruments:

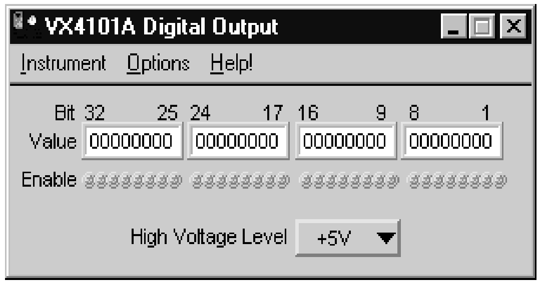
- Digital Input
- Digital Output
- Relay Drivers

1. Perform the Option 1D operational check by launching the following instruments:

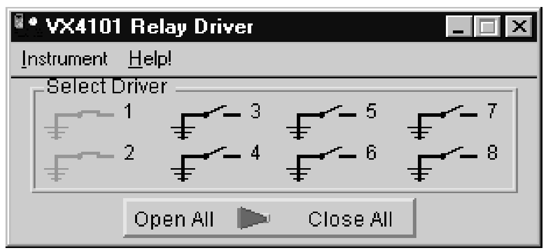
   a. Digital Input



b. Digital Output:



c. Relay Driver:



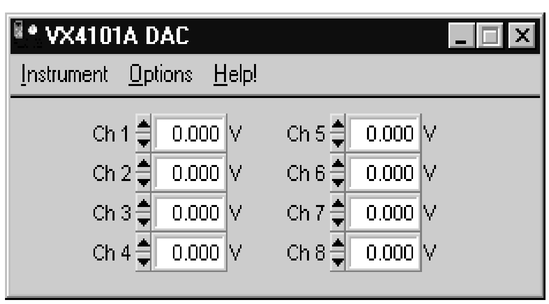
2. Close the Digital Input, Digital Output, and Relay Driver and do one of the following:

- Exit the Soft Front Panel
- Perform the operational check(s) for option 1A

Option 1A Operational Check

If you purchased Option 1A, your VX4101A includes the Digital to Analog Converter (DAC) . Perform the operational check as follows:

1. Launch the DAC:



2. Close the the DAC and exit the Soft Front Panel

Installation Checklist

Revision Level:___________

Serial No.: ___________

Mainframe Slot Number: ___________

Switch Settings:___________

VXIbus Logical Address Switch: ___________

Cable Installed (if any): ___________

VXI*plug&play* software installed: ___________

Notes

Performed by: _________________________ Date: ______________

Operating Basics

About Global and Instrument Commands

The SCPI command set included with the VX4101A controls both overall functions of the MultiPaq™ instrument, and commands with unique behaviors for each individual instrument associated with the MultiPaq™ Instrument. The command set includes the following types of commands:

- Global commands
- Instrument commands

For detailed descriptions of global commands and specific commands for each instrument module, see *Syntax and Commands*.

About Global Commands

Global commands and queries have the same behavior regardless of which instrument you select. Examples of global commands include *RST, *IDN?, and INSTrument:SELect.

About Instrument Commands

Instrument commands and queries have different behavior depending upon which instrument you select. An example of an instrument class query is MEASure:VOLT:DC?. This query is valid for both the DMM and the Counter, but each instrument uses them differently.

The SCPI INSTrument:SELect command enables the VX4101A to differentiate the same command, such as MEAS:VOLT:DC?, from its unique functionality with each instrument. This command permits the MEASure command to be directed to a specific instrument. An example of how each instrument uses the associated INSTRument command is described in *Syntax and Commands* for each each instrument.

***NOTE**. Only one instrument can be selected at a time. This is referred to as the active instrument. The active instrument is the only device able to receive commands. Other instruments can be executing previously entered commands or queries, such as scan or measure commands, in parallel with the active instrument.*

VX4101A Operational Modes

The VX4101A MultiPaq™ Instrument incorporates multiple instruments in a single C-Size VXIbus slot. This maximizes functionality while minimizing cost and space. The innovative design enables you to use each instrument of the VX4101A as a conventional instrument compliant with IEEE 488.2 protocols. You can also override IEEE 488.2 protocols and query multiple instruments simultaneously. The two operational modes are as follows:

- Synchronous mode
- Asynchronous mode

For an example of how to use Asynchronous mode, see *Using the VX4101A MultiPaq™ Instrument*.

About Synchronous Mode

The Synchronous mode of the VX4101A is based on strict IEEE-488.2 compliance. In the 488.2 model, all instrument activity is sequential.

The IEEE 488.2 standard requires that a query be completed prior to the receipt of any new commands or queries. A query is completed after the controller has read the response. For more information on 488.2 compliance, see Chapter 6, *Message Exchange Control Protocol*, of *the IEEE 488.2 Standard*.

Under 488.2 protocols, the other instruments are unavailable during a query. The 488.2 Message Exchange Protocol Enforcer (MEPE) restricts commands and queries to another instrument module from being sent while a query is in progress.

***NOTE**. The default settings for the VX4101A are fully IEEE-488.2 compliant.*

About Asynchronous Mode

The multiple instrument architecture of the VX4101A enables you to send commands to one instrument while receiving query responses from other instruments. You receive the responses non-sequentially, which decreases test times and lowers system costs. The instrument accomplishes this with an alternate mode of operation, the Asynchronous Protocol.

Asynchronous Protocol allows asynchronous query and command of instruments. When you enable the Asynchronous Mode, the VX4101A stores responses in a queue. When you request data from this queue via a VXI read, you receive responses in first-in, first-out (FIFO) order, regardless of which instrument produced the response. Since these responses can be returned in a different order than the queries, the VX4101A uses a different method of associating a response with the associated query. This is described in a later section entitled *Response Formats***.**

The Asynchronous Protocol ensures that the response queue never blocks the parser, as follows:

1. One entry in the queue is reserved for the overflow message.

2. If the queue becomes full, any subsequent writes to the queue will result in a loss of the message.

   If a message has been lost in this fashion, then the next VXIbus read will result in transmission of the error message in the buffer overflow. Data which is already in the response queue will remain there until it is read.

The 488.2 synchronization commands, *WAI, *OPC, and *OPC? are handled as global synchronization devices by the Asynchronous Protocol. This allows you to explicitly control the order of commands to multiple instruments on a single device.

NOTE. *Although it is possible to switch back and forth between strict 488.2 enforcement and the Asynchronous Protocol during instrument operation, Tektronix recommends that you choose one mode or the other and leave the instrument in that mode until it is reset. Switching back and forth during operation could cause the instrument to enter an undesirable state under some circumstances.*

Asynchronous Mode Commands

This section lists and describes the SCPI command set used in the Asynchronous Protocol. The command structure is as follows:

```
SYSTem:LANGuage [["SYNChronous"] | "ASYNchronous"]
```

This command allows you to choose between strict 488.2 protocol enforcement and the Asynchronous Mode. The default state of this feature will be *SYNChronous***,** which implies strict 488.2 enforcement. An argument of *ASYNchronous* enables the Asynchronous Mode. Both query and command are supported.

If the currently selected language is *SYNChronous*, any attempt to use an Asynchronous Mode command or query will generate the following error message:

-210,“Settings conflict;Execute <:SYST:LANG ASYN> first”

```
SYSTem:RQUeue:SNUMber:STATe "ON"|"OFF"|1|0
```

This command turns the *<sequence #>* style of tags on or off. See the section *Response Formats* for detailed information. Both query and command are supported.

```
SYSTem:RQUeue:SNUMber[:SET] <seq #>
```

This command sets the next *<sequence #>* to be used. See the section *Response Formats* for detailed information. Both query and command are supported.

```
SYSTem:RQUeue:QMODe [[ "NEXT "] | "ALL"]
```

This command selects the instrument’s query mode while using the Asynchronous Protocol. Both command and query are supported.

The default is the *NEXT* mode. In this mode, a VXIbus read is interpreted as having been preceded by an implicit *SYSTem:RQUeue[:NEXT]?* query requesting the next entry from the queue. For every VXIbus read supplied, one response will be returned from the response queue.

In the *ALL* mode, a VXIbus read is interpreted as having been preceded by an implicit *SYSTem:RQUeue:ALL?* query requesting all entries currently residing in the response queue. For every VXIbus read supplied, all entries in the response queue will be returned in FIFO order, as one message separated by a semicolon. The response queue is emptied as described under the *SYSTem:RQUeue:ALL?* query.

Asynchronous Query Responses

When using Asynchronous Mode, query responses can be returned in a different order than their associated queries. Each response must be tagged with a unique identifier associating the response with the correct query. The general form of a query response in the Asynchronous mode is as follows:

```
<tag>,<response>
```

The *<tag>* field can have two possible user-selectable formats:

- Sequence name
- Sequence number

Sequence Name Format

This format shows each instrument name in ASCII format. In the default, the <tag> field is defined as follows:

```
"<instrument name>:<canonical query string>"
```

The *<instrument name>* is the same name which would be used by the command *INSTrument:SELect <instrument name>*. For global queries not associated with a specific instrument, the *<instrument name>* used is *VX4101A*.

The *<canonical query string>* is based upon the original query which generated this response. The canonical form is the query string expanded to include all default nodes, with each node represented in its short form in upper case ASCII.

For example, if the MEAS:FREQ? query had been issued to the Counter, the response from the queue using this format would be:

```
"Counter:MEAS:FREQ?", 1.00000000000000E+07
```

When the VX4101A is using the Asynchronous Protocol and the response queue is empty, a VXI word serial read will cause the VX4101A to return a default message. This ensures that a query will not tie up the VXI backplane.

The *<response>* field is identical to the query response returned by the instrument under strict IEEE 488.2 enforcement.

***NOTE**. For queries taking a significant amount of time to return, it is possible to use a software function, such as a service request generated by activity in the SCPI Status registers, rather than polling for the response. This determines when a response is available.*

When there is no data in the response queue, the following response will be generated in this mode:

```
"VX4101A:RQU?","EMPTY"
```

When there is an overflow in the response queue, the following response will be generated in this mode:

```
"VX4101A:RQU?","OVERFLOW"
```

Sequence Number Format

The second <tag> field format, which is selected with *SYSTem:SNUMber:STATe ON*, is as follows:

```
<instrument number>:<sequence number>
```

The *<instrument number>* is the same number which would be used by the command *INSTrument:NSELect <instrument number>*. For global queries not associated with a specific instrument, the instrument number used is *0*. The sequence number is a one-up number assigned to a query as it is received by the card. As a query is received, it is assigned the current sequence number. The next query received will have a sequence number one larger than the last.

NOTE. *Note that sequence numbers are global to the card and have no direct correlation with the instrument number.*

The sequence number defaults to zero on power-on. To set the sequence number to a desired value, use the following command:

```
SYSTem:RQUeue:SNUMber[SET]<sequence number>
```

After the instrument receives this command, the next received query will be assigned the <*sequence number*> specified in the command. Each subsequent sequence number will increment by 1 each time a query is received.

The query *SYSTem:RQUeue:SNUMber[:SET]?* can be used to determine what the next sequence number will be. Since the response to this query will contain a sequence number itself (assuming numerical tags are enabled), the response to this query will be the sequence number which will be used by the next query received.

For example, if the message is the sixth message in the queue and the Counter identification number is 3, the response for the example above would be:

```
3:6,1.00000000000000E+07
```

The sequence number formats enable you to identify the instruments according to your own unique requirements. Using the longer ASCII format allows you to type commands into a talker/listener to get easily readable feedback about the source of the current response. The more terse numerical format is easier to parse in automated test software.

When there is no data in the response queue, the following response will be generated in numeric mode:

```
0:-1,"EMPTY"
```

When the response queue has overflowed, the following response will be generated in numeric mode:

```
0:-2,"OVERFLOW"
```

About Instrument Triggering

For digital multimeters, scanners, and sources such as the DAC and Digital I/O, it is common to think of the devices as triggering to start an operation. For counters, on the other hand, it is more common to think of the counter as arming to take a measurement.

Triggering and arming the VX4101A and its associated instrument modules can be considered to be the same operation. Any differences will be noted. For example, in addition to the STARt arming source, the Counter implements a STOP arming source. Counting is enabled when the start source is asserted and disabled when the stop source is asserted.

Trigger Sources

For the SurePath™ master control, triggers are implemented in firmware, while for the other instruments, the triggers are implemented in hardware. For this reason, the SurePath™ timers have less resolution and accuracy than the other instrument triggers. Since the SurePath™ trigger times are of the same order as typical settling delays, impact to instrument performance is negligible.

Table 2–1: VX4101A Global Trigger Sources

Trigger	Source
HOLD	Instrument trigger source is disconnected from all trigger sources. If you INITiate an instrument with a trigger source of HOLD, the instrument will only enter the triggered state upon receipt of a TRIGger:IMMediate command. This trigger is fixed.
IMMediate	Device enter the triggered state immediately after receiving the INITiate command. This trigger is fixed.
BUS	Trigger source is either a word serial trigger command or the 488.2 common command *TRG. This trigger is delayable.
EXTernal	This trigger source is derived from the front panel MultiPaq™ External Trigger In, Pin24B signal. This input signal is common to all instruments. This trigger is delayable.
TTLTrigger<0-7>	One of the VXIbus TTL trigger lines. This trigger is delayable. TTLT sources cannot be routed to other TTLT sources.
COMMand<0-4>	One of five software command triggers. TRIG:FIR<N> is used to send these triggers. This trigger is delayable.
TIMer	Trigger source is the programmable periodic timer. This trigger is fixed.
SUREPATH	Trigger is generated whenever a scan line has switched and settled. This trigger is delayable.

Table 2–1: VX4101A Global Trigger Sources (Cont.)

Trigger	Source
DMM	Trigger is generated whenever the DMM has completed a measurement. This trigger is delayable.
COUNTER	Trigger is generated whenever the Counter has completed a measurement. This trigger is delayable.
CTR_EXTARM	This trigger is the digital representation of the analog signal input into the Counter front panel arm signal. This trigger is delayable.
DAC	This trigger can have one of three modes, depending on how the DAC is configured in trigger mode: it is generated either after each sample is output, at the end of the current segment, or at the end of the sequence. This trigger is delayable.
DIGI	This trigger is generated either after each sample is collected or when the specified data has been collected. This trigger is delayable.
DIGO	This trigger can have one of three modes, depending on how the trigger mode is configured: it is generated after each sample is output, at the end of a current segment, or at the end of the sequence. This trigger is delayable.

Table 2–2: VX4101A Counter-Specific Trigger Sources

Trigger	Source
CTR_CHAN2	The signal on channel two of the Counter can start a measurement on channel one of the counter. This source is fixed and can only be used as an ARM:STARt source on channel one of the Counter.
INTernal	This fixed source is only valid as an ARM:STOP source and is used to specify that other internal settings on the Counter (either the aperture or delay-by-events counter) will be used to stop the Counter.
LEVel	This fixed source is valid as an ARM:STOP source. It is used to specify that the ARM:STARt source is level-sensitive instead of edge-sensitive. The Counter will remain armed as long as the start source is asserted.

Additional Instrument-Specific Trigger Sources

In addition to the above trigger sources, the Digital Multimeter (DMM), Digital to Analog Converter (DAC), Digital Input, and Digital Output use the following fixed source:

Table 2–3: VX4101A Fixed Trigger Sources for the DMM, DAC, Digital Input and Digital Output

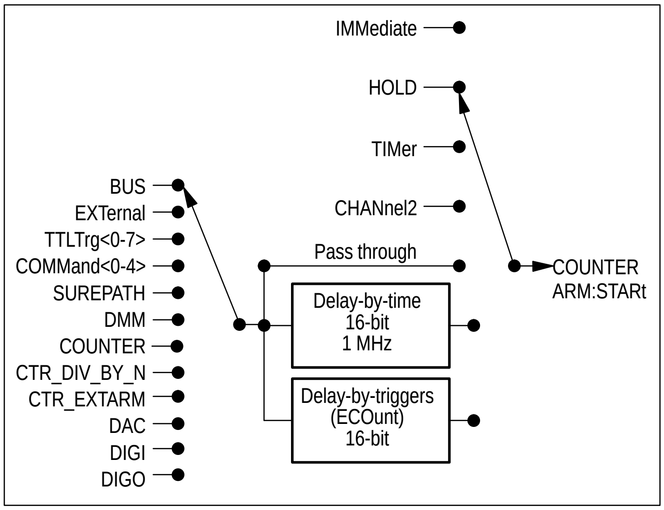
Trigger	Source
HANDshake	Derived from one of the front panel REQUEST signals. There is one source for each instrument. When you select the handshake or trigger source, the operation triggers on the leading edge of a programmable polarity pulse of this signal. When the operation ends, the instrument generates an acknowledge signal on the front panel ACKNOWLEDGE STROBE signal associated with the generating instrument. You can also program the polarity of the ACKNOWLEDGE STROBE pulse. For information on handshake pin assignments, see Appendix *B:Input/Output Connections*. See Appendix A: Specifications for additional information.

VX4101A Trigger Architecture

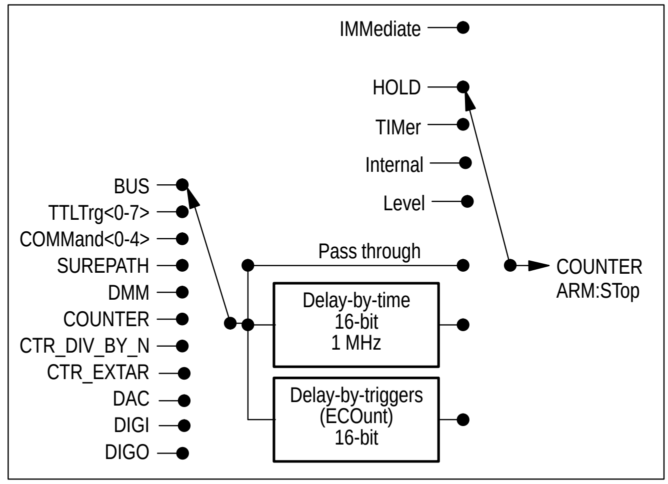
Figure 1 shows the trigger architecture of the VX4101A. Note that not all trigger modes and sources are common to all instruments. Most trigger sources can be routed to any of the VXIbus TTL trigger lines or to an external trigger output on the front panel. The two types of trigger sources are as follows:

- *Fixed* trigger sources are always in pass-through mode
- *Delayable* trigger source supports three trigger modes: pass-through, delay by time, and delay by trigger

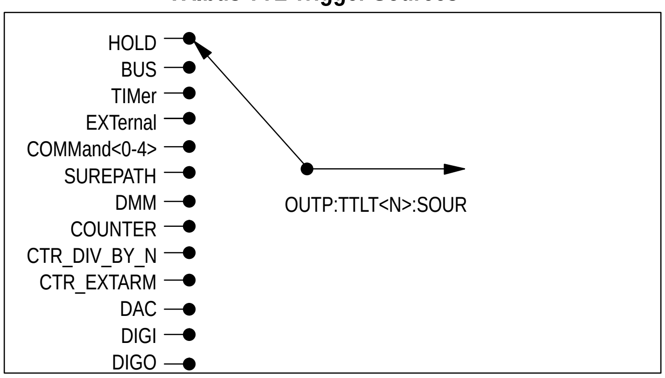
ARM:STARt Sources for Counter



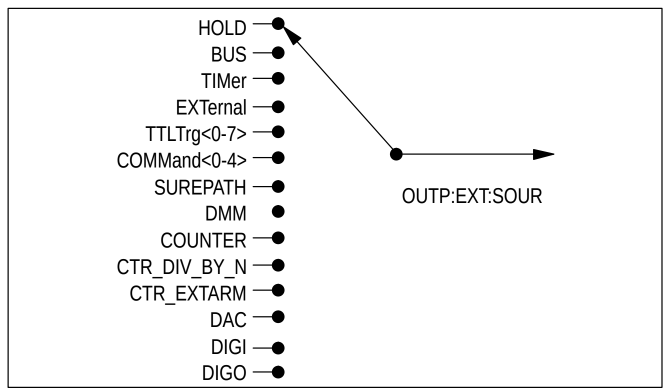
ARM:STOP Sources for Counter



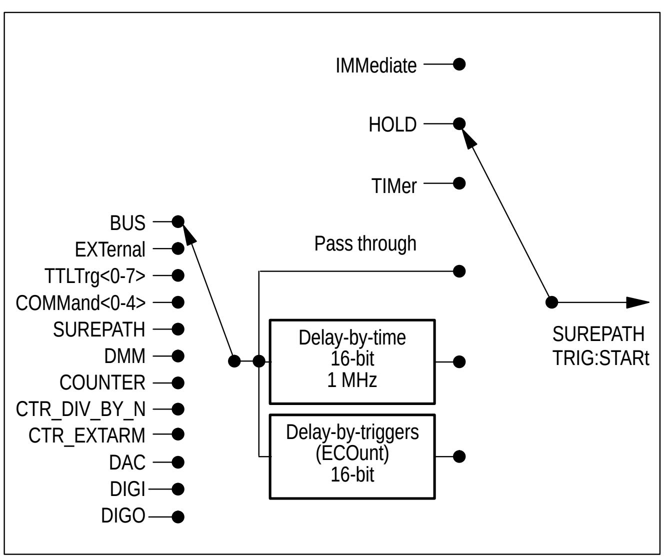
VXIbus TTL Trigger Sources



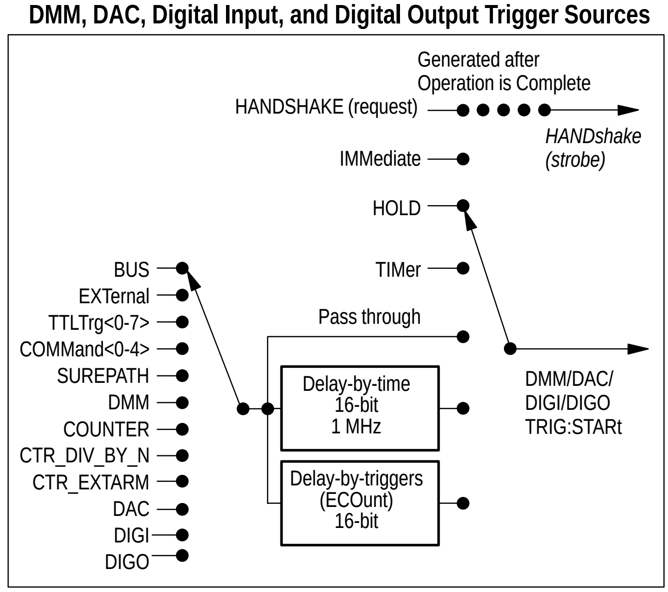
VX4101A Trigger Output Sources



SurePath™ Trigger Sources



DMM, DAC, Digital Input, and Digital Output Trigger Sources



About Fast Data Channel (FDC) Operation

The following section tells you about using the Fast Data Channel (FDC), a high-speed protocol for data transfer, with the VX4101A MultiPaq™ Instrument.

The VXI*plug&play* driver provided with the VX4101A implements FDC and does not require full knowledge of the protocol. The description that follows is only for users who choose not to use the driver.

***NOTE**. For more information on using VXIplug&play drivers to implement FDC, see the online help files accompanying the driver software.*

About FDC

Fast Data Channel (FDC) is a communications protocol for transferring data between a VXIbus Commander device and its Servant device. The protocol is standardized, approved by the VXIbus Consortium in a standard known as VXI-10. It is supported by VXIbus instrument hardware and software vendors. FDC was recognized as a supported protocol by the VXI*plug&Play* Alliance in 1996.

FDC is a bi-directional, block oriented, data transfer mechanism that uses shared memory and is well suited for transferring large buffers of data between a host and a servant device. Bit flags in the FDC header word allow transfer of data larger than the FDC buffer memory. Drivers supporting the protocol must exist on each Host/Servant pair employing FDC.

The FDC protocol defines the establishing of FDC channels, setting up and coordinating of the data transfers, channel termination, and error handling. In order to use the FDC protocol for accessing instrumentation data, you must use your application code to establish FDC channels, invoke FDC I/O drivers to actuate the data transfer, close the the channels at termination, and handle any errors that resulted from the data transfer. Vendor supplied I/O libraries are generally required.

The FDC Process

In a typical VXIbus instrumentation system, the Commander device is your host computer, and the Servant device is the instrument. A typical user test application employing FDC would contain many of the following operations:

1. Establish a communication session with the instrument

2. Configure and open the FDC channel

3. Transfer data via the FDC channel

4. Close the FDC channel

5. Terminate the instrument session

***NOTE**. The Commander/Servant hierarchical structure of the VXIbus architecture requires the Commander device to initiate communication with the Servant device.*

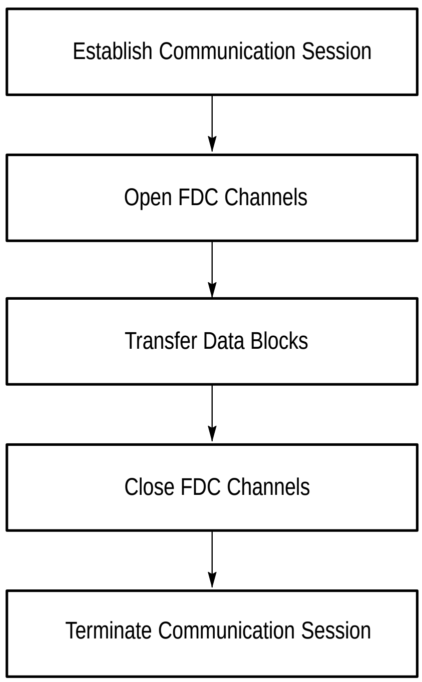


Figure 2–1: Typical FDC Process

Establishing a Communications Session

Before VXIbus instruments can communicate with each other, you must establish an I/O linkage between the instruments. The complexity of this communications session varies with the host environment, but typically involves Operating System calls that return I/O driver reference handles. You must use both Word Serial Protocol (WSP) as well as Fast Data Channel I/O drivers to set up, configure, and terminate FDC channels. Only actual data buffer transfers use the physical FDC channel.

Opening and Configuring an FDC

After the WSP and FDC I/O drivers are established between the Commander and Servant Device (typically a host computer and the instrument), an FDC channel has to be configured and opened for operation. Configuring the channel defines the transfer characteristics, such as access protocols, data width, and others. Opening the channel identifies the particular FDC channel to be used, its location in shared memory, and its maximum size.

NOTE. *Multiple FDC channels may be established concurrently between a Commander and Servant pair.*

Transferring Data Via the FDC Channel

Data can be downloaded from the host to the instrument as well as extracted from the instrument and transferred back to the host. Prior to the actual data transfer, the host must define the direction of the data transfer. The transfer operations work as follows:

Transferring Data from the Host to the Instrument. Source instruments are candidates for FDC data transfers from the host to the instrument. In this scenario, a buffer of data is transferred from the host application to the instrument for processing.

Data Transfers from the Instrument to the Host. Measurement instruments are candidates for FDC data transfers from the instrument to the host. In this scenario, a buffer of data is transferred from the instrument back to the host for processing.

Bi-Directional Transfers. FDC channels can be bi-directional. For example, some applications might require retrieval of blocks of data from a source instrument, transmit blocks of data to a measurement instrument, or repeatedly transmit and receive data between instruments.

Closing the FDC Channel

When your application program completes its FDC data transfer, you must close the FDC channel to recover the system resources required to maintain the channel, as well as to put the I/O connection in an appropriate state to re-establish a channel for the next FDC transfer.

Terminating the Instrument Session

This activity terminates the I/O session between the VXIbus instruments and releases the I/O driver's system resources.

FDC Operation with the DMM and DAC

The VX4101A MultiPaq™ Instrument uses FDC protocol only with the Digital Multimeter (DMM) and the Digital to Analog Converter (DAC). The DMM is a measurement device and generates data that is transferred back to the host. The DAC is a source device and accepts data from the host. The DAC can also be programmed to return data back to the host.

About Physical and Logical Channels

The VX4101A FDC incorporates both physical and logical channels. The FDC I/O drivers on the host and instrument device communicate with each other via the physical FDC channel established between them.

Depending on the host application and the instrument configuration, you should refer to the FDC channel in your application program with a logical channel number. The logical channel number should always be the same for each instrument that employs FDC.

Each instrument's logical FDC channel numbers range from 1 to the number of FDC channels used by the instrument.

***NOTE**. Since the DMM and DAC each only support one FDC channel, their logical FDC channel number is always 1.*

To support the logical channel, however, your application software must retrieve configuration information from the Servant device channel to supply to the Host's FDC drivers prior to using the FDC channel.

FDC SCPI Commands

The VX4101A supports the FDC protocol with the VXI:FDC SCPI Subsystem. The following commands comprise the FDC command set:

```
VXI[:SERVant]:FDC?
```

Returns comma separated list of physical FDC channel numbers assigned to the VX4101A.

```
VXI[:SERVant]:FDC:SEL [<channel number>]
```

Selects the FDC logical channel number.

```
VXI[:SERVant]:FDC:SEL?
```

Returns the FDC current logical FDC channel number.

```
VXI[:SERVant]:FDC:OPEN <mode> [,<channel number>]
```

Opens the selected FDC channel in the selected mode (Read Only, Write Only or Read/Write).

```
VXI[:SERVant]:FDC:CLOSe [<channel number>]
```

Closes the selected FDC channel.

```
VXI[:SERVant]:FDC:CONFiguration?
```

Returns configuration data on FDC channel.

```
VXI[:SERVant]:FDC:BUFFer <buffer size>
```

Defines the size of data for a source instrument to return.

NOTE. *See the Syntax and Commands section for the DMM or the DAC for a full explanation of the VXI:FDC command subsystem.*

FDC Example

The following scenario and example code demonstrates how you can use the FDC data transfer capabilities of the VX4101A.

In this scenario, the Unit Under Test (UUT) is a device that processes a variety of waveforms. You must write a production test that sequences through various patterns to verify that the device processes the waveforms properly. The VX4101A DAC instrument will be the source of the waveform pattern to the UUT.

Because of the large amount of data to construct the waveforms, you will use FDC to transmit the raw data from the host machine to the DAC. You will use a VXIbus embedded computer as the host machine and will run the Commander-side FDC drivers supplied by the embedded computer's VISA software.

NOTE. *Please refer to the National Instrument's NI-VISA Programmer's Reference Manual for a complete description of the command syntax used in these examples.*

Establishing a Communications Session

You must establish the I/O linkage between the host and the VX4101A DAC. You use both Word Serial Protocol (WSP) and FDC I/O drivers. Two communications sessions will be required: one for WSP and one for FDC.

Command Syntax. You use the following command syntax:

```
viOpen (session, resource name, access mode, timeout, vi);
```

Example Code. You can use the following code to open and close the FDC session:

```
errs = viOpen (visaRM, VX4101Desc, VI_NULL,VI_NULL, &fdc);

errs = viOpen (visaRM, VX4101Desc, VI_NULL, VI_NULL , &wsp);
```

Successful execution of these commands returns the associated I/O driver reference handle in the parameter pointed to by *wsp* or *fdc*.

Configuring the FDC Channel

In this section, you will prepare the FDC channel to transmit data.

Before You Begin. Before the channel can be configured, the host needs to know which physical FDC channels are supported by the instrument. You can retrieve this information from the instrument by issuing the following query:

```
VXI:FDC?
```

This query returns a comma separated list of integers that represent the numeric values of the physical FDC channels supported by the instrument.

Command Syntax. This is an ASCII command and is sent through the WSP channel using the VISA functions, viRead and viWrite. Use the following the syntax to program FDC to read and write data:

```
viRead (session, buff, max count, return count);

viWrite (session, buff, length(buff), return count);
```

Example Code for Configuring the FDC Channel. You can use the following code to prepare the FDC channel to transmit data:

```
errs = viWrite (wsp, "VXI:FDC?", 8, &RetCnt);

errs = viRead (wsp, buff, sizeof(buff), &RetCnt);

fdc_physical_channel = atoi(buff);
```

Since the DMM and DAC only have 1 FDC channel, the list returned by either device contains a single channel value. This physical FDC channel number is passed to the host FDC drivers through the Host session with a viSetAttribute command. Use the following command syntax:

```
viSetAttribute (vi, attribute, attribute state);
```

Example Code for Passing Values to the Drivers. You can use the following sample code to pass the channel values to the FDC drivers:

```
errs = viSetAttribute (fdc, VI_ATTR_FDC_CHNL,
fdc_physical_channel);
```

The Commander and Servant FDC drivers are now assigned to the same physical channel.

Other Attributes

Other FDC attributes may also be specified. The host computer vendor's FDC support documentation will detail all the configuration parameters required for your FDC driver implementation.

After the FDC drivers are configured, you must command the servant device to open the logical FDC channel associated with its physical channel. The FDC:SEL? query returns the currently selected logical fdc channel:

```
errs = viWrite (wsp, "VXI:FDC:SEL?", 12, &RetCnt);

errs = viRead (wsp, buff, sizeof(buff), &RetCnt);

fdc_logical_channel = atoi(buff);
```

***NOTE**. For the VX4101A DAC and DMM, the above query always returns a 1. You do not need to use this query.*

Opening the FDC Channel

The Open command opens the selected FDC channel in the selected mode, in this case, Read/Write.

Example Code for Opening the Channel.

```
sprintf(buff,"VXI:FDC:OPEN RW %u",fdc_logical_channel);

errs = viWrite (wsp, buff, strlen(buff), &RetCnt);
```

The VX4101A is comprised of a number of independently operating instruments. The previous OPEN command requires the servant device to set up and initialize some of the instrument's internal resources. The time it takes to complete this process varies with the VX4101A configuration and behavior at the time the OPEN command is issued. To ensure that an opened FDC channel is not used to transfer data until it is ready to do so, you should poll the FDC channel to determine if it is ready for data transfers.

The configure query returns the state of the currently selected FDC channel. In this case, the channel state must be READ_WRITE before data transfer can reliably commence.

Example Code for Querying the Channel.

```
while(strncmp(buff,"READ_WRITE",10))
    {
    errs = viWrite (wsp, "VXI:FDC:CONF?", 13, &RetCnt);
    errs = viRead (wsp, buff, sizeof(buff), &RetCnt);
    buff[RetCnt] = 0;
    }
```

Transferring Data

After configuring and opening the FDC channel, data transfers can begin. You can use FDC to quickly and efficiently program an output waveform.

Using ASCII Values. The VX4101A DAC can also be programmed for an output waveform with easy to use, but slow, ASCII values. When programmed in ASCII, the DAC converts the ASCII voltage values to binary data prior to driving its output hardware. When programmed with FDC, the data transferred to the DAC must already be in a binary format suitable for driving the hardware.

In a typical application, you develop waveform patterns with ASCII data. Then, using FDC, the instrument reads back the binary representations of the waveform patterns. During the execution of the production test, the binary representations of the waveforms can be downloaded to the DAC with FDC. This provides a mechanism to rapidly and efficiently transmit a variety of different waveforms from the DAC to the Unit Under Test.

The DAC is specialized for outputting waveforms in real time. In order to read back its output buffers, you must define the amount of data to return. Reception of the return count command signals the DAC to present its output data to the FDC drivers. In the following example code, assume the DAC has already been programmed with ASCII voltage values. The BUFF command defines the return count of the data that was read back and the viRead command invokes the host FDC driver's read function to retrieve the data via FDC.

Example Code for Retrieving Data. Use the following example code to retrieve the data:

```
errs = viWrite (wsp, "VXI:FDC:BUFF 1024", 17, &RetCnt);

errs = viRead(fdc, buff, sizeof(buff), &RetCount);
```

Once the waveform has been returned to the host in binary format, the data can be re-sent to the VX4101A DAC with the host FDC driver's write function.

Example Code for Re-Sending Data.

```
errs = viWrite (fdc, buff, strlen(buff), &RetCnt);
```

Closing the FDC Channels

Reading and writing the FDC channel can be repeated until you have completed testing the unit. At that point the program could close down the FDC channels in preparation for connecting and testing another device.

Example Code for Closing the FDC Channels. You can use the following code to close the FDC channels:

```
errs = viWrite (wsp, "VXI:FDC:CLOSE",13, &RetCnt);
```

Terminating the Communication Session

When all testing is complete, you can terminate the I/O sessions and shut down the test equipment. Use the following command syntax:

```
viClose(vi);
```

Example Code for Terminating the Communication Session. You can use the following code to terminate the session:

```
errs = viClose (fdc);

errs = viClose (wsp);
```

Using the VX4101A MultiPaq™ Instrument

This section includes guidelines and procedures for the following operations of the VX4101A MultiPaq™ Instrument:

- Using Asynchronous Mode

Using Asynchronous Mode

Using Asynchronous Mode, the VX4101A can return responses in a different chronological order than their associated queries. To keep track of the actual order in which the responses are returned, the VX4101A MultiPaq™ Instrument assigns a unique tag to each response. You can define two different formats in which to receive tags:

- ASCII format, which provides feedback in a more easily readable format
- Numeric format, which is more terse, but is easier to parse in automated test software

For more information on VX4101A query formats, see *VX4101A Operations*.

Starting Sequence Number

In the following example the instrument is set up to return queries using ASCII tags.

This example shows how you can use the Asynchronous Protocol to send multiple queries without intervening VXIbus reads and still associate queries and responses, regardless of the order in which they are sent and received.

NOTE*. This example assumes that a 1 MHz signal is connected to channel 1 of the counter and a 0.1 Volt DC signal is connected to the DMM.*

Prerequisites In order to get the responses shown in the following procedures, the instrument must have previously performed a word serial read.

NOTE*. In the strict 488.2 example, sending the word serial read prematurely could cause a bus timeout. In the Asynchronous Protocol examples, performing a word serial read would simply cause the default empty message to be returned.*

---

1. Enter the following command to select Asynchronous Mode:

   ```
   SYST:LANG ASYN
   ```

2. Enter the following query for error conditions:

```
SYST:ERR?
```

After a VXIbus read, the instrument returns the following response (note the <tag> field preceding the response):

```
"VX4101A:SYST:ERR?",0,"No error"
```

3. Send the following two queries without performing an intervening VXIbus read:

```
*IDN?
```

```
INST:CAT?
```

Two VXIbus reads in succession will return the following responses:

```
"VX4101A:*IDN?",Tektronix, VX4101A, B010101, Firmware
v.2.0.0/SCPI:95.0
```

```
"VX4101A:INST:CAT?",VX4101A,SurePath,DMM,Counter,DAC,DIGI,
DIGO,RDriv
```

4. Enter the following command to select the DMM:

```
INST:SEL DMM
```

5. Send the following query to set up and acquire a 10V DC measurement:

```
MEAS:ARR:VOLT:DC? 10,MIN,MAX
```

NOTE*. This measurement takes approximately 35 seconds to complete.*

6. Without attempting to read the response to the DMM query, immediately select the Counter with the following command:

```
INST:SEL Counter
```

7. Send the following query to measure the frequency of the signal on channel one of the Counter:

```
MEAS:FREQ?
```

8. A VXIbus read will return the result of the counter measurement:

```
"Counter:MEAS:SCAL:FREQ?",1.00000000000000E+06
```

9. Another VXIbus read returns a message indicating that the message queue is empty:

```
"VX4101A:RQU?","EMPTY
```

10. Approximately 30 seconds after the DMM measurement was initiated, the results will be ready. At this point a VXIbus read will return these results:

```
"DMM:MEAS:ARR:VOLT:DC?",#3130+1.00000E-01,+1.00000E-01,+1.000
00E-01,+1.00000E-01,+1.00000E-01,+1.00000E-01,+1.00000E-01,+1
.00000E-01,+1.00000E-01,+1.00000E-01
```

11. Another VXIbus read returns a message indicating that the message queue is once again empty:

```
"VX4101A:RQU?","EMPTY
```

Using the Counter

This chapter contains procedures and hardware tips for using the Counter for the following measurements:

- Measuring frequency
- Measuring time interval
- Measuring rise time

What You Should Know About

See the Appendix *Counter Architecture* for a block diagram of how the Counter front end is constructed.

Measuring Frequency

In this procedure, you will make a frequency measurement using a 10 nS aperture.

1. Select the Counter:

   ```
   INST:SEL COUNTER
   ```

2. Set the input 1 coupling to DC:

   ```
   INPut:COUPling DC
   ```

3. Set input 1 impedance to 50 Ω:

   ```
   INPut:IMPedance 50
   ```

4. Set the aperture to 10 nanoseconds:

   ```
   SENSe:FREQ:APER 1e-8
   ```

5. Configure the Counter to read frequency:

   ```
   SENSe:FUNCtion "FREQuency"
   ```

6. Initiate the configuration and take the measurement:

   ```
   INIT
   ```

7. Retrieve the measured data:

   ```
   FETCh?
   ```

Measuring Time Interval

In this procedure, you will make a time interval measurement using external gating.

1. Select the Counter:

```
INST:SEL COUNTER
```

2. Set the input 1 coupling to DC:

```
INPut:COUPling DC
```

3. Set input 1 impedance to 50 Ω:

```
INPut:IMPedance 50
```

4. Set the input 1 signal lowpass filter state On:

```
INPut:FILT ON
```

5. Select the start arming source to be used after the Counter is initiated:

```
ARM:SOURce CTR_EXTARM
```

6. Select the start arming source to be used when the Counter is initiated:

```
ARM:STOP:SOURce COUNTER
```

7. Select the Time Interval function without changing most of the Counter setup:

```
SENSe:FUNCtion "TINTerval"
```

***NOTE**. Input coupling and impedance are not changed.*

8. Set the aperture to minimum:

```
SENS:APER MIN
```

9. Initiate the configured measurement:

```
INIT
```

***NOTE**. Need application specific delay to allow time for triggers to occur.*

10. Retrieve the measured data:

```
FETC?
```

Measuring Rise Time

In this procedure, you will make a rise time measurement using a 10 ns aperture time.

Programming Example

1. Select the Counter:

   `INST:SEL COUNTER`

2. Set input 1 coupling to DC:

   `INPut:COUPling DC`

3. Set input 1 impedance to 50 Ω:

   `INPut:IMPedance 50`

4. Sets the aperture to 10 nanoseconds:

   `SENSe:RTIMe:APER 1e-8`

   or

   `SENSe:RISE:TIME:APER 1e-8`

5. Configure the Counter to read signal Rise Time:

   `SENSe:FUNCtion "RTIMe"`

   or

   `SENSe:FUNCtion "RISE:TIME"`

6. Initiate the configuration and take the measurement:

   `INIT`

7. Retrieve the measured data:

   `FETCh?`

Measuring Time Interval with Delay

Programming Example

1. Select the Counter:

```
INST:SEL COUNTER
```

2. Set input 1 coupling to DC:

```
INPut:COUPling DC
```

3. Set input 1 impedance to 50 Ω:

```
INPut:IMPedance 50
```

4. Set the input 1 signal lowpass filter state to On:

```
INPut:FILT ON
```

5. Set the input 1 to an expected peak-to-peak input voltage and an expected input offset voltage:

```
INPut1:SETup 2,1
```

6. Set input 2 coupling to DC:

```
INPut2:COUPling DC
```

7. Set input 2 impedance to 50 Ω:

```
INPut2:IMPedance 50
```

8. Set the input 2 signal lowpass filter state to On:

```
INPut2:FILT ON
```

9. Set the Input 2 to an expected peak to peak input voltage and an expected input offset voltage:

```
INPut2:SETup 2,1
```

10. Select the Time Interval with delay by time function:

```
SENSe[1,2]:FUNCtion "TINT:DEL:TIME"
```

NOTE*. This command will not change most of the setup for the Counter. Input coupling and impedance are not changed*

11. Set the delay to 1 msec:

```
SENS:TINT:DELay: 1e-3
```

This will return the time interval from the first edge on channel one to the first edge on channel two that is at least 1 msec later.

12. Set the aperture to minimum:

```
SENS:APER MIN
```

13. Initiate the configured measurement:

```
INIT
```

14. Place the result of the measurement in the output buffer so you can read it:

```
FETch?
```

15. Set the time interval with delay by events:

```
SENSe12:FUNCtion "TINT:DEL:EVEN"
```

16. Set the number of events to 10:

```
SENSe:TINTerval:DELay:EVENTS 10
```

This will return the time interval from the first edge of channel one to the tenth edge on channel two.

17. Initiate the configured measurement:

```
INIT
```

18. Place the results of the measurement in the output buffer so you can read them:

```
FETC?
```

Using the Digital Input

This section includes guidelines and procedures for the following operations of the Digital Input:

- Program the Digital Input
- Read the current input
- Read points using external handshake

What You Should Know About

Start-Up

Upon start-up, all Digital Input pins are active.

Shared Pin Assignments with Digital Output

Since both the Digital Input and Output share the same pins, the Digital Input detects any voltage applied to the Digital Output. The 32 pins of the Digital Input physically share the same pins with the Digital Output. If you are using both the Digital Input and Digital Output, each common pin will be set to low if either the Digital Output or your connection to the Digital Input drives it low.

Digital Output Source

If the Digital Output source is external and no external voltage is applied, the Digital Input level of unconnected Digital inputs will be affected by other connected Digital Inputs.

Pin Usage for Only the Digital Input. If you are not using the Digital Output, then you should select one of the following internal excitation values:

- 5 VDC
- 12 VDC
- 24 VDC

Select the level based on whether the measured inputs are nominal 5 VDC or less, 12 VDC or less, or higher than 12 VDC.

About Triggering

You can program the Digital Input to trigger as follows:

- Normal triggers
- On a pattern

Programming the Digital Input

This section describes how to program the Digital Input using the described command set.

1. Select the Digital Input as the active instrument:

```
INST:SEL DIGI
```

2. Set the return format to hexadecimal:

```
FORMAT HEX
```

3. Enable all bits as input:

```
SENSE:PSELECT:DIGLOBAL ENAB
```

4. Set up the Digital Input to take 1000 postmatch measurements:

```
sens:arr 1000
```

5. Set up the threshold value to 2.8 V:

```
sense:threshold 2.8
```

6. Set the match pattern:

```
trig:matc #h00007e38
```

7. Set the mask for the match pattern:

```
trigger:mask #h0000FFFF
```

8. Start the measurements:

```
initiate
```

***NOTE**. Until the match occurs, the instrument will not take the 1000 post-match measurements. You must manually enter the match pattern in order to use this example.*

9. Query the amount of prematch data

```
FETch:COUNt?
```

10. Retrieve all postmatch data:

```
FETch?
```

Reading Current Input

This procedure will show you how to read current input to the Digital Input.

1. Select the Digital Input as the active instrument:

   `INST:SEL DIGI`

2. Set the data format to hexadecimal:

   `FORMat HEX`

3. Enable all input bits:

   `INPut:PSelect:DIGLobal ENAB`

4. Set up the read operation:

   `CONF:DIGL 2.5`

5. Initiate measurement:

   `INIT`

6. Retrieve the most recent values:

   `FETch?`

Reading Points Using the External Handshake Feature

This procedure will show you how to read input points using handshake signals.

1. Select the Digital Input as the active instrument:

   `INST:SEL DIGI`

2. Set the data format to hexadecimal:

   `FORMat HEX`

3. Set up the read:

   `CONF:DIGL 2.5`

4. Set up the instrument to make 20 measurements:

   `sense:arr 20`

5. Program the instrument for one trigger and handshake:

   `trig:mode once`

6. Program the instrument for the external handshake:

   ```
   trig:source handshake
   ```

7. Manually send 20 external input strobes

8. Retrieve the number of measurements taken:

   ```
   fetch:count?
   ```

9. Retrieve the data taken by the measurements:

   ```
   Fetch?
   ```

Using the Digital Output

This section includes guidelines and procedures for the following operations of the Digital Output:

- Outputing one 32-bit word
- Outputing a sequence

What You Should Know About

About Segments and Sequences

You should understand the following definitions:

Segments: a programmable number of points

Sequence: a programmable number of segments

Shared Pins With the Digital Input

The Digital Output is connected to the Digital Input. The 32 pins of the Digital Input physically share the same pins with the Digital Output.

Pin Usage With Digital Input and Output Together. Common pins used by both the Digital Input and Digital Output will be driven low if either instrument drives it low.

***NOTE**. If you are using the Digital Input and want to use a pin only as an input, you must reset the Digital Output for pin to the default state of High.*

Outputing One 32-Bit Word

This procedure will show you how to continuously output a 32-bit word at a specified frequency with the Digital Output.

1. Select the Digital Output:

```
INST:SEL DIGO
```

2. Load the first memory location:

```
TRACe 1,<data>
```

3. Load the sample rate at the specified frequency:

```
TRACe:SRATe <sample output frequency>
```

4. Set the trace length:

```
TRACe:Points 1
```

5. Specify continuous output:

```
INIT:CONTinuous ON
```

6. Begin the sampling operation:

```
TRIGger:IMMEDiate
```

Outputing a Sequence

This procedure will show you how to output a sequence of x segments of y points with the Digital Output. The internal clock is set to 1 kHz.

1. Select the Digital Output:

```
INST:SEL DIGO
```

2. Define the number of data points:

```
TRACe (1:Y),<data 1,...,data Y>
```

3. Set the output sample frequency to 1 kHz:

```
TRACe:SRATe 1000
```

4. Set the segment length to the number of data points Y:

```
TRACe:Points Y
```

5. Set the sequence number:

```
TRIG:MODE SEQuence, <number of segments>
```

6. Initiate the configuration:

```
INIT:IMM
```

Using the Digital Multimeter

This section shows you how to make the following types of measurements with the Digital Multimeter (DMM):

- DC Voltage
- AC Voltage
- Resistance
- Current

What You Should Know About

MEASure and CONFigure Commands

The following command subsystems enable you to take measurements:

- MEASure
- CONFigure

The MEASure subsystem requires less specific knowledge of the DMM. The MEASure commands configure the instrument for the measurement, initiates the measurement, and returns the data.

The CONFigure command configures the instrument for a specific measurement and allows more specific instrument parameters to be set before a measurement is made. For example, you can control aperture settings and initialization. You use CONFigure along with two other command subsystems:

  - INIT
  - FETCh?

The INIT command initiates the measurement. The FETCh? commands return the measured data. MEASure can be used in place of the CONF, INIT, and FETCh? commands in the measurement procedures below.

The READ? query initiates a measurement and returns the measured data in one step. READ? can be used in place of INIT and FETCh? in the procedures below.

Selecting the Input Impedance

The VX4101A has a default 10 MΩ input impedance (± 5%) following power up for all voltage ranges. This input impedance is the result of the 10 MΩ resistance ladder in the VX4101A provided for attenuation of voltages greater than 3 VDC. This is usually not an issue for most measurements. However, if the signal you are measuring has a source or output impedance of greater than 100 Ω, you might receive errors resulting from this impedance. A voltage divider circuit is created by the measurement source impedance and the VX4101A input impedance. For example, a 1000 Ω source impedance loaded by the VX4101A 10 MΩ input impedance will result in a .01% error.

For voltage measurements less than 3 VDC, this error may be essentially eliminated by sending the command INPut:IMPedance 10e9. Since an attenuator is not required for the lower voltage ranges, this command is provided to switch out that attenuator, if desired.

About Low Level DC Measurements

The VX4101A in the 30 mV DC measurement range has a resolution of 100 nV. The accuracy at 0 VDC is between 8 μV and 15 μV depending on aperture selection. At this low level of operation, the primary contributions to the error in ambient temperature applications are the DC low range input amplifier voltage and current noise and drift with time and temperature, and thermal EMF drops in the range changing relays. These errors are minimized with the use of low drift amplifiers and low thermal EMF relays in the VX4101A.

Cabling and aperture selection are vital in order to attain these measurement accuracies. The primary source of additional error when you make an actual measurement is normal mode power line noise. Normal mode power line noise is differential noise between the high and low input that can easily be coupled into your cabling. Since it is differential noise between the high and low input it looks to the DMM like an actual input voltage.

To minimize these errors at low-level measurements, you should use shielded twisted pair cabling to connect to the DMM. Since the VX4101A has a floating front end, the low input of the VX4101A is not grounded. If your measurement is referenced to ground, the best place to provide the ground reference is at your equipment. If the measurement is referenced to ground, it is probably already grounded within your unit somewhere. The DMM low side should not be grounded. This will only create a ground loop adding potential error. Finally, using an aperture that is a power line period multiple will provide significant power line rejection as discussed in *About Power Line Noise*, below.

Low level errors are also minimized by using the autozero capability of the VX4101A. Autozero minimizes long term drift errors in the input amplifier by comparing a null reading taken during autozero with a reference null reading taken when the VX4101A was last calibrated and compensating for any difference. The autozero function also eliminates a portion of temperature drift errors in applications where the temperature drifts slowly.

About Power Line Noise and Common Mode Rejection

Three sets of common mode specifications are provided for the VX4101A:

- DC Common Mode Rejection
- AC Common Mode Rejection
- AC Effective Common Mode Rejection

Power line noise is generally at a frequency of 60 Hz or 50 Hz depending on the power system prevalent in the area of the world where you are located. 400 Hz is often prevalent in avionics applications. The VX4101A minimizes errors due to power line noise at these frequencies. This rejection is provided by a combination of the VX4101A floating front end and the resulting common mode rejection, and by the aperture programming provided.

Power line errors (typically injected into the measurement cabling) are of two types, differential mode errors and common mode errors. Differential mode errors, as described above, may be reduced by proper cabling and by selection of aperture. Common mode errors are those equally injected onto both lines of the measurement cabling. They are reduced by proper cabling, by selection of aperture, and by the VX4101A AC common mode rejection.

Proper aperture selection for power line noisy environments may be easily selected by using the SENS:NPLC command. This permits selection of apertures that are an integer number of power line cycles. The VX4101A is initially shipped assuming operation in a 60 Hz environment. For 50 Hz and 400 Hz (a multiple of 50 Hz), the CALibrate:LFRequency command is provided to select multiples of 50 Hz periods when the SENS:NPLC command is used. The CAL:LFR command is included in the CALibrate command subsystem so the latest selection may be stored in non-volatile memory on the VX4101A. As such, you need only send the command once on receipt of your equipment, although you may find it easier just to include it in the initialization sequence of your test program.

Normal mode rejection specifications are specified at the particular frequency of interest within some tolerance. You can reject frequencies outside that tolerance, if they are high enough, by choosing an aperture that is a multiple of that frequency, or for broadband noise by choosing as large an aperture as is consistent with test time requirements. The amount of rejection for any frequency and aperture may be determined by integrating the noise signal over the aperture time and determining the resulting DC error.

Making a DC Volt Measurement

This procedure will show you how to make a DC measurement with the DMM.

Hardware Tips

DC voltage measurements are made by connecting to pins 6 V+ and 1 V– of the D-sub connector.

Programming Example

Do the following to perform a DC voltage measurement on the DMM:

1. Select the DMM:

   ```
   INST:SEL DMM
   ```

2. Configure the DMM to take a DC measurement in the 30 V range with minimum resolution:

   ```
   CONF:VOLT:DC 30, MIN
   ```

3. Initiate the measurement:

   ```
   INIT
   ```

4. Retrieve the measurement results:

   ```
   FETCh?
   ```

Making AC Volt Measurements

Hardware Tips

AC voltage measurements are made by connecting to pins 6 (V+) and 1 (V–) of the D-sub connector. The following measurement example is AC coupled. Any DC component is blocked.

The default range change time for AC measurements is very slow, approximately 2 seconds, to permit accurate measurements of low frequencies (below 100 Hz). If you know the frequency you are measuring is above 100 Hz, you can use the SENSe:BANDwidth command to speed up range changes and autoranging. 60 Hz may also be measured in the Fast Bandwidth Mode if an additional .3% error is added to the accuracy spec.

Programming Example

1. Select the DMM:

   ```
   INST:SEL DMM
   ```

2. Configure the DMM to take a AC Volt measurement in the 30 volt range with minimum resolution:

   ```
   CONF:VOLT:AC 30, MIN
   ```

***NOTE**. To make a DC coupled AC voltage measurement, use ACDC in place of AC in the CONF:VOLT:AC command*

3. Initiate and acquire the AC V measurement:

   `INIT`

4. Return the AC V measurement:

   `FETCh?`

Making a 2-Wire Resistance Measurement

Hardware Tips

It is possible to make both 4-wire and 2-wire measurements. The hardware considerations for both types of measurements are as follows:

2-Wire Measurements. 2-wire resistance measurements are made by connecting to pins 6 (R+) and 1 (R–) of the D-sub connector. For making 4-wire resistance measurements pins 9 (R+) and 5 (R–) are also used.

4-Wire Measurements. 4-wire measurements are useful when making low-resistance or null resistance measurements. The additional two wires eliminate cabling resistance in the measurement. For effective use of the four-wire measurement, connect the pins as follows:

- Connect pins 6 and 9 as close as possible to the positive side of the resistance to be measured
- Connect pins 1 and 5 as close as possible to negative side of the resistance

Programming Example

1. Select the DMM:

   `INST:SEL DMM`

2. Configure the DMM to take a resistance measurement in the 30 Ohm range with minimum resolution:

   `CONF:RES 30, MIN`

3. Initiate and acquire the resistance measurement and retrieve results:

   `READ?`

Making a Current Measurement

Hardware Tips

Current measurements are made by connecting to pins 7 (I+) and 3 (I–) of the D-sub connector.

Making a current measurement results in a voltage drop, known as burden voltage, across the VX4101A current terminals. The voltage drop may affect your circuit for which you are measuring current. Check the burden specification in Appendix A to determine if this is a concern.

Programming Example

1. Select the DMM:

```
INST:SEL DMM
```

2. Configure the DMM to take a Current measurement in the 1.0 Amp range with minimum resolution:

```
CONF:CURR 1.0, MIN
```

3. Initiate and acquire the measurement:

```
INIT
```

4. Retrieve the measurement results:

```
FETCh?
```

Using the Digital to Analog Converter (DAC)

This section contains the following procedures for using the DAC:

- Generating continuous and multiple waveforms
- Generating repetitive waveforms
- Reading a trace from the DAC in binary
- Using the sample handshake mechanism
- Programming a trace with a numeric array list

What You Should Know About

Key Concepts

Sample. A single DAC output point. At the end of a sample, the DAC output amplitude remains at the last sample value until a new sample is initiated. For example, you can use the command TRIG:MODE SAMPLE to externally trigger sample generation.

Segment. A group of DAC points that generate a waveform. The maximum number of points in a waveform is 1024. You use the TRAC:POIN <number of points> command to enter the segment length. At the end of a segment, the DAC output amplitude remains at the last segment value until you initiate a new segment. You can initiate a new segment continuously, by the repetition frequency, or by a trigger. You can use the command TRIG:MODE SEGMENT to externally trigger segment generation, or the TRACe:FREQuency to internally program a segment repetition rate.

Sequence. A programmed repetition of segments. The maximum number of segments is 4096. At the end of a sequence, the DAC output amplitude remains at the last sequence value until a new sequence is triggered. You use the TRIG:MODE SEQ,<sequence length> command to define the number of segments. You can initiate a new sequence either by a trigger, or continuously.

Waveform Period. The waveform period is defined as the number of points in the waveform times the sample period.

If the waveform repetition mechanism is enabled, the repetition period must be greater than or equal to the waveform period. Calculate the waveform period as follows:

*Sample Period * Number of Points*

DAC Trigger Modes

The DAC has the following trigger modes.

- Sample mode allows an external or handshake trigger to clock the DAC samples when an external or handshake trigger source is selected.
- Segment triggering allows an external, handshake, or immediate trigger source to initiate an externally timed segment.
- Sequence triggering allows an external, handshake or immediate trigger source to initiate an externally timed sequence.

DAC Channel Common Functions

All channels share the following functions:

- Common sample clock
- Common trigger source
- Common trigger output
- Common repetition mechanism
- Common trigger mode

Using the Internal Clock

When the trigger source is programmed to IMMediate, the internal sample clock is selected. This is the default condition.

Generating Continuous and Multiple Waveforms

This procedure shows you how to generate four waveforms simultaneously with the DAC. You will generate the following waveform types:

- 62.5 Hz triangle waveform on channel two
- 62.5 ramp waveform on channel one
- 500 Hz and 250 Hz square waveforms on channel three and channel eight

You will use Numeric Array Lists.

Programming Example

Perform the following steps:

1. Select the DAC with the following command:

```
INST:SEL DAC
```

2. Enter the following command to set the number of data points used in the waveforms:

```
TRACe:POINts 32
```

3. Enter the next command to set the sample rate for the waveforms:

```
TRACe:SRATe 2000
```

4. Enter the following command to create a Ramp waveform on Channel one:

```
TRACe1,1,0,.625,1.25,1.875,2.5,3.125,3.75,4.375,5,5.625,6.25,
6.875,7.5,8.125,8.75,9.375,10,0,.625,1.25,1.875,2.5,3.125,3.7
5,4.375,5,5.625,6.25,6.875,7.5,8.125
```

5. Enter the next command to create a Triangle waveform on Channel two:

```
TRACe 2,1,0,.625,1.25,1.875,2.5,3.125,3.75,4.375,5,4.375,
3.75,3.125,2.5,1.875,1.25,.625,0,-.625,-1.25,-1.875,-2.5,-3.1
25,-3.75,-4.37,-5,-4.375,-3.75,-3.125,-2.5,-1.875,-1.25,-.625
```

6. Enter the following two commands to create a 500Hz Square wave on Channel three:

```
TRACe 3,(1:2,5:6,9:10,13:14,17:18,21:22,25:26,29:30),5
```

```
TRACe 3,(3:4,7:8,11:12,15:16,19:20,23:24,27:28,31:32),-5
```

7. Enter the following command to create a 250Hz Square wave on Channel eight:

```
TRACe 8,1,0,0,0,0,5,5,5,5,0,0,0,0,5,5,5,5,0,0,0,0,5,5,5,5,0,
0,0,0,5,5,5,5
```

8. Enter the following command to initialize a continuous DAC output of signals:

```
INIT:CONT ON
```

9. Enter the following command to abort the signals currently being generated:

```
INIT:CONT OFF
```

Generating Repetitive Waveforms

This procedure shows you how to generate the same waveforms as in the previous example, with continuous output, at a repetition frequency of 50 Hz.

As before, you will use Numeric Array Lists.

Programming Example

Perform the following steps:

1. Select the DAC with the following command:

```
INST:SEL DAC
```

2. Enter the following command to set the number of data points used in the waveforms:

```
TRACe:POINts 32
```

3. Enter the next command to set the sample rate for the waveforms:

```
TRACe:SRATe 2000
```

4. Enter the following command to set the repeat frequency of the waveforms:

```
TRACe:RFR 50
```

This command will repeat the waveforms at a repeat frequency of 50 Hz.

5. Enter the following command to create a Ramp waveform on Channel one:

```
TRACe 1,0,.625,1.25,1.875,2.5,3.125,3.75,4.375,5,5.625,6.25,
6.875,7.5,8.125,8.75,9.375,10,0,.625,1.25,1.875,2.5,3.125,3.7
5,4.375,5,5.625,6.25,6.875,7.5,8.125,8.75,9.375,10
```

6. Enter the next command to create a Triangle waveform on Channel two:

```
TRACe 2,1,0,.625,1.25,1.875,2.5,3.125,3.75,4.375,5,4.375,
3.75,3.125,2.5,1.875,1.25,.625,0,–.625,–1.25,–1.875,–2.5,–3.1
25,–3.75,–4.37,–5,–4.375,–3.75,–3.125,–2.5,–1.875,–1.25,–.625
```

7. Enter the following two commands to create a 500Hz Square wave on Channel three:

```
TRACe 3,(1:2,5:6,9:10,13:14,17:18,21:22,25:26,29:30),5
```

```
TRACe 3,(3:4,7:8,11:12,15:16,19:20,23:24,27:28,31:32),–5
```

8. Enter the following command to create a 250Hz Square wave on Channel eight:

```
TRACe 8,1,0,0,0,0,5,5,5,5,0,0,0,0,5,5,5,5,0,0,0,0,5,5,5,5,0,
0,0,0,5,5,5,5
```

9. Enter the following command to initialize a continuous DAC output of signals:

   ```
   INIT:CONT ON
   ```

10. Enter the following command to abort the signals currently being generated:

    ```
    INIT:CONT OFF
    ```

Reading a Trace from the DAC in Binary

1. Select the DAC as the active instrument:

   ```
   INST:SEL DAC
   ```

2. Set the data format to binary:

   ```
   FORMAT BIN
   ```

***NOTE**. ASCII is the default format. You can use the command FORMAT ASCII to change the data format back to ASCII. The queries in this procedure can be used in either ASCII or Binary.*

3. Read the current trace from channel 1, starting at index 1 through the currently programmed number of points:

   ```
   TRACe? 1
   ```

4. Read the value of the third data point from channel 8:

   ```
   TRACe? 8,3
   ```

5. Read from channel 3 data points 100 through 200:

   ```
   TRACe? 3,(100:200)
   ```

6. Set the data format to ASCII:

   ```
   FORMAT ASCII
   ```

Using the Sample Handshake Mechanism

The handshake mechanism is similar to external trigger except that it adds a second signal to notify the unit under test that the instrument is ready to accept the next trigger. The handshake trigger input is called DAC. A second signal DAC**,** provides a signal back to the user indicating the DAC is ready for the next trigger. The default polarity of both DAC and DAC are active low.

The following example demonstrates the handshake mechanism in sequence mode. Segment and sample modes are similar. In this example, a burst of 10 triangle waves will be triggered by a square wave signal generator. The triangle waves have a period of 4 milliseconds. The burst will have a duration of 40 milliseconds. The trigger source will be a TTL square wave with a period of 10 milliseconds (the trigger source simulates a trigger that may have a period shorter than the programmed sequence).

Hardware Tips

To ensure the trigger is not re-initiated during the burst, the square wave generator must be qualified by the DAC’s Analog Output Request signal. To accomplish this, do the following:

1. Connect the DAC as described above to a 100 Hz TTL signal generator.

2. Reset to the default settings:

```
*RST
```

Programming Example

Perform the following steps:

1. Place an R/S flip flop between the square wave trigger source and the DAC.

- **a.** Connect the flip-flop S input to the square wave signal source.
- **b.** The Q* output is connected to the DAC’s Analog Output Request.
- **c.** The flip-flop Clear* input is connected to the DAC’s Analog Output Strobe.
- **d.** The R input is connected to ground

2. Select the DAC as the active instrument:

```
INST:SEL DAC
```

3. Program the internal sample rate to 10 kHz:

```
TRAC:SRAT 10000
```

4. Program the number of sample points to 40:

```
TRAC:POIN 40
```

5. Program channel 1, starting at index 1 to a 10 Vpp triangle waveform:

```
TRAC
1,1,1,2,3,4,5,6,7,8,9,10,9,8,7,6,5,4,3,2,1,0,-1,-2,-3,-4,-5,-
6,-7,-8,-9,-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0
```

6. Program a sequence of 10 segments:

```
TRIG:MODE SEQ,10
```

7. Set the handshake polarity:

```
TRIG:HAND:POL:REQ INV
```

```
TRIG:HAND:POL:STR INV
```

8. Start the DAC:

```
INIT:CONT ON
```

9. Stop the DAC:

```
INIT:CONT OFF
```

Programming a Trace with a Numeric Array List

This programs a 5 Vpp square wave with zero offset and 256 points per period.

1. Select the DAC as the active instrument:

```
INST:SEL DAC
```

2. Use the numeric list format to set the array for channel one:

   a. Set the following channel one points to +5 V:

   ```
   TRAC 1,(1:128,257:384,513:640,769:896),5
   ```

   This sets points 1-128, 257–384, 513:640, and 769:896 to +5 V

   b. Set the following channel one points to –5V:

   ```
   TRAC 1,(129:256,385:512,641:768,897:1024),-5
   ```

   This sets points 129-256, 385-512, 641:768, and 897:1024 to –5 V

3. Set the waveform period to 10 Hz:

```
TRAC:SRAT 2560
```

4. Set the number of points to 1024:

```
TRAC:POIN 1024
```

5. Start the waveform:

```
INIT:CONT ON
```

6. Stop the waveform:

```
INIT:CONT OFF
```

Using the Relay Drivers

You can perform the following operation with the Relay Drivers:

- Opening and closing relays

Opening and Closing Relays

This procedure shows you how to open and close the relay drivers. The procedure is as follows:

Programming Example

1. Select the Relay Drivers as the active instrument:

   ```
   INSTrument:SELect RDRIV
   ```

2. Enter the following command to open relays on channel one, three, five, and seven:

   ```
   OPEN (@1,3,5,7)
   ```

3. Enter the following command to verify which relays are closed:

   ```
   CLOSe:STATe?
   ```

   The instrument returns the following string: 10101010

4. Enter the following command to close the relays:

   ```
   CLOSe (@1,3,5,7)
   ```

   A subsequent [ROUTe]:CLOSe:STATe? query will show that all relays are closed.

Using the SurePath™ Modules

You can perform the following operations with the SurePath™ modules:

- Closing a relay
- Opening a relay

What You Should Know About

SurePath™ Module

The procedure demonstrates closing a relay on a VX4330 SurePath™ module. You can use a similar procedure for the other SurePath™ modules compatible with the VX4101A. For a list of the modules that will work with the VX4101A, see *Product Description*.

Closing a Relay

In this procedure, you will assign a specific dwell to the scanner and set channels to multiplex (mux). The "mux" mode allows you to close multiple relays (or channels) at the same time. Perform the following procedure to close a relay with the SurePath™ master:

1. Select SurePath:

```
INSTrument:SELect SUREPATH
```

2. Enter the following command to assign the close dwell:

```
close:Dwell m1,.5
```

3. Enter the following command to set the close mode of specific sections of the scanner matrix to "mux."

```
close:mode mux,m1,(1:3)
```

4. Enter the following command to close a specific section of the matrix:

```
close(@m1(1!1:10!1,1!2:5!2,5!3:10!3))
```

This closes the following:

Channels 1-10 in section one
Channels 1-5 in section two
Channels 5-10 in section three

5. When the relays close, pulse the front panel encode signals corresponding to sections one, two, and three.

Syntax and Commands

Command Syntax

About Protocol and Syntax

Command protocol and syntax for the VX4101A Module are as follows:

- The following is a command for the DMM:

  ```
  CONFigure[:SCALar][:VOLTage]:DC [<expected value>
  [,<resolution>]]
  ```

  Each of the following commands is a valid form of this command. This is only a partial list of commands and is intended only for illustrative purposes:

  ```
  CONFIGURE:SCALAR:VOLTAGE:DC 5.0,.001
  conf:DC
  CONF:scal:DC 5.0
  Conf:Dc MAX,min
  confIGURE:scalAR:voltAGE:DC 21
  conf:DC 10,maximum
  ```

***NOTE**. Examples throughout this manual use various forms of the command syntax to further illustrate these concepts.*

- A command string consists of a string of ASCII-encoded characters terminated by a <program message terminator>. The <program message terminator> is optional white space, followed by any one of the following command terminations:

  A line feed <LF> or new line <NL> character (hexadecimal 0A, decimal 10)

  The END bit set

  The END bit with a line feed <LF> or new line <NL>

  The command string is buffered until the terminator is encountered, at which time the entire string is processed.

- In addition to terminating a command, the semi-colon character directs the SCPI command parser to interpret the next command with the assumption that all characters up to and including the last colon in the previous command have just been parsed.

- White space characters can be used to make a command string more readable. These characters are ASCII-encoded bytes in the range hexadecimal 00-09 and 0B-20 (decimal 0-9 and 11-32). This range includes the ASCII control characters and the space, but excludes the line feed <NL>. White space characters are ignored when legally encountered in a command string. White space is allowed anywhere in a command string, except for the following:

  Within a program mnemonic (for example ROUTE)

  Around a colon (:) mnemonic separator (for example ROUTE: CLOSE or ROUTE :CLOSE)

  Between a mnemonic and a (?) (for example CLOSE ?)

  Following an asterisk (*) (for example * STB?)

  Within a number (for example 12 34)

  At least one white space character is required between a command/query header and its associated arguments. For example in the command

  ```
  route:configure:join m1,(1:6)
  ```

  the command header is the string "route:configure:join". The arguments associated with this command are the module name "m1" and the section list "(1:6)". At least one white space character must be sent before the first argument.

  In the query

  ```
  route:close? (@m1(1:64))
  ```

  the query header is the string "route:close?". The argument associated with this query is the channel list "(@m1(1:64))". At least one white space character must be sent before the channel list argument.

- A SCPI command or query is composed of one or more keywords separated by colons. A keyword can be sent in either short or long form. The short form of a keyword is composed of capital letters in the command descriptions. The long form is composed of all characters in the keyword. The keywords in a command or query can be a combination of long and short forms. Commands and queries are parsed in a case-independent manner

- Multiple data parameters passed by a command are separated by a comma (,)

- A question mark (?) following a command indicates that a response will be returned. All responses from the VX4101A are terminated with a line feed <LF> (hexadecimal 0A) character

- In the command descriptions, the following special characters are used. Except for the colon (:), these characters are not part of the command and should not be sent. If an optional field is omitted, the default for the command is applied

[ ]	Brackets indicate an optional field
|	A bar indicates a logical OR choice
:	A colon is used to separate command fields
< >	Field indicator

SCPI/IEEE 488.2 Command Elements

The definition of elements used in SCPI/IEEE 488.2 commands and command descriptions is as follows:

`<NR1>`

ASCII integer representation of a decimal number.

`<NRf>`

ASCII integer, fixed point or floating point representation of a decimal number.

`Error/Event queue`

When the command parser detects a syntax error or data range error, it places an error message describing the error in the Error/Event queue. Bit 2 of the Status Byte Register is set to indicate that this queue is not empty. Bit 5 of the Standard Event Status Register (the Command Error bit) is set if the parser detects a syntax error. Bit 4 of the Standard Event Status register (the Execution Error bit) is set if the parser detects a numeric argument that is out of range. When a SYSTem:ERRor? query is received, an error message is moved from the Error/Event queue and placed in the Output queue.

SCPI Commands for the VX4101A

This section summarizes the SCPI commands that control overall functions of the VX4101A, including:

- Querying different firmware versions
- Querying measurement options
- Selecting which instrument to use
- Selecting the IEEE 488.2 Synchronous Mode or the Asynchronous Mode

Command Summary

The following is a listing of the available command subsystems and syntax:

ABORt Subsystem

Commands

```
ABORt:COMPlete
ABORt[:IMMediate]
```

CALibrate Subsystem

Commands

```
CALibrate:ROSCillator

   :CLEar
   :MANual <frequency>
```

INSTrument Subsystem

Commands

```
INSTrument:NSELect <instrument id>
INSTrument[:SELect] <instrument name>
```

Queries

```
INSTrument:CATalog?

   :FULL?
   :LONG?

INSTrument:COUNt?
INSTrument:NSELect?
INSTrument[:SELect]?
```

OUTput Subsystem

Commands

```
OUTPut:EXTernal:SOURce <ext source>
OUTPut:TTLTrg[<N>]:SOURce <trig source>
```

Queries

```
OUTPut:EXTernal:SOURce:CATalog?
OUTPut:EXTernal:SOURce?
OUTPut:TTLTrg[<N>]:SOURce?
OUTPut:TTLTrg[<N>]:SOURce:CATalog?
```

SOURce Subsystem

Commands

```
SOURce:ROSCillator[:SOURce] <source>
SOURce:ROSCillator:VALue <frequency>
```

Queries

```
SOURce:ROSCillator[:SOURce]?
SOURce:ROSCillator:VALue?
```

SYStem Subsystem

Commands

```
SYSTem:LANGuage <language>
SYSTem:RQUeue

    :QMODe <qmode>
    :SNUMber[:SET] <snumber>
    :SNUMber:STATe <state>

SYSTem:TIMeout <timeout in seconds>
```

Queries

```
SYSTem:ERRor?
SYSTem:LANGuage?
SYSTem:OPTions?
SYSTem:OPTions:DESCription? <option code>
SYSTem:READy?
SYSTem:RQUeue?

    :QMODe
    :SNUMber[:SET]?
    :SNUMber:STATe?

SYSTem:SNUMber?
SYSTem:TIMeout?
SYSTem:VERSion?
```

TRIGger Subsystem

Commands

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:FIRe<command trigger>
TRIGger ([:SEQuence1]|:STARt)[:LAYer]:TIMer <timer seconds>
```

Queries

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:TIMer?
```

ABORt Subsystem

Command Syntax

```
ABORt:COMPlete
ABORt[:IMMediate]
```

Query Syntax

N/A

***RST Value**

N/A

Limits

N/A

Related Commands

```
INSTrument:ABORt
INITiate
INITiate:CONTinuous
```

Command Description

ABORt[:COMPlete]

Place all instruments in the IDLE state, aborting after the current SEGMent or SEQuence is complete. The instrument configuration is unchanged and a subsequent INIT command will cause the instrument to re-start the same type of measurement. Regardless of whether the instrument is in Synchronous or Asynchronous Mode, you can send this command while a query is in progress and abort any measurement.

For more information on how the VX4101A handles queries in both Synchronous and Asynchronous mode, see *About VX4101A Operational Modes*.

ABORt[:IMMediate]

Places all instruments in the IDLE state, immediately aborting any measurement or other instrument activity currently in progress. The instrument configuration is unchanged and a subsequent INIT command will cause the instrument to re-start the same type of measurement.

***NOTE**. While the instrument is initiated, any command or query which would cause the instrument to change its configuration or start a new measurement will first cause the instrument to abort immediately. After this implicit abort has been issued, the ABORt bit of the Operational Status Register will toggle to signal that the abort has completed and the new command or query will be executed.*

***NOTE**. If a query (as opposed to a command) is currently in progress, the behavior of the instrument will depend upon the communications protocol in effect. If the instrument is in asynchronous mode, a new command or query can be sent while a query is in progress and the implied abort will occur. If, however, the instrument is in synchronous mode, the command or query will be queued while the current query completes. This is a ramification of the IEEE 488.2 Message Exchange Protocol Enforcer (MEPE).*

Query Response N/A

Examples

ABORt[:COMPlete]

Command	Response
INST:SEL DAC	
trac:srat 10000	
trac:poin 10	
trac:data 1,1,1,2,3,4,5,6,7,8,9,0	
trig:mode seq,100	
init:cont	
trac:ind?	21,3
abor:comp	
trac:ind?	1,1

ABORt[:IMMediate]

Command	Response
INST:SEL DMM	
CONF:ARR:VOLT:DC 512	
INIT	
FETC:COUN?	127

Command	Response
ABOR	
FETC:COUN?	153
FETC:COUN?	153

***NOTE**. After the ABORt command, the instrument will take no more measurements.*

CALibrate Subsystem

Command Syntax

```
CALibrate:ROSCillator

   :CLEar
   :MANual <frequency>
```

Query Syntax

N/A

***RST Value**

Parameter	Values
<frequency>	Calibration Value Stored in non-volatile memory

Limits

Parameter	Value
<frequency>	9 MHz ≤ Frequency ≤ 11 MHz

Related Commands

SOUR:ROSC[?]
SOUR:ROSC:VAL[?]

Command Description

CALibrate:ROSCillator:CLEar
This command negates the calibration for the currently selected reference oscillator. The reference oscillator is flagged as uncalibrated in nonvolatile memory and any time the current source is reselected, an error message will be generated noting this fact. An uncalibrated reference oscillator is assumed to be exactly 10 MHz. This command would typically be used when a reference oscillator source has been changed (such as changing a Slot 0 controller) and the instrument has not yet been calibrated with the new source.

CALibrate:ROSCillator:MANual <frequency>
This command allows you to manually specify the calibrated frequency to be used for the currently selected reference oscillator. The specified frequency value is stored in non-volatile memory and overwrites the calibrated value stored previously.

***NOTE**. To temporarily override the calibrated value without storing in non-volatile memory, use the the SOUR:ROSC:VAL command.*

Examples

CALibrate:ROSCillator:CLEar

Command	Response
SOUR:ROSC INT:SYST:ERR?	–313,“Calibration memory lost; VX4101A; Reference Oscillator Uncalibrated”
SOUR:ROSC:VAL?	10000000.0000
INST:SEL COUNTER	
CAL:ROSC 10.001E6	
SOUR:ROSC:VAL?	9999999.0000
CAL:ROSC:CLE	
SOUR:ROSC:VAL?	10000000.000
SOUR:ROSC:VAL?	10000000.0000

CALibrate:ROSCillator:MANual

Command	Response
SOUR:ROSC INT	
SYST:ERR	–300 “Device specific error; VX4101A; Reference Oscillator Not Calibrated”
CAL:ROSC:MAN	9.999999E6
SOUR:ROSC INT	
SOUR:ROSC:VAL?	9999999.0000
SYST:ERR	0, “No error”

INSTrument Subsystem

Command Syntax

```
INSTrument:NSELect <instrument id>

INSTrument[:SELect] <instrument name>
```

Query Syntax

```
INSTrument:CATalog?

   :FULL?
   :LONG?

INSTrument:COUNt?

INSTrument:NSELect?

INSTrument[:SELect]?
```

***RST Value**

Parameter	Values
<instrument id>	0
<instrument name>	VX4101A

Limits

Parameter	Values
<instrument id>	0–99
<instrument name>	VX4101A, SUREPATH, DMM, COUNTER, DAC, DIGI, DIGO, RDRV
<number of instruments>	0-99

Related Commands

N/A

Command Description

INSTrument:NSELect <instrument id>
Selects an instrument by its assigned instrument number. The VX4101A is always instrument 0, selecting this number deselects all other instruments.

INSTrument[:SELect] <instrument name>
This command selects an instrument by its assigned instrument name.

Query Response

INSTrument:CATalog?
This query lists the names of the individual instruments which comprise the VX4101A. These names can be used to select the instrument using the INST:SEL command.

INSTrument:CATalog:FULL?
This query lists the names and numbers of the individual instruments which comprise the VX4101A.

***NOTE**. To select an instrument by either name or number, see the INSTrument:NSELect or the INSTrument:SELect command in this section.*

INSTrument:CATalog:LONG?
This query is primarily for the use of the VXI*plug&play* soft front panel. It lists information on an individual instrument including the instrument name, number, soft front panel executable, and ASCII description. Specify an instrument number in the argument to return information about that instrument. Specify no argument to return information on all instruments. The information is returned separated by commas.

INSTrument:COUNt?
This query returns the number of instruments which are listed when a INST:CAT? query is performed.

INSTrument:NSELect?
Queries the current instrument based upon the instrument number. The VX4101A is always instrument 0 and selecting this number effectively deselects all instruments.

INSTrument[:SELect]?
This query returns the current instrument based upon the instrument name. Selecting the VX4101A effectively deselects all other instruments.

Examples

INSTrument:CATalog?

Command	Values
INST:CAT?	VX4101A, SurePath, DMM, Counter

INSTrument:CATalog:FULL?

Parameter	Values
INST:CAT:FULL?	VX4101A, 0, SurePath, 1, DMM, 2, Counter, 3

INSTrument:CATalog:LONG?

Parameter	Values
INST:CAT:LONG 2	DMM,2,tksfdmm.exe,"DMM"

INSTrument:COUNt?

Parameter	Values
INST:CAT?	VX4101A, SurePath, DMM, Counter
INST:COUN?	4

INSTRument:NSELect

Command	Values
*RST	
INST:NSEL?	0
INST:NSEL 2	
INST:NSEL?	2

INSTrument[:SELect]

Command	Values
*RST	
INST:SEL?	VX4101A
INST:SEL COUNTER	
INST:SEL?	COUNTER

OUTPut Subsystem

Command Syntax

`OUTPut:EXTernal:SOURce <ext source>`

`OUTPut:TTLTrg[<N>]:SOURce <trig source>`

Query Syntax

`OUTPut:EXTernal:SOURce:CATalog?`

`OUTPut:EXTernal:SOURce?`

`OUTPut:TTLTrg[<N>]:SOURce?`

`OUTPut:TTLTrg[<N>]:SOURce:CATalog?`

Command Class

Global

***RST Value**

Parameter	Values
<ext source>	Hold
<trig source>	Hold
<N>	1

Limits

Parameter	Values
<ext source>	HOLD,IMMEDIATE,BUS,EXTERNAL,TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COM-MAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
<trig source>	HOLD, COMMAND0, COMMAND1, COMMAND2, COM-MAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM
<N>	0-7 are valid TTL Trigger Suffixes. Default is 1

Command Description

OUTPut:EXTernal:SOURce <ext source>
Selects the trigger source for the MultiPaq™ external output trigger on pin 24E.

OUTPut:TTLTrg[<N>]:SOURce <trig source>
Selects the trigger source for the specified VXIbus TTL trigger.

Query Response

OUTPut:EXTernal:SOURce:CATalog?
Lists available trigger sources for use with the OUTP:EXT:SOUR command.

OUTPut:EXTernal:SOURce?
Lists the current external trigger source.

OUTPut:TTLTrg[<N>]:SOURce?
Queries the trigger source for the specified VXIbus TTL trigger.

OUTPut:TTLTrg[<N>]:SOURce:CATalog?
Lists available trigger sources for use with the OUTP:TTLT:SOUR command. The possible values are:

HOLD, BUS, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM

Examples

OUTPut:EXTernal:SOURce:CATalog?

Command	Response
OUTP:EXT:SOUR:CAT?	HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO

OUTPut:TTLTrg[<N>]:SOURce:CATalog?

Command	Response
OUTP:TTLT:SOUR:CAT?	HOLD, BUS, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM

OUTPut:TTLTrg[<N>]:SOURce

Parameter	Response
OUTP:TTLT:SOUR:CAT?	HOLD, BUS, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM
OUTP:TTLT5:SOUR DMM	
OUTP:TTLT5:SOUR?	DMM

SOURce Subsystem

Command Syntax

```
SOURce:ROSCillator[:SOURce] <source>
SOURce:ROSCillator[:SOURce]:VALue <frequency>
```

Query Syntax

```
SOURce:ROSCillator[:SOURce]?
SOURce:ROSCillator[:SOURce]:VALue?
```

***RST Value**

Parameter	Values
<source>	INTernal
<frequency>	Calibrated internal source value

Limits

Parameter	Values
<source>	INTernal|CLOCk10|USER1|USER2|USER3|USER4| USER5|USER6|USER7|USER8|USER9|USER10
<frequency>	DEFault or 9 MHz ≤ value ≤ 11 MHz

Related Commands N/A

Command Description

SOURce:ROSCillator[:SOURce] <source>
This command selects the source of the instrument's 10 MHz reference oscillator. Valid sources are the VX4101A on-board 10 MHz crystal and the VXIbus backplane 10 MHz signal. Because you might have more than one possible backplane source, the VX4101A allows calibration factors for up to ten back reference oscillator sources (USER1 to USER10) to be stored in non-volatile memory. CLOCk10 is an alias for USER1. When the source has never been calibrated, then a warning is issued that the source is not calibrated. Sources that are not calibrated are assumed to be ideal (10 MHz).

***NOTE**. The Counter uses the 10 MHz signal to drive most of its own control circuitry. If the backplane 10 MHz signal is selected and no signal is present, any attempt to use the Counter will result in a communications failure with the* VX4101A. *To recover from this state, cycle power on the* VX4101A.

***NOTE**. The reference oscillator should only be changed when the instrument is not active. Switching the reference oscillator source during measurements will result in indeterminate instrument operation.*

SOURce:ROSCillator:VALue <frequency>
This command provides a temporary override of the calibrated oscillator frequency with a frequency supplied that you enter. The device uses the frequency to provide a calibrated value for a USER oscillator without running a calibration cycle or to simulate an ideal oscillator. This value is lost whenever the oscillator source is switched. Choosing DEFault as the value re-initializes the oscillator frequency to the calibrated value, if available. If the calibrated value is not available, 10 MHz is used.

Query Response

SOURce:ROSCillator [:SOURce?]
INTernal|CLOCk10|USER1|USER2|USER2|USER3|USER4|
USER5|USER6|USER7|USER8|USER9|USER10

SOURce:ROSCillator:VALue?
Returns the calibrated value of the internal frequency source.

Examples

SOURce:ROSCillator:VALue?

Command	Response
*RST	
SOUR:ROSC?	INTernal
SOUR:ROSC:VAL?	9.99999999999999E+06
SOUR:ROSC CLOC	
SOUR:ROSC?	CLOCk10
SOUR:ROSC:VAL?	1.00000010000000E+07
SOUR:ROSC USER2	
SOUR:ROSC?	USER2
SYST:ERR?	–300 “Device specific error; VX4101A; Reference Oscillator Not Calibrated”
SOUR:ROSC:VAL?	1.00000000000000E+07

SYSTem Subsystem

Command Syntax

```
SYSTem:LANGuage <language>

SYSTem:RQUeue

    :QMODe <qmode>
    :SNUMber[:SET] <snumber>
    :SNUMber:STATe <state>

SYSTem:TIMeout <timeout in seconds>
```

Query Syntax

```
SYSTem:ERRor?
SYSTem:LANGuage?
SYSTem OPTions?
SYSTem:OPTions:DESCription? <option code>
SYSTem:READy?
SYSTem:RQUeue?

    :QMODe?
    :SNUMber[:SET]?
    :SNUMber:STATe?

SYSTem:SNUMber?
SYSTem:TIMeout?
SYSTem:VERSion?
```

***RST Value**

Parameter	Values
<language>	SYNC
<qmode>	NEXT
<snumber>	0
<state>	OFF
<timeout>	0 (infinite timeout)

Limits

Parameter	Values
<language>	SYNChronous | ASYNchronous
<qmode>	NEXT | ALL
<snumber>	0 to 2,147,483,647 for normal sequence numbers –1 for empty queue –2 for queue overflow
<state>	0, 1, ON, OFF

Parameter	Values
<timeout>	0, disable timeout 1–100,000 seconds
SYSTem:READy?	0|1

Related Commands N/A

Command Description

SYSTem:LANGuage <language>
This command enables or disables the IEEE 488.2 Message Exchange Protocol Enforcer (MEPE). MEPE is enabled when you select Synchronous Protocol. The MEPE is disabled and Query Responses are tagged when the you select ASYNchronous Protocol. See *Instrument Functions* for more information on Synchronous and Asynchronous Protocols.

SYSTem:RQUeue:QMODe <qmode>
This command provides a fast way to retrieve all responses in the Response Queue. When the mode is NEXT, a word serial read will retrieve a single response (assuming one exists). If the mode is ALL, then a word serial response will retrieve all responses, separated by semicolons, currently in the response queue. See the *Theory of Operations* in *Operating Basics* for a detailed description of this mechanism.

SYSTem:RQUeue:SNUMber[:SET] <snumber>
When in ASYNchronous mode, this command specifies what the next sequence number should be. By default, the sequence numbers start from zero and increment by one after each query is received. Since the query for the current sequence number has a sequence number in its tag, the query itself is the sequence number which will be used to tag the next Query Response. See the *Theory of Operations* in *Operating Basics* for a detailed description of this mechanism.

SYSTem:RQUeue:SNUMber:STATe <state>
When in ASYNchronous mode, this command specifies whether Query Responses are tagged with ASCII labels, or with the numeric instrument number and an incrementing sequence number. See the *Theory of Operations* section for a detailed description of this mechanism.

SYSTem:TIMeout <timeout in seconds>
Specifies the maximum amount of time a query is allowed to take. If the query has not completed within the specified time, the instrument is aborted and the TIMEOUT response is placed in the response queue. This command has no

effect if the VX4101A is using the Asynchronous Protocol. For more information on Asynchronous Protocol, see *Theory of Operation* in *Operating Basics*.

Query Response

SYSTem:ERRor?
This query returns returns in first-in first-out order any error messages which have been queued. Error messages are of the form *<error #>,error string*. If the queue is empty, this is reported as 0, *No Error*. Depending upon the type of error, the error string may indicate which instrument generated the error.The two possible language

SYSTem:LANGuage?
This query returns the language, or mode currently selected for the instrument. The possible languages are Synchronous or Asynchronous.

SYSTem:OPTions?
This query returns the options for which the VX4101A is currently configured. The options are returned as the following codes:

- 1C (Counter 500 MHz Frequency Option)
- 2C (Counter Channel 3 Prescalar Option)
- 1D (32 Digital I/O and eight Relay Drivers)
- 1A (16 Channel Digital to Analog Converter)
- 1T (High stability TCX0 reference)
- NOOPT (no option installed)

SYSTem:OPTions:DESCription? <option code>
Provides textual description of the option codes returned by the SYSTem:OPTions? query. See *SYSTem:OPTions?* for a complete listing of the possible options.

SYSTem:READy?
The VX4101A negates SYSFAIL* approximately three seconds after power-on and is ready for the word serial *Begin Normal Operations* command. Full instrument initialization completes in approximately twenty seconds from power-on to completion.

***NOTE**. Polling, or querying the instrument as quickly as possible, might slow the completion of the second initialization stage. A delay of one second between SYSTem:READy? is recommended.*

You use this query to determine if full power-on initialization has completed for all instruments and that all commands and queries are available. A return value of (1) indicates that initialization is complete and that all commands are loaded. A return value of (0) indicates that only the following set of commands are available:

Table 3–1: Commands Available at Power-On

Command Syntax	Command Type	Description
*CLS	IEEE 488.2	Clears all event status registers and queues.
*ESE	IEEE 488.2	Sets the contents of the IEEE 488.2 Standard Event Status Enable Register.
*ESE?	IEEE 488.2	Queries the contents of the IEEE 488.2 Standard Event Status Enable Register.
*ESR	IEEE 488.2	Sets the contents of the IEEE 488.2 Standard Event Status Register.
*ESR?	IEEE 488.2	Queries the contents of the IEEE 488.2 Standard Event Status Register.
*RST	IEEE 488.2	Resets VX4101A components to their reset values.
NOTE: *See the command summary for the *RST command in the section* IEEE 488.2 Commands *for the specific reset values.*		
*SRE	IEEE 488.2	Sets the contents of the IEEE 488.2 Service Request Enable Register.
*SRE?	IEEE 488.2	Queries the contents of the IEEE 488.2 Service Request Enable Register.
*STB	IEEE 488.2	Queries the contents of the IEEE 488.2 Status Byte Register.
STATus:OPERation:CONDition?	Status and Events	Returns the current operational status of the VX4101A
STATus:OPERation:ENABle	Status and Events	Sets the Operational Enable Register for the VX4101A
STATus:OPERation:ENABle?	Status and Events	Queries the Operational Enable Register for the for the VX4101A

Table 3–1: Commands Available at Power-On (Cont.)

Command Syntax	Command Type	Description
STATus:OPERation[:EVENt]?	Status and Events	Returns contents of Operational Event Register for the VX4101A.
STATus:OPERation:NTRansition	Status and Events	Sets the Operational Negative Transition Filter for the VX4101A.
STATus:OPERation:NTRansition?	Status and Events	Queries the Operational Negative Transition Filter for the VX4101A.
STATus:OPERation:PTRansition	Status and Events	Sets the Operational Positive Transition Filter for the VX4101A
STATus:OPERation:PTRansition?	Status and Events	Queries the Operational Positive Transition Filter for the VX4101A
STATus:PRESet	Status and Events Reporting System	Clears the enable registers of all Operational Status Registers, sets all Positive Transition Filters, and clears all Negative Transition Filters
STATus:QUEue:ENABle	Status and Events Reporting System	Allows you to specify which errors and events, by error number, should be placed in the error/event queue
STATus:QUEue[:NEXT]?	Status and Events Reporting System	Returns next item from error/event queue in FIFO order
STATus:QUEue:ENABle?	Status and Event Reporting System	Queries the Questionable Enable Register for the VX4101A
STATus:QUEStionable[EVENt]?	Status and Event Reporting System	Returns contents of Questionable Event Register for the VX4101A
STATus:QUEStionable:CONDition?	Status and Event Reporting System	Returns contents of Questionable Condition Register for the VX4101A
STATus:QUEStionable:ENABle	Status and Event Reporting System	Sets the Questionable Enable Register for the VX4101A
STATus:QUEStionable:ENABle?	Status and Event Reporting System	Queries the Questionable Enable Register for the VX4101A
STATus:QUEStionable:PTRansition	Status and Event Reporting System	Sets the Questionable Positive Transition Filter for the VX4101A

Table 3–1: Commands Available at Power-On (Cont.)

Command Syntax	Command Type	Description
STATus:QUEStionable:PTRansition?	Status and Event Reporting System	Queries the Questionable Positive Transition Filter for the VX4101A
STATus:QUEStionable:NTRansition	Status and Event Reporting System	Sets the Questionable Negative Transition Filter for the VX4101A
STATus:QUEStionable:NTRansition?	Status and Event Reporting System	Queries the Questionable Negative Transition Filter for the VX4101A
SYSTem:ERRor?	VX4101A MultiPaq ™ Instrument Commands	Returns in first-in first-out order any error messages which have been queued for any instruments
SYSTem:OPTions?	VX4101A MultiPaq ™ Instrument Commands	Returns the options for which the VX4101A is currently configured
SYSTem:READy?	VX4101A MultiPaq ™ Instrument Commands	Queries for completion of the power-on initialization sequence for all instruments
***NOTE:** Polling the instrument, i.e. querying it as fast as possible, might slow the completion of the second initialization stage. You should have a delay of at least one second between SYSTem:READy? queries.*		
SYSTem:SNUMber?	VX4101A MultiPaq ™ Instrument Commands	This query returns the serial number of the VX4101A
SYSTem:VERSion?	VX4101A MultiPaq ™ Instrument Commands	Returns the firmware version of the VX4101A

***NOTE**. Until initialization is complete, the VX4101A will recognize only the commands listed above. Attempting to enter any other command will return a Command Not Found Error.*

SYSTem:RQUeue:QMODe?
Queries the mode in which the instrument returns responses in the Response Queue.

SYSTem:RQUeue:SNUMber[:SET]?
This query returns the next sequence number for Asynchronous Mode.

SYSTem:RQUeue:SNUMber:STATe?
Returns the SNUMber state of 0|1|ON|OFF when the VX4101A is in Asynchronous Mode.

SYSTem:SNUMber?
This query returns the serial number of the VX4101A.

SYSTem:TIMeout?
This query returns the maximum amount of time a query is allowed to take when the VX4101A is in Synchronous Mode.

SYSTem:VERSion?
Returns the firmware version of the instrument in <major>.<minor>.<sub-minor> form.

Examples

SYSTem:ERRor?

Command	Response/description
SYST:ERR?	–113,“Undefined header; Command not found; XXX\0D”
SYST:ERR?	0,“No error”

SYSTem:LANGuage?

Command	Response
*RST	
SYST:LANG?	SYNC
SYST:LANG ASYN	
SYST:LANG?	ASYN

SYSTem:OPTions?

Command	Response
SYST:OPT?	NOOPT

SYSTem:OPTions:DESCription

Command	Response
SYST:OPT?	1C
SYST:OPT:DESC? “1C”	Counter 500 MHz option

SYSTem:READy?

Command	Response
BNO (*Begin Normal Operations)* command sent	
SYST:READy?	0
INST:CAT?	
SYST:ERR?	–113, "Undefined header; Command not found; inst:cat\n?
Wait approximately 20 seconds	
SYST:READY?	1
INST:CAT?	VX4101A. SurePath, DMM, Counter, DAC, DIGI, DIGO, RDriv

SYSTem:RQUeue:QMODe

Command	Response
SYST:LANG ASYN	
SYST:RQU:QMOD ALL	
*IDN?	
SYST:ERR?	"VX4101A:*IDN?", Tektronix, VX4101A, B00000021, Firmware v.2.0.0/SCPI:95.0; "VX4101A:SYST:ERR?", 0, "No error"

SYSTem:RQUeue:SNUMber[:SET]

Command	Response
SYST:LANG ASYN	
SYST:RQU:SNUM 100	
SYST:RQU:SNUM?	0:100,101

SYSTem:RQUeue:SNUMber:STATe

Command	Response
SYST:LANG ASYN	
SYST:RQU:SNUM:STAT?	"VX4101A:SYST:RQU:SNUM:STAT?",0
SYST:RQU:SNUM:STAT ON	
SYST:RQU:SNUM:STAT?	0:1,1

SYSTem:SNUMber?

Command	Response
SYST:SNUM?	B0000001

SYSTem:TIMeout

Command	Response
SYST:TIM 10	
INST:SEL DMM	
CONF:ARR:VOLT:DC 512,MAX	
READ?	
<wait 10 seconds>	
	TIMEOUT

SYSTem:VERSion?

Command	Response
SYST:VERS?	2.0.0

TRIGger Subsystem

Command Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:FIRe <command trigger>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:TIMer <timer seconds>
```

Query Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:TIMer?
```

***RST Value**

Parameter	Values
<command trigger>	0–4
<timer seconds>	0 seconds, <off>

Limits

Parameter	Values
<command trigger>	0-4
<timer seconds>	1e–6 seconds to 65.535e–3 seconds in 1e–6 second steps

Related Commands N/A

Command Description

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:FIRe <command trigger>
Generates one of five software triggers: COMMAND0 through COMMAND4.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:TIMer <timer seconds>
Sets or queries the value of the period trigger source. Note that this is a global class command and there is only one periodic timer for the VX4101A. Any time you program the TRIG:TIM command, the period of the timer trigger will be changed for all instruments using it as a source.

Table 3–2: Trigger Resolution (in µs)

Lower Bound	Upper Bound	Resolution
≥ 2	< 510	2
≥ 510	< 1275	5
≥ 1275	< 2040	8
≥ 2040	< 6375	25
≥ 6375	< 8160	32
≥ 8160	< 12750	50
≥ 12750	< 25500	100

Query Response

TRIGger ([:SEQuence1]|:STARt)[LAYer]:TIMer
This query returns the value of the periodic trigger source in seconds.

Examples

TRIGger ([:SEQuence1]|:STARt)[:LAYer]:FIRe <0-4>

Command	Response
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
INST:SEL DMM	
CONF:VOLT:DC	
TRIG:SOUR COMM0	
INIT	
FETC:COUN?	0
TRIG:FIR0	
FETC:COUN?	1
FETC?	–6.03720E+00

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:TIMer

Command	Response
TRIG:TIM 500E–6	
TRIG:TIM?	500E–6
INST:SEL DMM	
TRIG:TIM?	500E–6
TRIG:TIM 45E–3	
INST:SEL COUNTER	
TRIG:TIM?	45E–3
TRIG:TIM .001	
TRIG:TIM?	1E–3
INST:SEL DMM	
TRIG:TIM?	1E–3

SCPI Commands for the Counter

This section contains SCPI commands for the Universal Counter (Counter). The commands are organized by command subsystem. Some commands within each subsystem are organized by type of command rather than in alphabetical order. The command subsystems for the Counter are as follows:

Command Summary

The following is a listing of the available command subsystems and syntax:

ARM Subsystem

Commands

```
ARM([:SEQuence1]|:STARt)[:LAYer]

    :DELay <time delay>
    :ECOunt <event delay>
    :IMMediate
    :LEVel <threshold>
    :MODE <mode>
    :SLOPe <slope>
    :SOURce <source>

ARM(:SEQuence2|:STOP)[:LAYer]

    :DELay <time delay>
    :ECOunt <event delay>
    :IMMediate
    :SOURce <source>
```

Queries

```
ARM([:SEQuence1]|:STARt)[:LAYer]

    :DELay?
    :ECOunt?
    :LEVel?
    :MODE?
    :SLOPe?
    :SOURce?

        :CATalog[:ALL]?

            :DELayable?
            :FIXed?
```

```
ARM(:SEQuence2|:STOP) [:LAYer]

    :DELay?
    :ECOunt?
    :SOURce?

        :CATalog[:ALL]?

            :DELayable?
            :FIXed?
```

CALCulate Subsystem

Commands

```
CALCulate:LIMit

    :ENVelope[:DATA] <threshold1>,<threshold2>
    :LOWer[:DATA] <threshold>
    :UPPer[:DATA] <threshold>

CALCulate:TRANsform:HISTogram:COUNt <numeric_value>
CALCulate:TRANsform:HISTogram:POINts <numeric_value>
CALCulate:TRANsform:HISTogram:RANGe <min>,<max>
CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>
```

Queries

```
CALCulate:AVERage? [<count>[,<offset>[,<step_size>]]]
CALCulate:LIMit

    ENVelope
    :FCOunt?
    :LOWer [DATA]?
    :REPort [:DATA]?
    :UPPER [DATA]

CALCulate:MEDian? [<count>[,<offset>[,<step_size>]]]
CALCulate:MINimum? [<count>[,<offset>[,<step_size>]]]
CALCulate:MAXimum? [<count>[,<offset>[,<step_size>]]]
CALCulate:SDEViation? [<count>[,<offset>[,<step_size>]]]
CALCulate:TRANsform:HISTogram?

    :ABOVe?
    :BELow?
    :COUNt?
    :POINts?
    :RANGe?
    :RANGe:AUTO?

CALCulate:VARiance? [<count>[,<offset>[,<step_size>]]]
```

CALibrate Subsystem

Commands

```
CALibrate:ARM[:VALue] <arm input voltage>
CALibrate[<channel>]

    :DELay <cross channel specifications>
    :HYSTeresis
    :LFCOmp
    :LINearity
    :VALue <input voltage>
    :ZERO

CALibrate:DTI
CALibrate3:BIAS
CALibrate[1|2|3]:ROSCillator <input frequency>
```

CONFigure Subsystem

Commands

```
CONFigure[1|2|3]([:SCALar]|:ARRay)

    :FREQuency [<array size>[,<expected value>[,resolution>]]]

    :FREQuency:RATio [<array size>[,<expected
    value>[,resolution>]]]

    :PERiod [<array size>[,<expected value>[,resolution>]]]

CONFigure[1|2]([:SCALar]|:ARRay)

    :NDUTy cycle|PDUTCycle|DCYCle [<array size>[,<dcycle
    reference>[,<expected value>[,resolution>]]]

    :NWIDth|PWIDth [<array size>[,<pwidth reference>[,<expected
    value>[,resolution>]]]

    :PHASe [<array size>[,<expected value>[,resolution>]]]

    :RTIMe|FTIMe|RISE:TIME|FALL:TIME [<array size>[,<low
    reference>[,<high reference> [,<expected
    value>[,resolution>]]]]]

    :TINTerval [<array size>[,<expected value>[,resolution>]]]

    [:VOLTage]

        :AC [<array size>[,<expected value>[,resolution>]]]
        :DC [<array size>[,<expected value>[,resolution>]]]
        :MINimum [<array size>[,<expected value>[,resolution>]]]
        :MAXimum [<array size>[,<expected value>[,resolution>]]]
        :PTPeak [<array size>[,<expected value>[,resolution>]]]
```

```
CONFigure[1|2|11|12|21|22]([:SCALar]|:ARRay)

    :TINTerval:DELay:TIME|EVENts [<array size>[,<delay
    time>|<delay events>[,<expected value>[,resolution>]]]]]

CONFigure[1|2|10|20]([:SCALar]|:ARRay)

    :TOTalize
```

***NOTE**. If you specify SCALar in the command, <array size> is not a valid parameter. If ARRay is specified, <array size> is a required parameter.*

Queries

```
CONFigure?
```

FETCh? Subsystem

Queries

```
FETCh[:...]? [<count>[,<start>[,<step>]]]

    :DCYCle?
    :FALL:TIME?
    :FREQuency?
    :FREQuency:RATio?
    :FTIMe?[<count>
    :NDUTycycle?
    :PDUTycycle?
    :NWIDth?
    :PWIDth?
    :PHASe?
    :PERiod?
    :RISE:TIME?
    :RTIMe?
    :TINTerval?
    :TINTerval:DELay?

FETCh:COUNt?
FETCh:TOTalize?
FETCh[:VOLTage][:...]? [<count>[,<start>[,<step>]]]

    :AC?
    :DC?
    :MINimum?
    :MAXimum?
    :PTPeak?
```

INITiate Subsystem

Commands

```
INITiate[:IMMediate]
INITiate:CONTinuous [ON]|OFF|0|1
```

Queries

```
INITiate:CONTinuous?
```

INPut Subsystem

Commands

```
INPut[1|2]

        :ATTenuation <attenuation>|DEFault|MINimum|MAXimum
        :COMParator[1|2]:LEVel

            [:ABSolute] <absolute level>|DEFault|MINimum|MAXimum
            :RELative <relative level>

        :HYSTeresis

            [:ABSolute] <absolute level>|DEFault|MINimum|MAXimum
            :RELative <relative level>

    :SLOPe <slope>|DEFault
    :COUPling <coupling>
    :FILTer[:LPASs]

        :FREQuency <frequency>|DEFault|MINimum|MAXimum
        [:STATe] <filter state>|DEFault

    :GAIN <gain>|DEFault|MINimum|MAXimum
    :IMPedance <impedance>|DEFault
    :OFFSet

        [:ABSolute] <absolute offset>|DEFault|MINimum|MAXimum
        :RELative <relative offset>

    :SETup <expected ptp>[,<expected offset>]

INPut:SETup:AUTO <auto>
```

Queries

```
INPut[1|2]

    :ATTenuation? [DEFault|MINimum|MAXimum]
    :COMParator[1|2]

        :LEVel[:ABSolute]? [DEFault|MINimum|MAXimum]
        :LEVel:RELative?

    :SLOPe? [DEFault]
```

```
    :HYSTeresis

        [:ABSolute]? [DEFault|MINimum|MAXimum]
        :RELative?

    :COUPling? [DEFault]
    :IMPedance? [DEFault]
    :FILTer[:LPASs]

        :FREQuency?[MINimum|MAXimum|DEFault]

        [:STATe]?[DEFault]

    :GAIN? [DEFault|MINimum|MAXimum]
    :OFFSet

        [:ABSolute]? [MINimum|MAXimum|DEFault]
        :RELative?

INPut:SETup:AUTO?[DEFault]
```

INSTrument Subsystem

Commands

```
INSTrument

    :ABORt
    :RESet
```

MEASure? Subsystem

Queries

```
MEASure[1|2|3]([:SCALar]|:ARRay)

    :FREQuency? [<array size>[,<expected value>[,resolution>]]]

    :FREQuency:RATio? [<array size>[,<second channel>[,<expected
    value>[,resolution>]]]]

    :PERiod? [<array size>[,<expected value>[,resolution>]]]

MEASure[1|2]([:SCALar]|:ARRay)

    :NDUTycycle|PDUTycycle|DCYCle? [<array
    size>[,<reference>[,<expected value>[,resolution>]]]]

    :NWIDth|PWIDth? [<array size>[,<reference>[,<expected
    value>[,resolution>]]]]

    :PHASe? [<array size>[,<expected value>[,resolution>]]]

    :RTIMe|FTIMe|RISE:TIME|FALL:TIME? [<array size>[,<low
    reference>[,<high reference> [,<expected
    value>[,resolution>]]]]]
```

```
   :TINTerval? [<array size>[,<expected value>[,resolution>]]]

   [:VOLTage]

      :AC? [<array size>[,<expected value>[,resolution>]]]

      :DC? [<array size>[,<expected value>[,resolution>]]]

      :MINimum? [<array size>[,<expected value>[,resolution>]]]

      :MAXimum? [<array size>[,<expected value>[,resolution>]]]

      :PTPeak? [<array size>[,<expected value>[,resolution>]]]

MEASure[1|2|11|12|21|22]([:SCALar]|:ARRay)

   :TINTerval:DELay:TIME|EVENts? [<array size>[,<delay
   time>|<delay events>[,<expected value>[,resolution>]]]]

MEASure[1|2|10|20]([:SCALar]|:ARRay)

   :TOTalize?
```

***NOTE**. If SCALar is specified, <array size> is not a valid parameter. If ARRay is specified, <array size> is a required parameter.*

OUTPut Subsystem

Commands

```
OUTPut:TRIGger:SOURce <channel>,<count>
```

Queries

```
OUTPut:TRIGger:SOURce?
```

READ? Subsystem

Queries

```
READ?
```

SENSe Subsystem

Commands

```
SENSE:APERture <time>|DEFault|MINimum|MAXimum
SENSe:COUNt <array size>|DEFault|MINimum|MAXimum
SENSe:MODe <mode>
SENSe:EVENts <# of events>
SENSe[1|2|3]:FUNCtion

   "TOTALize"
   "FREQuency"
```

```
    "FREQuency:RATio <second channel>"
    "PERiod"

SENSe[1,2]:FUNCtion

  "DCYCle"
  "FALL:TIME"
  "FTIMe"
  "NDUTycycle"
  "NWIDth"
  "PDUTycycle"
  "PHASe"
  "PWIDth"
  "RISE:TIME"
  "RTIMe"
  "TINTerval"
  "VOLTage:AC"
  "VOLTage:DC"
  "VOLTage:MINimum"
  "VOLTage:MAXimum"
  "VOLTage:PTPeak"

SENSe[1|2|11|12|21|22]:FUNCtion

  "TINTerval:DELay:TIME"
  "TINTerval:DELay:EVENts"

SENSe[1|2|10|20]:FUNCtion "TOTalize"
SENSe:TINTerval:DELay:EVENts <event
delay>|MINimum|MAXimum|DEFault
SENSE:TINTerval:DELay:TIME <time delay>|DEFault|MINimum|MAXimum
```

Queries

```
SENSe:APERture? [DEFault|MINimum|MAXimum]
SENSe:COUNt? [DEFault|MINimum|MAXimum]
SENSe:EVENts? [DEFault|MINimum|MAXimum]
SENSe:MODe? [DEFault]
SENSe:FUNCtion?
SENSe:TINTerval:DELay:EVENts? [DEFault]|MINimum]|MAXimum]
```

SOURce Subsystem

Commands

```
SOURce:COSCillator[:SOURce] <source>
```

Queries

```
SOURce:COSCillator[:SOURce]?

SOURce:COSCillator:VALue?
```

STATus? Subsystem

```
STATus:OPERation:CONDition?
```

TEST Subsystem

```
TEST: ALL?
```

UNIT Subsystem

Commands

```
UNIT:ANGLe <units>[,<zero>]
```

Queries

```
UNIT:ANGLe?
```

ARM Subsystem

Command Syntax

```
ARM([:SEQuence1]|:STARt)[:LAYer]

    :DELay <time delay>
    :ECOunt <event delay>
    :IMMediate
    :LEVel <threshold>
    :MODE <mode>
    :SLOPe <slope>
    :SOURce <source>

ARM(:SEQuence2|:STOP)[:LAYer]

    :DELay <time delay>
    :ECOunt <event delay>
    :IMMediate
    :SOURce <source>
```

Query Syntax

```
ARM([:SEQuence1]|:STARt)[:LAYer]

    :DELay?
    :ECOunt?
    :LEVel?
    :MODE?
```

```
    :SLOPe?
    :SOURce?

        :CATalog[:ALL]?

            :DELayable?
            :FIXed?

ARM(:SEQuence2|:STOP) [:LAYer]

    :DELay?
    :ECOunt?
    :SOURce?

        :CATalog[:ALL]?

            :DELayable?
            :FIXed?
```

Command Class

Instrument (Counter)

***RST Value**

Parameter	Value
<time delay>	0 seconds (pass-through)
<event delay>	0 triggers (pass-through)
<threshold>	1.4 V
<mode>	ALL
<slope>	POSitive
<source>	IMMEDIATE

Limits

Parameter	Value
<time delay>	<delay in seconds> 0 = pass through, 1 to 65,535 triggers in 1 µs steps
<event delay>	<triggers to count> 0 = pass through, 1 to 65,535 triggers
<threshold>	TTL = 1.6 V ECL = –1.3 V ZERO = 0.0 V MAXimum = 20.0 V MINimum = –20.0 V AUTO = 1.4 V –20 to +20 V
<mode>	ONCE|ALL

Parameter	Value
<slope>	POSitive | NEGative
<source>	BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, CTR_CHAN2, DAC, DIGI, DIGO, INTERNAL LEVEL

Related Commands N/A

Description The following summarizes the ARM subsystem functionality:

ARM ([:SEQuence1] |:STARt) [:LAYer]:DELay <time delay>
Specifies a time delay to occur after receipt of an arm signal prior to actually arming the Counter. If the arming source selected is fixed, this command will have no effect on the Counter arming. This command always zeros the event count delay, so specifying a delay of zero places the arm subsystem in pass-through mode. In this mode, the instrument arms immediately upon receipt of a trigger.

ARM ([:SEQuence1] |:STARt) [:LAYer]:ECOunt <time delay>
Specifies the number of arm signals to count prior to arming the Counter. Upon receipt of arming signal N (where N is the number specified in the command), the Counter will be armed to take measurements. If you are using a fixed arm source, this command will have no effect on the Counter arming. This command always zeros the delay by time parameter, so specifying an event count of zero places the arm subsystem in pass-through mode. In this mode, the instrument arms immediately upon receipt of a trigger.

ARM ([:SEQuence1] |:STARt) [:LAYer]:IMMediate
Causes a one-time entry by the Counter into the armed state without receiving the specified start arm signal.

ARM ([:SEQuence1] |:STARt) [:LAYer]:LEVel <threshold>
Specifies the comparator level of Counter External Arm Signal. When the External Arm Signal crosses this level with the appropriate slope, the instrument activates the CTR_EXTARM trigger source.

ARM ([:SEQuence1] |:STARt) [:LAYer]:MODE <mode>
When the Counter has been configured for an array measurement, this command specifies whether the Counter will perform one or all operations when an ARM command is received. If the mode is ALL, then all operations will be completed

upon receipt of one arm signal. If the mode is ONCE, then the instrument will perform one operation and reenter the initiated state. This will continue until the specified number of arm signals have been received. At that point, the specified number of operations have been completed.

ARM ([:SEQuence1] |:STARt) [:LAYer]:SLOPe <slope>

Specifies the slope of the Counter External Arm Signal. When the slope is POSitive, the CTR_EXTARM trigger source will be activated by a rising edge passing through the specified level. When the slope is NEGative, the CTR_EXTARM trigger source will be activated by a falling edge passing through the specified level. For more information see the ARM:LEVel command.

ARM ([:SEQuence1] |:STARt) [:LAYer]:SOURce <source>

Selects or queries the start arming source to be used when the Counter is initiated.

ARM (:SEQuence2 |:STOP) [:LAYer]:DELay <time delay>

Specifies a time delay to occur after receipt of a stop arm signal prior to actually disarming the Counter. If the stop arming source selected is fixed, this command will have no effect on the Counter arming. This command always zeros the event count delay, so specifying a delay of zero places the arm subsystem in pass-through mode. In this mode, the instrument disarms immediately upon receipt of an arming signal.

ARM (:SEQuence2 |:STOP) [:LAYer]:ECOunt <time delay>

Specifies the number of stop arm signals the instrument will count prior to disarming the Counter. Upon receipt of an arming signal, the Counter will exit the armed state. If the arm source selected is fixed, this command will have no effect on disarming the Counter. This command always zeros the delay by time parameter, so specifying an event count of zero places the arm subsystem in pass-through mode. In this mode, the instrument arms immediately upon receipt of a trigger.

ARM (:SEQuence2 |:STOP) [:LAYer]:IMMediate

Cause a one-time exit by the Counter from the armed state without receiving the specified stop arm signal.

ARM (:SEQuence2 |:STOP) [:LAYer]:SOURce <source>

Selects or queries the stop arming source to be used when the Counter is initiated.

Query Response

Query	Response
ARM ([:SEQuence1] |:STARt) [:LAYer]:DELay?	<delay in seconds>
ARM ([:SEQuence1] |:STARt) [:LAYer]:ECOunt?	<triggers to count>
ARM ([:SEQuence1] |:STARt) [:LAYer]:LEVel?	TTL | ECL | ZERO | MAXimum | MINimum | AUTO | <desired voltage>
ARM ([:SEQuence1] |:STARt) [:LAYer]:MODE?	ONCE|ALL
ARM ([:SEQuence1] |:STARt) [:LAYer]:SLOPe?	POSitive | NEGative
ARM ([:SEQuence1] |:STARt) [:LAYer]:SOURce?	<current source>
ARM ([:SEQuence1] |:STARt) [:LAYer]:SOURce:CATalog [:ALL]?	Lists all trigger sources available for use with the ARM:SOUR command. Specifies the start arming source for the Counter. The possible responses are: HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, CTR_CHAN2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7
ARM ([:SEQuence1] |:STARt) [:LAYer]:SOURce:CATalog: DELayable?	Lists all delayable trigger sources available for use with the ARM:SOUR command. This command specifies the start arming source for the Counter. The possible responses are as follows: BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
ARM ([:SEQuence1] |:STARt) [:LAYer]:SOURce:CATalog: FIXed?	Lists all fixed trigger sources available for use with the ARM:SOUR command. This command specifies the start arming source for the Counter. The possible responses are as follows: HOLD,IMMEDIATE,TIMER,CTR_CHAN2
ARM(:SEQuence2 |:STOP) [:LAYer]:DELay?	<delay in seconds>
ARM (:SEQuence2 |:STOP) [:LAYer]:ECOunt?	<triggers to count>
ARM(:SEQuence2 |:STOP) [:LAYer]:SOURce?	<current source>

Query	Response
ARM(:SEQuence2 |:STOP) [:LAYer]:SOURce:CATalog[:ALL]?	Lists all trigger sources available for use with the ARM:STOP:SOUR command. This command specifies the stop arming source for the Counter. The possible responses are as follows: HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, INTERNAL, LEVEL
ARM(:SEQuence2 |:STOP) [:LAYer]:SOURce:CATalog: DELayable?	Lists all delayable trigger sources available for use with the ARM:STOP:SOUR command. This command specifies the stop arming source for the Counter. The possible responses are as follows: BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
ARM (:SEQuence2 |:STOP) [:LAYer]:SOURce:CATalog: FIXed?	Lists all fixed trigger sources available for use with the ARM:STOP:SOUR command. This command specifies the stop arming source for the Counter. The possible responses are as follows: HOLD, IMMEDIATE, TIMER, INTERNAL, LEVEL

Examples

ARM ([:SEQuence1] |:STARt) [:LAYer]:DELay <delay in seconds>

Command	Description
INST:SEL COUNTER	Selects the Counter
CONF:VOLT:DC	Sets the Counter for DC voltage measurements
ARM:SOUR TTLT0	Arms the TTLT0 source
ARM:DEL 1E–3	Sets time delay for 1E–3 seconds
INIT	Begins the measurement

ARM ([:SEQuence1] |:STARt) [:LAYer]:ECOunt

Command	Response
INST:SEL COUNTER	Selects the Counter
CONF:VOLT:DC	Sets the Counter for DC voltage measurements
ARM:SOUR TTLT0	Arms the TTLT0 source
ARM:ECO 100	Sets the event counter for 100 events
INIT	Begins the measurement

ARM ([:SEQuence1] |:STARt) [:LAYer]:IMMediate

Command	Response
INST:SEL COUNTER	Selects the Counter
ARM:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, CTR_CHAN2, DAC, DIGI, DIGO
CONF:FREQ	Configures the Counter to read frequency
ARM:SOUR TTLT0	Arms source TTLT0
INIT	Begins the measurement
FETC:COUN?	0 Returns the number of measurements
TRIG:IMM	Arms Counter to take an immediate measurement
FETC:COUN?	1 Returns the number of measurements
FETC?	1.234567890E6

ARM ([:SEQuence1] |:STARt) [:LAYer]:LEVel

Command	Response
INST:SEL COUNTER	Selects the Counter
ARM:SLOP NEG	
ARM:LEV TTL	
INST SEL DMM	
TRIG:SOUR CTR_EXTARM	
CONF:VOLT:DC	
INIT	
<wait until falling edge of external arm crosses 1.6 V>	
FETC?	+1.00000E–01

ARM ([:SEQuence1] |:STARt) [:LAYer]:MODE

Command	Response
INST:SEL COUNTER	Selects the Counter
CONF:ARR:FREQ 3	Configures the Counter to take three array measurements

Command	Response
ARM:MODE ONCE	Sets up Counter to take one measurement and then re-enter the initiated state.
ARM:SOUR COMM0	Selects COMM0 as the start arming source
INIT	Begins the measurement
FETC:COUN?	0
TRIG:FIR0	
FETC:COUN?	1
TRIG:FIR0	
TRIG:FIR0	
FETCH:COUN?	3
FETC?	#2411.234567891E6, 1.234567890E6, 1.234567892E6

ARM ([:SEQuence1] |:STARt) [:LAYer]:SLOPe

See ARM:LEVel example

ARM ([:SEQuence1] |:STARt)[:LAYer]:SOURce:CATalog[:ALL]?

Command	Response
INST:SEL COUNTER	
ARM:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, CTR_CHAN2
CONF:FREQ	
ARM:SOUR BUS	
INIT	
FETC:COUN?	0
*TRG	
FETC:COUN?	1
FETC?	1.234567890E6

ARM ([:SEQuence1] |:STARt)[:LAYer]:SOURce:CATalog:DELayable?

Command	Response
ARM:SOUR:CAT:DEL?	BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO

ARM (:SEQuence2 |:STOP) [:LAYer]:SOURce:CATalog:[ALL]

Command	Response
ARM:STOP:SOUR:CAT?	HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, CTR_CHAN2, INTERNAL, LEVEL

ARM ([:SEQuence1] |:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Command	Response
ARM:SOUR:CAT:FIX?	HOLD,IMMEDIATE,TIMER,CTR_CHAN2

ARM (:SEQuence2 |:STOP) [:LAYer]:DELay

Command	Response
INST:SEL COUNTER	
CONF:VOLT:DC	
ARM:STOP:SOUR TTLT0	
ARM:STOP:DEL 1E–3	
INIT	

ARM (:SEQuence2 |:STOP) [:LAYer]:ECOunt

Command	Response
INST:SEL COUNTER	
CONF:VOLT:DC	
ARM:STOP:SOUR TTLT0	
ARM:STOP:ECO 100	
INIT	

ARM (:SEQuence2 |:STOP) [:LAYer]:IMMediate

Command	Response
CONF:FREQ	
ARM:STOP:SOUR TTLT0	
INIT	
FETC:COUN?	0
ARM:STOP:IMM	
FETC:COUN?	0

ARM (:SEQuence2 |:STOP) [:LAYer]:SOURce:CATalog:ALL

Command	Response
INST:SEL COUNTER	
ARM:STOP:SOUR:CAT?	HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, INTERNAL, LEVEL, DAC, DIGI, DIGO
CONF:ARR:FREQ 512	
ARM:SOUR BUS	
ARM:STOP:SOUR BUS	
INIT	
FETC:COUN?	0
*TRG	
FETC:COUN?	23
FETC:COUN?	47
*TRG	
FETC:COUN?	53

ARM(:SEQuence2|:STOP)[:LAYer]:SOURce:CATalog[:ALL]?

Command	Response
ARM:STOP:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, CTR_CHAN2, INTERNAL, LEVEL

ARM (:SEQuence2 |:STOP) [:LAYer]:SOURce:CATalog:DELayable?

Command	Response
ARM:STOP:SOUR:CAT:DEL?	BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO

ARM (:SEQuence2 |:STOP) [:LAYer]:SOURce:CATalog:FIXed?

Command	Response
ARM:STOP:SOUR:CAT:FIX?	HOLD,IMMEDIATE,TIMER,INTERNAL,LEVEL

CALCulate Subsystem

The CALCulate commands enable you to process data acquired by the instrument with SENSe commands.

Command Syntax

```
CALCulate:LIMit

   :ENVelope[:DATA] <threshold1>,<threshold2>
   :LOWer[:DATA] <threshold>
   :UPPer[:DATA] <threshold>

CALCulate:TRANsform:HISTogram:COUNt <numeric_value>
CALCulate:TRANsform:HISTogram:POINts <numeric_value>
CALCulate:TRANsform:HISTogram:RANGe <min_value>, <max_value>
CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>
```

Query Syntax

```
CALCulate:AVERage? [<count>[,<offset>[,<step_size>]]]

CALCulate:LIMit

   :ENVelope [:DATA]?
   :FCOunt [:DATA]?
   :LOWer [:DATA]?
   :UPPer [:DATA]?
   :REPort [:DATA]?

CALCulate:MEDian? [<count>[,<offset>[,<step_size>]]]
CALCulate:MINimum? [<count>[,<offset>[,<step_size>]]]
CALCulate:MAXimum? [<count>[,<offset>[,<step_size>]]]
CALCulate:SDEViation?[<count>[,<offset>[,<step_size>]]]
CALCulate:TRANsform:HISTogram?
```

```
:ABOVe?
:BELow?
:COUNt?
:POINts?
:RANGe?
:RANGe:AUTO?
```

```
CALCulate:VARiance? [<count>[,<offset>[,<step_size>]]]
```

Command Class Instrument

***RST Value**

Parameter	Value
<threshold1>	N/A
<threshold2>	N/A
<threshold>	N/A
<numeric value>	N/A
<min>	N/A
<max>	N/A
<count>	size of last array measurement
<offset>	1
<step size>	1

Limits

Parameter	Value
<threshold1>	N/A
<threshold2>	N/A
<threshold>	N/A
<numeric value>	N/A
<min>	N/A
<max>	N/A
<count>	1 to 1000
<offset>	1 to 1000
<step size>	1 to 1000

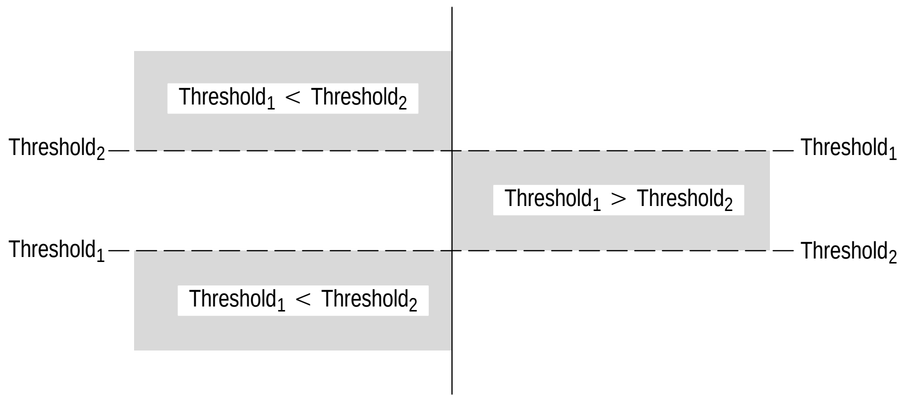
Related Commands N/A

Command Description

CALCulate:LIMit:ENVelope[:DATA] <threshold1>,<threshold2>
This command searches for all the input data values within an *envelope* of values defined as being above, below, or in between, a set of boundary values. The

range of the envelope is determined by the <*threshold1*> and the <*threshold2*> values. For example:

- If the <*threshold2*> is greater than the <*threshold1*>, the range of data values searched for is above the <*threshold1*> value or below the <*threshold1*> value.
- If the <*threshold2*> is less than <*threshold1*>, the range of data values searched for is between the <*threshold1*> and <*threshold2*>.



CALCulate:LIMit:LOWer[:DATA] <threshold>
This command searches for all the input data values below a certain <threshold>.

CALCulate:LIMit:UPPer[:DATA] <threshold>
This command searches for all the input data values above a certain threshold.

CALCulate:TRANsform:HISTogram:COUNt <threshold>
This command determines the number of data points to include in a histogram calculation.

CALCulate:TRANsform:HISTogram:POINts <threshold>
This command sets the number of intervals in a histogram calculation. If a data value is exactly the same value as a limit dividing two bins, the data value will be counted in the next bin, unless the value matches the last interval limit.

Bin #	Limits
1	5 –10
2	10 –15
3	15 –20
4	20 –25

Bin #	Limits
5	25 –30
6	30 –35

Data	Bin
6	1
33	6
15	3
35	6

CALCulate:TRANsform:HISTogram:RANGe <minimum>,<maximum>
This command sets the minimum and maximum values to use in a histogram calculation.

CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>
This command sets the minimum and maximum values to use in a histogram calculation to be automatically determined. The minimum and maximum data points will be used.

Query Response

Query	Response
CALCulate:AVERage? [< count >[,<offset> [,<step_size>]]]	Averages a specified number of measurements in the memory buffer. The optional parameters are for averaging selected values in the memory buffer. <count> specifies the number of data points to calculate. <offset> determines at which point the instrument will begin averaging stored values. <step_size> determines the number of measurements the instrument will skip before taking a value to be averaged.
NOTE: *In order to specify <step size>, you must first enter <count> and <offset> values>. If you do not specify a <count> value, then the instrument will use the most recent <count> as the default.*	
CALCulate:LIMit:ENVelope[: DATA]?	Returns the <threshold1> and <threshold2> values
CALCulate:LIMit:FCOunt?	Performs a limit test on the current available data and returns the number of data points that failed a limit test.
NOTE: *You must send a CALC:limit:upper, lower, or envelope command before this query.*	
CALCulate:LIMit:REPort[: DATA]?	Returns the <memory_index> and <failed_value> values collected in the most recent CALCulate:LIMit:...[:DATA] command.

Query	Response
NOTE: *You must execute the CALCulate:LIMit:...[DATA] command before running the CALCulate:REPort command. If no data values were found, the instrument will return a value of zero. (The first point in memory is “1”)*	
NOTE: *You must send a CALC:LIMit:UPPer, LOWer, or ENVelope command before sending the above query.*	
CALCulate:LIMit:LOWer[:DATA]?	The <threshold> value
CALCulate:LIMit:UPPer[:DATA]?	The <threshold> value
CALCulate:MAXimum? [<count>[<offset> [,<step size>]]]	This query calculates and returns the maximum value for a set of data. The <data_index> returned indicated which data point in memory where the maximum value is. The <maximum_value> returns the value of the maximum data point. If no valid data values were found “No Data” will be returned. The optional parameters are for comparing selected values in the memory buffer. The optional <count> parameter specifies the number of data points to search. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be searched. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be compared. Note: In order to specify a <step_size>, the user must enter the <count> and the <offset> information.

Query	Response
CALCulate:MEDian? [<count>[,<offset>[, <step size>]]]	This query calculates and returns the median value for a set of data. The <data_index> returned indicated which data point in memory where the median value is. The <median_value> returns the value of the median data point. If no valid data values were found “No Data” will be returned. The optional parameters are for searching selected values in the memory buffer. The optional <count> parameter specifies the number of data points to search. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be searched. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be compared. Note: In order to specify a <step_size>, the user must enter the <count> and the <offset> information.
CALCulate:MINimum? [<count>[<offset> [,<step size>]]]	This query calculates and returns the minimum value for a set of data. The <data_index> returned indicated which data point in memory where the minimum value is. The <minimum_value> returns the value of the minimum data point. If no valid data values were found “No Data” will be returned. The optional parameters are for comparing selected values in the memory buffer. The optional <count> parameter specifies the number of data points to compare. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be searched. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be compared. Note: In order to specify a <step_size>, the user must enter the <count> and the <offset> information.

Query	Response
CALCulate:SDEViation? [<count>[,<offset> [,<step size>]]]	This query calculates and returns the standard deviation for a set of data. The formula used for standard deviation is: σ = sqrt (Σ ($data_i$ – average)2/(n–1) (n = amount of data summed) If no valid data values were found “No Data” will be returned. The optional parameters are for calculating selected values in the memory buffer. The optional <count> parameter specifies the number of datapoints to calculate. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be calculated. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be used. ***NOTE:*** *In order to specify a <step_size>, you must enter the <count> and the <offset> information.*
CALCulate:TRANsform: HISTogram?	This query calculates and returns the histogram for a set of data. The format is as follows: <bin_data{bin_data>}
CALCulate:TRANsform: HISTogram:ABOVe?	Returns the number of points above the maximum value in a histogram calculation.
CALCulate:TRANsform: HISTogram:BELow?	Returns the number of points below the minimum value in a histogram calculation.
CALCulate:TRANsform: HISTogram:COUNt?	Returns number of points in a histogram calculation.
CALCulate:TRANsform: HISTogram:POINts?	Returns number of points in a histogram calculation.
CALCulate:TRANsform: HISTogram:RANGe?	Returns the <minimum_value> and <maximum_value> in a histogram calculation.

<table>
<tr><th>Query</th><th>Response</th></tr>
<tr><td>CALCulate:TRANsform:
HISTogram:AUTO?</td><td><ON|OFF></td></tr>
<tr><td>CALCulate:VARiance?
[<count>[,<offset>
[,step size>]]]</td><td>This query calculates and returns the variance for a set of data. The variance is the standard deviation squared or:

$\sigma = sqrt (\Sigma (data_i - average)2/(n-1)$

(n = amount of data summed)

If no valid data values were found “No Data” will be returned. The optional parameters are for calculating selected values in the memory buffer.

The optional <count> parameter specifies the number of datapoints to calculate. If no <count> is specified, then the number measurements last taken will be the default.

The optional <offset> parameter determines at which point in memory values will start to be calculated.

The optional <step_size> parameter determines the number of measurements to skip before taking a value to be used.

NOTE: In order to specify a <step_size>, you must enter the <count> and the <offset> information.</td></tr>
</table>

Examples

CALCulate:AVERage?[<count>[,<offset>[,<step_size>]]]

Command	Response
inst:sel ctr	Selects the Counter
calc:lim:upper 5.0	Sets upper data value limit
calc:aver?	Averages all the data that has been collected
calc:aver? 20,3,2	Averages 20 measurement readings. Starts with the 3rd measurement. Averages every other data point

CALCulate:LIMit:ENVelope[:DATA]

Command	Response
inst:sel ctr	Selects the Counter
calc:lim:env 5000,1000	Sets an envelope value. All data points above 05.0 kHz and below 1.0 kHz are targeted.
calc:lim:env?	1000, 5000<lf> returns the envelope threshold value
calc:lim:env 1000,5000	Sets an envelope value. All data points between 5.0 kHz and 1.0 kHz are targeted.
calc:lim:env?	5000, 1000<lf> returns the envelope threshold value

CALCulate:LIMit:FCOunt?

Command	Response
inst:sel ctr	Selects the Counter
calc:lim:upp 5000	Sets upper threshold value
calc:lim:fco?	Returns the number of data points outside of a defined threshold.

CALCulate:LIMit:LOWer[:DATA]

Command	Response
inst:sel ctr	Selects the Counter
calc:lim:lower 5.0	Sets lower threshold value
calc:lim:lower?	5.0 Returns the lower threshold value

CALCulate:LIMit:REPort[:DATA]?

Command	Response
inst:sel ctr	Selects the Counter
calc:lim:upp 5000	Sets <threshold2> value
calc:lim:rep?	2, 6002.0; 5,5001.0 Returns the number of data points and data values outside of the defined threshold

CALCulate:LIMit:UPPer[:DATA]

Command	Response
inst:sel ctr	Selects the Counter
calc:lim:upper 5.0	Sets the <threshold2> value
calc:lim:upper?	5.0 Returns the <threshold2> value

CALCulate:MEDian

Command	Response
inst:sel ctr	Selects the Counter
CALC:MEDian?	Finds the median value

CALCulate:MINimum

Command	Response
inst:sel ctr	Selects the Counter
CALC:MINimum?	Finds the minimum value

CALCulate:MAXimum

Command	Response
inst:sel ctr	Selects the Counter
CALC:MAXimum?	Finds the maximum value

CALCulate:SDEViation

Command	Response
inst:sel ctr	Selects the Counter
CALC:SDEViation?	Calculates and returns the standard deviation on a set of data

CALCulate:TRANsform:HISTogram

Command	Response
inst:sel ctr	Selects the Counter
CALC:TRAN:HIST:COUNt 5	Sets the number of data to use in a histogram calculation to 5
CALC:TRAN:HIST:POINts 3	Sets the number of intervals in a histogram calculation to 3
CALC:TRAN:HIST:RANGe 10,00	Sets the range of a histogram calculation
CALC:TRAN:HIST?	Returns the histogram distribution

CALCulate:TRANsform:HISTogram:ABOVe?

Command	Response
inst:sel ctr	Selects the Counter
CALC:TRAN:HIST:RANGE: AUTO OFF	Turns off autoranging
CALC:TRAN:HIST:RANGE: 10,50	Sets the range of the histogram calculation for 10-50
CALC:TRAN:HIST:ABOV?	Returns the number of data points above 50

CALCulate:TRANsform:HISTogram:BELow

Command	Response
inst:sel ctr	Selects the Counter
CALC:TRAN:HIST:BEL?	Returns the number of data points below 10

CALCulate:TRANsform:HISTogram:COUNt

Command	Response
inst:sel ctr	Selects the Counter
CALC:TRAN:HIST:COUNt 5	Sets the number of data to use in a histogram calculation to 5
CALC:TRAN:HIST:COUNt?	Returns the number of data points to use in a histogram calculation

CALCulate:TRANsform:HISTogram:POINts

Command	Response
inst:sel ctr	Selects the Counter
CALC:TRAN:HIST:POINts 3	Sets the number of intervals in a histogram calculation to 3
CALC:TRAN:HIST:POINts?	Returns the number of intervals in a histogram calculation

CALCulate:TRANsform:HISTogram:RANGe

Command	Response
inst:sel ctr	Selects the Counter
CALC:TRAN:HIST:RANGe	Sets the range of a histogram calculation
CALC:TRAN:HIST:RANGe?	Returns the range of a histogram calculation

CALCulate:TRANsform:HISTogram:RANGe:AUTO

Command	Response
inst:sel ctr	Selects the Counter
CALC:TRAN:HIST:RANGe: AUTO ON	Automatically sets the range of a histogram calculation
CALC:TRAN:HIST:RANGe: AUTO?	Tells if the histogram auto range is on or off

CALCulate:VARiance

Command	Response
inst:sel ctr	Selects the Counter
CALC:VARiance?	Calculates and returns the variance on a set of data

CALibrate Subsystem

The CALibrate commands enable you to set the Counter functions to their correct values.

Command Syntax

```
CALibrate:ARM[:VALue] <arm input voltage>

CALibrate[<channel>]

    :DELay <cross channel specification>
    :HYSTeresis
    :LFCOmp
    :LINearity
    :VALue <input voltage>
    :ZERO

CALibrate:DTI

CALibrate3:BIAS

CALibrate[1|2|3]:ROSCillator <input frequency>
```

Query Syntax N/A

***RST Value** N/A

Limits

Parameter	Value
<arm input voltage>	0.0 or 20.0
<cross channel specification>	12 or 21
<channel>	1 or 2
<input voltage>	–50 to +50
<input frequency>	9000000.0000 to 11000000.0000

Related Commands

SOUR:ROSC[?]
SOUR:ROSC:VAL[?]

Command Description

CALibrate:ARM[:VALue] <arm input voltage>

The CALibrate:ARM:VALue command performs a gain and zero calibration on the ARM input. This command requires an <arm_input_voltage> calibration source set to 0.0 ±0.001 V for calibrating the offset, and 20.0 V ±0.1% for calibrating the gain.

CALibrate[<channel>]:DELay <cross channel specification>

Performs calibration of the internal delay cabling for the channel specified. This command requires an input frequency source set to a 10 MHz square wave at ±1.0 V ±0.1 V.

This command performs cross-channel delay calculations between the two input channels. It requires the 10 MHz square wave to be connected to both channel inputs of the instrument using a "T" connector and cables of equal length.

The <channel> argument specifies which of the two standard Counter channels you want to calibrate.

CALibrate[<channel>]:HYSTeresis

The CALibrate[<channel>]:HYSTeresis command calculates the hysteresis linearization factors. This command uses an internal reference, and so does not require any external input.

The <channel> argument specifies the channel being calibrated, and can be specified as either 1 or 2. If <channel> is not specified, channel one is assumed.

CALibrate[<channel>]:LFCOmp

The CALibrate:LFCOmp command performs a low frequency compensation on the inputs. This command requires an input frequency source set to a square wave of ±2.5 V ±0.1 V at 1 kHz.

The <channel> argument specifies the channel being calibrated, and can be specified as either 1 or 2. If [<channel>] is not specified, channel one is assumed.

CALibrate[<channel>]:LINearity

The CALibrate[<channel>]:LINearity command calculates linearity corrections for the preamplifier section of the Counter hardware. This command requires an input calibration source set to 0.5 V ±0.1%.

The <channel> argument specifies the channel being calibrated, and can be specified as either 1 or 2. If [<channel>] is not specified, channel one is assumed.

CALibrate[<channel>]:VALue <input voltage>

The CALibrate[<channel>]:VALue <input voltage> command calculates the gain correction factors for the Counter attenuation and gain settings. This command requires an input calibration source set to the <input voltage> values below (±0.1%).

Table 3–3: Input Calibration Source Settings

<input voltage>
+50.0
+20
+10
+5.0
+2.0
+1.0
+0.5
+0.2
+0.1
+0.05
–50.0
–20
–10
–5.0
–2.0
–1.0
–0.5
–0.2
–0.1
–0.05

[<channel>] specifies the channel being calibrated, and can be specified as either 1 or 2. If [<channel>] is not specified, channel one is assumed.

CALibrate[<channel>]:ZERO

The CALibrate[<channel>]:ZERO command performs offset corrections for each of the gain and attenuation settings of the Counter. This command requires an input calibration source set to 0.0 V ±0.001 V.

[<channel>] specifies the channel being calibrated, and can be specified as either 1 or 2. If [<channel>] is not specified, channel one is assumed.

CALibrate:DTI

The CALibrate:DTI command performs a statistical analysis of the internal hardware for determining the setting of the digital time interpolation bit used by the Counter hardware. This command requires an input frequency source set to a 10 MHz square wave at ±0.5 V ±0.1 V.

CALibrate3:BIAS

The CALibrate3:BIAS performs a sensitivity adjustment for the channel three input of the card.

CALibrate[1|2|3]:ROSCillator <input frequency>

This command is used to calibrate the currently selected reference oscillator using an externally supplied signal with a known frequency and the VX4101A counter. The suffix on the Calibrate keyword specifies to which channel the external signal is connected. The argument specifies the frequency of this signal. After the calibration is complete, the calibrated reference oscillator frequency is stored in nonvolatile memory and will be used for all subsequent measurements.

Examples

CALibrate:ARM[:VALue]

Command	Description
CALIBRATE:ARM:VALUE 0	Performs the ARM input offset correction
cal:arm 20	Performs the ARM input gain correction

CALibrate[<channel>]:DELay

Command	Description
CALIBRATE1:DELAY	Performs the channel 1 cabling delay calibration
cal2:del	Performs the channel 2 cabling delay calibration
CAL:DELAY 12	Performs the channel 1 to 2 cross channel delay calibration
cal:delay 21	Performs the channel 2 to 1 cross channel delay calibration

CALibrate[<channel>]:VALue

Command	Description
CALIBRATE1:VALUE 5	Performs the gain correction for channel 1 for a 5 V input
CAL2:VAL –5.0	Performs the gain correction for channel 2 for a –5.0 V input

CALibrate[<channel>]:HYSTeresis

Command	Response
CALIBRATE1:HYSTERESIS	Performs the hysteresis linearization for channel 1
CAL2:HYST	Performs the hysteresis linearization for channel 2

CALibrate[<channel>]:LFCOmp

Command	Response
CALIBRATE1:LFCOMP	Performs the low frequency compensation for channel 1
CAL2:LFCO	Performs the low frequency compensation for channel 2

CALibrate[<channel>]:LINearity

Command	Response
CALIBRATE1:LINEARITY	Performs the linearization correction for channel 1
CAL2:LIN	Performs the linearization correction for channel 2

CALibrate[<channel>]:ZERO

Command	Response
CALIBRATE1:ZERO	Performs the offset calibration for channel 1
CAL2:ZERO	Performs the offset calibration for channel 2

CALibrate:DTI

Command	Response
CALIBRATE:DTI	Performs the DTI calculations.

CALibrate[3]:BIAS

Command	Response
CALIBRATE3:SENSITIV	N/A, Perform the sensitivity adjustment

CALibrate [3] :ROSCillator <input frequency>

Command	Response/Description
SOUR:ROSC INT; SYST:ERR?	–313,“Calibration memory lost; VX4101; Reference Oscillator Uncalibrated” Display the reference oscillator error
SOUR:ROSC:VAL?	10000000.0000
INST:SEL COUNTER	Selects the Counter
CAL:ROSC 10.001E6	
SOUR:ROSC:VAL?	9999999.0000

CONFigure Subsystem

The CONFigure commands tell the Counter what type of measurement to make and the input channel(s) to use. The measurement will not be made until a INITiate or READ? command is given.

The input coupling and impedance are not changed by these commands. You must select the coupling and impedance that makes sense for the input signal(s) and the desired measurement.

The input attentuation, offset, gain, and comparator hysteresis are not changed by these commands. However, the comparator slopes and thresholds are changed to defaults by these commands. If INPut:SETup:AUTO (autotrigger) is set to ON or ONCE, the input attenuation, offset, gain, and comparator hysteresis may change when the measurement is started. The coupling and impedance are set with the INPut:COUPling and INPut:IMPedance commands.

After a CONFigure command, you can use the INPut and SENSe commands to modify the CONFigure setup before the measurement is started by an INITiate or READ? command. These modifications can be undone at INITiate by some types of measurements.

The Counter may be programmed to make and store up to 1000 measurements. The SCALar commands program the Counter for one measurement. The ARRAy commands program the Counter for a maximum of 1000 measurements as specified by the <array size> parameter.

Command Syntax

```
CONFigure[1|2|3]([:SCALar]|:ARRay)

   :FREQuency [<array size>[,<expected value>[,resolution>]]]

   :FREQuency:RATio [<array size>[,<expected
   value>[,resolution>]]]

   :PERiod [<array size>[,<expected value>[,resolution>]]]

CONFigure[1|2]([:SCALar]|:ARRay)

   :NDUTy cycle|PDUTCycle|DCYCle [<array size>[,<dcycle
   reference>[,<expected value>[,resolution>]]]

   :NWIDth|PWIDth [<array size>[,<pwidth reference>[,<expected
   value>[,resolution>]]]

   :PHASe [<array size>[,<expected value>[,resolution>]]]

   :RTIMe|FTIMe|RISE:TIME|FALL:TIME [<array size>[,<low
   reference>[,<high reference> [,<expected
   value>[,resolution>]]]

   :TINTerval [<array size>[,<expected value>[,resolution>]]]

   [:VOLTage]

      :AC [<array size>[,<expected value>[,resolution>]]]
      :DC [<array size>[,<expected value>[,resolution>]]]
      :MINimum [<array size>[,<expected value>[,resolution>]]]
      :MAXimum [<array size>[,<expected value>[,resolution>]]]
      :PTPeak [<array size>[,<expected value>[,resolution>]]]

CONFigure[1|2|11|12|21|22]([:SCALar]|:ARRay)

   :TINTerval:DELay:TIME|EVENts [<array size>[,<delay
   time>|<delay events>[,<expected value>[,resolution>]]]

CONFigure[1|2|10|20]([:SCALar]|:ARRay)

   :TOTalize
```

NOTE*. If you specify SCALar in the command, <array size> is not a valid parameter. If ARRay is specified, <array size> is a required parameter.*

Query Syntax

```
CONFigure?
```

***RST Value**

Parameter	Value
<array size>	1
<delay time>	1E–6
<delay events>	1000
<dcycle reference>	50
<pwidth reference>	50
<low reference>	10
<high reference>	90

Limits

Parameters	Value
<array size>	1 to 1000
<delay time>	1E–9 to 9E6
<delay events>	1 to 9E15
<dcycle reference>	10 to 90
<pwidth reference>	10 to 90
<low reference>	10 to 90
<high reference>	10 to 90

Related Commands

MEASure, SENSe

Command Description

The following describes specific characteristics and behavior of CONFigure commands applies to all measurement functions.

Characteristics of All CONFigure Commands

If a reference is specified, then the instrument must perform an autotrigger to determine the appropriate trigger levels. Thus, if INPut:SETup:AUTO is set to OFF, then INPut:SETup:AUTO will be set to ONCE. If INPut:SETup:AUTO is set to ONCE or ON, then INPut:SETup:AUTO will be unchanged.

If no reference is specified and the INPut:SETup:AUTO is set to OFF, then the counter front end will be changed to the following:

gain is set to 1

attenuation is set to 1

threshold is set to 0 V for frequency, period, and totalize and to –0.03 V for all other measurement functions (so that the upper threshold of the hysteresis is set to 0 V)

Aperture Characteristics

The aperture is set for all measurement commands except for totalize as follows:

If the optional resolution argument is not specified, then aperture is set to the default value. If the optional expected value and resolution arguments are not specified, then the aperture is set to the default value. If both the expected value and resolution arguments are specified, then the aperture is calculated as follows:

aperture = 10^(–9+log<expected value>–log<resolution>)

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:APERture command.

The aperture for totalize is 9e6 seconds. An abort must be issued to terminate the measurement.

CONFigure[1|2|3]([:SCALar]|:ARRay):FREQuency [<array size>[,expected value>[,<resolution>]]]

This command configures the Counter to measure the frequency of the signal on the CONFigure suffix input channel. The units of <expected value> and <resolution> are both Hz.

This command will modify the following:

function - set to FREQuency
aperture - described below
aperture/events mode - set to APERture
if autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified

If neither of the optional arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:FREQuency:APERture command.

CONFigure[1|2|3]([:SCALar]|:ARRay):FREQuency:RATio [<array size>[,expected value>[,<resolution>]]]

This command configures the Counter to measure the ratio of the frequencies of the signals on the CONFigure suffix input channel and <second channel>.

The <expected value> and <resolution> arguments have no units. The parameters <expected value> and <resolution> are accepted but are not used.

This command will modify the following:

function - set to FREQ:RAT
aperture - set to default
aperture/events mode - set to APERture

if autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level, slope can be modified

The SENSe suffix selects the input channel for the numerator, the <second channel> selects the input channel for the denominator. Ratios of a channel to itself are always one and the Counter will generate an error if programmed to do it. <second channel> can be 1, 2 or 3. When channel 3 is not being used and the Counter mode is EVENts, the <second channel> will be used as the input to the EVENts counter. In the case where input channel 3 is used and the counter mode is EVENts, the other specified input channel will be used as the input to the EVENts counter.

CONFigure[1|2]([:SCALar]|:ARRay):NDUTycycle|PDUTCycle|DCYCLE [,dcycle reference>[,<expected>[<array size>[,expected value>[,<resolution>]]]]

***NOTE**. Duty cycle measurements use all available hardware timers. When the Counter is in this mode, the aperture is time controlled by software and has a minimum period of approximately 10 ms. For event apertures, only a measurement.*

This command configures a positive or negative duty cycle measurement. The reference is the percent of the signal's peak value, e.g. percent of peak to peak as measured from the signal's minimum.

This command will modify the following:

function - set to PDUT or NDUT

aperture - described below

aperture/events mode - set to APERture

The input channel attenuation, offset, gain, level and slope can be modified.

CONFigure[1|2]([:SCALar]|:ARRay):NWIDth|PWIDth [<array size>[,<pwidth reference>[,expected value>[,<resolution>]]]

This command configures the Counter to measure the positive or negative pulse width time of the signal on the CONFigure suffix input channel. The units of the <reference> are percentage. The units of <expected value> and <resolution> are seconds. If the <expected value> argument is used, the expected period of the input signal should be used, not the expected pulse width. This measurement uses both comparators of the input channel. The reference value is used to set the comparator threshold levels to a percentage of the peak-to-peak signal.

If a reference value is specified and INPut:SETup:AUTO is set to OFF, INPut:SETup:AUTO will be set to ONCE. If a reference value is not specified, the input channel comparators 1 and 2 levels will be set to 0 V.

This command will modify the following:

function - set to PWID or NWID
aperture - described below
aperture/events mode - set to APERture

The input channel attenuation, offset, gain, level and slope can be modified

If neither of the optional arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log <expected> - \log <resolution>)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:PWIDth:APERture command.

CONFigure[1|2|3]([:SCALar]|:ARRay):PERiod [<array size>[,expected value>[,<resolution>]]]

This command configures the Counter to measure the period of the signal on the CONFigure suffix input channel. The units of <expected value> and <resolution> are seconds.

This command will modify the following:

function - set to PERiod
aperture - described below
aperture/events mode - set to APERture

if autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level, and slope can be modified

If neither of the optional arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log <expected> - \log <resolution>)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:FREQuency:APERture command.

CONFigure[1|2]([:SCALar]|:ARRay):PHASe [<array size>[,expected value>[,<resolution>]]]

This command configures a phase measurement. The suffix selects the channel for the beginning of the measurement. For example, CONF1:PHASe will set up the instrument to perform a phase measurement from channel one to channel two, whereas CONF2:PHASe will set up the instrument to perform a phase measurement from channel two to channel one.

This command will modify the following:

Function set to PHAS

Aperture/events mode set to aperture

If auto setup is ON or ONCE, the input attenuation, offset, gain, level, and slope can be modified.

***NOTE**. Phase measurements use all available hardware timers. When the Counter is in this mode, the aperture is controlled by software and has a minimum period of approximately 10 ms.*

CONFigure[1|2]([:SCALar]|:ARRay):RTIMe|FTIMe|RISE:TIME|FALL:TIME [<array size>[,<low reference>[,<high reference>[expected value>[,<resolution>]]]]

This command configures the Counter to measure the rise or fall time of the signal on the CONFigure suffix input channel. The units of the <low reference> and <high reference> are a percentage. The units of <expected value> and <resolution> are seconds. If the <expected value> argument is used, the expected period of the input signal should be used, not the expected rise/fall time. This measurement uses both comparators of the input channel.

If a reference value is specified and INPut:SETup:AUTO is set to OFF, INPut:SETup:AUTO will be set to ONCE. If a reference value is not specified, the input channel comparator 1 level will be set to –.25 V and input channel comparator 2 level will be set to +.25 V for a rise time measurement. For fall time, the reverse is set.

This command will modify the following:

function - set to RTIMe or FTIMe
aperture - described below
aperture/events mode - set to APERture

The input channel attenuation, offset, gain, level and slope can be modified

If neither of the optional arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:RTIMe:APERture command.

CONFigure[1|2]([:SCALar]|:ARRay):TINTerval [<array size>[,expected value>[,<resolution>]]]

This command sets the Counter to make a time interval measurement. This measurement is made between input channels 1 and 2. The CONFigure suffix selects the input channel for the beginning of the interval. The end of the interval will be from the remaining channel. The measurement is made from the first detected rising edge on the first channel to the first following rising edge on the second channel. If the <expected value> argument is used, the expected period of the input signal should be used, not the expected time interval.

This command will modify the following:

function - set to TINT
aperture - described below
aperture/events mode - set to APERture

If autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified

If neither of the optional arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:TINTerval:APERture command.

CONFigure[1|2|11|12|21|22]([:SCALar]|:ARRay):TINTerval:DELay:TIMe|EVENts [<array size>[,<delay time>|<delay events>[,expected value>[,<resolution>]]]
This command configures a time interval measurement where the second channel is ignored for a specified period of time (delay by time) or for a specified number of second channel events (delay by events). If neither time or events are specified, the instrument assumes delay by time. Selecting 1 or 12 will configure a time interval measurement with delay from channel one to channel two. Selecting 2 or 21 will configure a time interval measurement with delay from channel two to channel one. Selecting 11 will configure a time interval with delay measurement from channel one to itself. Likewise, selecting 22 will configure a time interval with delay measurement from channel two to itself.

***NOTE**. Time Interval With Delay uses all available hardware timers. When the counter is in this mode, the aperture is controlled by software and has a minimum period of approximately 10 ms.*

This command will modify the following:

function - set to TINT:DEL:EVEN|TM
aperture - described below
aperture/events mode - set to APERture

If autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified

If neither of the optional <expected value> and <resolution> arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:TINTerval:APERture command.

CONFigure[1|2|10|20][:SCALar]:TOTalize
Configures a totalize measurement. The counter totalizes two selected channels simultaneously. Channels 1 and 2 are selected by either 1 or 2. Channel one and the 1 GHz VCO are selected by 10. Channel two and the 1 GHz VCO are selected by 20. If you select either channel 1 or 2, and the VCO measurement provide an elapsed time in nanoseconds after the totalized results for the channel.

CONFigure[1|2]([:SCALar]|:ARRay)[:VOLTage][:...] [<array size>[,expected value>[,<resolution>]]]
These commands configure the Counter to measure the voltage on the CONFigure suffix input channel. The <expected value> and <resolution> parameters are accepted but ignored.

The maximum and minimum voltage of the input signal are determined assuming either a DC signal or a repetitive signal with a frequency of at least 1000 Hz. The user must select the proper INPut:COUPling and INPut:IMPedance separately. The voltage measurements are calculated as follows:

AC = (maximum – minimum) / 2.828
DC = (maximum + minimum) / 2
MAXimum = maximum
MINimum = minimum
PTPeak = maximum – minimum

These commands will modify the following:

input setup - no effect
function - set to AC, DC, MIN, MAX or PTP
autosetup mode - no effect
aperture/events mode - no effect

The actions of these commands are modified by the following:

<expected value> - no effect
<resolution> - no effect
autosetup mode - no effect
aperture/events mode - no effect

Query Response

Query	Response
CONFigure?	Returns the current measurement configuration. This includes any channels, SCALar or ARRay settings, and measurements selected.

Examples

Command/Query	Response/Description
CONF2:ARR:NDUT 1000,25	Configures the instrument to take 1000 negative duty cycle measurements with a trigger point of 25% of the signal's peak on Channel 2.
CONFigure1: ARRay :Dcycle 100,50	Configures the instrument to take 100 positive duty cycle measurements with a trigger point of 50% of the signal's peak on Channel 1.
CONF1:PHAS	Configures the instrument to take a phase measurement from channel 1 to channel 2.
CONF1:TINT:DEL:EVEN 100	Configures a time interval with delay measurement from channel 1 to the 100th edge on channel 2.
CONFigure10:TOTalize	Sets up the instrument to totalize channel 1 and the 1 GHz VCO.
CONFigure?	"10:SCAL:TOT"

FETCh? Subsystem

The FETCh? subsystem is used to retrieve measurement results. If a measurement or array of measurements haven't completed, the VX4101 will delay the read until the measurement(s) have completed unless <count> is less than or equal to the number of measurements completed.

The results of some measurements can be returned as a different measurement,such as frequency and period. If a frequency measurement was made, it can be also be read back as a period by the command FETC:PER? If autotrigger is ON or ONCE, voltages for the primary measurement channel can be read back.

If the measurement was an array measurement specified with commands such as MEAS:ARR:func or CONF:ARR:func or SENS:func:COUN, the optional parameters <count>, <start> and <step> apply.

The default value of <count> is the number of measurements requested for the array. The default value of <start> is 1. The default value of <step> is 1. The number of values returned is <count>. The first of the values returned is value <start>. Then every <step> values after start are returned.

For instance, the command FETC? 4,3,2 will return four values starting with value three and stepping 2 values. The values returned would be 3, 5, 7, and 9.

If the range of requested values is beyond the number of measurements, an error will be generated and no values returned.

The format of the returned values varies with the measurement function and whether it was an array measurement or not.

NOTE. *The Counter will not perform an implied abort for a fetch query that is received during a measurement.*

Command Syntax N/A

Query Syntax

```
FETCh[:...]? [<count>[,<start>[,<step>]]]

    :DCYCle?
    :FALL:TIME?
    :FREQuency?
    :FREQuency:RATio?
    :FTIMe?
    :NDUTycycle?
    :PDUTycycle?
    :NWIDth?
    :PWIDth?
    :PHASe?
    :PERiod?
    :RISE:TIME?
    :RTIMe?
    :TINTerval?
    :TINTerval:DELay?

FETCh:COUNt?

FETCh:TOTalize?

FETCh[:VOLTage][:...]? [<count>[,<start>[,<step>]]]

    :AC?
    :DC?
    :MINimum?
    :MAXimum?
    :PTPeak?
```

***RST Value**

Parameter	Value
<count>	The number of measurements requested for the array.
<start>	1
<step>	1

Limits

Parameter	Value
<count>	1 to 1000
<start>	1 to 1000
<step>	1 to 1000

Related Commands

CONFigure, SENSe

Description

N/A

Query Response

FETCh[:...]? [<count>[,<start>[,<step>]]]
If a single measurement is made, the format of the returned value is:

n.nnnnnnnnnnnnnnnesxx<lf>

If an array measurement is made, the format of the returned values is:

#abbrn.nnnnnnnnnnnnnnnesxx,rn.nnnnnnnnnnnnnnnesxx, ... ,
rn.nnnnnnnnnnnnnnnesxx<lf>

where

a = number of b digits
b = number of characters in the returned data not including #abb
r = space or –
n = value
s = + or –
xx = exponent

The default value of the <count> parameter is the number of measurements requested for the array. The default value of <start> is 1. The default value of <step> is 1. The number of values returned is <count>. The first of the values returned is value <start>. Then every <step> values after start are returned.

FETCh:COUNt?
This command returns the number of measurements completed. This command may be used to monitor the progress of a measurement or an array of measurements.

CAUTION. *Do not continuously issue FETCh?COUNt or STAT:OPER:COND? queries to check if the measurement is complete. Provide a minimum 100 ms delay between each query to permit the multitasking system to process the measurement efficiently or use *SRQ to avoid processing delays caused by polling.*

FETCh:TOTalize?
The totalize measurement returns two integers of up to 15 digits. The first integer is the total counts from input channel one and the second integer is the total counts from input channel two.

FETCh[:VOLTage][:...]? [<count>[,<start>[,<step>]]]
Sets up to return the results of the previous voltage measurement. If a single voltage measurement has been made, any of the other voltage function results may be fetched. If an array of voltage measurements have been made, only the results of the requested voltage measurement function may be fetched.

The default value of <count> is the number of measurements requested for the array. The default value of <start> is 1. The default value of <step> is 1. The number of values returned is <count>. The first of the values returned is value <start>. Then every <step> values after start are returned.

If a single measurement is made, the format of the returned voltage is:

n.nnnn<lf>

If an array measurement is made, the format of the returned values is:

#abbrn.nnnnesxx,rn.nnnnesxx, ... ,rn.nnnnesxx<lf>

where

a = number of b digits
b = number of characters in the returned data not including #abb
r = – or space
n = value
s = + or –
xx = exponent

for example:

meas:arr:DC? 10

#3119 2.5712e+01, 1.5392e+02,–1.0252e+02, 2.5642e+02,–9.7442e+01, 3.0777e+01,–2.2567e+02, 2.5647e+02,–9.7447e+01, 3.0777e+01

Examples

FETCh[:...]?

Command	Response
meas:arr:freq? 5	#3114 1.000028872529214e+07, 1.000028882397919e+07, 1.000028897200977e+07, 1.000028892266624e+07, 1.000028879930743e+07

FETCh:TOTalize?

Command	Response
meas:tot?	"9179, 0"
fetc?	"62710055, 0"
fetc?	"72161385, 0"
fetc?	"81704262, 0"
fetc?	"100501585, 0"
fetc?	"112160067, 0"
abort	

INITiate Subsystem

Command Syntax

`INITiate[:IMMediate]`

`INITiate:CONTinuous [ON]|OFF|0|1`

Query Syntax N/A

***RST Value** N/A

Limits N/A

Related Commands

`READ`
`MEASure`
`ARM`

Command Description

INITiate[:IMMediate]
Initiates its current trigger sequence. After the instrument has completed the current trigger sequence, it enters the idle state.

INITiate:CONTinuous
Measurements are continuously made until a *RST or ABORt command is used while the Counter is continuously making measurements, the FETCh? command may be used to return the results of the most recently completed measurement.

Initiates its current trigger sequence. After the instrument has completed the current trigger sequence, it re-enters the initiated state. It will continue this cycle until an abort, reset, or INIT:CONT OFF is received.

Query Response

INITiate:CONTinuous?
Returns a 0 if the Counter is not taking continuous measurements, a 1 if it is.

INPut Subsystem

This subsystem controls the characteristics of the instrument's input channels.

Command Syntax

```
INPut[1|2]

    :ATTenuation <attenuation>|DEFault|MINimum|MAXimum

    :COMParator[1|2]

        :LEVel

            [:ABSolute] <absolute level>|DEFault|MINimum|MAXimum
            :RELative <relative level>

        :HYSTeresis

            [:ABSolute] <absolute
            hysteresis>|DEFault|MINimum|MAXimum

            :RELative <relative hysteresis>

        :SLOPe <slope>|DEFault

    :COUPling <coupling>

    :FILTer[:LPASs]

        :FREQuency <frequency>|DEFault|MINimum|MAXimum
        [:STATe] <filter state>|DEFault

    :GAIN <gain>|DEFault|MINimum|MAXimum

    :IMPedance<impedance>|DEFault
```

```
  :OFFSet

    [:ABSolute] <absolute offset>|DEFault|MINimum|MAXimum
    :RELative <relative offset>

  :SETup <expected ptp>[,<expected offset>]

INPut [1|2]:SETup:AUTO <auto>
```

Query Syntax

```
INPut[1|2]

  :ATTenuation? [DEFault|MINimum|MAXimum]
  :COMParator[1|2]:LEVel

    [:ABSolute]? [DEFault|MINimum|MAXimum]
    :RELative?

  :HYSTeresis

    [:ABSolute]?[DEFault|MINimum|MAXimum]

    :RELative?

  :SLOPe? [DEFault]

  :COUPling? [DEFault]

  :IMPedance? [DEFault]

  :FILTer[:LPASs]:FREQuency? [MINimum|MAXimum|DEFault]
  [:STATe]?[ON|OFF|DEFault]

  :GAIN? [DEFault|MINimum|MAXimum]

  :OFFSet

    [:ABSolute]? [MINimum|MAXimum|DEFault]
    :RELative?

INPut:SETup:AUTO? [DEFault]
```

***RST Value**

Parameter	Value
<attenuation>	1
<absolute level>	0
<relative level>	N/A
<absolute hysteresis>	0.6
<relative hysteresis>	N/A
<slope>	POSitive
<coupling>	AC

Parameter	Value
<frequency>	20e6
<filter state>	OFF
<gain>	1
<impedance>	1E6
<absolute offset>	0
<relative offset>	N/A
<expected PTP>[, <expected offset>]	N/A
<auto>	OFF

Limits

Command or Query	Value	
<attenuation>	1 to 100	DEFault = 1 MINimum = 1 MAXimum = 100
<absolute level>	–0.5 to .5	
<relative level>	–300 to +300	
<absolute hysteresis>	0.01 to 0.06	DEFault = 0.06 MINimum = 0.01 MAXimum = 0.06
<relative hysteresis>		
<slope>	Positive|Negative	DEFault = positive
<coupling>	AC/DC	DEFault = AC
<frequency>	20E6/100E6	DEFault = 20E6 MINimum = 20E6 MAXimum = 100E6
<filter state>	ON/OFF	DEFault = OFF
<gain>	0.4 to 10.0	DEFault = 1.0 MINimum = 0.4 MAXimum = 10.0
<impedance>	50/1E6	DEFault = 1E6
<absolute offset>	–1.0 to +1.0	DEFault = 0 MINimum = –1.0 MAXimum = 1.0
<relative offset>	–100 to +100	
<expected PTP>[, <expected offset>]	N/A	
<auto>	On/Off/Once	

Related Commands N/A

Command Description The following summarizes the INPut commands:

INPut[1|2]:ATTenuation <attenuation>|DEFault|MINimum|MAXimum
This command sets the input block signal attenuator for the specified channel.

If <attenuation> is less than 2, the attenuator is set to 1. If <attenuation> is between 2 and 20, the attenuator is set to 10. If <attenuation> is greater than 20, the attenuator is set to 100.

INPut[1|2]:COMParator[1|2]:LEVel[:ABSolute] <absolute level>|DEFault|MINimum|MAXimum
This command sets the threshold level for the input channel and comparator selected. The units are volts.

DEFault = 0
MINimum = –0.5
MAXimum = 0.5

To calculate what a particular input voltage will be at the comparator, the following equation may be used:

VL = S*G*(VO + VI/A)

where

VL = voltage at the comparator
S = 1 for positive slope, –1 for negative slope
G = gain setting
VO = offset voltage
VI = input voltage
A = attenuator setting

INPut[1|2]:COMParator[1|2]:LEVel:RELative <relative level>
This command sets the comparator threshold level voltage of the channel and comparator selected. This value will be used at INITiate time to set the comparator level to a voltage that is the same as if the voltage was fed to the input. This compensates for attenuation, offset and gain settings. The unit for <relative level> is V. The comparator level hardware will be set when the measurement is INITiated. The following formula will be used:

VL = S*G*(VLR/A + VO)

where

VL = comparator level voltage
S = slope, 1 for positive, –1 for negative
G = current gain setting
VLR = relative comparator level previously set
A = current attenuator setting
VO = current offset setting (converted from a relative offset if necessary)

INPut[1|2]:COMParator[1|2]:HYSTeresis[:ABSolute] <absolute hysteresis>|DEFault|MINimum|MAXimum

This command sets the hysteresis of the channel and comparator selected. The units are in volts. If an <absolute hysteresis> is specified, it is rounded to the nearest of the values 0.01, 0.02, 0.03, 0.04, 0.05 and 06.

DEFault = .06
MINimum = .01
MAXimum = .06

INPut[1|2]:COMParator[1|2]:HYSTeresis:RELative <relative hysteresis>

This command sets the hysteresis voltage for the selected input channel and comparator. This value will be used at INITiate time to set the comparator hsyteresis to a voltage that is the same as if the hysteresis voltage was implemented at the input. This compensates for attenuation and gain settings. The unit for <relative hysteresis> is V. The comparator hysteresis hardware will be set when the measurment is INITiated. The following formula will be used:

VH = G(VHR/A)

where

VH = comparator hysteresis set at INITiate time
G = current gain setting
VHR = relative comparator hsyteresis set here
A = current attenuator setting

INPut[1|2]:COMParator[1|2]:SLOPe <slope>|DEFault

This command sets the slope for the selected input channel and comparator. The slope may be either positive or negative.

INPut[1|2]:COUPling <coupling>

This command sets the input block signal coupling for the specified channel to AC or DC.

DEFault = AC

INPut[1|2]:IMPedance <impedance>|DEFault

This command sets the input terminating impedance for the specified channel. If <impedance> is less than 60, the impedance is set to 50 Ω. If <impedance> is greater than 60, the impedance is set to 1E6 Ω (1 MΩ).

DEFault = 1E6

INPut[1|2]:FILTer[:LPASs]:FREQuency <frequency>|MINimum|MAXimum|DEFault

This command sets the input block signal lowpass filter for the selected channel. If <frequency> is less than 30E6, the filter is set to 20E6 Hz. If <frequency> is greater than 30E6, the filter is set to 100E6 Hz.

***NOTE**. This setting has no effect on the signal unless the INPut<channel>:FILTer[:LPASs][:STATe] ON command is given. The units for <frequency> are in Hz*

DEFault = 20E6
MINimum = 20E6
MAXimum = 100E6

INPut[1|2]:FILTer[:LPASs][:STATe] <filter state>|DEFault

This command sets the input block signal lowpass filter state for the selected channel to ON or OFF.

DEFault = OFF

INPut[1|2]:GAIN <gain>|DEFault|MINimum|MAXimum

This command sets the input block signal gain for the specified channel.

DEFault = 1.0
MINimum = 0.4
MAXimum = 10.0

***NOTE**. When using the X10 gain, the 20 MHz filter should be turned on. This combination provides the best sensitivity for measurements below 20 MHz. Also, when using the X5 gain, the 100 MHz filter should be turned on. This combination provides the best sensitivity for measurements up to 100 MHz. The X1 and X2.5 gains can be used up the full 500 MHz bandwidth of the product.*

Table 3–4 shows the settings for optimum sensitivity.

Table 3–4: Optimum Sensitivity Settings

Measurement	Gain	Filter
<20 MHz	X10	20 MHz
<100 MHz	X5	100 MHz
<500 MHz	X2.5	None

INPut[1|2]:OFFSet[:ABSolute] <absolute offset>|MINimum|MAXimum|DEFault
This command sets the offset voltage for the channel specified by the INPut suffix. The units are volts. The offset voltage is subtracted from the signal after the input attenuator and before the input gain.

DEFault = 0
MINimum = –1.0
MAXimum = 1.0

This command will abort any command in progress.

INPut[1|2]:OFFSet:RELative <relative offset>
This command sets the offset voltage for the channel specified by the INPut suffix. This value will be used at INITiate to set the offset to a voltage that is relative to the input.

INPut[1|2]:SETup <expected PTP>[,<expected offset>]
This command allows you to set up input channel one or 2 by specifying an expected peak-to-peak input voltage and, optionally, an expected input offset voltage. The Counter will set the input channel attenuation, offset and gain to settings that would center the expected signal in 80% of the comparator range. Both comparators' slope is set to POSitive, level to 0 V and hysteresis to MAXimum.

INPut:SETup: <auto>
This command controls the autosetup of input channels 1 and 2. If it is set ON, each measurement will be proceeded by an automatic setup of the input channel(s) involved in the measurement. If it is set OFF, the current settings are used for the measurement. For ARRay measurements, the ONCE setting will cause an autosetup to occur only for the first measurement. For SCALar measurements, ONCE will cause an autosetup with each measurement.

An autosetup adjusts the input channel's attenuation, offset and gain so that the input signal's peak-to-peak voltage into the comparator is centered on about 80% of the range of the comparator.

INPut commands which manually set the input channel hardware other than COUPling, IMPedance and FILTering will remain in effect when a measurement is initiated only if autosetup is set to OFF.

The time a measurement is taken is influenced by the ARM subsystem.

Query Response

INPut[1|2]:ATTenuation?
Without one of the optional parameters, this command moves to the output-buffer the current setting of the input block signal attenuator for the specified channel. If one of the optional parameters is used, the default, minimum or maximum value for attenuation is moved to the output-buffer. The possible values returned are 1, 10 or 100.

INPut[1|2]:COMParator[1|2]:LEVel[:ABSolute]?
Without one of the optional arguments, this command moves the current threshold level setting to the output buffer. The setting is of the currently selected channel and comparator. If one of the optional arguments is included, the MINimum, MAXimum or DEFault value for level is moved to the output buffer instead. The value returned is calculated as follows:

VL = S*G*(VLR/A + VO)

INPut[1|2]:COMParator[1|2]:LEVel:RELative?
The threshold level corrected for attenuation, slope, gain and offset of the channel and comparator selected is moved to the output buffer. If the level was not previously set by the relative command, the value returned is calculated by the formula:

VLR = A((S*VL)/G – VO)

where

VLR = comparator trigger level relative to the input
A = current attenuator setting
S = slope, 1 for positive, –1 for negative
VL = comparator level previously set
G = current gain setting
VO = current offset setting (converted from a relative offset if necessary)

INPut[1|2]:COMParator[1|2]:HYSTeresis[:ABSolute]?
Without one of the optional arguments, this command moves to the output-buffer the current setting in volts of the hysteresis of the currently selected channel and comparator. If one of the optional arguments is included, the MINimum, MAXimum or DEFault value for hysteresis is moved to the output-buffer instead.

If the hysteresis was previously set with the relative hysteresis command, the query will calculate the hysteresis voltage to put in the output buffer by the formula:

VH = G(VHR/A)

where

VH = comparator hysteresis that will be output for query
G = current gain setting
VHR = relative comparator hysteresis voltage previously set
A = current attenuator setting

INPut[1|2]:COMParator[1|2]:HYSTeresis:RELative?
This query places in the output buffer the selected input channel and comparator hysteresis setting in volts. If the hysteresis was not set by the relative command, the value returned is calculated by the formula:

VHR = A(VH/G)

where

VHR = comparator hysteresis relative to the input
A = current attenuator setting
VH = comparator hysteresis previously set
G = current gain setting

INPut1|2]:COMParator[1|2]:SLOPe?
Without one of the optional parameters, this command moves to the output buffer the current setting of the slope of the selected input channel and comparator. If the optional parameter is used the default value for slope is moved to the output buffer.

INPut[1|2]:COUPling?
Without the optional parameter, this command moves the current setting of the input block signal *coupling* for the specified *channel* to the output buffer. If the optional argument DEFault is included, the default coupling is moved to the output-buffer. The possible values returned are “AC” and “DC”.

INPut[1|2]:IMPedance?
Without the optional parameter, this command moves to the output buffer the current setting of the input terminating impedance for the specified channel. If the optional argument DEFault is included, the default impedance is moved to the output-buffer. The possible values returned are 50 Ω or 1E6 Ω.

INPut[1|2]:FILTer[:LPASs]:FREQuency?
Without one of the optional arguments, this command moves to the output buffer the current setting in Hz of the input lowpass filter. If one of the optional arguments is included, the MINimum, MAXimum or DEFault value for filter is moved to the output-buffer instead.

INPut[1|2]:FILTer[:LPASs][:STATe]?
Without the optional parameter, this command moves to the output buffer the current setting of the input lowpass filter state for the specified channel. If the optional argument DEFault is included, the default filter state is moved to the output_buffer. The possible values returned are 1 for ON and 0 for OFF.

INPut[1|2]:GAIN?
Without one of the optional parameters, this command moves to the output buffer the current setting of the input gain for the specified channel. If the one optional parameters is used, the DEFault, MINimum or MAXimum value for gain is moved to the output buffer.

INPut[1|2]:OFFSet[:ABSolute]?
Without one of the optional arguments, this command moves to the output buffer the current setting in volts of the offset. If one of the optional arguments is used, the MINimum, MAXimum or DEFault value for offset is moved to the output buffer instead.

If the offset was set with the INPut:OFFSet:RELative command, the value returned will be the voltage the offset will be set to at INITiate time.

INPut[1|2]:OFFSet:RELative?
This query moves the offset voltage of the input channel to the output buffer. In the process, the offset voltage is corrected for the attenuation.

INPut:SETup:AUTO?
This command moves the current setting of autosetup to the output buffer. The returned value will be 0 for OFF, 1 for ON and ONCE for ONCE.

Examples

INPut[1|2]:OFFSet[:ABSolute]

The channel one input signal is a 0.5 V_{p-p} sine wave with a +.25 VDC component and the user wishes to remove the DC component with the INPut:OFFset command.

Command	Response
INSTrument:SELect Counter	Selects the Counter
INPut:SETup:AUTo off	
SENSe:FUNCtion “FREQuency”	
INPut:COUPling DC	
INPut:OFFSet .25	

INPut[1|2]:COMParator[1|2]:LEvel[:ABSolute]

In this example, the channel one input signal varies between +0.5 and –0.5 V and you will cause the comparator to trigger at 0.25 V.

Command	Response
INSTrument:SELect Counter	
INPut:SETup:AUTo off	
SENSe:FUNCtion “FREQuency”	
INPut:COUPling DC	
INPut:COMParator:LEVel .25	

INPut[1|2]:OFFSet:RELative

The channel one input signal is a 0.5 V_{p-p} sine wave with a +0.25 VDC component. The user wishes to remove the DC component with the INPut:OFFset:RELative command.

Command	Response
INSTrument:SELect Counter	
INPut:SETup:AUTo off	
SENSe:FUNCtion “FREQuency”	
INPut:COUPling DC	
INPut:OFFSet:RELative .25	

INSTrument Subsystem

The Instrument subsystem controls the instrument's status. You can use it to end a measurement or to make the instrument available for a measurement.

Command Syntax

```
INSTrument
    :ABORt
    :RESet
```

Query Syntax

N/A

Command Class

Instrument

***RST Value**

N/A

Limits

N/A

Related Commands

ABORt
INITiate
INITiate:CONTinuous

Command Description

The following summarizes the INSTrument subsystem functionality:

INSTrument:ABORt
Places active instrument in the IDLE state, aborting any measurement or other instrument activity in progress. The instrument configuration is unchanged and a subsequent INIT command will cause the instrument to re-start the same type of measurement. If the instrument is in Asynchronous mode, this command can be sent while a query is in progress and the measurement will be aborted.

If the instrument is in Synchronous mode, this command will be queued while a query is in progress. This is a ramification of the IEEE 488.2 Message Exchange Protocol Enforcer (MEPE).

INSTrument:RESet
This command resets the currently selected instrument without affecting other instruments. The instrument returns to its *RST state. The instrument remains selected.

Query Response

N/A

Examples

INSTrument:ABORt

Command	Response/Description
INST:SEL Counter	Selects the Counter
CONF:ARR:FREQ 512	
INIT	Begins the measurement
FETC:COUN?	127
INST:ABOR	Ends the measurement
FETC:COUN?	153 Shows the number of measurements made prior to receiving the ABORt command

NOTE. *After abort, no more measurements are taken.*

INSTrument:RESet

Command	Response
INST:SEL Counter	Selects the Counter
CONF:DC	Configures for DC voltage measurements
CONF?	“CONF1:SCAL:VOLT DC”
INST:RES	Resets the instrument to default values
INST:SEL?	Counter
CONF?	“CONF1:SCAL:FREQ” Instrument displays the default configuration

MEASure? Subsystem

The MEASure queries tell the Counter what type of measurement to make and the input channel(s) to use. It also INITiates a measurement and moves to the results of the completed measurement to the output buffer.

The input filtering, coupling, and impedance are not changed by these commands. You must select the coupling and impedance that make sense for the input signal(s) and the desired measurement. The filtering, coupling and impedance are set with the INPut:FILter, INPut:COUPling and INPut:IMPedance commands. The input attenuation, offset, gain, and comparator hysteresis are not changed by these commands. The comparator slopes and thresholds are changed to defaults by these commands. If INPut:SETup:AUTO (autotrigger) is set to ON or ONCE, the input attenuation, offset, gain, and comparator hysteresis can change when the measurement is started.

The following applies to all measurement functions:

If a reference is specified, then the instrument must perform an autotrigger to determine the appropriate trigger levels. Thus, if INPut:SETup:AUTO is set to OFF, then INPut:SETup:AUTO will be set to ONCE. If INPut:SETup:AUTO is set to ONCE or ON, then INPut:SETup:AUTO will be unchanged.

If no reference is specified and the INPut:SETup:AUTO is set to OFF, then the counter front end will be changed to the following:

gain is set to 1

attenuation is set to 1

threshold is set to 0 V for frequency, period, and totalize and to –0.03 V for all other measurement functions (so that the upper threshold of the hysteresis is set to 0 V)

The aperture is set for all measurement commands except for totalize as follows:

- If the optional resolution argument is not specified, then aperture is set to the default value.
- If the optional expected value and resolution arguments are not specified, then the aperture is set to the default value.
- If both the expected value and resolution arguments are specified, then the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1E–8 and a maximum of 5 seconds. You can use the SENSe:APERture command to set a longer aperture.

The aperture for totalize is 9E6 seconds. An ABORt command must be issued to terminate the measurement.

***NOTE**. The Counter will perform an implied abort for a measurement query that is received during a measurement. The measurement in progress will be terminated and the measure command will be executed.*

You can program the Counter to make and store up to 1000 measurements. The SCALar commands program the Counter for one measurement. The ARRAy commands program the Counter for 1 to 1000 measurements as specified by the <array size> parameter.

Command Syntax N/A

Query Syntax

```
MEASure[1|2|3]([:SCALar]|:ARRay)

   :FREQuency? [<array size>[,<expected value>[,resolution>]]]

   :FREQuency:RATio? [<array size>[,<second channel>[,<expected
   value>[,resolution>]]]]

   :PERiod? [<array size>[,<expected value>[,resolution>]]]

MEASure[1|2]([:SCALar]|:ARRay)

   :NDUTycycle|PDUTycycle|DCYCle? [<array
   size>[,<reference>[,<expected value>[,resolution>]]]]

   :NWIDth|PWIDth? [<array size>[,<reference>[,<expected
   value>[,resolution>]]]]

   :PHASe? [<array size>[,<expected value>[,resolution>]]]

   :RTIMe|FTIMe|RISE:TIME|FALL:TIME? [<array size>[,<low
   reference>[,<high reference> [,<expected
   value>[,resolution>]]]]]

   :TINTerval? [<array size>[,<expected value>[,resolution>]]]

   [:VOLTage]

      :AC? [<array size>[,<expected value>[,resolution>]]]

      :DC? [<array size>[,<expected value>[,resolution>]]]

      :MINimum? [<array size>[,<expected value>[,resolution>]]]

      :MAXimum? [<array size>[,<expected value>[,resolution>]]]
```

```
    :PTPeak? [<array size>[,<expected value>[,resolution>]]]
MEASure[1|2|11|12|21|22]([:SCALar]|:ARRay)
   :TINTerval:DELay:TIME|EVENts? [<array size>[,<delay
   time>|<delay events>[,<expected value>[,resolution>]]]]
MEASure[1|2|10|20]([:SCALar]|:ARRay)
   :TOTalize?
```

NOTE*. If SCALar is specified, <array size> is not a valid parameter. If ARRay is specified, <array size> is a required parameter.*

NOTE*. If SCALar is specified, <array size> is not a valid parameter. If ARRay is specified, <array size> is a required parameter.*

*RST Value

Parameter	Value
<array size>	1
<low reference>	10
<high reference>	90
<second channel>	N/A
<reference>	50
<delay time>	1e –6
<delay events>	1000

Limits

Parameter	Value
<array size>	1 to 100
<reference>	10 to 90
<low reference>	10 to 90
<high reference>	10 to 90
<second channel>	1|2|3
<delay time>	1E–9 to 9E–6
<delay events>	1 to 9E15
<reference>	1 to 90

Parameter	Value
<expected value>	none or ignored
<resolution>	none or ignored

Related Commands

CONFigure, SENSe

Command Description

N/A

Query Response

MEASure[1|2][:SCALar|:ARRay]:DCYCle|NDUTycycle|PDUTycycle? [<array size>[,<reference>[,expected value>[,<resolution>]]]
This query performs a positive or negative duty cycle measurement. The reference is the percent of the signal's peak value, e.g. percent of peak to peak as measured from the signal's minimum. Note that DCYCle is the same as PDUTycycle The <expected value> and <resolution> are accepted but not used.

This command will modify the following:

function - set to PDUT or NDUT

aperture - described below

aperture/events mode - set to APERture

The input channel attenuation, offset, gain, level and slope can be modified.

MEASure[1|2|3]([:SCALar]|:ARRay):FREQuency? [<array size>[,expected value>[,<resolution>]]]
Configures the Counter to measure the frequency of the signal on the MEASure suffix input channel. The measurement is initiated and the result placed in the output buffer. The units of <expected value> and <resolution> are both Hertz.

This command will modify the following:

function - set to FREQuency
aperture - described below
aperture/events mode - set to APERture

If autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified.

If neither of the optional arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log <expected> - \log <resolution>)}$$

with a minimum of 1e–8 and a maximum of five seconds. A longer aperture may be set with the SENSe:FREQuency:APERture command.

MEASure[1|2|3]([:SCALar]|:ARRay):FREQuency:RATio? [<array size>[,<second channel>[,<expected value>[<resolution>]]]]

This query configures the Counter to measure the ratio of the frequencies of the signals on the MEASure suffix input channel and <second channel>. The measurement is initiated and the result placed in the output buffer. Ratio, <expected value>, and <resolution> have no units. The parameters <expected value> and <resolution> are accepted but are not used.

This command will modify the following:

function - set to FREQ:RAT
aperture - set to default
aperture/events mode - set to APERture

If autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified.

The SENSe suffix selects the input channel for the numerator. The <second channel> selects the input channel for the denominator. Ratios of a channel to itself are always one and the Counter will generate an error if programmed to do it. The <second channel> can be 1, 2 or 3. When channel 3 is not being used and the Counter mode is EVENts, the SENSe suffix channel will be used as the input to the EVENts Counter. In the case where input channel 3 is used and the Counter mode is EVENts, the other specified input channel will be used as the input to the EVENts Counter.

MEASure[1|2|3]([:SCALar]|:ARRay):PERiod? [<array size[,expected value>[,resolution>]]]

Configures the Counter to measure the period of the signal on the MEASure suffix input channel. The measurement is initiated and the result placed in the output buffer. The units of <expected value> and <resolution> are seconds.

This command will modify the following:

function - set to PERiod
aperture - described below
aperture/events mode - set to APERture

If autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified

If the optional arguments <expected value> and <resolution> are not used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:FREQuency:APERture command.

MEASure[1|2]([:SCALar]|:ARRay):NWIDth|PWIDth? [<array size>[,<reference>[,<expected value>[<resolution>]]]

This query configures the Counter to measure the positive or negative pulse width time of the signal on the MEASure suffix input channel. The measurement is initiated and the result placed in the output buffer. The units of the <reference> are percentage. The units of <expected value> and <resolution> are seconds. If the <expected value> argument is used, the expected period of the input signal should be used, not the expected pulse width. This measurement uses both comparators of the input channel. The <reference value> is used to set the comparator threshold levels to a percentage of the peak-to-peak signal.

If a reference value is specified and INPut:SETup:AUTO is set to OFF, INPut:SETup:AUTO will be set to ONCE. If a reference value is not specified, the input channel comparators 1 and 2 levels will be set to 0 V.

This command will modify the following:

function - set to PWID or NWID
aperture - described below
aperture/events mode - set to APERture

The input channel attenuation, offset, gain, level and slope can be modified
If the optional arguments <expected value> and <resolution> are not used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of five seconds. A longer aperture may be set with the SENSe:PWIDth:APERture command.

MEASure[1|2]([:SCALar]|:ARRay):PHASe? [<array size>[,<expected value>[,resolution>]]]

This query performs a phase measurement. The suffix selects the channel for the beginning of the measurement. For example, MEAS1:PHASe will perform a phase measurement from channel one to channel two, whereas MEAS2:PHASe will perform a phase measurement from channel two to channel one.

This command will modify the following:

Function set to PHAS

Aperture?events mode set to APERture

The input channel attenuation, offset, gain, level, and sloped can be modified.

***NOTE**. Time Interval With Delay, Phase, and Duty Cycle use all available hardware timers. When the counter is in one of these modes, the aperture is controlled by software and has a minimum period of approximately 10 ms.*

MEASure[1|2]([:SCALar]|:ARRay):RTIMe|FTIMe? [<array size>[,<low reference>[,<high reference>[,<expected value>[,<resolution>]]]]]

This query configures the Counter to measure the rise or fall time of the signal on the MEASure suffix input channel. The measurement is initiated and the result placed in the output buffer. The units of the <low reference> and <high reference> are percentage. The units of <expected value> and <resolution> are seconds. If the <expected value> argument is used, the expected period of the input signal should be used, instead of the expected rise/fall time. This measurement uses both comparators of the input channel.

If a reference value is specified and INPut:SETup:AUTO is set to OFF, INPut:SETup:AUTO will be set to ONCE. If a reference value is not specified, the input channel comparator 1 level will be set to –.25 V and input channel comparator 2 level will be set to +.25 V for a rise time measurement. For fall time, the reverse is set.

This command will modify the following:

function - set to RTIMe or FTIMe
aperture - described below
aperture/events mode - set to APERture

if autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified

MEASure[1|2]([:SCALar]|:ARRay):TINTerval?
Sets the Counter to make a time interval measurement and fetch the result. This measurement is made between input channels 1 and 2. The MEASure suffix selects the input channel for the beginning of the interval, the end of the interval will be from the remaining channel. The measurement is made from the first detected rising edge on the first channel to the first following rising edge on the second channel. If the <expected value> argument is used, the expected period of the input signal should be used, not the expected time interval.

This command will modify the following:

function - set to TINT
aperture - described below
aperture/events mode - set to APERture

If autosetup mode is ON or ONCE, the input channel attenuation, offset, gain, level and slope can be modified

If neither of the optional arguments are used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:TINTerval:APERture command.

MEASure[1|2|11|12|21|22]([:SCALar]|:ARRay):TINTerval:DELay[:TIME|EVENts]? [<array size>[,<delay time>|<delay events>[,<expected value>[,<resolution>]]]]
This query performs a time interval measurement where the second channel is ignored for a specified period of time (delay by time) or for a specified number of second channel events (delay by events). If neither time or events are specified, the delay remains unchanged. Selecting 1 or 12 will perform a time interval measurement with delay from channel one to channel two. Selecting 2 or 21 will perform a time interval measurement with delay from channel two to channel one. Selecting 11 will perform a time interval with delay measurement from channel one to itself. Likewise, selecting 22 will perform a time interval with delay measurement from channel two to itself.

***NOTE**. Time Interval With Delay, Phase, and Duty Cycle use all available hardware timers. When the counter is in one of these modes, the aperture is controlled by software and has a minimum period of approximately 10 ms.*

MEASure[1|2|10|20][:SCALar]:TOTalize?

This query performs a totalize measurement. The counter totalizes two selected channels simultaneously. Channels 1 and 2 are selected by either 1 or 2. Channel one and the 1 GHz VCO are selected by 10. Channel two and the 1 GHz VCO are selected by 20. Selecting either channel one or 2 and the VCO provides an elapsed time in nanoseconds after the totalized results for the selected channel.

This measurement is unique. It allows you to fetch the totals while it is still counting. Each fetch will return updated count values. This command has a default aperture of 99 days. Use the ABORt command to end this measurement. Since this measurement allows reading the Counter hardware while it is counting, there is some risk that the value returned will be in error.

If the optional <expected value> and <resolution> arguments are not used:

aperture is set to the default value

If only the <expected value> argument is used:

aperture is set to the default value

If both <expected value> and <resolution> are used the aperture is calculated as follows:

$$aperture = 10^{(-9 + \log < expected > - \log < resolution >)}$$

with a minimum of 1e–8 and a maximum of 5 seconds. A longer aperture may be set with the SENSe:RTIMe:APERture command.

MEASure[1|2]([:SCALar]|:ARRay)[:VOLTage][:...]? [<array size>[,<expected value>[,<resolution>]]]

These queries configure the Counter to measure the voltage on the MEASure suffix input channel. The measurement is initiated and the result placed in the output buffer. The <expected value> and <resolution> parameters are accepted but ignored.

The maximum and minimum voltage of the input signal are determined assuming either a DC signal or a repetitive signal with a frequency of at least 1000 Hz. The user must select the proper INPut:COUPling and INPut:IMPedance separately. The voltage measurements are calculated as follows:

AC = (maximum – minimum) / 2.828
DC = (maximum + minimum) / 2

MAXimum = maximum
MINimum = minimum
PTPeak = maximum – minimum

These commands can or will modify the following:

function - set to AC, DC, MIN, MAX or PTP

autosetup mode - no effect
aperture/events mode - no effect

Examples

Command/Query	Response/Description
MEAS2:ARR:NDUT? 1000,25	Takes 1000 negative duty cycle measurements with a trigger point of 25% of the signal's peak.
MEASure2:ARRay: PDUTycycle? 100,50	Take 100 positive duty cycle measurements with a trigger point of 50% of the signal's peak.
MEASure1:PHASe?	Take a phase measurement from channel 1 to channel 2.
MEAS1:TINT:DEL:EVEN? 100	Take a time interval with delay measurement from channel 1 to the 100th edge on channel 2.
MEASure10:TOTalize?	Totalize channel 1 and the 1 GHz VCO.

OUTPut Subsystem

The OUTPut subsystem specifies the output trigger source and division ratio.

Command Syntax `OUTPut:TRIGger:SOURce <channel>,<count>`

Query Syntax `OUTPut:TRIGger:SOURce?`

***RST Value**

Parameter	Value
<channel>	0
<count>	1.0

Limits

Parameter	Value
<channel>	0-2 (note 0 = off)
<count>	1-9E15

Related Commands N/A

Command Description

OUTPut:TRIGger:SOURce <channel>,<count>

This command allows a Counter input to be used as a trigger source. Either channel 1 or 2 can be selected. Selecting channel 0 turns this feature off. The divisor is the number by which the input signal is divided (e.g. a divisor of 10 on a 10 MHz signal will result in a 1 MHz trigger output). You can set up the front end with SENSe commands to provide attenuation, gain, offset, and filtering as you would for any other measurement. Likewise, the trigger threshold and hysteresis may also be selected.

SENSe subsystem commands can be used to configure the front-end to condition the signal appropriately or if INP:SET:AUTO is ON or ONCE an auto-level operation will occur.

The Counter will perform an implied abort for an output command that is received during a measurement. The measurement in progress will be terminated and the output command will be executed.

The Counter will not perform an implied abort for an output query that is received during a measurement.

***NOTE**. When the Counter is in this mode, it cannot be used to perform frequency and period measurements. Sending any command which would alter the current configuration of the counter will result in an implied abort of this operation. This operation can also be stopped by sending an ABORt or by resending this command with the channel set to zero (0).*

Query Response

OUTPut:TRIGger:SOURce?

0	disabled
1	channel 1
2	channel 2

Examples

OUTPut:TRIGger:SOURce <channel>,<count>

Command	Response
INST:SEL COUNTER	
OUTPut:TRIG:SOUR?	0,1.00000000000000E+00
OUTPut:TRIG:SOUR 1,5	
OUTPut:TRIG:SOUR?	1,5.00000000000000E+00

READ? Subsystem

Command Syntax N/A

Query Syntax `READ?`

Query Response The read query causes an INITiate:IMMediate action and a FETCh? query. See the INITiate and FETCh command descriptions.

***RST Value** N/A

Limits N/A

Related Commands INITiate, FETCh?

Description N/A

SENSe Subsystem

The SENSe commands enable you to select the input channel, type of measurement to be made and the manner in which it is made. It does not cause a measurement to be made.

Only the SENSe:FUNCtion command has a SENSe suffix. This suffix will select an input channel to be used for the FUNCtion. If a SENSe suffix is used on the other SENSe commands, a "No suffix allowed error" will be set.

Implied Abort
The counter will perform an implied abort for a sense command that is received during a measurement. The measurement in progress will be terminated and the sense command will be executed.

The counter will not perform an implied abort for a sense query that is received during a measurement.

Command Syntax

```
SENSE:APERture <time>|DEFault|MINimum|MAXimum

SENSe:COUNt <array size>|DEFault|MINimum|MAXimum

SENSe:MODe <mode>
```

```
SENSe:EVENts <# of events>

SENSe[1|2|3]:FUNCtion

   "TOTalize"
   "FREQuency"
   "FREQuency:RATio <second channel>"
   "PERiod"

SENSe[1,2]:FUNCtion

   "DCYCle"
   "FALL:TIME"
   "FTIMe"
   "NDUTycycle"
   "NWIDth"
   "PDUTycycle"
   "PHASe"
   "PWIDth"
   "RISE:TIME"
   "RTIMe"
   "TINTerval"
   "VOLTage:AC"
   "VOLTage:DC"
   "VOLTage:MINimum"
   "VOLTage:MAXimum"
   "VOLTage:PTPeak"

SENSe[1|2|11|12|21|22]:FUNCtion

   "TINTerval:DELay:TIME"
   "TINTerval:DELay:EVENts"

SENSe[1|2|10|20]:FUNCtion "TOTalize"

SENSe:TINTerval:DELay:EVENts <event
delay>|MINimum|MAXimum|DEFault

SENSe:TINTerval:DELay:TIMe <time delay>|DEFault|MINimum|MAXimum
```

Query Syntax

```
SENSe:APERture?[DEFault|MINimum|MAXimum]

SENSe:COUNt?[DEFault|MINimum|MAXimum]

SENSe:EVENts? [DEFault|MINimum|MAXimum]

SENSe:MODe? [DEFault]

SENSe:FUNCtion?

SENSe:TINTerval:DELay:EVENts? [DEFault]|MINimum]|MAXimum]
```

***RST Value**

Parameter	Value
<array size>	1
<time>	1e–01
<#of events>	1E to 3
<mode>	APERture
SENSe[1|2|3]:FUNCtion	"FREQ"
SENSe[1|2|10|11|12|20|21|22] :FUNCtion	"FREQ"
<time delay>	le–6
<event delay>	1

Limits

Parameter	Value
<array size>	1 to 1000
<time>	1e–8 to 9e6
<# of events>	1 to 9E+15
<mode>	APERTURE|EVENTS
<time delay>	1e–9 to 9e+6
<event delay>	1 to 9e15

Command Description

SENSe:APERture <time>
This command sets the Counter measurement aperture <time>. Whether the aperture time is used or not depends on the mode set with one of the SENSe:MODE command. The default mode is to use an aperture. The units are in seconds and range from 1E–8 to 9E6 in 1E–9 size steps.

CONFigure and MEASure commands also set the aperture.

***NOTE**. Time Interval With Delay, Phase, and Duty Cycle use all available hardware timers. When the counter is in one of these modes, the aperture is controlled by software and has a minimum period of approximately 10 ms.*

SENSe:COUNT <array size>
This command sets the Counter to do <array size> number of measurements. CONFigure and MEASure commands also set this count.

SENSe:MODe <mode>
This command sets the Counter to make a measurement for a length of time

(APERture) or for a number of cycles of the input signal (EVENTs). However some functions don't have one or both of the APERture and EVENts modes. Those functions ignore the mode setting.

CONFigure and MEASure commands set the mode to APERture.

***NOTE**. Event mode with greater than 1 event is invalid for Time Interval with Delay, Duty Cycle, or Phase Angle measurements.*

SENSe:EVENts <# of events>

This command sets the Counter <# of events>. If the SENSe:<function>:MODE is set to EVENTs, the Counter will make a measurement for a number of cycles of the input signal. The range of events is 1 to 9E15 step 1, however the measurement must complete in 9E6 seconds (99 days) to avoid errors.

***NOTE**. Event mode with greater than 1 event is invalid for Time Interval with Delay, Duty Cycle, or Phase Angle measurements.*

SENSe[1|2|3]:FUNCtion

The SENSe:FUNCtion command selects a function and input channel without changing most of the Counter setup. The input filtering, coupling, and impedance are not changed by these commands. You must select the filtering, coupling and impedance that makes sense for the input signal. The input attenuation, offset, gain, and comparator hysteresis are not changed by these commands.

However, the comparator slopes and thresholds are changed to defaults by these commands. If INPut:SETup:AUTO (autotrigger) is set to ON or ONCE, the input attenuation, offset, gain, and comparator hysteresis may change when the measurement is started. See the CONFigure or MEASure command descriptions for detail of these commands.

The possible parameters for this command are as follows:

“TOTalize”
“FREQuency”
“FREQuency:RATio <second channel>”
“PERiod”

SENSe[1,2]:FUNCtion

Selects a function and input channel without changing most of the set up of the Counter. The input coupling and impedance are not changed by these commands. The user must select the coupling and impedance that makes sense for the input signal. The input attenuation, offset, gain, and comparator hysteresis are not changed. However, the comparator slopes and thresholds are changed to defaults. If INPut:SETup:AUTO (autotrigger) is set to ON or ONCE, the input attenuation, offset, gain, and comparator hysteresis may change when the measurement is started.

The possible parameters for this command are as follows:

“DCYCle”
“FALL:TIME”
“FTIMe”
“NDUTycycle”
“NWIDth”
“PDUTycycle”
“PHASe”
“PWIDth”
“RISE:TIME”
“RTIMe”
“TINTerval”
“VOLTage:AC”
“VOLTage:DC”
“VOLTage:MINimum”
“VOLTage:MAXimum”
“VOLTage:PTPeak”

***NOTE**. “DCYCle” is the same as “PDUTycycle,” “FTIMe” is the same as “FALL:TIME,” and “RTIMe” is the same as “RISE:TIME.”*

SENSe[1|2|11|12|21|22]:FUNCtion

Selects a function and input channel without changing most of the set up of the Counter. The input coupling and impedance are not changed by these commands. You must select the coupling and impedance that makes sense for the input signal. The input attenuation, offset, gain, and comparator hysteresis are not changed. However, the comparator slopes and thresholds are changed to defaults. If INPut:SETup:AUTO (autotrigger) is set to ON or ONCE, the input attenuation, offset, gain, and comparator hysteresis may change when the measurement is started.

The possible parameters for this command are as follows:

“TINTerval:DELay:TIMe”
“TINTerval:DELay:EVENTS”

SENSe[1|2|10|20]:FUNCtion
The SENSe:FUNCtion command selects a function and input channel without changing most of the Counter setup. The input filtering, coupling, and impedance are not changed by these commands. You must select the filtering, coupling and impedance that makes sense for the input signal. The input attenuation, offset, gain, and comparator hysteresis are not changed by these commands.

However, the comparator slopes and thresholds are changed to defaults by these commands. If INPut:SETup:AUTO (autotrigger) is set to ON or ONCE, the input attenuation, offset, gain, and comparator hysteresis may change when the measurement is started. See the CONFigure or MEASure command descriptions for detail of these commands.

The possible parameters for this command are as follows:

“TOTalize”

SENSe:TINTerval:DELay:EVENTs <event delay>
This command sets the number of events to use for the time interval with delay.

SENSe:TINTerval:DELay:TIME <time delay>
This command sets the delay time for time interval with delay by time function.

Query Response

SENSe:APERture? [DEFault|MINimum|MAXimum]
This query moves the currently set aperture time, set by the most recent SENSe:...:APERture, CONFigure or MEASure command, to the output buffer. If one of the optional arguments is used, the DEFault, MINimum or MAXimum value of aperture is moved to the output buffer instead.

SENSe:COUNT? [DEFault|MINimum|MAXimum]
This query moves the currently set <array size> count, as set by the most recent SENSe:...:COUNt, CONFigure or MEASure command, to the output buffer. If one of the optional arguments is used, the DEFault, MINimum or MAXimum value of count is moved to the output buffer instead.

SENSe:EVENts? [DEFault|MINimum|MAXimum]
This query moves the number of events, as set by the most recent SENSe:...:EVENts command, to the output buffer . If one of the optional arguments is used, the DEFault, MINimum or MAXimum value of aperture is moved to the output buffer instead.

SENSe:MODE? [DEFault]
This query moves the currently set mode, as set by the most recent SENSe:...:MODE, CONFigure or MEASure command, to the output buffer. If the optional DEFault argument is used, then the default value of aperture mode will be moved to the output buffer instead.

SENSe:FUNCtion?
This query moves the currently selected function to the output buffer. Functions are selected by a SENSe:FUNCtion, CONFigure or MEASure command.

SENSe:TINTerval:DELay:EVENts? [DEFault]|MINimum]|MAXimum]
Returns the current value of the time interval delay by events of the default, minimum or maximum value. The default value is 1 event.

Examples

SENSe:APERture

Command	Response
SENS:APER DEF	Returns set aperture to default
SENSe:APERture?	1e–1 Query aperture value

SENSe:COUNt

Command	Response
SENSE:COUNT 100	Set array size to 100 measurements
SENS:COUN?	100 Query array size

SENSe:EVENts

Command	Response
SENS:EVENT MIN	Returns set events to minimum
SENSe:EVENts?	1 Queries events value

SENSe:MODe

Command	Response
SENSE:MODE EVENTS	Set mode to events
SENS:MOD?	EVEN Queries for the current mode

SENSe[1|2|10|11|12|20|21|22]:FUNCtion

Command	Response
SENSE1:FUNCTION “FRE-QUENCY”	Set mode to events
SENS1:FUNC?	FREQ Query Function

SENSe:TINTerval:DELay:EVENts

Command	Response
SENS:TINT:DEL:EVEN MIN	Set event delay to minimum
SENSE:TINT:DELAY:EVENTS?	1 Query event delay

SOURce Subsystem

The SOURce subsystem commands are used to command the TCXO1 option (if available).

Command Syntax

```
SOURce:COSCillator[:SOURce] <source>
```

Query Syntax

```
SOURce:COSCillator[:SOURce]?
SOURce:COSCillator:VALue?
```

***RST Value**

Parameter	Value
<source>	TCX01 with the Option 1T installed, ROSCILLATOR otherwise

Limits

Parameter	Value
<source>	ROSCILLATOR|TCXO1

Related Commands N/A

Command Description

SOURce:COSCillator[:SOURce] ROSCillator|TCXO1

This command selects the source of the Counter reference oscillator. Choices are the VX4101A Reference Oscillator (ROSCILLATOR) or the Temperature Controlled Crystal Oscillator (TCXO1). Specifying TCX01 as a source will generate an error if Option 1T is not available on the VX4101A.

Query Response

Query	Value
SOURce:COSCillator:VALue?	<oscillator frequency>
SOURce:COSCillator[:SOURce]?	This query returns the current clock source selected for the counter. Possible values are: ROSCILLATOR|TCXO1

Examples

SOURce:COSCillator[:SOURce] ROSCillator|TCXO1

Command	Response
*RST	
SOUR:COSC?	TCXO1

SOURce:COSCillator:VALue?

Command	Response
SOUR:COSC:VAL?	9.9999999999999E+06

STATus Subsystem

The STATus commands for the VX4101A Counter includes the use of the *coincidence bit*. This bit indicates that a time interval measurement has occurred that is approximately coincident with the built-in delay for the counter front end. This delay is calibrated and is usually about 1.8 ns. There are three cases in which the coincidence bit is used:

- During single-shot measurements, the firmware checks the coincidence bit, and if it is set, the time interval result to be returned will be set to the calibrated delay value. There are no errors generated.

- During continuous single-shot, or array measurements, one of the following actions take place:

  a. The firmware behaves as described above for single-shot measurements. It checks the coincident bit for each measurement and returns the calibrated delay value if it is set. No errors are generated.

- During gated measurements, using either an external gate, events mode where events are larger than 1, or aperture mode where the aperture is longer than just a single event:

  The firmware will check the coincidence bit at the end of the gate. If it is set, it will still return its calculated answer, but it will also generate an error condition. You can set up the status registers to interrupt on such an error.

Command Syntax N/A

Query Syntax `STATus:OPERation:CONDition?`

Command Class Instrument

Query Response The operational condition register value.

***RST Value** 0

Limits N/A

Formats Query Response
Numeric

Related Commands MEASure?
READ?
INITiate
ABORt

Description The STATus:OPERation:CONDition query returns the current operational status of the Counter. The bit definitions of the value are (bit 0 = the least significant bit):

Bit	Definition	Function
0	Calibrating	Set when any CALibration operation is running. Cleared when the CALibration operation is complete
1	Settling	Set when the instrument changes its function or range. Cleared when the all circuitry has settled
2	Ranging	Set when the instrument is auto-ranging. Cleared when the input range has been found
3	Sweeping	Not used
4	Measuring	Set when an INITiate command is executed. Cleared when the command is complete or aborted
5	Triggering	Not used
6	Arming	Set when the instrument is waiting for an arm signal. Cleared when the arm is received
7	Correcting	Set when the instrument is performing an auto-zero operation. Cleared when the auto-zero operation is complete
8	Testing (User 1)	Set when the instrument is performing a self-test. Cleared when the self-test is complete
9	Aborting (User 2)	Set when the instrument is in the process of aborting an operation. Cleared when the abort is complete
10	User 3	Not used
11	User 4	Not used
12	User 5	Reserved
13	Instrument Summary	Not used
14	Program Running	Not used
15	Reserved	Always 0

Examples

Command	Response/Description
stat:oper:cond?	16 Makes a measurement (0010 hex)
stat:oper:cond?	3072 Measurement complete because of ABORt (0C00 hex)

TEST Subsystem

The TEST subsystem handles the self test operations of the instrument. The Counter self test tests the Counter memory, the Read/Write hardware control registers, the analog front end, and a 2.5 MHz test signal. The query returns pass/fail information. In a failed situation, additional failure information can be obtained with the SYStem:ERRor? query.

Command Syntax N/A

Query Syntax `TEST:ALL?`

***RST Value** N/A

Limits N/A

Related Commands `*TST?`

Query Response Initiates the Counter self-test operation. If the test fails, an error message is placed in the error queue and the error LED blinks. The self test tests the following:

- Two 4 Kb Counter measurement buffers
- Logic registers
- Analog front end pre-amp offset, pre-amp inverter, and pre-amp gain digital to analog converters (DACs)
- A 2.5 MHz signal is routed in through a test source and checked for accuracy

Examples

Command	Response/Description
INST:SEL Counter	Counter
TEST:ALL?	“CTR: Self-Test Passed” ”CTR: Self-Test Failed Initiates the Counter self test operation and returns either a pass or failure message.

UNIT Subsystem

The UNIT subsystem command specifies the units for the phase measurements as either degrees or radians and determines whether the units will be positive or centered around zero.

Command Syntax

```
UNIT:ANGLe <units>[,<zero>]
```

Query Syntax

```
UNIT:ANGLe?
```

***RST Value**

Parameter	Value
<units>	RADIAN
<zero>	MINIMUM

Limits

Parameter	Value
<units>	DEGREE|RADIAN
<zero>	MINIMUM|CENTER|AUTO

Related Commands

CONFigure|MEASure|SENSe[:...]:PHASe

Command Description

Specifies the units for the phase measurements as either degrees or radians and determines whether the units will be positive or centered around 0.

MINimum will set 0 as the minimum measurement and return measurements as 0 to 360 degrees or 0 to 2 π radians.

CENTer will set 0 as the center measurement and return measurements from –180 to +180 or – π to + π radians. Selecting AUTo will start as MINimum, returning positive values only. The instrument will automatically switch to CENTer if crossing 0 degrees (0 radians) in the negative direction and automatically switch back to MINimum if crossing 180 degrees (π radians) in the positive direction.

Query Response

DEG|RAD,MIN|CENT|AUT

Examples

Command	Response
UNIT:ANGL DEG,CENT	values from –180 to +180 Sets the units to degrees

SCPI Commands for the Digital Input

This section contains the SCPI commands for the Digital Input. You can use the commands to calibrate the instrument, prepare it for operation, or to trigger a measurement.

Command Summary

The following is a listing of the available command subsystems and syntax:

CALibration Subsystem

```
CALibration:VALue <cal value>
```

CONFigure Subsystem

Commands

```
CONFigure[:SCALar]

    :DIGLobal

        [:NORMal] <voltage_threshold>
        :INVerted <voltage_threshold>

    :DIPort

        [:NORMal] <voltage_threshold>,<numeric_port_list>
        :INVerted <voltage_threshold>,<numeric_port_list>

    :DIBit

        [:NORMal] <voltage_threshold>,<channel_bit_list>
        :INVerted <voltage_threshold>,<channel_bit_list>

CONFigure[:ARRay]

    :DIGLobal

        [:NORMal] <repetitions>,<voltage_threshold>
        :INVerted <repetitions>,<voltage_threshold>

    :DIPort

        [:NORMal]
        <repetitions>,<voltage_threshold>,numeric_port_list>

        :INVerted
        <repetitions>,<voltage_threshold>,numeric_port_list>
```

```
    :DIBit

        [:NORMal]
        <repetitions>,<voltage_threshold>,<channel_bit_list>

        :INVerted
        <repetitions>,<voltage_threshold>,channel_bit_list>
```

Queries

```
CONFigure?
```

FETch? Subsystem

Queries

```
FETCh? [<count>[<offset>[,<step_size>]]]

FETCh:COUNt?
```

FORMat Subsystem

Commands

```
FORMat[:DATA] <format>
```

Queries

```
FORMat[:DATA]?
```

INITiate Subsystem

Command

```
INITiate:[IMMediate] <control>
```

INSTrument Subsystem

Commands

```
INSTrument:ABORt[:IMMediate]
INSTrument:RESet
```

MEASure Subsystem

Queries

```
MEASure[:SCALar]

    :DIGLobal

        [:NORMal]? <voltage_threshold>
        :INVerted? <voltage_threshold>

    :DIPort

        [:NORMal]? <voltage_threshold>,<numeric_port_list>
        :INVerted? <voltage_threshold>,<numeric_port_list>
```

```
:DIBit

   [:NORMal]? <voltage_threshold>,<numeric_port_list>
   :INVerted? <voltage_threshold>,<numeric_port_list>

MEASure[:ARRay]

:DIGLobal

   [:NORMal]? <repetitions>,<voltage_threshold>
   :INVerted? <repetitions>,<voltage_threshold>

:DIPort

   [:NORMal]?
   <repetitions>,<voltage_threshold>,<numeric_port_list>
   :INVerted?
   <repetitions>,<voltage_threshold>,<numeric_port_list>

:DIBit

   [:NORMal]?
   <repetitions>,<voltage_threshold>,<channel_bit_list>
   :INVerted?
   <repetitions>,<voltage_threshold>,<channel_bit_list>
```

READ? Subsystem

```
READ?
```

SENSe Subsystem

Commands

```
SENSe:ARRay <array size>
SENSe:MODE <mode>
SENSe:PSELect

:DIBit

   [:NORMal]
   <ENABle|DISable>,<channel_bit_list>|<numerical_bit_mask>
   :INVerted
   <ENABle|DISable>,<channel_bit_list>|<numerical_bit_mask>

:DIGLobal

   [:NORMal] <ENABle|DISable>
   :INVerted <ENABle|DISable>

:DIPort

   [:NORMal] <ENABle|DISable>,<port list>
   :INVerted <ENABle|DISable>,<port list>
```

```
SENSe:SRATe <sample rate>
SENSe:THReshold <voltage_threshold>
```

Queries

```
SENSe:ARRay?
SENSe:MODE?
SENSe:PSELect?
SENSe:SRATe?
SENSe:THReshold?
```

STATus Subsystem

```
STATus:OPERation:CONDition?
```

TEST Subsystem

Queries

```
TEST:ALL?
```

TRIGger Subsystem

Commands

```
TRIGger:MASK <mask>
TRIGger:MATCh <pattern>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event delay>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

    :REQuest NORMal|INVerted <polarity>
    :STRobe NORMal|INVerted <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
```

Queries

```
TRIGger:MASK?
TRIGger:MATCh?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

    :REQuest?
    :STRobe?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

CALibration Subsystem

The CALibration subsystem handles the calibration operations of the instrument.

Command Syntax

`CALibration:VALue <cal value>`

Query Syntax

N/A

***RST Value**

N/A

Limits

Parameter	Value
<cal value>	2.5|12

Related Commands

N/A

Command Description

This command calibrates the Digital Input threshold. This is a 2-point calibration. The voltages used for calibration are 2.5 VDC and 12 VDC. You must use external 2.5 V and 12 V sources to perform the calibration. You must hook up the external source to Discrete Input 1 and set the Digital Output source to 1 (inactive).

***NOTE**. The Digital Output instrument uses the same pins as the Digital Input, so the Digital Output should be reset prior to calibrating the Digital Input to ensure that the outputs are inactive.*

Query Response

N/A

Examples

Command	Response/Description
CAL:VAL 2.5	Sets the 2.5 V calibration factor using an external 2.5 V source.

CONFigure Subsystem

The CONFigure subsystem commands sets up the instrument to take a measurement.

Command Syntax

```
CONFigure[:SCALar]

    :DIGLobal

        [:NORMal] <voltage_threshold>
        :INVerted <voltage_threshold>

    :DIPort

        [:NORMal] <voltage_threshold>,<numeric_port_list>
        :INVerted <voltage_threshold>,<numeric_port_list>

    :DIBit

        [:NORMal] <voltage_threshold>,<channel_bit_list>
        :INVerted <voltage_threshold>,<channel_bit_list>

CONFigure[:ARRay]

    :DIGLobal

        [:NORMal] <repetitions>,<voltage_threshold>
        :INVerted <repetitions>,<voltage_threshold>

    :DIPort

        [:NORMal]
        <repetitions>,<voltage_threshold>,numeric_port_list>

        :INVerted
        <repetitions>,<voltage_threshold>,numeric_port_list>

    :DIBit

        [:NORMal]
        <repetitions>,<voltage_threshold>,<channel_bit_list>

        :INVerted
        <repetitions>,<voltage_threshold>,channel_bit_list>
```

Queries

```
CONFigure?
```

Query Syntax

```
CONFigure?
```

***RST Value**

Parameter	Value
<NORMal|INVerted>	NORM
<voltage_threshold>	2.3 V
<input_mask>	#hffffffff
<repetitions>	1
<numeric_port_list>	(1:4)
<channel_bit_list>	(@1:32)

Limits

Parameter	Value
<NORMal|INVerted>	NORMal|INVerted
<voltage_threshold>	0.0–20.0 V
<input_mask>	#h00000000 to #hffffffff
<repetitions>	1–4096
<numeric_port_list>	1–4
<channel_bit_list>	1–32

Related Commands

READ?
INITiate
FETCh?
MEASure?

Command Description

The '[:SCALar]' commands set up the Digital Input for a single measurement read, while the 'ARRay' commands set up the Digital Input for a multiple read sequence. The rate at which data is read in is determined by the Digital Input's sample rate or handshake rate, depending on whether handshaking is enabled or not. For ARRay commands, the sample rate is set to default. When a NORMal command is used, data is returned as a 1 if the input is higher than the input threshold and a 0 if the input is lower than the threshold. For an INVerted command, the opposite is true. The data mode will be formatted, i.e. all disabled bits will be returned as 0.

If a port list is used, selected ports will be enabled or disabled. Only bits enabled as input ports can be queried for input. The port list can be comma separated or listed as (x:y) for ports x through y.

Port	Bits
1	1 through 8
2	9 though 16

Port	Bits
3	17 through 24
4	25 through 32

If a bit_list is used, specific input bits will be enabled or disabled. Only bits enabled as input bits can be read from and queried for its input. The bit list can be comma separated or listed as (@x:y) for bits x through y.

Query Response

Command	Value
CONFigure?	"<SCALar|ARRay>;<DIGlobal|DIPort|DIBit>:NORMal|INVerted>",<voltage_threshold>V,<input_mask>,<repetitions>

Examples

Command	Response/Description
FORMAT HEX	Sets the return format to hex.
SENS:PSEL:DIGL ENAB	Enables all bits as input.
CONFIGURE:DIGLOBAL 2.5	Sets up the: Voltage threshold to 2.5 V Read from all inputs Normal One reading
CONF?	"SCALar:DIGLobal:NORMal", 2.5 V, #hffffffff ,1 Queries the input setting.

FETCh? Subsystem

The FETCh? queries retrieve completed measurements. You can use FETCh:COUNt? to monitor the progress of a measurement or an array of measurements.

Command Syntax

N/A

Query Syntax

```
FETCh? [<count>[<offset>[,<step_size>]]]
FETCh:COUNt?
```

***RST Value**

Parameter	Value
<count>	Currently specified array size, which is 1+postmatch count if pattern match is used.
<offset>	(without match pattern) 1
<step_size>	1
<postmatch measurement count, prematch count>	0,0 (match not found yet)

Limits

Parameter	Value
<count>	If match pattern is used or currently specified array size if match isn't used, the value is: Prematch count + postmatch count +1 (pattern matched)
<offset>	–1 times prematch count to postmatch count if pattern match is used or 1 to currently specified array size if match isn't used.
<step_size>	Same limits as <count>
<postmatch measurement count, prematch count>	0 ≤ postmatch count ≤ 4096, 0 ≤ prematch count ≤ 4095

***NOTE**. Data starts at 1; a postmatch value of 0 is used to indicate that no match has occurred yet).*

Related Commands

READ?
MEASure?
CONFigure

Query Response

FETCh? [<count>[<offset>[,<step size>]]]
Fetches the last measurement(s) taken after an INITiate. If you don't use the pattern match, FETCh can return all or part of the array size specified. If a pattern match is used, then FETCh can also retrieve pretrigger data. For example, if the pattern match is used and the prematch and postmatch buffers are full, then FETCh? 8192,–4095,1 would retrieve all the data. For the same case, FETCh? 4096,1,1 retrieves all the postmatch data, while FETCh? 4097,0,1 would retrieve the pattern match in addition to the postmatch data. To determine how many prematch and postmatch points are available, use the FETCh:COUNt? query.

FETCh:COUNt?
This query fetches the number of postmatch and prematch measurements taken since the last INITiate.

***NOTE**. If you enable the Digital Input pattern match, both the prematch and postmatch values remain at zero until a match is found.*

Examples

FETCh?

Command	Response/Description
FORM HEX	Sets the return format to hexadecimal.
SENSE:PSEL:DIGL ENAB	Enables all bits as input
CONF:ARRAY:DIGLOBAL 2,2.5	Sets up the: Voltage threshold to 2.5 V Read from all inputs, Normal, Two readings
READ?	#hf0f0f0f0,#h1111111 Orders a read in the current configuration.
CONF:ARR:DIGL 2,18.0	Setting up a different configuration.
INITiate	Arm the instrument.
FETCh?	#h0000ffff,#hfffff0000 Getting last read values.

FETCh:COUNt?

Command	Response/Description
FORMAT HEX	Sets the return format to hexadecimal
SENSE:PSELECT: DIGLOBAL ENAB	Enables all bits as input
sens:arr 1000	Take 1000 postmatch measurements
sense:threshold 2.8	Set the threshold to 2.8V
trig:matc #h00007e38	Set the match pattern
trigger:mask #h0000FFFF	Set the mask for the match pattern
initiate	Start taking measurements
fetc, coun?	7,42 Reads how many measurements have been taken so far

FORMat Subsystem

This subsystem defines the format for returned data.

Command Syntax

`FORMat[:DATA] <format>`

Query Syntax

`FORMat[:DATA]?`

***RST Value**

Parameter	Value
<format>	HEX

Limits

Parameter	Value
<format>	INTeger|HEXadecimal|BINary

Related Commands

N/A

Query Response

`INT|HEX|BIN`

Command Description

The FORMat[:DATA] command specifies the format of the data returned from all queries returning numeric data. Data can be returned in an ASCII integer, ASCII hexadecimal or binary format.

The binary trace format is based upon the IEEE 488.2 Indefinite Arbitrary Block Program Data type. The data is preceded by a preamble of <#0>, followed by binary data. Each sample is encoded in four bytes. The most significant byte is transmitted first, followed by the least significant byte (big endian). The series of samples is terminated by a new line (‘\n’) and the end bit. The most significant bit corresponds to bit 32, and the least significant bit corresponds to bit 1.

Examples

Command	Response/Description
format hex	Sets the return format to hexadecimal
form?	HEX Queries the return format

INITiate Subsystem

Command Syntax `INITiate:IMMediate`

Query Syntax N/A

Command Class Instrument

***RST Value** N/A

Limits N/A

Related Commands ABORt
INSTrument:ABORt

Command Description **INITiate:IMMediate**
This command initiates the current programed trigger sequence. After the instrument has completed the current trigger sequence, it enters the idle state. See instrument documentation for details on instrument state after an INITiate.

Query Response N/A

Examples See TRIGger([: SEQuence1] |:STARt) [:LAYer]:IMMediate command for an example.

INSTrument Subsystem

The INSTrument subsystem aborts, resets, or selects the instrument.

Command Syntax `INSTrument:ABORt[:IMMediate]`

`INSTrument:RESet`

Query Syntax N/A

***RST Value** N/A

Limits N/A

Command Description

INSTrument:ABORt [:IMMediate]
This command places the Digital Input in the idle state, and terminates any activities that are currently in progress. A subsequent INIT command will cause the Digital Input to initiate a new operation based upon the current configuration, which is unchanged by the ABORt. After execution, the Digital Input Instrument is still selected.

You can use this command for terminating an unsuccessful pre-match capture.

INSTrument:RESet
This command resets the Digital Input to its power-on (default) state. After execution, the Digital Input remains selected. After reset, the Digital Input memory is initialized to zero. The sample rate and points are at their default values.

Query Response N/A

Examples

INSTrument:ABORt

Command	Response/Description
inst:sel digi	Selects the Digital Input (Digital Input configuration commands)
inst:abor	Aborts the Digital Input activity
instrument:select?	DIGI Queries the current instrument

INSTrument:RESet

Command	Response/Description
inst:sel digi	Selects the Digital Input
instrument:reset	Resets the Digital Input
inst:sel?	DIGI Queries the instrument selected

MEASure Subsystem

The MEASure subsystem configures and initiates a measurement, and then returns the results.

Command Syntax N/A

Query Syntax

```
MEASure[:SCALar]

    :DIGLobal

        [:NORMal]? <voltage_threshold>
        :INVerted? <voltage_threshold>

    :DIPort

        [:NORMal]? <voltage_threshold>,<numeric_port_list>
        :INVerted? <voltage_threshold>,<numeric_port_list>

    :DIBit

        [:NORMal]? <voltage_threshold>,<numeric_port_list>
        :INVerted? <voltage_threshold>,<numeric_port_list>

MEASure[:ARRay]

    :DIGLobal

        [:NORMal]? <repetitions>,<voltage_threshold>
        :INVerted? <repetitions>,<voltage_threshold>

    :DIPort

        [:NORMal]?
        <repetitions>,<voltage_threshold>,<numeric_port_list>
        :INVerted?
        <repetitions>,<voltage_threshold>,<numeric_port_list>

    :DIBit

        [:NORMal]?
        <repetitions>,<voltage_threshold>,<channel_bit_list>
        :INVerted?
        <repetitions>,<voltage_threshold>,<channel_bit_list>
```

***RST Value**

Parameter	Value
<NORMal|INVerted>	NORM
<voltage_threshold>	2.3 V
<input_mask>	#hffffffff
<repetitions>	1
<numeric_port_list>	(1:4)
<channel_bit_list>	(@1:32)

Limits

Parameter	Value
<NORMal|INVerted>	NORMal|INVerted
<voltage_threshold>	0.0–20.0 V
<input_mask>	#h00000000 to #hffffffff
<repetitions>	1–4096
<numeric_port_list>	1–4
<channel_bit_list>	1–32

Related Commands

CONFigure
READ?

Query Response

The MEASure:SCALar queries set up the Digital Input for a single measurement, reads a value, and returns the requested data.

The MEASure:ARRay queries set up the Digital Input for multiple reads, reads the values, and returns the requested data.

The voltage threshold can be specified or set to default. For ARRay: queries, the sample rate is set to default. When a NORMal query is used, data is returned as a 1 of the input is higher than the input threshold and a 0 if the input is lower than the threshold. For an INVerted query, the opposite is true. The data mode will be formatted, i.e. all disabled bits will be returned as 0.

If a port list is used, selected ports will be enabled or disabled. Only bits enabled as input ports can be queried for input. The port list can be comma separated or listed as (x:y) for ports x through y.

Data is returned in the following format:

```
<data>{,<data>}
```

Port	Bits
1	1 through 8
2	9 though 16
3	17 through 24
4	25 through 32

If a bit_list is used, specific input bits will be enabled or disabled. Only bits enabled as input bits can be read from and queried for its input. The bit list can be comma separated or listed as (@x:y) for bits x through y.

MEASure returns the complete array (see READ?). To retrieve prematch data, use a CONFigure or SENSe command followed by an INITiate command and then a FETCh? query.

Examples

Command	Response/Description
format hex	Sets the return format to hexadecimal
sens:psel:digl enab	Enables all bits as input
meas:digl? 2.3	#h00000000 Initiates a measurement with voltage threshold at 2.3 V
meas:digl:inv? 4.4	#hffffffff Initiates a measurement with voltage threshold at 4.4 V and inverts the bits.

READ? Subsystem

The Read subsystem initiates a measurement and returns the results.

Command Syntax N/A

Query Syntax READ?

***RST Value** N/A

Limits N/A

Related Commands

CONFigure
FETCh?
INITiate

Query Response

This query performs an INITiate, followed by a FETCh? query. The number of data points read, voltage threshold and bits read are setup with a CONFigure or SENSe command. READ? returns the complete array if no pattern match is used, or the match and all the postmatch data if pattern match is used. To retrieve prematch data, use a CONFigure or SENSe command followed by an INITiate command and then a FETCh? query. Data is returned in the following format:

```
<data_point>{,<data_point>}
```

Examples

Command	Response/Description
format hex	Sets the return format to hexadecimal
sens:psel:digl enable	Enables all bits as input
configure:diglobal 2.5	Sets up the following: Voltage threshold to 2.5 V Read all inputs Normal One reading
conf	“<SCALar|ARRay>;<DIGlobal|DIPort|DIBit>:NORMal|INVerted>”,<voltage_threshold>V,<input_mask>,<repetitions>
READ?	f0f0f0f0 Orders a read in the current configuration

SENSe Subsystem

The SENSe subsystem sets up the instrument to take a measurement.

The SENSe: commands can also configure the instrument following a CONFigure command. However, the CONFigure command or MEASure? query will reset all SENSe command parameters to the default state.

Command Syntax

```
SENSe:ARRay <array size>

SENSe:MODe <mode>

SENSe:PSELect

   :DIBit

      [:NORMal] <ENABle|DISable>,<channel_bit_list>|
      <numerical_bit_mask>
```

```
      :INVerted <ENABle|DISable>,<channel_bit_list>|
      <numerical_bit_mask>

   :DIGLobal

      [:NORMal] <ENABle|DISable>
      :INVerted <ENABle|DISable>

   :DIPort

      [:NORMal] <ENABle|DISable>,<port_list>
      :INVerted <ENABle|DISable>,<port_list>

SENSe:SRATe <sample rate>

SENSe:THReshold <voltage_threshold>
```

Query Syntax

```
SENSe:ARRay?

SENSe:MODE?

SENSe:PSELect?

SENSe:SRATe?

SENSe:THReshold?
```

*RST Value

Parameter	Value
<array size>	1
<mode>	FORMatted
<ENABle|DISable>	ENABle
<channel_bit_list>	(@1:32)
<port_list>	(1:4)
<numerical_bit_mask>	#hffffffff
<sample rate>	48000.0
<voltage threshold	2.3

Limits

Parameter	Value
<array size>	1-4096
<mode>	FORMatted|UNFormatted
<channel_bit_list>	1-32
<numerical_bit_mask>	#h00000000 to #hffffffff
<ENABle|DISable>	ENABle|DISable

Parameter	Value
<channel_bit_list>	1-4
<port_list>	1-4
<sample rate>	3.662-48000.0
<voltage threshold	0.00-20.000

Resolution

Command or Query	Value
SENSe:SRATe?	(240e+03/n) where $5 \leq n \leq 65536$
SENSe:THReshold?	5 mV

Related Commands

FETCh?
READ?
CONFigure
MEASure

Description

SENSe:ARRay <array size>
Sets the number of postmatch data points. The match pattern is defined as index 0. Prematch will automatically store up to 4095 patterns before the match.

SENSe:MODE <mode>
This command determines whether the return data is masked with the bits defined as inputs. Selecting FORMatted with this command masks all readbacks to 0 for those bits not programmed as input.

Selecting UNFormatted with this command sets all readbacks to return raw data from all 32 bits, whether they are defined as input bits or not. Since all 32 bits of Digital Input are common with the Digital Output Instrument, a low from either the UUT or from the Digital Output Instrument will result in a low on that bit in UNFormatted mode (the result will be a high if the data is specified as INVerted).

SENSe:PSELect:DIBit [:NORMal|:INVerted] <ENABle|DISable>,<channel_bit_list>|<numeric_bit_mask>
Enables or disables specific input bits defined by the <channel_bit_list>. It can be comma separated or listed as (@x:y) for bits x through y. The query returns the enable/disable state of all the input bits. A one in a bit position indicates that the bit is enabled.

SENSe:PSELect:DIGLobal [:NORMal]|:INVerted <ENABle|DISable>
Enables or disables all 32 bits of the Digital Input. The query returns the

enable/disable state of all the input bits. A one in a bit position indicates that the bit is enabled.

SENSe:PSELect:DIPort [:NORMal]|:INVerted <ENABle|DISable>,<port_list>
Enables or disables selected ports as input bits. The port list can be comma separated or listed as (x:y) for ports x through y.

Port	Bits
1	1 through 8
2	9 though 16
3	17 through 24
4	25 through 32

The query returns the enable/disable state of all the input bits. A one in a bit position indicates that the bit is enabled.

NOTE*. The SENSe:PSel commands change any previously-defined NORMal|INVerted setting. See the CONFigure or MEASure commands.*

SENSe:SRATe <sample rate>
Sets the sample (frequency) rate of input in Hz. The SRATe command is used to set the sample rate of array SENSe commands and the input frequency of samples taken in a pattern match operation.

If a numeric value is specified that does not correlate to an integer “n” as specified above, the instrument will automatically select the closest value.

SENSe:THReshold <voltage_threshold>
This command sets the threshold for all digital input bits. If the SENSe:PSELect command selects NORMal, then inputs above the threshold will read 1 and those below will read as 0. If the SENSe:PSELect command specifies INVerted, then the opposite is true.

Query Response

Query	Response
SENSe:ARRay?	<array size>
SENSe:MODE?	0 formatted, 1 unformatted
SENSe:PSELect?	<input_word_mask>
SENSe:SRATe?	<sample_rate>
SENSe:THReshold?	<voltage_threshold>

Examples

SENSe:ARRay

Command	Response/Description
SENS:ARR 5	Sets the number of postmatch points to 5
SENSe:ARRay?	5 Queries the number of specified postmatch data points.

SENSe:PSELect:DIGLobal

Command	Response/Description
FORMAT HEX	Sets the return format to hexadecimal.
SENS:PSEL:DIGL ENAB	Enables all bits as input bits.
SENS:PSELECT?	#hffffffff Queries all bit states.
SENS:PSEL:DIGL DIS	Disables all bits as input bits.
SENSE:PSELECT	#h00000000 Queries all bit states.

SENSe:PSELect:DIPort

Command/Query	Response/Description
FORM HEX	Sets the return format to hexadecimal.
SENSe:PSELect:DIPort ENAB,(1,4)	Enables ports 1 & 4 as input bits
SENS:PSELECT?	#hff0000ff Queries all bit states.
SENS:PSEL:DIP ENAB,(1:4)	Enables ports 1 thru 4 as input bits.
SENS:PSEL:DIP DIS,(1)	Disables port 1 as an input port.
SENSe:PSELect?	#hffffff00 Queries all bit states.

SENSe:PSELect:DIBit

Command/Query	Response/Description
FORM HEX	Sets the return format to hexadecimal.
SENSe:PSELect:DIBit ENAB,(@1:8)	Enables bits 1 – 8 as input bits.
SENS:PSEL?	Queries all bit states. #h000000ff

Command/Query	Response/Description
SENS:PSEL:DIB DIS-ABLE,#hf0	Disables bits 5 – 8 as input bits.
SENSe:PSELect?	#h0000000f Queries all bit states.

SENSe:MODE <FORMatted\|UNFormatted>

Command/Query	Response/Description
SENS:MODE FORM	Sets the data returned to be formatted.
SENS:MODE?	FORMATTED Queries the mode for the data format.

SENSe:SRATe?

Command/Query	Response/Description
SENSE:SRATE 48000.0	Sets the sample rate to 48 kHz.
SENS:SRAT?	48000.0 Queries the sample rate.
SENSE:SRATE 24100.0	Sets the sample rate to 24.1 kHz
SENS:SRAT?	24100.0 Queries the sample rate.

SENSe:THReshold?

Command/Query	Response/Description
SENSE:THRESHOLD 10	Sets the threshold.
SENS:THR?	10.000 Queries the threshold.

STATus? Subsystem

The STATus queries enable you to inquire on the current operational state of the instrument.

Many STATus commands are available for use for all instruments. For a summary of those commands, see the *Status and Events* section.

Command Syntax

N/A

Query Syntax

`STATus:OPERation:CONDition?`

Command Class

Instrument

***RST Value**

0

Limits

N/A

Formats

Query Response
Numeric

Related Commands

MEASure?
READ?
INITiate
ABORt

Query Response

The STATus:OPERation:CONDition query returns the current operational status of the Digital Input. The bit definitions of the value are (bit 0 = the least significant bit):

Bit	Definition	Function
0	Calibrating	Set when any CALibration operation is running. Cleared when the CALibration operation is complete
1	Settling	Not used
2	Ranging	Not used
3	Sweeping	Not used
4	Measuring	Set when the INITiate command is executed. Cleared when the command is complete or aborted

Bit	Definition	Function
5	Triggering	Set when the instrument is waiting for a trigger signal. Cleared when the trigger is received
6	Arming	Not used
7	Correcting	Not used
8	Testing (User 1)	Set when the instrument is performing a self-test. Cleared when the self-test is complete
9	Aborting (User 2)	Set when the instrument is in the process of aborting an operation. Cleared when the abort is complete
10	User 3	Not used
11	Pattern Match	Set when pattern match has occurred Cleared when INITiated
12	User 5	Reserved
13	Instrument Summary	Not used
14	Program Running	Not used
15	Reserved	Always 0

Examples

Command	Response
status:operation:condition?	16 Measurement in progress (0010 hex).
stat:oper:cond?	3072 Measurement complete because an ABORt was received (0C00 hex).

TEST Subsystem

The TEST subsystem handles the self test operations of the instrument. The Digital Input self test tests the Digital Input memory and the Read/Write hardware control registers. The query returns pass/ fail information. In a failed situation, additional failure information can be obtained with the SYStem:ERRor? query.

Command Syntax N/A

Query Syntax `TEST:ALL?`

***RST Value** N/A

Limits N/A

Related Commands *TST?

Query Response Initiates the Digital Input self test operation and returns one of two possible responses:

"DIGI: Self Test Passed"
"DIGI: Self Test Failed"

NOTE*. If the self-test fails, you can obtain further information with a SYST:ERR? query.*

Examples

Command/Query	Response/Description
INSTRUMENT:SELECT DIGI	Selects the Digital Input instrument.
TEST:ALL?	"Digital Input passes self-test." Runs the self-test.

TRIGger Subsystem

Command Syntax

```
TRIGger:MASK <mask>

TRIGger:MATCh <pattern>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event delay>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest NORMal|INVerted <polarity>
   :STRobe NORMal|INVerted <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
```

Query Syntax

```
TRIGger:MASK?

TRIGger:MATCh?
```

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest?
   :STRobe?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

Command Class Instrument

***RST Value**

Command	Values
<mask>	#h00000000
<pattern>	N/A
<time delay>	0 seconds
<event delay>	0 triggers (pass-through)
<polarity>	INVerted
<mode>	ALL
<source>	IMMEDIATE
<mode>	ONCE, ALL

Limits

Command	Values
<mask>	#h00000000 – #hffffffff
<pattern>	#h00000000–#hffffffff
<time delay>	0 seconds=pass through 1E–06 to 65.535 E–03M in 1E–06 steps
<event delay>	0 = pass through, 1 to 65,535 triggers in step of 1
<polarity>	NORMal|INVerted
<mode>	ONCE, ALL
<source>	BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

Related Commands N/A

Command Description

TRIGger:MASK <mask>
Defines a mask for the match pattern. Bits set to 0 in the mask are “don’t care” bits for the match pattern. Thus, setting the mask to #h00000000 will disable pattern matching completely.

See the TRIGger:MASK <mask> example for more information on changing the mask.

TRIGger:MATCh <pattern>
Defines a match pattern to use for a search. Until the instrument finds a match pattern, all read data will be stored in prematch memory. When the instrument finds the match pattern, it is placed in prematch memory.

See the TRIGger:MASK <mask> example for more information on changing the mask.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay <time delay>
Specifies a time delay to occur after receipt of a trigger prior to actually triggering. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the event count delay, so specifying a delay of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger command.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt <event delay>
Specifies the number of triggers to count prior to triggering. Upon receipt of trigger N (where N is the number specified in the command), the instrument will enter the triggered state. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the delay by time parameter, so specifying an event count of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity:REQuest|:STRobe NORMal|INVerted <polarity>
Specifies the active edge of the signals to be received and driven on the external handshake lines. NORMal indicates that the rising edge is active. INVerted indicates that the falling edge is active.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate
Causes a one time entry into the triggered state without receiving the specified

trigger. This command is often used to “prime the pump” in cases such as setting up a scan list measurement.

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
When an instrument has been configured for some type of array measurement, this command specifies whether the instrument will perform one or all operations when a trigger is received. If the mode is ALL, then all operations will be completed upon receipt of one trigger condition. If the mode is ONCE, then the instrument will perform one operation and reenter the initiated state. This will continue until the specified number of triggers has been received (and hence, the specified number of operations has been completed).

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce <source>
Selects or queries the trigger source to be used when the instrument is initiated.

Query Response

Query	Response
TRIGger:MASk	<mask>
TRIGger:MATCh?	<mask>
TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay?	<delay in seconds>
TRIGger([:SEQuence1]|: STARt)[:LAYer]:ECOunt	<triggers to count>
TRIGger([:SEQuence1]|: STARt)[:LAYer]:HANDshake: POLarity:REQuest|STRobe	NORMal | INVerted
TRIGger([:SEQuence1]|: STARt)[:LAYer]:SOURce	<current source>
TRIGger([:SEQuence1]|: STARt)[:LAYer]:SOURce: CATAlog[ALL]?	Lists all trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

NOTE: *Depending upon which options were purchased with the VX4101A, some of the trigger sources listed above may not be available for use.*

Query	Response
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?	Lists all delayable trigger sources available for use with the TRIG:SOUR command. BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?	Lists all fixed trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, TIMER, HANDSHAKE

Examples

TRIGger:MASK <mask>

Command	Response
TRIG:MATC #h00000000	Sets the value of the match pattern for all input bits. All bits are set to 0.
TRIGGER:MASK #hffffffff	Sets the match mask to "care" bits.
TRIG:MASK?	#hffffffff Queries the mask

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay

Command	Response
INST:SEL DIGIN	Selects the Digital Input instrument
CONF:VOLT:DC	Configures the Digital Input for a DC voltage reading
TRIG:SOUR TTLT0	Sets the trigger source to be TTLT0
TRIG:DEL 1E-3	Sets the trigger delay to 1 mSec
INIT	Initiates the measurement

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt

Command	Response
INS:SEL DIGIN	Selects the Digital Input instrument
CONF:VOLT:DC	Configures the Digital Input for a DC voltage reading
TRIG:SOUR TTLT0	Sets the trigger source to be TTLT0
TRIG:ECO 100	Sets the trigger count to 100
INIT	Initiates the measurement

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate

Command	Response
TRIG:SOUR:CAT?	IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
INST:SEL DMM	
CONF:VOLT:DC	
TRIG:SOUR TTLT0	
INIT	
FETC:COUN?	0
TRIG:IMM	
FETC:COUN?	1
FETC?	–6.03720E+00

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE

Command	Response
INST:SEL DMM	
CONF:ARR:VOLT:DC 3	
TRIG:MODE ONCE	
TRIG:SOUR COMM0	
INIT	
FETC:COUN?	0
TRIG:FIR0	
FETC:COUN?	1
TRIG:FIR0	
TRIG:FIR0	
FETCH:COUN?	3
FETC?	#239–6.04180E+00,–6.04180E+00,–6.04180E+00

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce

Command	Response
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
INST:SEL DMM	
CONF:VOLT:DC	
TRIG:SOUR BUS	
INIT	
FETC:COUN?	0
*TRG	
FETC:COUN?	1
FETC?	–6.03720E+00

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

Command	Response
INST:SEL DMM	
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

Command	Response
TRIG:SOUR:CAT:DEL?	BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, DAC, DIGI, DIGO, CTR_EXTARM

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Command	Response
INST:SEL:DMM	
TRIG:SOUR:CAT:FIX?	HOLD,IMMEDIATE,TIMER

SCPI Commands for the Digital Output

The Digital Output command syntax enables you to control the 32 output bits of the instrument. The bits can be programmed with pattern segment lengths to a maximum of 4096 sample points. If not all of the 32 bits are required, you can use the commands to turn the appropriate bits on or off.

Command Summary

The following is a listing of the available command subsystems and syntax:

FORMat Subsystem

Commands

```
FORMat[:DATA] <format>
```

Queries

```
FORMat[:DATA]?
```

INITiate Subsystem

Commands

```
INITiate
   :CONTinuous <control>
   [:IMMediate]
```

Queries

```
INITiate:CONTinuous?
```

INSTrument Subsystem

Commands

```
INSTrument:ABORt ([:IMMediate]|:COMPlete)
INSTrument:RESet
```

OUTPut Subsystem

Commands

```
OUTPut:LEVel <voltage level>
OUTPut:TRIGger:SOURce <source>
```

Queries

```
OUTPut:LEVel?
OUTPut:TRIGger:SOURce?
```

STATus? Subsystem

```
STATus:OPERation:CONDition?
```

TEST Subsystem

Queries

```
TEST:ALL?
```

TRACe Subsystem

Commands

```
TRACe:CLEar
TRACe|DATA[:DATA] <indices>{,<ascii_hex_data>}
TRACe|DATA[:DATA] <index>,{,ascii_hex_data, indefi-
nite_binary_data}
TRACe:POINts <number_of_points>
TRACe:RFRequency <repeat_frequency>
TRACe:RPERiod <repeat_period>
TRACe:SRATe <sample_rate>
```

Queries

```
TRACe|DATA[:DATA]? <numeric_list>
TRACe:INDice?
TRACe:POINts?
TRACe:RFRequency?
TRACe:RPERiod?
TRACe:SRATe?
```

TRIGger Subsystem

Commands

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time_delay>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event_delay>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest <polarity>
   :STRobe <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
```

Queries

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest?
   :STRobe?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

***NOTE**. The Digital Output shares pins with the Digital Input. When using a pin as an input, Tektronix recommends that you clear and initialize the Digital Output for High (inactive) output, unless the pin is intended to be a bi-directional input/output pin.*

FORMat Subsystem

This subsystem formats return data.

Command Syntax

`FORMat[:DATA] <format>`

Query Syntax

`FORMat[:DATA]?`

***RST Value**

Parameter	Value
<format>	HEX

Limits

Parameter	Value
<format>	INTeger|HEXadecimal|BINary

Related Commands

N/A

Command Description

The FORMat[:DATA] command specifies the format of the data returned from all TRACe:DATa queries. Data can be returned in an ASCII integer, ASCII hexadecimal, or binary format.

Query Response INTeger|HEXadecimal|BINary

Examples

Command	Response/Description
TRACe 1,#hffffffff	Stores data in 1 memory location.
FORMat HEX	Sets the return format to hexadecimal.
FORMat?	HEX Queries the return format
TRACe? 1	#hffffffff Queries the first data point in memory.
FORMat BIN	Sets the return format to binary.
NOTE: *Consult the IEEE 488.2 Standard, section 7.7.6 for indefinite format.*	
FORMat?	BIN Queries the return format.
TRACe? 1	<binary data> Queries the first data point in memory.
FORMAT INT	Sets the return format to integer
FORM?	INT Queries the return format
TRACe 2,#hf	Stores hex data “f” in the second memory location.
TRAC? 2	15 Queries the second sample point in memory.

INITiate Subsystem

Command Syntax

```
INITiate
   :CONTinuous <control>
   [:IMMediate|
```

Query Syntax `INITiate:CONTinuous?`

Command Class Instrument

***RST Value**

Parameter	Value
<control>	0

Limits

Parameter	Value
<control>	0|1|OFF|ON
NOTE: *0=OFF, 1=ON*	

Related Commands

ABORt

Command Description

INITiate:CONTinuous <control>
Initiates its current trigger sequence. After the instrument has completed the current trigger sequence, it re-enters the initiated state. It will continue this cycle until an abort, reset, or INIT:CONT OFF is received. See instrument documentation for details on instrument state after an initiate.

This command applies to all trigger sources.

INITiate[:IMMediate]
Initiates its current trigger sequence. After the instrument has completed the current trigger sequence, it enters the idle state.

NOTE*. This command applies only to an IMMediate TRIGger source .This command returns an error for all other trigger sources. Either instrument will accept triggers and generate data until the instrument receives one of the following commands:*

ABORt

INSTrument:ABORt

An implied abort

INITiate:CONTinuous OFF.

Consult the summaries for each of these commands for more information.

Query Response

Command	Response
INITiate:CONTinuous?	0=OFF=continuous initiate not enabled 1=ON=continuous initiate enabled

Examples **INITiate:CONTinuous**

Command	Response
INST:SEL:DIGO	
INITCONT:ON	

INSTrument Subsystem

The INSTrument subsystem aborts, resets, or selects the Digital Output instrument.

Command Syntax

```
INSTrument:ABORt([:IMMediate]|COMPlete)
INSTrument:RESet
```

Query Syntax N/A

***RST Value** N/A

Limits N/A

Related Commands ABORt

Command Description

INSTrument:ABORt([:IMMediate]|COMPlete)
This command places the Digital Output in the idle state, and terminates any activities that are currently in progress. A subsequent INIT command will cause the Digital Output to initiate a new operation based upon the current configuration, which is unchanged by the ABORt. After execution, the Digital Output Instrument is still selected. INST:ABORt:COMPlete terminates activity after the present sequence is completed. INST:ABORt:IMMediate terminates operation on the next sample.

INSTrument:RESet
This command resets the Digital Output to its power-on (default) state. After execution, the Digital Output remains selected. After reset, the Digital Output memory is initialized to zero. The sample rate and points are at their default values.

Examples

INSTrument:ABORt

Command	Response/Description
inst:sel digo	Selects the Digital Output (Digital Input configuration commands)
INIT	Initiates command
inst:abort	Ends instrument operation

INSTrument:RESet

Command	Response/Description
inst:sel digo	Selects the Digital Output
instrument:reset	Aborts instrument operation and resets the Digital Output to default state
inst:sel?	DIGO Queries the instrument selected

OUTPut Subsystem

The output subsection specifies the output voltage, output bits and handshaking mode.

Command Syntax

```
OUTPut:LEVel <voltage level>
OUTPut:TRIGger:SOURce <source>
```

Query Syntax

```
OUTPut:LEVel?
OUTPut:TRIGger:SOURce?
```

***RST Value**

Parameter	Value
<voltage level>	EXT
<source>	SEQuence

Limits

Parameter	Value
<voltage level>	5|12|24|EXT
<source>	SEQuence|SEGment|SAMPle

Related Commands N/A

Command Description

OUTPut:LEVel <voltage level>
This command sets the output voltage level for all 32 output bits. The output voltage can be set to 5V, 12V, 24V or can be input from an external source. The external input voltage source must be limited between 0V and 36V.

***NOTE**. The actual high voltage output will depend on the type of load placed on the pin. Internal loading limits the output to approximately 75% of the programmed nominal values. Any additional external load placed on the pins might limit the output further. For the load limits of the Digital Output, see Appendix A: Specifications.*

OUTPut:TRIGger:SOURce <source>
Selects when the Digital Output will generate output triggers. It can be set to generate a trigger every sample period, segment period, or once per sequence.

Query Response

Query	Response
OUTPut:LEVel <voltage level>	<5|12|24| EXT>
OUTPut:TRIGger:SOURCE <source>	SEQUENCE|SEGMENT|SAMPLE

***NOTE**. The output trigger in the segment or sequence mode occurs just prior to the start of the last sample and is approximately 70 nsec in width. In the trigger out sample mode, the trigger pulse occurs approximately 3 ms after the sample changes and is 1 ms in length.*

Examples

OUTPut:LEVel <voltage level>

Command	Response/Description
OUTPut:LEVel EXT	Selects an external source for the output voltage.
OUTPut:LEVel?	EXT Queries the output voltage level.

OUTPut:TRIGger:SOURce <source>

Command	Response/Description
OUTP:TRIG:SOUR SEGM	This outputs a trigger for each segment.
OUTP:TRIG:SOUR?	SEGM

STATus? Subsystem

The STATus queries enable you to inquire on the current operational state of the instrument.

Many STATus commands are available for use for all instruments. For a summary of those commands, see the *Status and Events* section.

Command Syntax N/A

Query Syntax `STATus:OPERation:CONDition?`

Command Class Instrument

***RST Value** 0

Limits N/A

Formats Query Response
Numeric

Related Commands MEASure?
READ?
INITiate
ABORt

Query Response The STATus:OPERation:CONDition query returns the current operational status of the Digital Output. The bit definitions of the value are (bit 0 = the least significant bit):

Bit	Definition	Function
0	Calibrating	Set when any CALibration operation is running. Cleared when the CALibration operation is complete
1	Settling	Not used
2	Ranging	Not used
3	Sweeping	Set when the INITiate command is executed. Cleared when the command is complete or aborted
4	Measuring	Not used
5	Triggering	Not used

Bit	Definition	Function
6	Arming	Not used
7	Correcting	Not used
8	Testing (User 1)	Set when the instrument is performing a self-test. Cleared when the self-test is complete
9	Aborting (User 2)	Set when the instrument is in the process of aborting an operation. Cleared when the abort is complete
10	User 3	Not used
11	User 4	Not used
12	User 5	Reserved
13	Instrument Summary	Not used
14	Program Running	Not used
15	Reserved	Always 0

Examples

Command	Response
init:cont on	
status:operation:condition?	8
init:cont off	disables trigger
stat:oper:cond?	0

TEST Subsystem

The TEST subsystem handles the self test operations of the instrument. The Digital Output self test tests the Digital Output memory and the Read/Write hardware control registers. The query return pass/ fail information. In a failed situation, additional failure information can be obtained with the SYStem:ERRor? query.

Command Syntax `N/A`

Query Syntax `TEST:ALL?`

***RST Value** N/A

Limits N/A

Related Commands `*TST?`

Query Response Initiates the Digital Output self test operation and returns one of two possible responses:

"DIGO: Self-Test Passed"

"DIGO: Self-Test Failed"

***NOTE**. If the test fails, you can obtain further information on the failure with the SYST:ERR? query.*

Examples

Command	Response
INSTRUMENT:SELECT DIGO	Select the Digital Output
TEST:ALL?	"DIGO: Self-Test Passed" Run self test.

TRACe Subsystem

The TRACe subsystem commands write output data into memory and specify the output frequency.

Command Syntax

```
TRACe:CLEar

TRACe|DATA[:DATA] <numeric list of indices>{,<ascii_hex_data>}

TRACe|DATA[:DATA] <index>{,ascii_hex_data|indefinite_binary_data}

TRACe:POINts <number_of_points>

TRACe:RFRequency <repeat_frequency>

TRACe:RPERiod <repeat_period>

TRACe:SRATe <sample_rate>
```

Query Syntax

```
TRACe[:DATA]? <numeric_list>

TRACe:INDice?

TRACe:POINts?

TRACe:RFRequency?
```

```
TRACe:RPERiod?

TRACe:SRATe?
```

***RST Value**

Parameter	Value
TRACe:INDice?	0,0
<number_of_points>	1
<repeat_frequency>	48000
<repeat_period>	0
<sample_rate>	48000

Limits

Parameter	Value
<indice>	1-4096
<index>	1-4096
TRACe:INDice?	$1 \leq$ segment length $\leq$ 4096 $1 \leq$ sample index $\leq$ 4096
<number_of_points>	1 - 4096
<repeat_frequency>	0.0625 - 48000.0 and greater than: <sample_rate> × <number_of_points>
<repeat_period>	2.0833e–5 – 16.0 and less than: <number_of_points>/<sample_rate>
<sample_rate>	3.622 - 48000

Related Commands

N/A

Command Description

TRACe:CLEar
Clears the memory buffer of all previous data and sets all bits to #hFFFFFFFF. This setting causes the Digital Output to send the discrete output bits High.

If the Digital Output is currently sending data, this command forces an implied abort, halting instrument operation. The memory buffer is initialized to its default state of #hFFFFFFFF.

The CLEAr command does not alter the points register.

TRACe|DATA[:DATA] <indices>{,<ascii_hex_data>}
TRACe|DATA[:DATA>] <index>{,ascii_hex_data|indefinite_binary_data>}
Fills the output memory buffer with data_values from starting memory location memory_index.

***NOTE**. The number of data values following the memory index must not exceed the number of memory locations (for example, a command argument of 4093 + 5 data_values specifies location 4097 which is not permitted).*

If the <index> is a numeric list of indices rather than a single index, then the memory locations indicated in the numeric list will be filled with the single specified data value.

If the index is a single value, then the memory locations starting at the index will be filled with successive values in the data list. The values may be in ASCII hex or indefinite binary format. For more information, consult the IEEE Standard, Section 7.7.6.

There are 4K words of memory available for data_values. To query several points from memory, enter the number of points in a numeric list.

If this command is sent while the instrument is sending data, or sweeping, then the instrument will automatically perform an implied abort and implement the trace command. The instrument must be initiated again to start sending data again.

If either parameter is omitted, the instrument generates an error.

The query will cause an implied abort if you enter it while the Digital Output is generating data.

TRACe:POINts <number_of_points>

Sets the number of trace data points in a segment and the number of data points returned in a query. The POINts command does not change the contents of memory either within or outside the number of points specified.

This command is used to define the number of sample points that comprise an output segment. Each segment can be comprised of up to 4096 sample points for each of the 32 bits. If the <number_of_points> field is omitted, or is out of limits, an error will be generated.

There is a second limitation on the trace buffer points. The number of points * the sample period (1/sample rate) must be less than 16 seconds. If set to be greater than this limit, the instrument will return an error upon INITiate.

TRACe:RFRequency <repeat_frequency>

Sets the repeat frequency, in Hz, of a trace pattern. The default value is Off (disabled).

The repeat frequency must be less than:

Sample Frequency x Number of Output Trace Points
or
<sample_rate> x <number_of_points>

TRACe:RPERiod <repeat_period>
Sets the repeat period of a trace pattern in seconds. The commands RFRequency and RPERiod program the same parameter. The RPERiod command is the reciprocal of the RFRequency command.

The repeat period must be greater than:

Sample Period * Number of Output Trace Points
or
<number_of_points>/<sample_rate>

TRACe:SRATe <sample_rate>
The sample (frequency) rate, in Hz, of a trace pattern. The sample rate times the number of data points must be greater or equal to the repeat frequency.

***NOTE**. The VX4101A will round the TRACe:RFRequency, TRACe:RPER and TRACe:SRATe values off to the nearest values supported by the VX4101A hardware. The rates are rounded to 240000/n, where n is an integer. The period is rounded to n/240000 where n is an integer. The queries for these commands will return actual programmed values.*

***NOTE**. The number of points multiplied by the sample period (or 1/sample rate) must be less than 16 seconds. If the number of points is more than 16 seconds, the instrument returns an error.*

Query Response

Query	Response
TRACe|DATA[:DATA]? <numeric_list>	<index>,{<numeric value>}
TRACe:INDice?	<segment_index>, <sample_index> of output memory buffer. You can also use RPERiod or RFRequency to create gaps between segments. During a gap, the Digital Output continues to output the final sample point, so this query will return the most recent segment output and final sample point.

Query	Response
NOTE: *For high sample traces, the <sample index> data might be old. You can use TRACe:INDice? query to determine the approximate sample index.* *After completing the output of a sequence, this query returns the final segment and final sample since the Digital Output will retain the data in the buffer until the instrument is reset or initiated.*	
TRACe:POINts?	<number_of_points>
TRACe:RFRequency?	<repetition frequency>
TRACe:RPERiod?	<repetition frequency>
TRACe:SRATe?	<sample rate>

Examples

TRACe:CLEar

Command	Response/Description
TRACe:CLEar	Clears the output memory buffer to #hFFFFFFFF

TRACe [:DATA]<numeric_list>{,<data_values>}

Command	Response/Description
TRACe:DATA 1 #hf0f0f0f0,#h0f0f0f0f, etc.	Fills the output memory buffer with data
TRACe:POINts 2	Sets the trace length to 2
TRACe:DATA? (1:2)	#hf0f0f0f0, #h0f0f0f0f Queries trace data
TRACe:DATA (1:4096), #ha0a0f0f0	Fills all 4096 memory locations with #ha0a0f0f0.
TRACe:DATA? (1:2)	#ha0a0f0f0 Queries trace data.

TRACe:INDice?

Command	Response/Description
*RST	Resets the entire card module.
TRACe:INDice?	0,0 Queries current output segment and sample index

TRACe:POINts <number_of_points>

Command	Response/Description
TRACe:POINts 11	Sets the number of trace data points to 11
TRACe:POINts?	11 Queries the number of trace data points.

TRACe:RFRequency <repeat_frequency>

Command	Response/Description
TRACe:RFRequency 10000.0	Sets the repeat frequency to 10 kHz
TRACe:RFRequency?	Queries the repeat frequency. 10000.0

TRACe:RPERiod <repeat_period>

Command	Response/Description
TRACe:RPERiod 0.001	Sets the repeat period to 1 ms
TRACe:RPERiod?	0.001 Queries the repeat period.

TRACe:SRATe<sample_rate>

Command	Response/Description
TRACe:SRATe 48000.0	Sets the sample rate to 48 kHz.
TRACe:SRATe?	48000.0 Queries the sample rate.

TRIGger Subsystem

Command Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time_delay>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event_count>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest <polarity>
   :STRobe <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
```

Query Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest?
   :STRobe?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

Command Class Instrument

***RST Value**

Parameter	Values
<time_delay>	0 seconds (pass-through)
<event_delay	0 triggers (pass-through)
<polarity>	INVerted
<mode>	SEQuence,1
<source>	IMMEDIATE

Limits

Parameter	Values
<time_delay	1E+06 to 65.535E–03 seconds in 1E–06 steps
<event_delay>	0 = pass through, 1 to 65,535 triggers
<polarity>	NORMal | INVerted
<mode>	SAMPle|SEGMent|(SEQuence, <sequence count>)
<source>	BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

Related Commands

N/A

Command Description

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay <time_delay>
Specifies a time delay to occur after receipt of a trigger prior to actually triggering. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the event count delay, so specifying a delay of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt <event_count>
Specifies the number of triggers to count prior to triggering. Upon receipt of trigger N (where N is the number specified in the command), the instrument will enter the triggered state. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the delay by time parameter, so specifying an event count of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity:REQuest|STRobe <polarity>
Specifies the active edge of the signals to be received and driven on the external handshake lines. NORMal indicates that the rising edge is active. INVerted indicates that the falling edge is active.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate
Causes a one time entry into the triggered state without receiving the specified trigger. This command is often used to "prime the pump" in cases such as setting up a scan list measurement.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:MODE <mode>
After receipt of an initiate, if the <mode> is SAMPle, then on receipt of a trigger,

the next location in the active trace will be output by the instrument and the location in the trace will be incremented. Each subsequent trigger will cause the instrument to output a new value.

If the <mode> is SEGMent, the number of points specified with the TRACe:POINts command will be output at the rate specified by the TRACe:SRATe command. Each subsequent trigger will cause the instrument to re-output the segment.

If the <mode> is SEQuence, the number of points specified with the TRACe:POINts command will be output at the rate specified by the TRACe:SRATe command. This will be repeated for the number of times specified in the sequence count specified when the mode was set. The spacing between each group of points will be such that the repeat period specified in the TRACe:RPERiod will be met. Each subsequent trigger will cause the instrument to re-output the entire sequence.

The instrument will remain in the initiated state, driving the output until an abort or a reset occurs.

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce <source>
Selects the trigger source to be used when the instrument is initiated.

Query Response

Query	Response
TRIGger([:SEQuence1] |:STARt)[:LAYer]:DELay?	<delay in seconds>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?	<triggers to count>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?	SAMPle|SEGMENT 1 SEQuence,<sequence count>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity:REQuest|STRobe?	NORMal | INVerted
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?	<current source>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?	Lists all trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

NOTE: *Depending upon which options were purchased with the VX4101A, some of the trigger sources listed above may not be available for use.*

Query	Response
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?	Lists all delayable trigger sources available for use with the TRIG:SOUR command. BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?	Lists all fixed trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, TIMER, HANDSHAKE

Examples

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay

Command	Response
INST:SEL DIGO	
TRIG:SOUR TTLT0	Use TTLT0 trigger
TRIG:DEL 1E-3	Sets trigger delay to 1 ms
INIT CONT ON	

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt

Command	Response
INS:SEL DIGO	
TRIG:SOUR TTLT0	Uses TTLT0 trigger source
TRIG:ECO 100	
INIT CONT ON	Triggers instrument after 100 TTLT0 triggers

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate

Command	Response
INST:SEL DIGO	
TRIG:SOUR:CAT?	IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
TRIG:SOUR HOLD	
TRIG:MODE SEQ,2	

Command	Response
INIT	
TRIG:IMM	Outputs one sequence of two segments

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE

Command	Response
INST:SEL DIGO	
TRIG:MODE SAMP	
TRIG:SOUR COMM0	
INIT	
TRIG:FIR0	Outputs one sample
TRIG:FIR0	Outputs one sample
TRIG:FIR0	Outputs one sample

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce

Command	Response
INST:SEL DIGO	
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
TRIG:MODE SEQ 2	
TRIG:SOUR BUS	
INIT	
*TRG	Outputs one sequence of two segments

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

Command	Response
INST:SEL DIGO	
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

Command	Response
INST:SEL DIGO	
TRIG:SOUR:CAT:DEL?	BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, DAC, DIGI, DIGO, CTR_EXTARM

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Command	Response
INST:SEL:DIGO	
TRIG:SOUR:CAT:FIX?	HOLD,IMMEDIATE,TIMER

SCPI Commands for the Digital to Analog Converter

This section includes a summary of SCPI commands you can use to control the Digital to Analog Converter (DAC) of the VX4101A MultiPaq™ Instrument. The commands enable you to calibrate the instrument, define the format for your input or output, prepare and initiate waveform generation, and query the instrument's current status.

Command Summary

The command syntax for the DAC is as follows:

CALibrate Subsystem

CALibrate[<channel>]:OUTput <channel >

CALibrate[<channel>]:VALue <channel>,<DVM reading>

CALibrate[<channel>]:SAVe

FORMat Subsystem

Commands
FORMat[:DATA] <format>

Queries
FORMat[:DATA]?

INITiate Subsystem

Commands
INITiate

:CONTinuous [0|OFF|1|ON]

[:IMMediate]

Queries
INITiate:CONTinuous?

INSTrument Subsystem

Commands
INSTrument:ABORt([:IMMediate]|:COMPlete)
INSTrument:RESet

OUTPut Subsystem

Commands
OUTPut:TRIGger:SOURce <trigger source>

Queries
OUTPut:TRIGger:SOURce?

STATus? Subsystem

STATus:OPERation:CONDition?

TEST Subsystem

TEST:ALL?

TRACe Subsystem

Commands
TRACe:Clear
TRACe|DATA[:DATA]

<Channel>,<index>{,ASCII Trace Voltages|BINary Indefinite Block Trace Voltages>}
<Channel>,<Numeric List of Indices>{,<ASCII Trace Voltages>}

TRACe:POINts <Trace Buffer Points>
TRACe:RFRequency <repeat frequency|OFF>
TRACe:RPERiod <repeat period|OFF>
TRACe:SRATe <sample_rate>

Queries
TRACE|DATA[:DATA]? <Channel Select>|<Numeric List of Indices>
TRACE|DATA[:DATA]? <Channel Select>

TRACe:INDice?
TRACe:POINts?
TRACe:RFRequency?
TRACe:RPERiod?
TRACe:SRATe?

TRIGger Subsystem

Commands
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event count>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

:REQuest <polarity>
:STRobe <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>

Queries
TRIGger:COUNt?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity
    :REQuest?
    :STRobe?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

VXI:FDC Subsystem

Commands

```
VXI[:SERVant]:FDC:CLOSe [<channel number>]
VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]
VXI[:SERVant]:FDC:SEL <channel number>
VXI[:SERVant]:FDC:BUFFer <buffer length>[,<channel number>]
```

Queries

```
VXI[:SERVant]:FDC?
VXI[:SERVant]:FDC:CONFiguration?[<channel number>]
VXI[:SERVant]:FDC:SEL?
VXI[:SERVant]:FDC:BUFFer?[channel number]
```

CALibrate Subsystem

You use these commands to calibrate the DAC. For more information on calibration, see *Calibration for the Digital to Analog Converter* in *Appendix G: Calibration*.

Command Syntax

```
CALibrate[<channel>]:OUTput <voltage>
CALibrate[<channel>]:VALue <channel>,<DVM reading>
CALibrate[<channel>]:SAVE
```

Query Syntax

N/A

*RST Value

Parameter	Value
<channel>	1
<DVM reading>	0

Limits

Parameter	Value
<channel>	1-8
<DVM reading>	Offset 0 Gain 8

Related Commands N/A

Command Description

CALibrate[<channel>]:OUTput <voltage>
Sets the DAC output for the selected channel at the specified nominal voltage. Enter a different <channel> to calibrate output for a different channel.

CALibrate[<channel>]:VALue <channel>,<DVM reading>
Sets the DAC for the selected channel at the most recent voltage reading from the digital voltmeter. Use this command to refine the voltage setting for the channel to an increasingly accurate tolerance.

For adjusting offset, you will repeat this command a total of six times.

For adjusting gain, you will repeat this command a total of eight times.

CALibrate[<channel>]:SAVE
Saves the best values found in the offset and gain adjustment. This command loads the values in EEPROM.

FORMat Subsystem

The FORMat subsystem specifies the format of command and response data.

Command Syntax `FORMat[:DATA] <format>`

Query Syntax `FORMat[:DATA]?`

***RST Value**

Parameter	Value
<format>	ASCII

Limits

Parameter	Value
<format>	ASCII|BINary

Related Commands
TRACe

Command Description
The command specifies how data is returned from a waveform (TRACe subsystem) voltage value query. Data can be returned in ASCII or binary format.

Query Response
The query returns the currently selected response data format.

Examples

Command	Response/Description
FORMat BINary	N/A Specifies the use of binary when returning trace voltage values. The binary format of the return trace data is based upon the IEEE 488.2 Indefinite Arbitrary Block Program Data type. The output amplitude = (x–2048) * 8 mV where x=256*MSByte+LSByte.

INITiate Subsystem

These commands cause the current TRIGger command sequence to begin.

Command Syntax

```
INITiate
    :CONTinuous [0|OFF|1|ON]
    [:IMMediate]
```

Query Syntax
`INITiate:CONTinuous?`

Command Class
Instrument

***RST Value**
N/A

Limits
0, 1, ON, or OFF

Related Commands
ABORt
INSTrument:ABORt

Description
INITiate:CONTinuous [0|OFF|1|ON]
This command initiates the current programmed trigger sequence. After the instrument has completed the current trigger sequence, it enters the initiated state

a second time. It will continue this cycle until an abort, reset, or INIT:CONT OFF command is received. This command applies to all trigger sources.

INITiate[:IMMediate]
This command initiates the current programmed trigger sequence. After the instrument has completed the current trigger sequence, it enters the idle state. See instrument documentation for details on instrument state after an INITiate. This command is applicable only to an IMMediate trigger source. INIT:IMMediate is functional only with the TRIG:SOURCE IMMediate command.

This command applies to TRIG:SOURCe:IMMEDiate only. This command returns an error for all other trigger sources.

***NOTE**. For the DAC and Digital Output, if you select a trigger source other than IMMediate, the INItiate:IMMediate command will not work. Either instrument will continue to accept triggers and generate data until one instrument receives one of the following commands:ABORt, INSTrument:ABORt, an implied abort, or INITiate:CONTinuous OFF. Consult the summaries for each of these commands for more information.*

Query Response

Command	Response
INITiate:CONTinuous [0|OFF|1|ON]	0 = continuous initiate not enabled 1 = continuous initiate enabled
INITiate:IMMediate	N/A

Examples See TRIGger([: SEQuence1] |:STARt) [:LAYer]:IMMediate command for an example.

INSTrument Subsystem

The INSTrument subsystem controls the operational state of the DAC.

Command Syntax

```
INSTrument:ABORt([:IMMediate]|COMPlete)

INSTrument:RESet
```

Query Syntax N/A

Limits N/A

***RST Value** N/A

Limits N/A

Related Commands ABORt:IMMEDiate
ABORt:COMPlete

Command Description

INSTrument:ABORt([:IMMediate]|COMPlete)
This command places the DAC in the idle state, and terminates any activities that are currently in progress. The instrument configuration is unchanged and a subsequent INIT command will cause the DAC to re-start its previous activity. After the activity completes, the DAC remains selected. INST:ABORt:COMPlete terminates activity after the present sequence is completed. INST:ABORt:IMMediate terminates operation on the next sample.

INSTrument:RESet
This command resets the DAC to its power-on state and sets the instrument to its default conditions. After execution, the DAC is remains selected. After reset, the DAC memory is initialized to zero; the sample rate and points are at their default values; each output is set to 0 V; the indice are 0,0; repeat period is set to off (i.e. no waveform segment gaps). The output will remain at 0V and the indice at 0,0 until the TRACe:DATA command is used to load memory and the instrument in initiated. The default trigger state is internal.

***RST Value** ABORt:IMMediate
ABORt:COMPlete

Examples

INSTrument:ABORt

Command	Response/Description
INST:SEL DAC	Selects the DAC
INIT:CONT ON	Initiates the DAC instrument
INST:ABOR	Aborts the DAC activity
INST:SEL?	DAC Queries for the selected instrument

INSTrument:RESet

Command	Response/Description
INST:SEL DAC	Selects the DAC
INIT:CONT ON	Initiates the DAC instrument
INST:RES	Resets the DAC to default state
INST:SEL?	Query returns DAC

OUTPUT Subsystem

The commands in this subsystem select when the DAC will generate an output trigger.

Command Syntax

`OUTPut:TRIGger:SOURce <trigger source>`

Query Syntax

`OUTPut:TRIGger:SOURce?`

***RST Value**

Parameter	Value
<trigger source>	SEQuence

Limits

Parameter	Value
<trigger source>	SEQuence|SEGMent|SAMPle

Resolution

N/A

Related Commands

N/A

Description

This command selects when the output trigger for the DAC will occur. It can be set to trigger every sample period, every waveform segment period, or once per waveform sequence.

Sample. A single DAC output point. At the end of a sample, the DAC output amplitude remains at the last sample value until a new sample is initiated. Use the command TRIG:MODE SAMPLE to program sample generation.

Segment. A group of DAC points that generate a waveform. The maximum number of points in a waveform is 1024. You use the TRAC:POIN <number of

points> command to enter the number of points in a segment. At the end of a segment, the DAC output amplitude remains at the last segment value until you initiate a new segment. You can initiate a new segment continuously, by the repetition frequency, or by a trigger. Use the command TRIG:MODE SEGMENT to program segment generation.

Sequence. A programmed repetition of segments. The maximum number of segments is 4096. At the end of a sequence, the DAC output amplitude remains at the last sequence value until a new sequence is triggered. You use the TRIG:MODE SEQ,<sequence length> command to define the number of segments. You can initiate a new sequence either by a trigger, or continuously.

Waveform Period. The waveform period is defined as the number of points in the waveform times the sample period.

If the waveform repetition mechanism is enabled, the repetition period must be greater than or equal to the waveform period. Calculate the waveform period as follows:

*Sample Period * Number of Points*

Query Response

SEQUENCE|SEGMENT|SAMPLE

***NOTE**. The output trigger in the segment and sequence mode occurs just prior to the start of the last sample and approximately 70 ns in width. The output trigger in the sample mode occurs about 6 ms after the new sample is strobed and is 2 ms in width.*

Examples

Command	Response
OUTP:TRIG:SOUR SEGM	Outputs a trigger from the DAC for each segment.

STATus? Subsystem

The STATus queries enable you to inquire on the current operational state of the instrument.

Many STATus commands are available for use for all instruments. For a summary of those commands, see the *Status and Events* section.

Command Syntax N/A

Query Syntax `STATus:OPERation:CONDition?`

Command Class Instrument

Query Response The operational condition register value

***RST Value** 0

Limits N/A

Formats Query Response
Numeric

Related Commands MEASure?
READ?
INITiate
ABORt

Description The STATus:OPERation:CONDition query returns the current operational status of the Digital Input. The bit definitions of the value are (bit 0 = the least significant bit):

Bit	Definition	Function
0	Calibrating	Set when any CALibration operation is running. Cleared when the CALibration operation is complete
1	Settling	Not used
2	Ranging	Not used
3	Sweeping	Set when the INITiate command is executed. Cleared when the command is complete or aborted

Bit	Definition	Function
4	Measuring	Not used
5	Triggering	Not used
6	Arming	Not used
7	Correcting	Not used
8	Testing (User 1)	Set when the instrument is performing a self-test. Cleared when the self-test is complete
9	Aborting (User 2)	Set when the instrument is in the process of aborting an operation. Cleared when the abort is complete
10	User 3	Not used
11	User 4	Not used
12	User 5	Reserved
13	Instrument Summary	Not used
14	Program Running	Not used
15	Reserved	Always 0

Examples

Command	Response
status:operation:condition?	16 Measurement in progress (0010 hex).
stat:oper:cond?	3072 Measurement complete because an ABORt was received (0C00 hex).

TEST Subsystem

The TEST subsystem handles the self test operations of the instrument.

Command Syntax N/A

Query Syntax `TEST:ALL?`

***RST Value** N/A.

Limits N/A

Related Commands `*TST?`

Command Description

Initiates the DAC self test operation and returns one of two possible responses:

"DAC: Self Test Passed"
"DAC: Self Test Failed"

Query Response

The query returns basic pass/fail information.

***NOTE**. In a failed situation, you can get more detailed failure information with the SYStem:ERRor? query.*

Examples

Command	Response/Description
TEST:ALL?	"DAC: Self-Test Passed" "DAC: Self-Test Failed" Initiates the DAC self test operation and returns either a pass or failure message.

TRACe Subsystem

The TRACe: and TRACe|DATA commands work with the arbitrary waveform generator capabilities of the DAC. The TRACe|DATA commands can be used with either the TRACe or DATA command

***NOTE**. For the TRACe|DATA commands, you can preface the command with either TRACe or DATa*

Command Syntax

```
TRACe:Clear

TRACe|DATA[:DATA]

   <Channel>,<index>{,ASCII Trace Voltages|BINary Indefinite
   Block Trace Voltages>}
   <Channel>,<Numeric List of Indices>{,<ASCII Trace Voltages>}

TRACe:POINts <Trace Buffer Points>

TRACe:RFRequency <repeat frequency|OFF>

TRACe:RPERiod <repeat period|OFF>

TRACe:SRATe <sample_rate>
```

Query Syntax

```
TRACE|DATA[:DATA]? <Channel Select>|<Numeric List of Indices>
TRACE|DATA[:DATA]? <Channel Select>

TRACe:INDice?

TRACe:POINts?

TRACe:RFRequency?

TRACe:RPERiod?

TRACe:SRATe?
```

Related Commands

FORMat

***RST Value**

Parameter	Value
<trace voltages>	0.0
TRACe:INDice?	0,0
***NOTE:** At reset, the DAC clears its memory and returns a segment index of 0. The DAC will not perform an output operation and remains at the 0,0 indice until you send a TRACe|DATA command and the instrument is initiated.*	
<Trace Buffer Points>	1
<repeat frequency>	OFF
<repeat period>	OFF
<sample_rate>	15000 Hz

Limits

Parameter	Limit
<Channel>,<Channel Select>	1-8
<Index>	1 ≤ index ≤ 1024
<Trace Voltage>	–14.0 ≤ voltage value ≤ + 14.0
TRACe:INDice?	1 ≤ sement ≤ 4096 1 ≤ sample index ≤ 1024
<Trace Buffer Points>	1 ≤ Trace Buffer Points ≤ 1024 and (Trace Buffer Points/Sample Rate) ≤ 16 seconds
<repeat frequency>	.0625 Hz ≤ Waveform Frequency ≤ 15000 Hz or OFF
<repeat period>	66.67E–6 seconds ≤ Waveform period ≤ 16 seconds, or OFF
<sample_rate>	3.662 Hz ≤ Sample Rate ≤ 15000 Hz and (Trace Buffer Points/Sample Rate) ≤ 16 seconds

Resolution

Parameter	Resolution
<ASCII Trace Voltages>	8 mV
<Channel> <Numeric List of Indices> <ASCII Trace Voltages>}	8 mV
<repeat frequency>	240 E3/n where 16 ≤ n ≤ 3,840,000
<repeat period>	n/240 E3 where 16 ≤ n ≤ 3,840,000
<sample_rate>	240 E3/n where 16 ≤ n ≤ 2^16

Related Commands `FORMat`

Command Description The TRACe: and TRACe|DATA commands perform the following operations:

TRACe:CLEAR
This command zeroes the DAC as follows:

- Sets the output array for each of the 8 DAC channels to zero.
- Sets all 1024 samples of each output array is set to zero.
- Sets all DAC output points to zero.

If the DAC is currently producing output, it stops immediately upon receiving this command. The instrument initializes the array as in the power-on sequence and returns to its default state. The Points register is not affected by the Clear command.

TRACe | DATA [:DATA] <Channel >,<Index>{,<ASCII Trace Voltages|BINary Indefinite Block Trace Voltages}
This command enables you to use the arbitrary waveform generator capabilities of the DAC. You can enter the command with either the TRACe or DATA prefix. The command loads the selected DAC channel with the list of sample voltage values. The command programs the DAC with ASCII or binary data.

You can use the parameters provided with the command, as follows:

If either the Channel Select or the Array Index integer parameters are omitted, an error is generated.

If this command is sent while the instrument is generating a waveform (e.g. Sweeping), then the instrument will automatically perform an implied abort and execute the TRACe command. The instrument must be re-initiated to start generating a waveform again.

TRACe|DATA[:DATA] <Channel>,<Numeric List of Indices>{,<ASCII Trace Voltages>

This command enables you to use the arbitrary waveform generator capabilities of the DAC. All commands use either the TRACe or DATA prefix. The commands load the selected DAC channel with the list of sample voltage values. The command programs the DAC to use ASCII data.

You can use the parameters provided with the command, as follows:

If either the Channel Select or the Array Index integer parameters are omitted, an error is generated.

If this command is sent while the instrument is generating a waveform (e.g. Sweeping), then the instrument will automatically perform and implied abort and implement the trace command. The instrument must be re-initiated to start generating a waveform again.

TRACe:POINts <Trace Buffer Points>

This command defines the number of sample points that comprise a segment. Each segment can have a maximum of 1024 data points (2048 byes). The points register is shared between each of the 8 DAC channels. Consequently, the Points command specifies the number of data points for each of the DAC waveforms. There is only one data point count register for all DAC channels. If the <Trace Buffer Points> field is omitted or is out of limits, the instrument generates an error.

***NOTE**. The number of points multiplied by the sample period (or 1/sample rate) must be less than 16 seconds. If the number of points more than 16 seconds, the instrument returns an error upon an INITiate command.*

TRACe:RFRequency <repeat frequency|OFF>

This command defines the repetition frequency of each waveform segment. It is the inverse of the TRACe:RPERiod. It defines the inverse of the waveform repeat period. For more information, see the TRACe:RPERiod command description.

Default units are in Hz. If the repeat period is set to OFF, the instrument will not allow any gaps, i.e. the repeat period will be calculated to equal the waveform period. Programming a repetition frequency is illegal if the trigger mode is SAMPle. An error will be generated on INITiate if the repetition frequency is enabled while the trigger mode is SAMPle.

TRACe:RPERiod <repeat period|OFF>

This command defines the repeat period of each waveform segment. This is the inverse of the TRACe:RFREQency command.

***NOTE**. The waveform segment period is the sample period times the number of data points. Thus, the waveform repeat period must be greater than or equal to the waveform segment period.*

If the waveform repeat period is set to be greater than the waveform segment period, then there will be a gap between waveform segments (the final point in the segment will be output until the next segment is started). If the waveform repeat period is equal to the waveform segment period, then there will be no gaps between the last point of a segment and the first point of the next segment. If the waveform repeat period is set to less than the waveform segment period, an error will be generated when INITiate is issued.

Default units are in seconds. If the repeat period is set to OFF, the repetition mechanism is disabled and no gap will begenerated. Programming a repetition frequency is illegal if the trigger mode is SAMPle. An error will be generated on an INITiate if the repetition frequency is enabled while the trigger mode is SAMPle.

TRACe:SRATe <sample_rate>

This command assigns the sample rate of the waveform data point. The value is in Hz. This is the frequency at which the waveform data points are output to the DAC output ports. Since there is only one sample clock and a single points register for all DAC channels, the sample rate and waveform frequency is the same for all 8 channels. The sample rate can range from 3.662 Hz to 15e3 Hz. The actual sample rate will be rounded to the closest value. Query responses will return the quantized value.

Changing the output frequency during the output of a waveform (e.g. sweeping), will result in an implied abort. The instrument accepts the change in frequency and go into a wait state until it is re-initiated.

***NOTE**. The number of points multiplied by the sample period (or 1/sample rate) must be less than 16 seconds. If the number of points is more than 16 seconds, the instrument returns an error.*

Query Response

The TRACe: subsystem queries have the following responses:

TRACE | DATA [:DATA]? <Channel Select>,<Numeric List of Indices>
TRACE | DATA [:DATA]? <Channel Select>
There are two forms for this query as shown above. In either case, the return data can be either ASCII decimal or binary data as defined by the FORMAT subsystem command. If either query is issued during operation, the device performs an applied abort and returns the query data.

The first query returns the array element values for the selected DAC channel. The Numeric List of Indices specifies which sample points to return for the selected DAC channel.

The second query above returns all of the values for the selected DAC channel.

TRACe:INDice?
Returns the current waveform segment index and sample index of the output memory buffer.

***NOTE**. The query response time might make this command impractical at high sample rates. For slow sample rates, you can use this query to determine the approximate position of the actual waveform.*

After completing the output of a waveform sequence, this query will return the final segment and final sample since the DAC will continue to hold that output until reset or initiated again.

RPERiod or RFREquency can be used to create gaps between waveform segments. During the gap, segment and sample numbers do not change.

TRACe:POINts?
This query returns the current number of active sample points in the DAC trace buffers.

TRACe:RFRequency?
Returns the frequency of the output cycle, in Hz.

TRACe:RPERiod?
Returns the period of the output cycle, in seconds.

***NOTE**. The VX4101A will round the TRACe:RFRequency, TRACe:RPER and TRACe:SRATe values off to the nearest values supported by the VX4101A hardware. The rates are rounded to 240000/n, where n is an integer. The period is rounded to n/240000 where n is an integer. The queries for these commands will return actual programmed values.*

***NOTE**. The number of points multiplied by the sample period (or 1/sample rate) must be less than 16 seconds. If the number of points is more than 16 seconds, the instrument returns an error.*

TRACe:SRATe?
Returns the current waveform data sample rate.

Examples

TRACe:CLEAR

Command	Response
TRAC:CLE	Zeroes all samples for all eight DAC channels.

TRACe | DATA [:DATA] <Channel >,<Index>{<ASCII Trace Voltages|BINary Indefinite Block Trace Voltages}
TRACe|DATA[:DATA] <Channel>,<Numeric List of Indices>{,<ASCII Trace Voltages>

Command	Response
DATA 1,10,1.0,2.0,3.0, 4.0,5.0	Fills the DAC channel one with five data values, 1-5, starting at index 10 Other memory locations are not modified.
TRAC 2,5,6.0,7.0	Fills DAC channel two with 2 values, 6 & 7, starting at index 5. The others values in the DAC are not modified.
TRAC 8, (1:10,64, 235:418), 5.0	Fills the following DAC channel 8 memory locations 1 through 10, location 64 and locations 235 through 418 all with 5.0 volts. Other memory locations are not modified
DATA? 1,(1:1000)	Request the voltages values in DAC 1 from index 1 through location 1000
TRAC 3,1 (#xxxxxxxxxxxxxxxx<new-line><endbit>)	Fills DAC channel three in memory locations 1-8 with binary data. Where xxxxxxxxxxxxxxxx is binary data.

TRACe:INDice?

Command	Response/Description
*RST	N/A Resets the output trace index
TRAC:IND?	1,1 Returns the output trace segment and sample index

TRACe:POINts <Trace Buffer Points>

Command	Response/Description
TRACE:POINTS 100	N/A Defines the trace buffer size to be 100 points for each DAC
TRAC:POIN?	100<LF> The allocated number of points in each DAC channel trace buffer.
trace:points 1.02e3	N/A Defines each DAC channel trace buffer size to 1020 points.

TRACe:RFRequency <repeat frequency\|OFF>

Command	Response/Description
TRACE:RFR 10e3	N/A Assigns a waveform frequency of 10 kHz.
TRAC:RFR?	10E3

TRACe:RPERiod <repeat period\|OFF>

Command	Response/Description
TRAC:RPER 100e–6	Assigns a waveform period of 100 μsec.
TRAC:RPER?	100E–6

TRACe:SRATe

Command	Response/Description
TRACe:SRATe 10e3	Programs the DAC hardware to output the waveforn data points at a 10 kHz rate.
TRAC:SRAT?	10E3 Queries for the sample rate

Trigger Subsystem

The TRIGger subsystem commands control the DAC trigger operation. The DAC supports trigger commands that initiate continuous DAC output operation.

Command Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event count>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest <polarity>
   :STRobe <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
```

Query Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest?
   :STRobe?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

Command Class Instrument

***RST Value**

Parameter	Values
<time delay>	0 seconds (pass-through)
<event delay>	0 triggers (pass-through)
<polarity>	INVerted
<mode>	SEQuence, 1
<source>	IMMEDIATE

Limits

Parameter	Values
<time delay>	0 seconds
<event count>	0 = pass through, 1 to 65,535 triggers in 1 μs steps
<polarity>	NORMal | INVerted
<mode>	SAMPle|SEGMent|(SEQuence, <sequence count>)
<source>	BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

Related Commands

`INITiate:IMMediate`

Command Description

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay <time delay>
Specifies a time delay to occur after receipt of a trigger prior to actually triggering. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the event count delay, so specifying a delay of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt <event count>
Specifies the number of triggers to count prior to triggering. Upon receipt of trigger N (where N is the number specified in the command), the instrument will enter the triggered state. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the delay by time parameter, so specifying an event count of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity:REQuest|:STRobe <polarity>
Specifies the active edge of the signals to be received and driven on the external handshake lines. NORMal indicates that the rising edge is active. INVerted indicates that the falling edge is active.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate
Causes a one time entry into the triggered state without receiving the specified trigger. This command is often used to "prime the pump" in cases such as setting up a scan list measurement.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:MODE <mode>
After receipt of an initiate, if the <mode> is SAMPle, then on receipt of a trigger,

the next location in the active trace will be output by the instrument and the location in the trace will be incremented. Each subsequent trigger will cause the instrument to output a new value.

If the <mode> is SEGMent, the number of points specified with the TRACe:POINts command will be output at the rate specified by the TRACe:SRATe command. Each subsequent trigger will cause the instrument to re-output the segment.

If the <mode> is SEQuence, the number of points specified with the TRACe:POINts command will be output at the rate specified by the TRACe:SRATe command. This will be repeated for the number of times specified in the sequence count specified when the mode was set. The spacing between each group of points will be such that the repeat period specified in the TRACe:RPERiod will be met. Each subsequent trigger will cause the instrument to re-output the entire sequence.

The instrument will remain in the initiated state, driving the output until an abort or a reset occurs.

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce <source>
Selects the trigger source to be used when the instrument is initiated.

Query Response

Query	Response
TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay?	<delay in seconds>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?	<triggers to count>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?	SAMPle|SEGMENT|SEQuence,<sequence count>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity:REQuest|STRobe?	NORMal | INVerted
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?	<current source>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?	Lists all trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

NOTE: *Depending upon which options were purchased with the VX4101A, some of the trigger sources listed above may not be available for use.*

Query	Response
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?	Lists all delayable trigger sources available for use with the TRIG:SOUR command. BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?	Lists all fixed trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, TIMER, HANDSHAKE

Examples

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay

Command	Response
INST:SEL DAC	
TRIG:SOUR TTLT0	
TRIG:DEL 1E-3	
INIT CONT ON	TTLT0 is delayed by 1 ms.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt

Command	Response
INS:SEL DAC	
TRIG:SOUR TTLT0	
TRIG:ECO 100	
INIT CONT ON	The DAC is triggered once per 100 TTLT0 triggers

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate

Command	Response
TRIG:SOUR:CAT?	IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
INST:SEL DAC	
TRIG:SOUR TTLT0	Selects TTLT0 trigger source
INIT	
TRIG:IMM	Triggers the instrument using software command

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE

Command	Response
INST:SEL DAC	
TRIG:MODE SAMP	Selects the sample mode
TRIG:SOUR COMM0	Selects the COMM0 trigger source
INIT	
FIR0	Issues trigger

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce

Command	Response
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
INST:SEL DAC	
TRIG:SOUR BUS	
INIT:CONT ON	
*TRG	Initiates the trigger

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

Command	Response
INST:SEL DAC	
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

Command	Response
TRIG:SOUR:CAT:DEL?	BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, DAC, DIGI, DIGO, CTR_EXTARM

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Command	Response
INST:SEL:DAC	
TRIG:SOUR:CAT:FIX?	HOLD,IMMEDIATE,TIMER

VXI:FDC Subsystem

This subsystem contains the command set that allows the setup and use of the DAC's Fast Data Channel.

Command Syntax

```
VXI[:SERVant]:FDC:CLOSe [<channel number>]

VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]

VXI[:SERVant]:FDC:SEL <channel number>

VXI[:SERVant]:FDC:BUFFer <buffer length>[,<channel number>]
```

Query Syntax

```
VXI[:SERVant]:FDC?

VXI[:SERVant]:FDC:CONFiguration? [channel number]

VXI[:SERVant]:FDC:SEL?

VXI[:SERVant]:FDC:BUFFer?
```

*RST Value

Parameter	Value
<buffer length>	0
<channel number>	1
<channel mode>	N/A

Limits

Parameter	Value
<buffer length>	$1 \leq$ buffer length $\leq$ 1024
<channel number>	1– total number of instrument FDC channels
<channel mode>	RO, WO, RW

Description

VXI[:SERVant]:FDC:CLOSe [<channel number>]
This command closes the selected logical FDC channel. If the optional channel number is omitted, the channel referenced by the VXI:FDC:SEL command is used. Closed FDC channels must be opened before they can be accessed for data transfer. The channel number parameter is the logical FDC channel number for the instrument.

VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]
This command opens the selected logical FDC channel. The mode parameter determines the direction of data flow, read only, write only or both read & write. The <channel mode> parameter can be one of the following unquoted SCPI labels: RO, WO, RW. Interpretation of Read and Write is from the servant device's point of view.; Data is *READ* off the VXIbus back plane into the instrument and *WRITTEN* from the instrument to the VXIbus back plane. If the optional channel number is omitted, the channel referenced by the VXI:FDC:SEL command is used. FDC channels must be opened before they can be accessed for data transfer.

Each opened FDC channel consumes system resources. Bidirectional channels require more resources than Read or Write only channels. Only open FDC channels in the operational mode appropriate for their intended use.

If the DAC FDC channel is opened for both reading and writing (mode parameter = RW) then the DAC can be loaded with ASCII data (see TRACE|DATA command description) and read back in binary. (see the FORMAT command description). You can then store the binary representation of the TRACE data on your hard drive and subsequently reload it into the DAC as a binary file using the FDC channel. This results in faster waveform loading operation for the DAC.

VXI[:SERVant]:FDC:SEL <channel number>
The command selects the FDC logical channel to be used by subsequent FDC commands. The channel number parameter is the logical FDC channel number for the instrument. It is not necessarily related to the physical FDC channel. It is not related to the DAC channels. All of the DAC channels use only one FDC channel. The power up default is logical channel 1.

VXI[:SERVant]:FDC:BUFFer <buffer length>,[,<channel number>]
This command specifies the amount of data to transfer from the servant back to the host during a FDC protocol data transfer; and, it configures the instrument for the transfer. This command is only needed when the controller is reading data back from a source instrument (a the source instrument typically accepts data from the host (via the VXI backplane) and sources it to the UUT). This command allows you to both define the size of the data to transfer in the reverse direction back to the host and to signal the instrument hardware to present the specified amount data to the FDC transfer mechanism. You must issue the

BUFFer command prior to each utilization of the host's FDC drivers to transfer data back from the servant to the host.

The buffer length is specified in number of DAC samples. Note that all 8 DAC channels are always transferred in any FDC transaction. Thus, a BUFFer setting of 1 will return 16 bytes (8 channels, 2 bytes per channel).

The channel number parameter is the logical FDC channel number for the instrument.

Note that this command is not necessary for measurement instruments, such as the DMM or counter, since a measurement instrument knows how much data it has ready to transfer via FDC. Nor is this command necessary for FDC writes to a source instrument, since the host knows how much data will be transferred. Thus, this command is only needed when the controller is utilizing FDC to retrieve data from a source instrument.

See also *FDC Operation* in *Instrument Functions*

Query Response

VXI[:SERVant]:FDC?
This query returns comma separated list of physical FDC channel numbers allocated to the instrument. Physical and logical channel numbers have a 1 to 1 correspondence. Channel assignment occurs as follows:

> Logical channel 1 is established on the first physical channel number returned by the query.
>
> Logical channel 2 is established on the second physical channel number returned by the query. This process continues until all logical channels are established on their corresponding channel number.

Physical FDC channel allocation is dependent upon the VX4101A option configuration. Knowledge of a logical FDC channel's physical identity is required by users writing their own low level Commander-side FDC drivers for their host computer. For more information, see *FDC Operation* in *Instrument Functions*

VXI[:SERVant]:FDC:CONFiguration? [<channel number>]
This query returns the configuration state of the selected FDC channel. If the optional channel number is omitted, the channel referenced by the VXI:FDC:SEL command is used. Return values are the following quoted ASCII strings:

"CLOSED", "OPEN", "INITIALIZED", "READ ONLY", "WRITE ONLY", "READ_WRITE"

Configuration states of READ ONLY, WRITE ONLY and READ WRITE imply an OPENED and INITIALIZED state. A channel that is in a Closed state must be Opened before it can be accessed. In practice, a channel will be in one of the four active states: (CLOSED, READ ONLY, WRITE ONLY, or both).

The <channel number> parameter is the logical FDC channel number for the instrument.

VXI[:SERVant]:FDC:SEL?
The query returns the currently selected logical FDC channel. The return value is a single integer, from 1 to the number of logical FDC channels the instrument supports.

VXI[SERVant]:FDC:BUFFer?
Returns the length of the last FDC transaction between a servant and a host. The buffer length is specified in number of samples. The return value is a single integer ranging from 1 to the number of samples that you have specified.

Examples

Command	Response/Description
VXI:FDC:SEL<1>	Selects logical FDC channel 1 for access.
VXI:FDC:Open, RO	Opens FDC channel 1 for read only operation
VXI:FDC?	2,4 Indicates physical FDC channel 2 is implemented on this devices logical channel 1; physical channel 4 on logical channel 2
VXI:FDC:CONF? 1	“READ ONLY” Indicates channel 1 is Opened, Initialized, and Read only.
VXI:FDC:CLOSe	Closes FDC channel 1

SCPI Commands for the Relay Drivers

This section contains the SCPI commands for the Relay Drivers of the VX4101A MultiPaq™ Instrument. You can use these commands to open and close relays, direct signals through the 8 relay drivers, and inquire on device status.

Command Summary

The following lists the available commands for the relay drivers:

ROUTe Subsystem

Commands

```
[ROUTe:]CLOSe <channel list>
```

```
[ROUTe:]OPEN <channel list>
```

Queries

```
[ROUTe:]CLOSe:STATe?
```

STATus Subsystem

Queries

```
STATus:OPERation:CONDition?
```

ROUTe Subsystem

The ROUTe commands enable you to control signals through any or all of the eight relay drivers.

Command Syntax

```
[ROUTe:]CLOSe <channel list>
```

```
[ROUTe:]OPEN <channel list>
```

Query Syntax

```
[ROUTe]:CLOSe:STATe?
```

*RST Value

N/A

Parameter	Value
<channel list>	N/A
STATe	Open

Limits

Parameter	Value
<channel list>	@1:8

Related Commands N/A

Command Description The ROUTe commands perform the following operations:

[ROUTe:]CLOSe <channel list>
This command closes between one and eight relay drivers on the VX4101A.

[ROUTe:]OPEN <channel list>
This command opens between one and eight relay drivers on the VX4101A.

Query Response **[ROUTe]CLOSe:STATe?**
This command queries the state of the relay drivers on the VX4101A. The MSB corresponds to relay 8 and the LSB corresponds to relay 1. The possible values are:

00000000–11111111

Examples **[ROUTe:]CLOSe**

Command	Response
ROUT:CLOS(@1,3,5,7)	Closes relay drivers 1,3,5, and 7
ROUTe:CLOSe:STATe?	"01010101" Queries which relay drivers are closed

[ROUTe:]OPEN

Command	Response
ROUT:CLOSe(@1,3,5,7)	Closes relay drivers 1,3,5, and 7
ROUTe:CLOSe:STATe?	"01010101" Queries which relay drivers are closed
ROUT:OPEN(@1,3,5,7)	Opens relay drivers 1,3,5, and 7

[ROUTe]CLOSe:STATe?

Command	Response
ROUT:CLOSe(@1,3,5,7)	Closes relay drivers 1,3,5, and 7
ROUTe:CLOSe:STATe?	"01010101" Queries which relay drivers are closed
ROUT:OPEN(@1,3,5,7)	Opens relay drivers 1,3,5, and 7

STATus? Subsystem

The STATus subsystem queries the status of the Relay Drivers

Command Syntax N/A

Query Syntax `STATus:OPERation:CONDition?`

***RST Value** 0

Limits 0

Related Commands N/A

Query Response The STATus query performs the following operations:

This query returns the status bits in the status register. A '0' is always returned. No bits are ever changed. This query was added for compatibility issues only.

Examples

Command	Action	Returns
STAT:OPER:COND?	Queries the status bits	0

SCPI Commands for the Digital Multimeter

This section contains summaries of SCPI commands for the Digital Multimeter (DMM) instrument in the VX4101A MultiPaq™ Instrument. You can use the commands to calibrate the instrument, define the types of measurements you want to make, the voltage levels you want to use, begin the measurement, and retrieve measurement results.

Command Summary

The following is a listing of the available command subsystems and syntax:

CALCulate Subsystem

Commands

```
CALCulate:LIMit

   :ENVelope[:DATA] <threshold1>,<threshold2>
   :LOWer[:DATA] <threshold>
   :REPort[:DATA]
   :UPPer[:DATA] <threshold>

CALCulate:TRANsform:HISTogram:COUNt <numeric_value>
CALCulate:TRANsform:HISTogram:POINts <numeric_value>
CALCulate:TRANsform:HISTogram:RANGe <min>,<max>
CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>
```

Queries

```
CALCulate:AVERage? [<count>[,<offset>[,<step_size>]]]
CALCulate:LIMit:ENVelope[:DATA]?
CALCulate:LIMit:FCOunt?
CALCulate:LIMit:LOWer[:DATA]?
CALCulate:LIMit:REPort[:DATA]?
CALCulate:LIMit:UPPer[:DATA]?
CALCulate:MEDian?[<count>[,<offset>[,<step_size>]]]
CALCulate:MINimum?[<count>[,<offset>[,<step_size>]]]
CALCulate:MAXimum?[<count>[,<offset>[,<step_size>]]]
CALCulate:SDEViation?[<count>[,<offset>[,<step_size>]]]
CALCulate:TRANsform:HISTogram?

   :ABOVe?
   :BELow?
   :COUNt?
   :POINts?
```

```
    :RANGe?
    :RANGe:AUTO?
```

```
CALCulate:VARiance?[<count>[,<offset>[,<step_size>]]]
```

CALibrate Subsystem

Commands

```
CALibrate:LFRequency <line frequency>
CALibrate:VALue <frequency>
CALibrate:ZERO:AUTO <auto>
```

Queries

```
CALibrate:LFREquency?
CALibrate:ZERO:AUTO?
```

CONFigure Subsystem

Commands

```
CONFigure([:SCALar]|:ARRay)[:...]

    [:VOLTage]
        :AC [<array size>[,<Expected Value>[,<Resolution>]]]
        :ACDC [<array size>[,<Expected Value>[,<Resolution>]]]
        :DC [<array size>[,<Expected Value>[,<Resolution>]]]
    :CURRent [:DC][<array size>[,<Expected
    Value>[,<Resolution>]]]
    :RESistance [<array size>[,<Expected Value>[,<Resolution>]]]
    :FRESistance [<array size>[,<Expected Value>[,<Resolution>]]]
```

***NOTE**. If SCALar is specified, the <array size is not valid. If ARRay is specified, <array size> is a required parameter.*

Queries

```
CONFigure?
```

FETCh? Subsystem

Queries

```
FETCh? [<Count>[,<Offset>[,<Step Size>]]]

FETCh:COUNt?
```

INITiate Subsystem

Commands

```
INITiate[:IMMediate]

INITiate:CONTinuous [<Boolean>]
```

Queries
INITiate:CONTinuous?

INPut Subsystem

Commands
INPut:IMPedance <impedance>

Queries
INPut:IMPedance?

INSTrument Subsystem

Commands
INSTrument:ABORt
INSTrument:RESet

MEASure? Subsystem

Queries
MEASure([:SCALar]|:ARRay)

[:VOLTage]

:AC? [<Array Size>[,<Expected Value>[,<Resolution>]]]
:ACDC? [<Array Size>[,<Expected Value>[,<Resolution>]]]
[:DC]? [<Array Size>[,<Expected Value>[,<Resolution>]]]

:CURRent[:DC]? [<Array Size>[,<Expected Value>[,<Resolution>]]]

:RESistance? [<Array Size>[,<Expected Value>[,<Resolution>]]]

:FRESistance? [<Array Size>[,<Expected Value>[,<Resolution>]]]

***NOTE**. If SCALar is specified, the <array size is not valid. If ARRay is specified, <array size> is a required parameter.*

READ? Subsystem

READ?

TEST Subsystem

TEST:ALL?

SENSe Subsystem

Commands
SENSe:BANDwidth:DETector <expected frequency>|DEFault|MINimum|MAXimum

SENSe:FUNCtion <Measurement Function>

```
SENSe:VOLTage

    [:DC]
    :AC
    :ACDC

        :RANGe[:UPPer] <Input Voltage Range>
        :AUTO <auto>
        :APERture <Aperture Value>
        :RPSecond <Readings/Second>
        :NPLCycles <Number of Power Line Cycles>
        :RESolution <Expected Resolution>
        :COUNt <Array Size>

SENSe:CURRent[:DC]

    :RANGe[:UPPer] <Input Current Range>
    :AUTO <auto>
    :APERture <Aperture Value>
    :RPSecond <Readings/Second>
    :NPLCycles <Number of Power Line Cycles>
    :RESolution <Expected Resolution>
    :COUNt <Array Size>

SENSe:[RESistance|FRESistance]

    :RANGe[:UPPer] <Input Resistance Range>
    :AUTO <Boolean|ONCE>
    :APERture <Aperture Value>
    :RPSecond <Readings/Second>
    :NPLCycles <Number of Power Line Cycles>
    :RESolution <Expected Resolution>
    :COUNt <Array Size>
```

Queries

```
SENSe[...]?
SENSe:BANDwidth:DETector?
```

STATus? Subsystem

```
STATus:OPERation:CONDition?
```

TEST Subsystem

```
TEST:ALL?
```

TRIGger Subsystem

Commands

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event delay>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest <polarity>
   :STRobe <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
```

Queries

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest?
   :STRobe?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

VXI:FDC Subsystem

Commands

```
VXI[:SERVant]:FDC:CLOSe [<channel number>]
VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]
VXI[:SERVant]:FDC:SEL <channel number>
```

Queries

```
VXI[:SERVant]:FDC?
VXI[:SERVant]:FDC:CONFiguration? [<channel number>]
VXI[:SERVant]:FDC:SEL?
```

CALCulate Subsystem

The CALCulate commands apply mathematical operations to a series of measurements, or retrieve measurements according to specific criteria.

Command Syntax

```
CALCulate:LIMit

   :ENVelope[:DATA] <threshold1>,<threshold2>
   :LOWer[:DATA] <threshold>
   :UPPer[:DATA] <threshold>

CALCulate:TRANsform:HISTogram:COUNt <numeric_value>
CALCulate:TRANsform:HISTogram:POINts <numeric_value>
CALCulate:TRANsform:HISTogram:RANGe <min>,<max>
CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>
```

Query Syntax

```
CALCulate:AVERage? [<count>[,<offset>[,<step_size>]]]

CALCulate:LIMit:ENVelope[:DATA]?

CALCulate:LIMit:FCOunt?

CALCulate:LIMit:LOWer[:DATA]?

CALCulate:LIMit:REPort[:DATA]?

CALCulate:LIMit:UPPer[:DATA]?

CALCulate:MEDian?[<count>[,<offset>[,<step_size>]]]

CALCulate:MINimum?[<count>[,<offset>[,<step_size>]]]

CALCulate:MAXimum?[<count>[,<offset>[,<step_size>]]]

CALCulate:SDEViation?[<count>[,<offset>[,<step_size>]]]

CALCulate:TRANsform:HISTogram?

   :ABOVe?
   :BELow?
   :COUNt?
   :POINts?
   :RANGe?
   :RANGe:AUTO?

CALCulate:VARiance?[<count>[,<offset>[,<step_size>]]]
```

***RST Value**

Parameter	Value
<count>	1
<offset>	1
<step_size>	1
<threshold>	User defined, depending on measurement
<threshold1>	User defined, depending on measurement
<threshold2>	User defined, depending on measurement
<numeric_value>	N/A
<min>	N/A
<max>	N/A

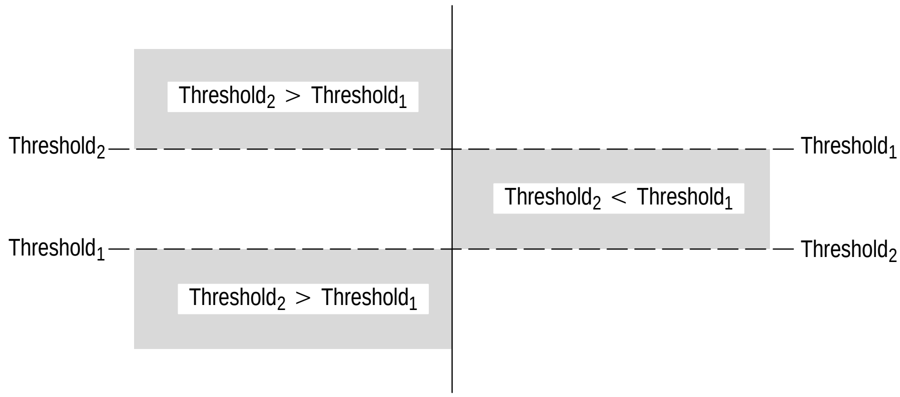
Limits

Parameter	Value
<count>	1–4096
<offset>	1–4096
<step_size>	1–4096
<threshold>	User defined, depending on measurement
<threshold1>	User defined, depending on measurement
<threshold2>	User defined, depending on measurement
<numeric_value>	N/A
<min>	N/A
<max>	N/A

Command Description

CALCulate:LIMit:ENVelope[:DATA] <threshold1>,<threshold2>
This command searches for all the input data values within an *envelope* of values defined as being above, below, or in between, a set of boundary values. The range of the envelope is determined by the *<threshold1>* and the *<threshold2>* values. For example:

- If the *<threshold2>* is greater than the *<threshold1>*, the range of data values searched for is above the *<threshold1>* value or below the *<threshold1>* value.
- If the *<threshold2>* is less than *<threshold1>*, the range of data values searched for is between the *<threshold1>* and *<threshold2>*.



CALCulate:LIMit:LOWer[DATA] <threshold>
This command sets a lower threshold limit. This command is different from the envelope command in that it checks only a lower threshold

CALCulate:LIMit:UPPer[DATA] <threshold>
This command sets an upper threshold limit. This command is different from the envelope command in that it checks only an upper threshold.

CALCulate:TRANsform:HISTogram:COUNt <numeric value>
This command determines the number of data points to include in a histogam calculation. The query version returns the programmed count.

CALCulate:TRANsform:HISTogram:POINts <numeric value>
This command sets the number of intervals in a histogram calculation. If a data value is exactly the same value as a limit dividing two bins, the data value will be counted in the next bin, unless the value matches the last interval limit. The query version returns the number of intervals programmed.

Bin #	Limits
1	5 –10
2	10 –15
3	15 –20
4	20 –25
5	25 –30
6	30 –35

Data	Bin
6	1
33	6
15	3
35	6

CALCulate:TRANsform:HISTogram:RANGe <min>,<max>
This command sets the minimum and maximum values to use in a histogram calculation. The

CALCulate:TRANsform:HISTogram:RANGe:AUTO <ON|OFF>
This command sets the minimum and maximum values to use in a histogram calculation to be automatically determined. The minimum and maximum data points will be used.

Query Response

CALCulate:AVERage? [<count>[,<offset>[,<step size>]]]
This query averages a specified number of measurements in the memory buffer. The optional parameters are for averaging selected values in the memory buffer.

The <count> parameter specifies the number of data points to calculate. If no <count> is specified, then the last number measurements taken will be the default.

The <offset> parameter determines at which point in memory values will start to be averaged.

The <step_size> parameter determines the number of measurements to skip before taking a value to be averaged.

***NOTE**. In order to specify a <step_size>, you must enter the <count> and the <offset> information.*

CALCulate:LIMit:FCOunt?
This query performs a limit test on the current available data and returns the number of data points that failed a limit test.

***NOTE**. You must call a CALCulate:LIMit:Upper, lower, or envelope command before this query.*

CALCulate:LIMit:LOWer[:DATA]?
This query returns the boundary value. The example below uses voltage.

NOTE*. You must call a CALCulate:LIMit:Upper, lower, or envelope command before you can use this query.*

CALCulate:LIMit:REPort[:DATA]?
This query returns the indices of the data values collected in the most recent CALCulate:LIMit:...[:DATA] command. Note:The CALCulate:LIMit:...[:DATA] command must be run before this command. If no data values were found a zero will be returned. (The first point in memory is “1”)

NOTE*. You must call a CALCulate:LIMit:Upper, lower, or envelope command before you can use this query.*

CALCulate:LIMit:UPPer[:DATA]?
This query returns the boundary value.

CALCulate:MAXimum? [<count>[,<offset>[,<step size>]]]
This query calculates and returns the maximum value for a set of data. The <data_index> returned indicated which data point in memory where the maximum value is. The <maximum_value> returns the value of the maximum data point. If no valid data values were found “No Data” will be returned. The optional parameters are for comparing selected values in the memory buffer. The optional <count> parameter specifies the number of data points to search. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be searched. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be compared. Note: In order to specify a <step_size>, you must enter the <count> and the <offset> information.

CALCulate:MEDian? [<count>[,<offset>[,<step_size>]]]
This query calculates and returns the median value for a set of data. The <data_index> returned indicated which data point in memory where the median value is. The <median_value> returns the value of the median data point. If no valid data values were found “No Data” will be returned. The optional parameters are for searching selected values in the memory buffer. The optional <count> parameter specifies the number of data points to search. If no <count> is specified, then the number measurements last taken will be the default. The

optional <offset> parameter determines at which point in memory values will start to be searched. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be compared. Note: In order to specify a <step_size>, you must enter the <count> and the <offset> information.

CALCulate:MINimum? [<count>[,<offset>[,<step_size>]]]
This query calculates and returns the minimum value for a set of data. The <data_index> returned indicated which data point in memory where the median value is. The <minimum_value> returns the value of the minimum data point. If no valid data values were found "No Data" will be returned. The optional parameters are for comparing selected values in the memory buffer. The optional <count> parameter specifies the number of data points to compare. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be searched. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be compared. Note: In order to specify a <step_size>, you must enter the <count> and the <offset> information.

CALCulate:SDEViation? [<count>[,<offset>[,<step_size>]]]
This query calculates and returns the standard deviation for a set of data. The formula used for standard deviation is:

$$\sigma = sqrt\ (\Sigma\ (datai - average)2\ /\ (n{-}1))$$

$$n = amount\ of\ data\ summed$$

If no valid data values were found "No Data" will be returned. The optional parameters are for calculating selected values in the memory buffer. The optional <count> parameter specifies the number of data points to calculate. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be calculated. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be used.

***NOTE**. In order to specify a <step_size>, you must enter the <count> and the <offset> information.*

CALCulate:TRANsform:HISTogram?
This query calculates and returns the histogram for a set of data.

CALCulate:TRANsform:HISTogram:ABOVe?
Returns the number of points above the maximum value in a histogram calculation.

CALCulate:TRANsform:HISTogram:BELow?
Returns the number of points below the minimum value in a histogram calculation.

CALCulate:VARiance? [<count>[,<offset>[,<step_size>]]]
This query calculates and returns the variance for a set of data. The variance is the standard deviation squared or:

$$\sigma = sqrt\ (\Sigma\ (datai - average)2\ /\ (n-1))$$

$$n = amount\ of\ data\ summed$$

If no valid data values were found “No Data” will be returned. The optional parameters are for calculating selected values in the memory buffer. The optional <count> parameter specifies the number of data points to calculate. If no <count> is specified, then the number measurements last taken will be the default. The optional <offset> parameter determines at which point in memory values will start to be calculated. The optional <step_size> parameter determines the number of measurements to skip before taking a value to be used. Note: In order to specify a <step_size>, you must enter the <count> and the <offset> information.

Examples

CALCulate:AVERage? [<count>[,<offset>[,<step_size>]]]

Command	Response
inst:sel dmm	Selects the DMM
calc:lim:upper 5.0	Sets upper voltage limit to 5
calc:aver?	Averages all the data that has been collected
calc:aver? 20,3,2	Averages 20 measurement readings. Starts with the 3rd measurement. Averages every other data point

CALCulate:LIMit:ENVelope[:DATA]

Command	Response
inst:sel dmm	Selects the DMM
calc:lim:env 5.0, 1.0	Sets an envelope voltage. All data points above 5.0 volts and below 1.0 volts are targeted.
calc:lim:env?	1.0, 5.0<lf> returns the envelope threshold voltages
calc:lim:env 1.0,5.0	Sets an envelope voltage. All data points between 5.0 volts and 1.0 volts are targeted.
calc:lim:env?	5.0, 1.0<lf> returns the envelope threshold voltage

CALCulate:LIMit:FCOunt?

Command	Response
inst:sel dmm	Selects the DMM
calc:lim:fco?	4 Returns the number of data points outside of a defined threshold.

CALCulate:LIMit:LOWer[:DATA]

Command	Response
inst:sel dmm	Selects the DMM
calc:lim:lower 5.0	Sets lower voltage threshold
calc:lim:lower?	5.0 Returns the lower threshold voltage

CALCulate:LIMit:REPort[:DATA]?

Command	Response
inst:sel dmm	Selects the DMM
calc:lim:rep?	2, 6.20; 5,5.01 Returns the number of data points and data values outside of the defined threshold

CALCulate:LIMit:UPPer[:DATA]

Command	Response
inst:sel dmm	Selects the DMM
calc:lim:upper?	5.0 Returns the upper threshold voltage

CALCulate:MEDian?

Command	Response
inst:sel dmm	Selects the DMM
CALC:MEDian?	Calculates and returns the median value

CALCulate:MINimum?

Command	Response
inst:sel dmm	Selects the DMM
CALC:MINimum?	Calculates and returns the minimum value

CALCulate:MAXimum?

Command	Response
inst:sel dmm	Selects the DMM
CALC:MAXimum?	Calculates and returns the maximum value

CALCulate:SDEViation?

Command	Response
inst:sel dmm	Selects the DMM
CALC:SDEViation?	Calculates and returns the standard deviation on a set of data

CALCulate:TRANsform:HISTogram?

Command	Response
inst:sel dmm	Selects the DMM
CALC:TRAN:HIST:COUNt 5	Sets the number of data to use in a histogram calculation to 5
CALC:TRAN:HIST:POINts 3	Sets the number of intervals in a histogram calculation to 3
CALC:TRAN:HIST:RANGE 10,50	Sets the range of a histogram calculation
CALC:TRAN:HIST?	Returns the histogram distribution

CALCulate:TRANsform:HISTogram:ABOVe?

Command	Response
inst:sel dmm	Selects the DMM
CALC:TRAN:HIST:COUNt 5	Sets the number of data to use in a histogram calculation to 5
CALC:TRAN:HIST:POINts 3	Sets the number of intervals in a histogram calculation to 3
CALC:TRAN:HIST:RANGE:AUTO Off	Turns off Autoranging.
CALC:TRAN:HIST:RANGe 10,50	Sets the minimum and maximum limit for a histogram calculation.
CALC:TRAN:HIST:ABOV?	Returns the number of data points above the maximum value

CALCulate:TRANsform:HISTogram:BELow?

Command	Response
inst:sel dmm	Selects the DMM
CALC:TRAN:HIST:COUNt 5	Sets the number of data to use in a histogram calculation to 5
CALC:TRAN:HIST:POINts 3	Sets the number of intervals in a histogram calculation to 3
CALC:TRAN:HIST:RANGE:AUTO Off	Turns off Autoranging.
CALC:TRAN:HIST:RANGe 10,50	Sets the minimum and maximum limit for a histogram calculation.
CALC:TRAN:HIST:BEL?	Returns the number of data points below the minimum value

CALCulate:TRANsform:HISTogram:COUNt

Command	Response
inst:sel dmm	Selects the DMM
CALC:TRAN:HIST:COUNt 5	Sets the number of data to use in a histogram calculation to 5
CALC:TRAN:HIST:COUNt?	Returns the number of data points to use in a histogram calculation

CALCulate:TRANsform:HISTogram:POINts

Command	Response
inst:sel dmm	Selects the DMM
CALC:TRAN:HIST:POINts 3	Sets the number of intervals in a histogram calculation to 3
CALC:TRAN:HIST:POINts?	Returns the number of intervals in a histogram calculation

CALCulate:TRANsform:HISTogram:RANGe

Command	Response
inst:sel dmm	Selects the DMM
CALC:TRAN:HIST:RANGe	Sets the range of a histogram calculation
CALC:TRAN:HIST:RANGe?	Returns the range of a histogram calculation

CALCulate:TRANsform:HISTogram:RANGe:AUTO

Command	Response
inst:sel dmm	Selects the DMM
CALC:TRAN:HIST:RANGe: AUTO ON	Automatically sets the range of a histogram calculation
CALC:TRAN:HIST:RANGe: AUTO?	Tells if the histogram auto range is on or off

CALCulate:VARiance?

Command	Response
inst:sel dmm	Selects the DMM
CALC:VARiance?	Calculates and returns the variance on a set of data

CALibrate Subsystem

The CALibrate commands enable you to enter and retrieve values to compare actual values with the device specifications. For complete procedures for calibrating the DMM, see *Calibrating the DMM* in the *Appendix*.

Command Syntax

```
CALibrate:LFRequency <line frequency>

CALibrate:SOURce

CALibrate:VALue <calibration input value>

CALibrate:ZERO:AUTO <auto>
```

Query Syntax

```
CALibrate:LFREquency?

CALibrate:ZERO:AUTO?
```

***RST Value**

Parameter	Value
<line frequency>	Last programmed line frequency rejection value
<frequency>	N/A
<auto>	Off

Limits

Parameter	Value
<Line Frequency>	50|60|400
CALibrate:VALue <Calibration Input Value>	0 to 300,000,000
<AUTO>	OFF|ON|ONCE

Related Commands

```
MEASure?
CONFigure
SENSe
READ?
INITiate
INP:IMP
```

Command Description

CALibrate:LFRequency <line frequency>
The CALibrate:LFRequency command defines the power line frequency at which the DMM performs noise rejection. You can define the frequency as 50, 60, or 400 Hz. With 400 Hz, the DMM actually uses 50 Hz rejection, since 50 Hz is a subharmonic of 400 Hz.

The line frequency rejection value (60 Hz factory default) is stored in nonvolatile memory for recall after power-on. Issuing the CALibrate:LFRequency command automatically stores the value specified into nonvolatile memory.

NOTE. *Because much of the setup of the DMM is dependent on the line frequency rejection value, the DMM function of the module returns to its power-on state when this command is issued.*

Specifying a value other than 50, 60, or 400 Hz will cause the DMM to default to 60 Hz.

CALibrate:SOURce <source>
Prior to calibration, the DMM must be programmed to a calibration source of External. An error will be generated if a CALibrate:VALue command when the calibration source is internal. This helps prevent inadvertent calibration of the DMM.

CALibrate:VALue <frequency>
A calibration of the DMM is recommended once a year to maintain published specifications accuracy. For more information, see *Calibration for the DMM* in *Appendix D: Calibration*.

The CALibrate:VALue command is used to perform the calibration. The set of modes and ranges that must be calibrated are as follows:

DC (10 GΩ input impedance), 0.030, 0.300, 3.00 V ranges
DC (10 MΩ input impedance, all ranges)
AC/DC, all ranges
4-wire Ω, all ranges
2-wire Ω, 30 Ω range at null only
Current, all ranges

The AC mode uses the AC/DC calibration information and is not calibrated separately.

The higher 2-wire Ω ranges depend on the 4-wire Ω gain calibration and the 30 Ω 2-wire null calibration, and are not calibrated separately. If a CALibration:VALue command is issued in the AC mode or higher 2-wire resistance ranges, an "Invalid DMM Calibration Mode" error is generated.

All calibrated modes and ranges except the 2-wire 30 Ω range require both a null and a gain calibration. The 2 wire 30 Ω range requires only a null calibration. The null calibration must be performed prior to the gain calibration. The limits of <Calibration Value Input> depend on the mode and range. For the DC, resistance, and current mode null calibrations, the argument must be exactly 0.

The <Calibration Input Value> is the external calibration voltage, resistance or current applied. Appendix E provides a set of external calibration specification requirements, environmental requirements, and warm-up requirements necessary for a valid calibration.

The DMM must be programmed for the desired mode/range prior to the calibration. Use a CONFigure command with the expected value equal to the range value, followed by an INITiate:[IMMediate] command or a READ? query.

CAUTION. *Do not send only the CONFigure command followed by a CALIbration:VALue command. You must include an INITiate or READ? command to place the DMM in the proper mode and range. You must allow sufficient time for both the calibration source to stabilize and the VX4101A to stabilize physically in the desired mode and range prior to performing the calibration.*

For the AC/DC "null" calibration and the gain calibration for all calibrated modes and ranges, the <Calibration Input Value> must be a positive value within the limit listed below. The limits shown are as a percentage of the full scale range value. For example, in the DC 30 V ranges, where the gain percentage must be between 65% and 100%, the CALibration:VALue argument must be between 19.5 and 30.0. A value outside of these limits will result in an "Invalid DMM Calibration Input" error.

Table 3–5: Limits of Calibration Input

Mode/Range	Low Limit	High Limit
DC Gain, All Ranges	65%	100%
AC/DC Null, 30 mV Range	30%	40%
AC/DC Null, other Ranges	3.33%	25%
AC/DC Gain, All Ranges	65%	100%
4 wire Resistance, Gain, All Ranges	30%	100%
Current, Gain, All Ranges	65%	100%

The AC/DC "null" calibrations are provided at a nonzero value due to the inherent null noise in any AC measurement system. The "null" calibration limits shown permit a more linear calibration for the region of specified operation for each range (10% to 100% of range value).

In order for the CALibration:VALue command to be used, the DMM must be programmed for CALibration:SOURce EXTernal. This provides a measure of protection against an unintentional calibration or calibration by unqualified personnel. In addition to the required limits above, a calibration reasonableness check is also provided. If the difference between the <Calibration Input Values> specified and the value measured by the VX4101 DMM is outside the required design limits, "DMM Calibration out of Range" error is generated.

The time for the CALibration:VALue command to complete depends on the mode and range and typically takes one or two seconds for all modes (except AC, which takes slightly more). In the DC, resistance and current modes, an auto zero base calibration is also performed with the null calibration. This additional calibration is used as a base reference for the CAL:ZERO:AUTO function.

CALibrate:ZERO:AUTO <auto>

The calibration procedure of the DMM includes commands to calculate and store, in nonvolatile memory, the offset value of every null calibrated function and range on the DMM. After power-on, the instrument reads these values from the nonvolatile memory, and uses them for null drift compensation.

Because offset drifts with time and temperature, the CALibrate:AUTO:ZERO command re-zeroes the DMM during operation.

If you program the CALibrate:ZERO:AUTO ONCE command, the DMM immediately performs an autozero operation for the function and range currently in effect. It will then compensate all subsequent measurements for that function by the calculated value.

***NOTE**. You must issue the CALibrate:SOURce INT command before autozero will work.*

If data sampling is in progress, issuing a CALibrate:ZERO:AUTO ONCE command will abort the acquisition.

If you program CALibrate:ZERO:AUTO ON, the DMM performs an autozero operation each time it receives a MEASure?, READ? or INITiate command *prior to the first measurement*, and whenever a measurement is taken which changes either the current function, or the current range. For example, if autoranging and autozeroing were enabled, issuing a MEASure? command would cause the DMM to first seek the appropriate range for the input, and once found, perform an autozero. If the range changes during the measurement process, the DMM performs an autozero operation automatically on the new range. The autozero routine uses the same aperture time as the measurement currently being taken. For example, if the current function were programmed to a 0.5 second aperture, the autozero routine will incur approximately 0.5 seconds of overhead each time it is invoked.

If CALibrate:ZERO:AUTO OFF is programmed, autozeroing is disabled.

***NOTE**. The autozero setup is valid for all functions and ranges of the DMM. That is, the autozeroing cannot be individually programmed for each function. Also the autozero values are not stored in nonvolatile memory. The calibration procedure of the DMM includes commands to calculate and store calibrated base offset values.*

Query Response

CALibrate:LFRequency?
Returns the rejection frequency value stored in nonvolatile memory.

CALibrate:SOURce?
Returns the calibration source, either EXT or INT.

CALibrate:ZERO:AUTO?
Returns a zero or a one, depending on whether you have enable autozeroing. The values are:

0=disabled, OFF
1=enabled, ON

Examples

Command	Response
calibrate:lfrequency 60	Defines the line rejection frequency to be 60 Hz
CAL:LFR?	60<LF> The programmed line rejection frequency
cal:lfr 400	Defines the line rejection frequency to be 400 Hz
CALibrate:LFRequency?	400<LF>
Calibrate:source ext	Specifies an external calibration source
CAL:SOUR?	EXT<LF> The programmed calibration source
CONF:FRES 300e3	Configures the DMM for the 4-wire 300 k resistance range
CAL:SOUR?	EXT<LF> The programmed calibration source
READ?	Returns the present resistance value and physically places the DMM in the specified range
Cal:Val 0	Performs a null calibration in the 300 kΩ range. A 0 Ω resistor input is required
CONF:FRES 300e3	Configures the DMM for the 4-wire 300 k resistance range
READ?	Returns the present resistance value and physically places the DMM in the specified range
CAL:VALUE 100.03e3	Performs a gain calibration assuming a 100.03 kΩ resistor has been placed on the input
Cal:sour int	Returns the DMM to internal Calibration source permitting an autozero to be performed
calibrate:zero:auto on	Enable the auto zero calibration for every measurement
CAL:ZERO:AUTO?	1<LF> Auto zero enabled
cal:zero:auto off	Disable the auto zero calibration for each measurement
cal:zero:auto?	0<LF> Auto zero disabled
Cal:Zero:Auto 1	1 and 0 can be substituted for ON and OFF respectively
Cal:zero:auto once	Perform an auto zero on the current function and range

CONFigure Subsystem

The CONFigure commands enable you to set up the DMM for a specific type of input measurement.

Command Syntax

```
CONFigure([:SCALar]|:ARRay)

   [:VOLTage]

      :AC [<Array Size>[,<Expected Value>[,<Resolution>]]]
      :ACDC [<Array Size>[,<Expected Value>[,<Resolution>]]]
      :DC [<Array Size>[,<Expected Value>[,<Resolution>]]]

   :CURRent[:DC] [<Array Size>[,<Expected
   Value>[,<Resolution>]]]

   :RESistance [<Array Size>[,<Expected Value>[,<Resolution>]]]

   :FRESistance [<Array Size>[,<Expected Value>[,<Resolution>]]]
```

***NOTE**. If SCALar is specified, the <array size> parameter is not valid. If ARRay is specified, <array size> is a required parameter.*

Query Syntax

```
CONFigure?
```

***RST Value**

Parameter	Value
CONFigure([:SCALar]|: ARRay)[:...]	300 VDC, 10 MΩ Input, 0.001 Resolution, Autoranging ON

Limits

Parameter	Value
<Array Size>	1 to 4096

***NOTE:** See MEASure? Command for more information on the <Expected Value> parameter. See [SENse][...]:RESolution command for more information on the RESolution parameter.*

Related Commands

```
MEASure?
INPut
READ?
SENse
FETCh?
CALibrate:LFR
CALibrate:ZERO:AUTO
```

Command Description

The CONFigure commands are sublevel commands to the MEASure? commands. They define the input configuration for a measurement without taking a measurement. If desired, you can customize the configuration setups via the lower level SENSe commands. The different input types you can set up are as follows:

CONFigure([:SCALar|:ARRay)[:VOLTage]:AC[<Array Size>[,<Expected Value>[,<Resolution>]]]
Sets up AC coupled, AC TRMS voltage measurements. The AC bandwidth is set to slow mode.

CONFigure([:SCALar|:ARRay)[:VOLTage]:ACDC[<Array Size>[,<Expected Value>[,<Resolution>]]](<Array Size>[,<Expected Value>[,<Resolution>]]]
Sets up DC coupled, AC TRMS voltage measurements. The AC bandwidth is set to slow mode.

CONFigure([:SCALar|:ARRay)[:VOLTage]:DC[<Array Size>[,<Expected Value>[,<Resolution>]]]
Sets up DC voltage measurements.

CONFigure([:SCALar|:ARRay):CURRent[:DC](<Array Size>[,<Expected Value>[,<Resolution>]]]
Sets up DC current measurements.

CONFigure([:SCALar|:ARRay):RESistance[<Array Size>[,<Expected Value>[,<Resolution>]]]
Sets up two-wire resistance measurements.

CONFigure([:SCALar|:ARRay):FRESistance[<Array Size>[,<Expected Value>[,<Resolution>]]]
Sets up four-wire resistance measurements.

***NOTE**. See the MEASure? command for detailed descriptions of the [<Expected Value>],[<Resolution>], and <Array Size> fields.*

The CONFigure[.....]? queries return detailed setup information for the requested function.

***NOTE**. If data sampling is in progress, it will be aborted on receipt of a CONFigure command to prevent ambiguous data interpretation. Typically the CONFigure (and SENSE) commands are used to predefine the setup of each of the six measurement functions.*

Query Response

CONFigure?

Returns the current measurement configuration information, with the actual rounded and/or calculated values used by the DMM.

Examples

Command	Response
CONFigure:SCALar:VOLTage:DC 2	Set to DC Volts, 1 Measurement
CONF?	":SCAL:VOLT:DC 3,1e-05"<LF>
conf:dc 10	Set to DC Volts
conf?	":SCAL:VOLT:DC 30,0.0001"<LF>
conf:volt:ac 5,0.01	Set to AC Volts
configure?	":SCAL:VOLT:AC 30,0.00154919"<LF>
conf:curr:dc max,minimum	Set to DC Amps maximum range and minimum resolution
CONFIGURE?	":SCAL:CURR:DC 1,7.74597e-05"<LF>
conf:res def	Set to two-wire Resistance
conf?	":SCAL:RES 3e+08,180000"<LF>
conf:fresistance	Set to four-wire Resistance
conf?	":SCAL:FRES 3e+08,180000"<LF>
conf:ARRAY:volt:dc 10,50,0.01	Set to DC Volts, 10 Measurements
configure?	":ARR:VOLT:DC 10,300,0.0109545"<LF>
conf:dc?	":ARR:VOLT:DC 10,300,0.0109545 AutoRange OFF AutoZero OFF Aperture=0.00166667 RPSec=600 NPLC=0.1 Rdgs2Avg=2 LFR=60"<LF> *

Table 3–6: Meaning of Returned String

Returned	Meaning
AutoRange OFF	The state of the autorange enable (ON | OFF | ONCE)
AutoZero OFF	The state of the autozero enable (ON | OFF)
Aperture=0.00166667	The aperture setting
RPSec=600	The number of readings per second corresponding to the aperture
NPLC=0.1	The number of power line cycles corresponding to the aperture
LFR=60	The line frequency rejection value

FETCh? Subsystem

The FETCh? subsystem is the lowest level of the measurement subsystem. It retrieves the measurement data acquired through higher level commands. The optional parameters selectively control which of the acquired value(s) the measurement subsystem returns. If the optional parameters are not specified, the command returns the amount of data specified by the <Array Size> field of the higher level commands.

For example, if the subsystem acquires thirty measurements, issuing a “FETCh?” returns the thirty values. Issuing a “FETCh? 10” would return the first 10 values acquired. Issuing a “FETCh? 10,21” would return 10 data values, beginning at offset 21 of the array (offset 1 = the first data value). Issuing a “FETCh? 5,1,2” would return five data values, beginning with the first acquired value, and stepping two locations for each value returned (values one, three, five, seven, and nine). If the parameters specify an index outside the number of measurements acquired, an error will be generated.

A subsequent FETCh? will return the same data if a new measurement has not been INITiated.

Command Syntax

N/A

Query Syntax

```
FETCh? [<Count>[,<Offset>[,<Step Size>]]]
```

```
FETCh:COUNt?
```

***RST Value**

Parameter	Value
<Count>	Size of last array measurement
<Offset>	0
<Step Size>	1

Limits

Parameter	Value
<Count>	1-4096
<Offset>	1-4096
<Step Size>	1-4096

Related Commands

```
MEASure?
CONFigure
SENSe
INPUt
INITiate
*STB?
```

Command Description

N/A

Query Response

FETCh:COUNt?
The FETCh:COUNt? query returns the number of measurements acquired since the last INITiate command was received. You can use this query to monitor the progress of a measurement or array of measurements.

***NOTE**. A CONFigure, SENSe, or INPut command will empty the measurement buffer. An error is generated if a FETCh? query is requested when no valid data is available.*

For multiple data measurements, the data is returned using SCPI block format.

The format is:

```
HEADERdata,data,...,data<LF>
```

where the HEADER field is

```
#xCOUNT
```

x = the size of the count field

COUNT = the number of bytes returned following the header field.

For example, the format of a “fetch? 2” response would be

```
#226+2.98999E+00,+2.99030E+00<LF>
```

The first numeral 2 following the # sign indicates that the COUNT size field is 2. The number 26 following it indicates there are 26 characters in the block following the header.

Examples

Command	Response
FETCh?	#239+1.23456E+00,+2.34567E+00<LF>
FETCh? 1	+1.23456E+00<LF>
FETCh? 2,2	#226+2.34567E+00,+3.45678E+00<LF>
FETCh? 2,1,2	#226+1.23456E+00,+3.45678E+00<LF>
Fetch:count?	3<LF>

INITiate Subsystem

The INITiate commands initiate the DMM to begin making a measurement.

Command Syntax

`INITiate[:IMMediate]`

`INITiate:CONTinuous [<Boolean>]`

Query Syntax

`INITiate:CONTinuous?`

***RST Value**

Parameter	Value
<Boolean>	0 (OFF)

Limits

Parameter	Value
<Boolean>	ON |1| OFF |0

Related Commands

```
MEASure?
CONFigure
SENSe
FETCh?
TRIGger
ABORt
```

Command Description

INITiate:CONTinuous [Boolean]
The INITiate:CONTinuous ON command internally triggers the DMM to operate in a free-running mode, where it continuously acquires and updates a single measurement. The FETCh? command can be issued any time after this command to read the current value of the input. The INITiate:CONTinuous command is independent of the trigger setup.

The INITiate:CONTinuous OFF command aborts any measurements currently in progress.

INITiate[:IMMediate]
Instrument initiates its current trigger sequence. After the instrument has completed the current trigger sequence, it enters the idle state. See instrument documentation for details on instrument state after an initiate.

***NOTE**. If data sampling is in progress, issuing an INITiate command aborts the acquisition and starts a new acquisition.*

Query Response

INITiate:CONTinuous?
Returns a zero or the number one indicating whether INITiate:CONTinuous is off or on.

Examples

Command	Response
INITiate:IMMediate	Arms the triggers for an acquisition
init:continuous	Put the DMM into free-running acquisitions

INPut Subsystem

The INPut command defines the input impedance for the lower three DC ranges of the DMM.

Command Syntax

`INPut:IMPedance <impedance>`

Query Syntax

`INPut:IMPedance?`

***RST Value**

Parameter	Value
<impedance>	10e6 Ω

Limits

Parameter	Value
<impedance>	10e6 Ω, 10e9 Ω

Related Commands

The related commands for INPut subsystem are as follows:

```
MEASure
CONFigure
SENSe
```

Command Description

INPut:IMPedance <impedance>
The INPut:IMPedance command defines the input impedance for DC voltage measurements on the 30 mV, 300 mV, and 3 V ranges. The impedance may be defined as either 10e6 or 10e9 Ω. Any value other than 10e6 or 10e9 is rounded up to the nearer of these values. The 30 V and 300 V ranges are fixed at a 10 MΩ input impedance. If <Input Impedance> is not specified, 10e6 is assumed. Specifying a value greater than 10e9 or less than zero generates an error.

***NOTE**. If Autoranging is active for DC Voltage measurements, the impedance value is automatically overwritten with 10.0e6 Ω when a DC Voltage measurement is taken.*

Query Response

INPut:IMPedance?
Returns the currently defined impedance value.

Examples

Command	Response
Input:Impedance 10e9	Define the input impedance as 10e9 Ω
INP:IMP?	1.00E+10<LF>

INSTrument Subsystem

The INSTrument commands enable you to control the operational state of the DMM.

Command Syntax

```
INSTrument:ABORt
INSTrument:RESet
```

Query Syntax

N/A

***RST Value**

N/A

Limits N/A

Related Commands The related commands of INSTrument subsystem are as follows:

```
ABORt
INITiate
INITiate:CONTinuous
```

Command Description

INSTRument:ABORt
This command places the active instrument in the IDLE state, aborting any measurement or other instrument activity in progress. The instrument configuration is unchanged and a subsequent INIT command will cause the instrument to restart the same type of measurement.

***NOTE**. If the instrument is in asynchronous mode, this command can be sent while a query is in progress and the measurements will be aborted. If however, the instrument is in synchronous mode, this command will be queued while a query is in progress.*

INSTrument:RESet
This command resets the currently selected instrument without effecting other instruments. The instrument returns to its *RST state. The instrument remains selected.

Query Response N/A

Examples

INSTRument:ABORt

Command	Response
INST:SEL DMM	
CONF:ARR:VOLT:DC	512
INIT	
FETC:COUN?	127
INST:ABOR	
FETC:COUN?	153
NOTE: After an ABORt command, the instrument stops taking measurements.	

INSTrument:RESet

Command	Response
INST:SEL DMM	
CONF:RES	
CONF?	":SCAL:RES 3e+08,180000"
INST:RES	
INST:SEL?	DMM
CONF?	":SCAL:VOLT:DC 300,0.001"

MEASure? Subsystem

The MEASure? queries enable you to set up and initiate a measurement and return the results.

Command Syntax N/A

Query Syntax

```
MEASure([:SCALar|Array)
   [:VOLTage]
      :AC? [<Array Size>[,<Expected Value>[,<Resolution>]]]
      :ACDC? [<Array Size>[,<Expected Value>[,<Resolution>]]]
      [:DC]? [<Array Size>[,<Expected Value>[,<Resolution>]]]
   :CURRent[:DC]? [<Array Size>[,<Expected
   Value>[,<Resolution>]]]
   :RESistance? [<Array Size>[,<Expected Value>[,<Resolution>]]]
   :FRESistance? [<Array Size>[,<Expected
   Value>[,<Resolution>]]]
```

***NOTE**. If SCALar is specified, the <array size> parameter is not valid. If ARRay is specified, <array size> is a required parameter.*

***RST Value**

300 V,
DC Coupled,
10 MΩ Input,
0.001 Resolution,
Autoranging ON,

Limits

Parameter	Value
<array_size>	1–4096
<expected_value>	See tables 3-7 - 3-10

NOTE*. The <array_size> limit is 2048 when the aperture time is one second.*

Related Commands

```
CONFigure?
INPut
SENSe
READ?
INITiate
FETCh?
CALibrate:LFRequency
CALibrate:ZERO:AUTO
```

Command Description N/A

Query Response

MEASure([:SCALar]|:ARRay)[:VOLTage][:DC]? [<Array Size>[,<Expected Value>]]]
Performs DC voltage measurements.

MEASure([:SCALar]|:ARRay)[:VOLTage]:AC? [<Array Size>[,<Expected Value>]]]
Performs AC coupled, AC TRMS voltage measurements. The AC bandwidth is reset to slow mode.

MEASure([:SCALar]|:ARRay)[:VOLTage]:ACDC? [<Array Size>[,<Expected Value>]]]
Performs DC coupled, AC TRMS voltage measurements. The AC bandwidth is reset to slow mode.

MEASure([:SCALar]|:ARRay):CURRent[:DC]? [<Array Size>[,<Expected Value>]]]
Performs DC current measurements.

MEASure([:SCALar]|:ARRay):RESistance? [<Array Size>[,<Expected Value>]]]
Performs two-wire resistance measurements.

MEASure([:SCALar]|:ARRay):FRESistance? [<Array Size>[,<Expected Value>]]]
Performs four-wire resistance measurements.

[:SCALar]:ARRay

The [:SCALar] and :ARRay fields define the number of measurements to be taken. Specifying [:SCALar] defines a single measurement acquisition (<Array Size> = 1). Specifying :ARRay defines multiple measurement acquisitions of count <Array Size>. If <Array Size> is less than 1 or greater than 4096, an error will be generated. Note that specifying an array size of 1 is equivalent to specifying scalar.

<Expected Value>

<Expected Value> is an estimate of the input signal amplitude. If a numeric value is specified, the DMM will use the range nearest to the value specified to make the measurement (rounded up). <Expected Value> can also be specified as MAXimum, MINimum, or DEFault to use the function's highest range, lowest range, or to default to autoranging respectively. If no value is specified, DEFault (autorange ON) is assumed.

Autoranging uses the currently defined range as the starting value for the autoranging search. If enabled, the DMM continuously checks and adjusts the input range whenever the signal exceeds the current range, or falls below 9.9% of the range. Once the range has been determined, the DMM will perform an autozero (if enabled). Specifying any value other than DEFault or none disables autoranging. See the SENSe:RANGe:AUTO command for more detail on autoranging.

Specifying <Expected Value> outside the function's range will generate an error.

<Resolution>

The <Resolution> field defines the resolution of a measurement by implicitly defining the aperture time of the measurement. The aperture is the sampling of the measurement acquisition. The greater the aperture, the more accurate the measurement. If MAXimum is specified as the resolution field, the DMM will use a 2 second aperture for each measurement. Specifying MINimum uses a 0.8333 ms (1 ms) aperture for 60 (50) Hz line frequency rejection. Specifying DEFault (or none) sets the aperture to 200 ms. See the SENSe[.....]:RESolution commands for the method used to determine the aperture when explicitly defining the <RESolution> field.

For the MEASure:VOLTage[:DC]? commands, <Expected Value> selects the range and input impedances as specified below. |EV| indicates the absolute value of the <expected value>.

Table 3–7: MEASure:VOLTage[DC]? and <Expected Value> ranges

<Expected Value>	Range	Input Impedance
+0.000 ≤ |EV| ≤ +0.030	±30 mV	10 MΩ or 10 GΩ
+0.030 < |EV| ≤ +0.300	±300 mV	10 MΩ or 10 GΩ
+0.300 < |EV| ≤ +3.000	±3.00 V	10 MΩ or 10 GΩ
+3.000 < |EV| ≤ +30.00	±30.0 V	10 MΩ
+30.00 < |EV| ≤ +300.0	±300.0 V	10 MΩ
MAXimum	±300.0 V	10 MΩ
DEFault	Autorange	10 MΩ
MINimum	±30 mV	10 MΩ

***NOTE**. Specifying DEFault (autorange) as the <expected value>* overwrites *the value specified by the INPut:IMPedance command with 10e6 Ω.*

For the MEASure:VOLTage:AC? and MEASure:VOLTage:ACDC? commands, the TRMS (true root-mean-square) value is returned. The :AC command defines AC coupled voltage measurements (DC rejection). The :ACDC command selects the coupling as AC + DC. EV selects the range as specified below:

Table 3–8: MEASure:VOLTage:AC? and <Expected Value> ranges

<Expected Value>	Range
+0.000 ≤ EV ≤ +0.030	±30 mV
+0.030 < EV ≤ +0.300	±300 mV
+0.300 < EV ≤ +3.000	±3.00 V
+3.000 < EV ≤ +30.00	±30.0 V
+30.00 < EV ≤ +300.0	±300.0 V
MAXimum	±300.0 V
DEFault	Autorange
MINimum	±30 mV

For the MEASure:CURRent? command, <expected value> selects the range as specified below:

Table 3–9: MEASure:CURRent? and <expected value> ranges

<Expected Value>	Range
0.000 ≤ |EV| ≤ 0.150	±150 mA
0.150 < |EV| ≤ 1.000	±1.00 A
MAXimum	±1.00 A
DEFault	Autorange
MINimum	±150 mA

For the MEASure:RESistance? and MEASure:FRESistance? commands, 2-wire or 4-wire resistance measurements are taken respectively. EV selects the range as specified below:

Table 3–10: MEASure:RESistance? and MEASure:FRESistance and <Expected Value> ranges

<Expected Value>	Range
+0 ≤ EV ≤ +30	30 Ω
+30 < EV ≤ +300	300
+300 < EV ≤ +3e3	3e3
+3000 < EV ≤ +30e3	30e3
+30000 < EV ≤ +300e3	300e3
+300e3 < EV ≤ +3e6	3e6
+3e6 < EV ≤ +30e6	30e6
+30e6 < EV ≤ +300e6	300e6
MAXimum	300e6
DEFault	Autorange
MINimum	30 Ω

***NOTE**. Each of the six functions of the DMM maintains a configuration table of its setup, which is independent of the other functions (DC Volts, AC Volts, ACDC Volts, DC Current, 2-wire Resistance, 4-wire Resistance). Issuing a MEASure? command defines the active measurement function for the READ? command, in addition to the array size, range, and resolution (aperture), values of the function's configuration table. The Autorange flag is also set appropriately.*

The MEASure[.....]? commands are executed immediately upon processing. If data sampling is in progress, it will be aborted, and a new acquisition initiated.

For array measurements, the data is returned using SCPI block format.

The format is:

```
HEADERdata,data,...,data<LF>
```

where the HEADER field is

```
#xCOUNT
```

x = the size of the count field

COUNT = the number of bytes returned following the header field.

For example, the format of a "meas:arr? 2" response would be

```
#226+2.98999E+00,+2.99030E+00<LF>
```

The first 2 following the # sign indicates the COUNT size field is 2. The following 26 indicates there are 26 characters in the block following the header.

Examples

Command/Query	Response/Description
MEASure:SCALar: VOLTage:DC? 2	+1.23456<LF>
CONFigure?	:SCAL:VOLT:DC 3,1e–05
	Performs a single DC Voltage measurement on the 3 V range. Use the default resolution to set the aperture to 200 ms. Explicitly defining an input range turns off autoranging. Issue the following commands:
meas:volt:ac? max, 0.01	+40.9873<LF>
conf?	:SCAL:VOLT:AC 300,0.0109545<LF>
	Performs a single AC Voltage measurement on the 300 V (Maximum) range. The resolution specified is rounded per the formula described in the SENSe[.....]:RESolution commands.
	This resolution is equivalent to an aperture of 0.00166667 for 60 Hz rejection

Command/Query	Response/Description
meas:resistance?	+12.3456<LF>
CONF?	:SCAL:RES 30,0.0001
	Perform a 2-wire, autorange resistance measurement. The default resolution is used to set the aperture to 200 ms. Issue the following commands:
Meas:arr:curr? 3,0.15, maximum	+0.109876,+0.109678,+0.109444<LF>
conf?	:ARR:CURR:DC 3,0.15,1.58114e-07
	Perform three current measurements on the 150 mA range. Specifying the maximum resolution sets the aperture value to 2 seconds. Issue the following commands:

TEST Subsystem

The TEST subsystem handles the self test operations of the instrument. The Digital Output self test tests the Digital Output memory and the Read/Write hardware control registers. The query returns pass/ fail information. In a failed situation, additional failure information can be obtained with the SYStem:ERRor? query.

Command Syntax N/A

Query Syntax TEST:ALL?

***RST Value** N/A

Limits N/A

Related Commands *TST?

Command Description Initiates the Digital Output self test operation and returns one of two possible responses:

“DIGO Self-Test Passed”

“DIGO Self-Test Failed”

***NOTE**. If the test fails, you can obtain further information on the failure with the SYST:ERR? query.*

Examples

Command	Response
INSTRUMENT:SELECT DIGO	Select the Digital Output
TEST:ALL?	"Digital Output passes self test." Run self test.

READ? Subsystem

The READ queries enable you to initiate and acquire a measurement.

Command Syntax N/A

Query Syntax `READ?`

***RST Value** N/A

Limits N/A

Related Commands
```
MEASure?
CONFigure
SENSe
INPUt
INITiate
FETCh?
```

Command Description N/A

Query Response

READ?
The READ? query is a sub-level command of the measurement subsystem, as its action is dependent on the configuration set up by the higher level MEASure, CONFigure, and SENSe commands. It performs a measurement acquisition by executing an INITiate:IMMediate command, followed by a FETCh? query. The measurement function which is performed is based on the current configuration set up by the higher level commands. For example, if a CONFig-ure:ARRay:VOLTage:DC? 10,3.0 command was previously issued, a READ? query would initiate, acquire, and return 10 DC V measurements in the 3.0 V

range. The READ? query always uses the currently active measurement function. If the CONFigure command above were followed by a SENSe:FUNCtion AC command, a READ? query would execute an AC V measurement.

***NOTE**. If data sampling is in progress, issuing a READ? will cause the acquisition to be aborted, and a new acquisition initiated.*

Examples

Command	Response	
conf?	:SCAL:VOLT:DC	3,1e-05<LF>
READ?	+1.23456<LF>	The measured DC voltage
meas:ac? 10	+9.87654<LF>	The measured AC voltage
conf?	:SCAL:VOLT:AC	30,0.0001<LF>
read?	+9.67890<LF>	The measured AC voltage
sense:function res	define the active configuration as 2W-resistance	
conf?	:SCAL:RES	3e+08,1<LF>
read?	+123<LF>	The measured resistance

SENSe Subsystem

The SENSe commands are low-level commands that customize the setup of the MEASure? and CONFigure commands.

For example, if you issued the command MEAS:VOLT? 3.0, and the measured value(s) are over-ranging, you can issue a SENSe:VOLT 30.0 command to switch to the 30 V range without modifying any of the other setups of the DC voltage configuration.

Each of the six functions (DC Volts, AC Volts, ACDC Volts, DC Current, 2-wire Resistance, 4-wire Resistance) of the DMM maintains a configuration table of its setup, which is independent of the other functions. The SENSe commands allow you to individually customize these parameters for each of the functions.

Command Syntax

```
SENSe:BANDwidth:DETector <expected frequency>|DEFault|MINi-
mum|MAXimum

SENSe:FUNCtion <Measurement Function>

SENSe:VOLTage[:DC]

    :AC
    :ACDC
```

```
        :RANGe[:UPPer] <Input Voltage Range>

           :AUTO <auto>

        :APERture <Aperture Value>
        :RPSecond <Readings/Second>
        :NPLCycles <Number of Power Line Cycles>
        :RESolution <Expected Resolution>
        :COUNt <Array Size>

SENSe:CURRent[:DC]

   :RANGe[:UPPer] <Input Current Range>

      :AUTO <auto>

   :APERture <Aperture Value>
   :RPSecond <Readings/Second>
   :NPLCycles <Number of Power Line Cycles>
   :RESolution <Expected Resolution>
   :COUNt <Array Size>

SENSe:RESistance|FRESistance

   :RANGe[:UPPer] <Input Resistance Range>

      :AUTO <Boolean|ONCE>

   :APERture <Aperture Value>
   :RPSecond <Readings/Second>
   :NPLCycles <Number of Power Line Cycles>
   :RESolution <Expected Resolution>
   :COUNt <Array Size>
```

Query Syntax

```
SENSe[...]?

SENSe:BANDwidth:DETector?
```

Command Class

Instrument

***RST Value**

Parameter	Values
<expected frequency>	20
<measurement function>	N/A
<Input Voltage Range>	300 VDC, 300 V range
<auto>	ON
<Aperture Value>	0.2 seconds
<readings/second>	50

Parameter	Values
<Number of Power Line Cycles>	12 (60 Hz),10 (50/400 Hz)
<Expected Resolution>	0.001
<array size>	N/A
<input resistance range>	300e6 Ω range
<input current range>	1 A range

Limits

Parameter	Value
<expected frequency>	20-100000 Hz
<measurement function>	VOLTage:DC VOLTage:AC VOLTage:ACDC CURRent RESistance FRESistance
<Input Voltage Range>	± 0.03, 0.3, 3.0, 30.0, 300.0 Volts
<auto>	ON|1|OFF|0|ONCE
<Aperture Value>	See the text below
<Readings/Second>	See the text below
<Number of Power Line Cycles>	See the text below
<expected resolution>	See the text below
<Array Size>	to 4096 (2048 for aperture >1 second)
<Input Current Range>	± 0.150, 1.0 Amps
<Input Resistance Range>	30, 300, 3e3, 30e3, 300e3, 3e6, 30e6, 300e6 Ω

***NOTE**. Input voltage ranges for SENSE:VOLTAGE [:AC] and [:ACDC] are positive only.*

Formats

<Measurement Function> DC | AC | ACDC | CURRent | RESistance | FRESistance

<Boolean> ON | 1 | OFF | 0

Related Commands

MEASure?
CONFigure
READ?

INPut
CALibrate:LFRequency

Command Description

SENSe:FUNCtion <measurement function>
The SENSe:FUNCtion commands define the measurement function for subsequent READ? commands. For example, if the present configuration mode is a DC voltage measurement, issuing a SENSe:FUNCtion RESistance command will cause the next measurement to be a 2-wire resistance measurement.

Issuing a SENSe:FUNCtion FRESistance command will cause the next measurement to be a 4-wire resistance measurement.

SENSe:BANDwidth:DETector <expected frequency>|DEFault|MINimum|MAXimum
This commands selects the bandwidth for the AC and ACDC voltage measurements. Frequency is specified as the expected frequency of the input signal in Hz. The DMM uses this to select the slow mode for frequencies below 100 Hz and the fast mode for frequencies of 100 Hz or higher. The slow mode has a longer time constant for the front end and takes longer for each measurement, particularly for range changes. The fast mode has a shorter time constant and takes measurements faster. If you are unsure of the frequency of the input signal, the slow mode should be chosen to optimize measurement accuracy. MINimum selects the slow mode by setting the expected frequency to 20 Hz. MAXimum selects the fast mode by setting the expected frequency to 100000Hz.

CONFigure and MEASure AC and ACDC commands reset the bandwidth to the slow mode.

SENSe[....]:COUNt <Array Size>
The SENSe[.....]:COUNt commands define the number of measurements to acquire. If <Array Size> is less than 1 or greater than 4096, an error will be generated. Specifying an <Array Size> of 1 is equivalent to specifying scalar.

SENSe[....]:RANGe <input range>
The SENSe[.....]:RANGe commands modify the applicable input range setup. Issuing this command disables autoranging. All range values are rounded up to the nearest applicable value. Specifying a value outside the function’s range will generate an error.

SENSe[.....]:RANGe:AUTO <auto>
The SENSe[.....]:RANGe:AUTO commands control when the DMM performs auto-ranging.

If you program a SENSe:RANGe:AUTO:ONCE, the DMM will perform an auto range each time it receives a MEASure?, READ? or INITiate command, *prior to the first measurement only.*

For example, the command MEAS:ARR? 100 would autorange prior to the first measurement, and would then use the range determined for making the 100 measurements. If any measurement exceeded the determined range, a +9.9E+37 (or –9.9E+37) value is stored as the measurement value.

If you program a SENSe:RANGe:AUTO ON command, the DMM continuously monitors the level of the input signal for every measurement taken. If the signal exceeds the current range, or falls below 9.9% of the range, the DMM will autorange to seek the appropriate range, and retake the measurement on the new range. If autozeroing is enabled, the DMM performs an autozero after each new range is determined.

If you enable autoranging for DC V measurements, it will overwrite the INPut:IMPedance value with 10e6 ohms.

***NOTE**. Note that autoranging is individually programmable for each function of the DMM.*

SENSe[.....]:APERture<aperture value>
SENSe[.....]:RPSecond<readings per second>
SENSe[.....]:NPLCycles<number of Powerline cycles>
SENSe[.....]:RESolution<Expected Resolution>
These commands define four different ways of specifying the aperture time of a measurement, which is the acquisition or sampling time for the measurement.

All calculated values are internally rounded to values based on apertures supported by the hardware.

You can use the CONFigure[.....]? queries to get the actual (rounded) values calculated by the DMM for these parameters. If any field corresponds to an aperture less than .0008333 seconds (.001 seconds for 50 Hz line frequency rejection), or an aperture greater than 2.0 seconds, the DMM will limit the value to that range. Within these limits, the aperture will be rounded to the next allowable aperture value.

The relationship between the aperture, readings per second, number of power line cycles, and expected resolution are per the formulae below.

***NOTE**. The rounding takes place within reasonable limits. For example, .0166 will be rounded up to .016666....*

The rounded aperture values permitted by the hardware design are as follows

- For 60 Hz line frequency rejection:

  From .000833... to .2725 seconds in .000833... second steps.

  From .2733... to 1.0933... seconds in .00333... second steps.

  From 1.10 to 2.0 seconds in .00833... second steps.

- For 50 Hz line frequency rejection:

  From .001 to .273 seconds in .001 second steps.

  From .276 to 1.092 seconds in .004 second steps.

  From 1.10 to 2.0 seconds in .01 second steps.

The readings per second and number of power line cycles are calculated based on the allowable aperture as follows:

Readings per Second = 1. / Aperture.

Number of Power Line Cycles = 60. * Aperture, for 60 Hz line frequency rejection, or 50. * Aperture, for 50 Hz line frequency rejection.

If the DMM is directly programmed for a specified Readings per Second, or Number of Power Line Cycles per second, aperture is solved for using the appropriate formula above, rounded to an allowable value. The Readings per Second and Number of Power Line Cycles will be recalculated based on the actual resulting aperture.

The expected resolution is also related to the measurement aperture. The intent of programming resolution is to provide a means of specifying the approximate accuracy of the measurement. It is not necessarily equal to the resolution of the digits in the readings returned by the DMM. The VX4101A calculates an expected resolution that roughly approximates the noise level of the measurement. Noise level is inversely proportional to the square root of the aperture.

***NOTE**. For read back consistency, 5.5 digits in exponential format is returned for all measurements. For an aperture less than one power line cycle, the least significant digit is always 0.*

The formula for expected resolution uses this relationship with a scaling that provides 5.5 digits resolution (10 μV in the 3 V range) for the default aperture of .200 seconds, and 4.5 digits expected resolution for an aperture of 2.0 msec, which approximates the noise level of the DMM. The formula used to calculate the expected resolution is as follows.

Resolution = (Range/300000.) * (.200/Aperture)^.5

***NOTE**. The formula to calculate aperture from expected resolution is the inverse.*

Aperture = .2 * Range^2 / (300000. * Resolution)^2

For the 1.5 A current range and the 300 MΩ resistance range, use a Range of 1.5 Amps and 54 Gohms respectively, in the formulas above.

Expected resolution is only an approximation. If you are concerned about optimizing the tradeoff between aperture/ test time and accuracy, use the Accuracy Specifications in the Appendix to determine an aperture. The accuracies and apertures listed in the Appendix are tested at the factory. The accuracy at apertures in between those tested may be assumed to change logarithmically from one specified aperture to the next aperture. For example , if the accuracy for some measurement level calculates to 120 μV for a .2 second aperture, and 140 μV for a .0167 second aperture, the accuracy for a .05 second aperture may be calculated as follows:

140 μV = 120 μV + k*log(.2/.0167), solving for k, k= 18.547 μV.

.05 second aperture accuracy = 120 μV + 18.547 μV * log(.2/.05) = 131.2 μV.

***NOTE**. Before choosing an aperture that is not a power line cycle integer multiple, it should be noted that the accuracy specifications assume a quiet power environment. In many testing environments, the power line noise contributes measurable error, particularly in the most sensitive voltage ranges and in the highest resistance ranges. An aperture that is a power line cycle integer multiple is recommended.*

Finally, if autoranging is enabled, the aperture value will remain unchanged if the DMM input range changes. However, the expected resolution value will be recalculated based on the new range found.

Example. If the command SENSe:VOLT:DC:RESolution 2.8e-4 were given, and the current DC voltage range were 30; the aperture of the measurement would be:

$$\text{Aperture}= .2 \times 30. \times 30./300000 \times 2.8c{-}4)^2 = 0.025510 \text{ seconds}$$

To determine the rounded aperture, divide 0.25510 by the aperture step size of .0008333 and round the result of 30.61 to 30.

The resulting aperture is: 30 × .0008333

The SENSe[.....]? query return the requested information.

***NOTE**. If data sampling is in progress, issuing a SENSE command for any function of the configuration currently in effect, or issuing a SENSe:FUNCtion command will cause the acquisition to be aborted.*

Query Response The response is the specific requested Information

Examples

Command/Query	Response/Description
SENSe:BANDwidth: DETector MAXimum	Set Bandwidth to Maximum (100 kHz)
SENSe:BANDwidth: DETector?	100000
SENSe:BANDwidth: DETector DEF	Set Bandwidth to Default (20 Hz)
SENSe:BANDwidth: DETector?	20
SENSe:VOLTage:DC: RANGe:UPPer 5	Sets the DC V range
sens:volt:range?	30<LF> This is the rounded up range value
voltage:ac:aperture 0.5	Sets the AC V aperture value
sens:volt:ac:aper?	0.5<LF> This is the aperture value
sense:volt:rps?	2<LF> This is the number of readings per second
SENSE:CURRENT: NPLCYCLES 120	Sets the DMM to 120 power line cycles
sens:curr:nplc?	120<LF>
sens:curr:a?	2<LF> Modifying the power line automatically modifies the aperture; the converse is also true
Sense:Resolution 0.1	Defines the resolution
sens:res:resolution?	56921<LF> This is the calculated resolution
sense:fres:res?	180000<LF> This is the current (default) resolution
sense:fres:range 30	Changes the 4-wire resistance range to 30

Command/Query	Response/Description
sense:fres:res?	0.0001<LF> The calculated resolution
sense:volt:range: auto 1	Enables DC V autoranging. 1 is interchangeable with ON
sense:volt:range:auto?	1<LF>

STATus? Subsystem

The STATus queries enable you to inquire on the current operational state of the instrument.

Many STATus commands are available for use for all instruments. For a summary of those commands, see the *Status and Events* section.

Command Syntax N/A

Query Syntax `STATus:OPERation:CONDition?`

Command Class Instrument

***RST Value** 0

Limits N/A

Formats Query Response
Numeric

Related Commands MEASure?
READ?
INITiate
ABORt

Query Response

The STATus:OPERation:CONDition query returns the current operational status of the DMM board. The bit definitions of the value are (bit 0 = the least significant bit):

Bit	Definition	Function
0	Calibrating	Set when any CALibration operation is running. Cleared when the CALibration operation is complete
1	Settling	Set when the instrument changes its function or range. Cleared when the all circuitry has settled
2	Ranging	Set when the instrument is autoranging. Cleared when the input range has been found
3	Sweeping	Not used
4	Measuring	Set when the INITiate command is executed. Cleared when the command is complete or aborted
5	Triggering	Set when the instrument is waiting for a trigger signal. Cleared when the trigger is received
6	Arming	Not used
7	Correcting	Set when the instrument is performing an autozero operation. Cleared when the autozero operation is complete
8	Testing (User 1)	Set when the instrument is performing a self-test. Cleared when the self-test is complete
9	Aborting (User 2)	Set when the instrument is in the process of aborting an operation. Cleared when the abort is complete
10	User 3	Not used
11	User 4	Not used
12	User 5	Reserved
13	Instrument Summary	Not used
14	Program Running	Not used
15	Reserved	Always 0

Examples

Command	Response
status:operation:condition?	16 Measurement in progress (0010 hex).
stat:oper:cond?	3072 Measurement complete because an ABORt was received (0C00 hex).

TEST Subsystem

The TEST queries enable you to initiate an instrument self-test.

Command Syntax N/A

Query Syntax `TEST:ALL?`

Command Class Instrument

***RST Value** N/A

Limits N/A

Related Commands *TST

Query Response This query is used to perform a self test of the DMM. If the test fails, an error message is placed in the error queue and the error LED begins to blink. The following are tested:

- Acquisition memory and control logic
- Programmable aperture circuitry
- Autozero test covering the following:

  DC amplifier
  AC amplifier
  TRMS converter
  A/D converter
  Serial to parallel conversion circuitry

Examples

Command	Response
INST:SEL DMM	Selects the DMM
TEST:ALL?	“DMM: Self-Test Passed” “DMM: Self-Test Failed”

TRIGger Subsystem

Command Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay <time delay>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt <event delay>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest <polarity>
   :STRobe <polarity>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:IMMediate

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce <source>
```

Query Syntax

```
TRIGger([:SEQuence1]|:STARt)[:LAYer]:DELay?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity

   :REQuest?
   :STRobe?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?
```

Command Class Instrument

***RST Value**

Parameter	Value
<time delay>	0 seconds
<event count>	0 triggers (pass-through)
<polarity>	INVerted
<mode>	ALL
<source>	IMMEDIATE

Limits

Parameter	Value
<time delay>	0 seconds
<event count>	0 = pass through, 1 to 65,535 triggers
<polarity>	INVerted
<mode>	ONCE|ALL
<source>	BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

Related Commands N/A

Description

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay <time delay>
Specifies a time delay to occur after receipt of a trigger prior to actually triggering. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the event count delay, so specifying a delay of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt <event delay>
Specifies the number of triggers to count prior to triggering. Upon receipt of trigger N (where N is the number specified in the command), the instrument will enter the triggered state. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the delay by time parameter, so specifying an event count of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity:REQuest|:STRobe <polarity>
Specifies the active edge of the signals to be received and driven on the external handshake lines. NORMal indicates that the rising edge is active. INVerted indicates that the falling edge is active.

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate
Causes a one time entry into the triggered state without receiving the specified trigger. This command is often used to “prime the pump” in cases such as setting up a scan list measurement.

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE <mode>
When an instrument has been configured for some type of array measurement, this command specifies whether the instrument will perform one or all operations when a trigger is received. If the mode is ALL, then all operations will be completed upon receipt of one trigger condition. If the mode is ONCE, then the instrument will perform one operation and reenter the initiated state. This will continue until the specified number of triggers has been received (and hence, the specified number of operations has been completed).

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce <source>
Selects or queries the trigger source to be used when the instrument is initiated.

Query Response

Query	Response
TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay?	<delay in seconds>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:ECOunt?	<triggers to count>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE?	Current state: ONCE|ALL
TRIGger([:SEQuence1]|:STARt)[:LAYer]:HANDshake:POLarity:REQuest|STRobe?	NORMal | INVerted
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce?	<current source>
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?	Lists all trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
NOTE: *Depending upon which options were purchased with the VX4101A, some of the trigger sources listed above may not be available for use.*	
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?	Lists all delayable trigger sources available for use with the TRIG:SOUR command. BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:FIXed?	Lists all fixed trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, TIMER, HANDSHAKE

Examples

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:DELay

Command	Response
INST:SEL DMM	Selects the DMM
CONF:VOLT:DC	Configures the DMM for DC measurements
TRIG:SOUR TTLT0	Selects TTLT0 as the trigger source
TRIG:DEL 1E-3	Sets the trigger delay
INIT	Initiates the measurement

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:ECOunt

Command	Response
INS:SEL DMM	Selects the DMM
CONF:VOLT:DC	Configures the DMM for DC measurements
TRIG:SOUR TTLT0	Selects TTLT0 as the trigger source
TRIG:ECO 100	Sets the event count to 100
INIT	Initiates the measurement

TRIGger ([:SEQuence1] |:STARt)[:LAYer]:IMMediate

Command	Response
TRIG:SOUR:CAT?	IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
INST:SEL DMM	Selects the DMM
CONF:VOLT:DC	Configures the instrument for DC measurements
TRIG:SOUR TTLT0	Selects TTLT0 as the trigger source
INIT	Initiates the measurement
FETC:COUN?	0 Returns the number of measurements completed
TRIG:IMM	Sets triggering to immediate
FETC:COUN?	1 Returns the number of measurements completed. This could vary depending on the actual measurements.
F?	–6.03720E+00

TRIGger([:SEQuence1]|:STARt)[:LAYer]:MODE

Command	Response
INST:SEL DMM	Selects the DMM
CONF:ARR:VOLT:DC 3	Configures for 3 DC voltage measurements
TRIG:MODE ONCE	Sets up the instrument to make one measurement and then re-enter the initiated state
TRIG:SOUR COMM0	Selects COMM0 as the trigger source
INIT	Starts the measurement
FETC:COUN?	0
TRIG:FIR0	
FETC:COUN?	1
TRIG:FIR0	
TRIG:FIR0	
FETCH:COUN?	3
FETC?	#239–6.04180E+00,–6.04180E+00,–6.04180E+00

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce

Command	Response
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE
INST:SEL DMM	
CONF:VOLT:DC	
TRIG:SOUR BUS	
INIT	
FETC:COUN?	0
*TRG	
FETC:COUN?	1
FETC?	–6.03720E+00

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATAlog[ALL]?

Command	Response
INST:SEL DMM	
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO, HANDSHAKE

TRIGger([:SEQuence1]|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

Command	Response
TRIG:SOUR:CAT:DEL?	BUS, EXTERNAL, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, DAC, DIGI, DIGO, CTR_EXTARM

TRIGger([:SEQuence1] |:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Command	Response
INST:SEL:DMM	
TRIG:SOUR:CAT:FIX?	HOLD,IMMEDIATE,TIMER

VXI:FDC Subsystem

These commands allow the setup and use of the instrument's Fast Data Channel (FDC).

Command Syntax

```
VXI[:SERVant]:FDC:CLOSe [<channel number>]

VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]

VXI[:SERVant]:FDC:SEL <channel number>
```

Query Syntax

```
VXI[:SERVant]:FDC?

VXI[:SERVant]:FDC:CONFiguration? [channel number]

VXI[:SERVant]:FDC:SEL?
```

*RST Value

Parameter	Value
<channel number>	1
<channel mode>	All closed FDC channels are opened

Limits

Parameter	Value
<channel number>	1 to total number of instrument FDC channels
<channel mode>	WO

Command Description

VXI[:SERVant]:FDC:CLOSe [<channel number>]
This command closes the selected logical FDC channel. If the optional channel number is omitted, the channel referenced by the VXI:FDC:SEL command is used. Closed FDC channels must be opened before they can be accessed for data transfer. The channel number parameter is the logical FDC channel number for the instrument. It is not necessarily related to the physical FDC channel.

VXI[:SERVant]:FDC:OPEN <channel mode>[,<channel number>]
This command opens the selected logical FDC channel. The mode parameter determines the direction of data flow, read only, write only or both read & write. The mode parameter can be one of the following unquoted SCPI labels: RO, WO, RW. Interpretation of Read and Write is from the servant device's point of view. Data is *READ* off the VXIbus back plane into the instrument and *WRITTEN* from the instrument to the VXIbus back plane. The DMM only supports the WO mode. If the optional channel number is omitted, the channel referenced by the VXI:FDC:SEL command is used. FDC channels must be opened before they can be accessed for data transfer.

VXI[:SERVant]:FDC:SEL<channel number>
The command selects the FDC logical channel to be used by subsequent FDC commands. The channel number parameter is the logical FDC channel number for the instrument. It is not necessarily related to the physical FDC channel. The power up default is logical channel 1.

Query Response

VXI[:SERVant]:FDC?
This query returns comma separated list of physical FDC channel numbers allocated to the instrument. Physical and logical channel numbers have a 1 to 1 correspondence. Channel assignment occurs as follows:

> Logical channel 1 is established on the first physical channel number returned by the query.

Logical channel 2 is established on the second physical channel number returned by the query. This process continues until all logical channels are established on their corresponding channel number.

Physical FDC channel allocation is dependent upon the VX4101A option configuration. Knowledge of a logical FDC channel's physical identity is required by users writing their own low level Commander-side FDC drivers for their host computer. For more information, see *FDC Operation* in *Instrument Functions*

VXI[:SERVant]:FDC:CONFiguration? [<channel number>]
This query returns the configuration state of the selected FDC channel. If the optional channel number is omitted, the channel referenced by the VXI:FDC:SEL command is used. Returned values are the following unquoted ASCII strings:

"CLOSED", "OPEN", "INITIALIZED", "READ ONLY", "WRITE ONLY", "READ_WRITE"

Configuration states of READ ONLY, WRITE ONLY and READ WRITE imply an OPENED and INITIALIZED state. A channel that is in a Closed state must be Opened before it can be accessed. In practice, a channel will be in one of the four active states: (CLOSED, READ ONLY, WRITE ONLY, or both).

The <channel number> parameter is the logical FDC channel number for the instrument.

VXI[:SERVant]:FDC:SEL?
This query returns the currently selected logical FDC channel. The return value is a single integer, from 1 to the number of logical FDC channels the instrument supports.

Examples

VXI[:SERVant]:FDC?

Command	Response/Description
VXI:FDC:SEL 1	Selects logical FDC channel 1 for access.
VXI:FDC:Open WO	Opens FDC channel 1 for write only operation
VXI:FDC?	2,4 Indicates physical FDC channel 2 is implemented on this devices logical channel 1; physical channel 4 on logical channel 2
VXI:FDC:CONF? 1	WRITE ONLY Indicates channel 1 is Opened, Initialized, and Write only.
VXI:FDC:CLOSe	Closes FDC channel 1

SCPI Commands for the SurePath™ Modules

This section lists the SCPI commands and queries for the SurePath™ Scanner master. This section is not a comprehensive listing of all SurePath™ commands. The commands listed in this section apply specifically to the VX4330 120-Channel Relay Multiplexor Module. The commands active in your specific module will vary with each model.

For commands specific to each individual SurePath™ module, consult the appropriate Tektronix manual for that module.

Command Summary

The following is a listing of the available command subsystems and syntax:

INITiate Subsystem

Commands

```
INITiate

    :CONTinuous [0|OFF|1|ON]

    [:IMMediate]
```

Queries

```
INITiate:CONTinuous?
```

INSTrument Subsystem

Commands

```
INSTrument:ABORt
INSTrument:RESet
```

ROUTe Subsystem

Commands

```
[ROUTe:]CLOSe <channel list>

    :DWELl <module name>, <nrf>
    :MODE <mode>,<module_name>,<section_list>

[ROUTe:]Configure <configuration>,<module_name>,<section_list>

    :DISJoin <module name>
    :JOIN <module_name>,<section_list>
```

```
[ROUTe:]MODule

    :DELete[:NAME] <module_name>
    :DELete:ALL
    [:DEFine] <module_name>,<nrf>

[ROUTe:]OPEN <channel_list>

    :ALL [module_name]
    :DWELl <module_name>,<nrf>

[ROUTe:]PFAil <action_at_powerfail>
[ROUTe:]SCAN <channel_list>

    :RATE <scan_rate>,<module_name>
```

Queries

```
[ROUTe:]CLOSe? <channel_list>
[ROUTe:]ID?
[ROUTe:]MODule[:DEFine]? <module_name
[ROUTe:]OPEN? <channel_list>
[ROUTe:]MODule:CATalog?
[ROUTe:]MODule:CATalog:SUPPorted?
```

STATus Subsystem

Queries

```
STATus:OPERation:CONDition?
```

TEST Subsystem

```
TEST:ALL?
```

TRIGger Subsystem

Commands

```
TRIGger([:SEQuence]|:STARt)[:LAYer]

    :DELay <delay in seconds>
    :COUNt <count>
    :ECOunt <triggers to count>
    :IMMediate
    :SOURce <source>
```

Queries

```
TRIGger([:SEQuence]|:STARt)[:LAYer]

    :COUNt?
    :DELay?
    :ECOunt?
    :MODE?
    :SOURce?
```

```
:CATalog[:ALL]?
:CATalog:DELayable?
:CATalog:FIXed?
```

***NOTE**. The Examples used in the command summaries are for illustrative purposes only and apply only to the VX4380 model of the SurePath™ Modules.*

INITiate Subsystem

These commands cause the current TRIGger command sequence to begin.

Command Syntax

```
INITiate

   :CONTinuous] <control>
   [:IMMediate|
```

Query Syntax

```
INITiate:CONTinuous?
```

Command Class

Instrument

***RST Value**

Parameter	Value
<control>	OFF

Limits

Parameter	Value
<control>	0|1|OFF|ON
***NOTE:** 0=OFF, 1=ON*	

Related Commands

ABORt
INSTrument:ABORt

Description

The INItiate commands perform the following operations:

INITiate:CONTinuous <control>
This command initiates the current trigger sequence. After the instrument has completed the current trigger sequence, it enters the initiated state a second time.

It will continue this cycle until an abort, reset, or INIT:CONT OFF command is received.

INITiate:IMMediate
This command initiates the current trigger sequence. After the instrument has completed the current trigger sequence, it enters the idle state.

Query Response

Command	Response
INITiate:CONTinuous?	0 = continuous initiate not enabled 1 = continuous initiate enabled

Examples See TRIGger([: SEQuence1] \|:STARt) [:LAYer]:IMMediate command for an example.

INSTrument Subsystem

The INSTrument commands control the current state of the modules. You can cause the instrument to cease making measurements, and either return to its default conditions or to retain the current setup.

Command Syntax

```
INSTrument:ABORt
INSTrument:RESet
```

Query Syntax N/A

Command Class Instrument

SurePath™ Module Valid Commands for all SurePath™ modules

Query Response N/A

***RST Value** N/A

Limits N/A

Related Commands
ABORt
INITiate
INITiate:CONTinuous

Command Description

INSTrument:ABORt

Places active instrument in the IDLE state and ceases any measurement or other instrument activity in progress. The instrument configuration is unchanged. A subsequent INIT command will cause the instrument to re-start the same type of measurement.

If the instrument is in Asynchronous mode, this command can be sent while a query is in progress and the measurement will be aborted.

If the instrument is in Synchronous mode, this command will be queued while a query is in progress.

NOTE*. After abort, no more measurements are taken.*

INSTrument:RESet

Resets the currently selected instrument without affecting other instruments. The instrument returns to its *RST state. The instrument remains selected.

Examples

INSTrument:ABORt

Command	Response
INST:SEL DMM	
CONF:RES	
CONF?	“:SCAL:RES 3e+08,180000”
INST:RES	
INST:SEL?	DMM
CONF?	“:SCAL:VOLT:DC 300,0.001”

INSTrument:RESet

See the TRIGger([: SEQuence1] |:STARt) [:LAYer]:IMMediate command for an example.

ROUTe Subsystem

These commands determine the specific path that signals will take through the specific modules. The parameters used in SCPI/IEEE 488.2 commands and command descriptions for the `[ROUTe:]` commands are as follows:

<NR1>
ASCII integer representation of a decimal number.

<NRf>
ASCII integer, fixed point or floating point representation of a decimal number.

<module_name>
A user-defined ASCII string to be associated with the local bus address of a relay module. <module_name> strings must start with a letter and may consist of alphanumeric characters, underscores, and digits. The maximum length of a <module_name> is 12 characters.

<channel_spec>
One or more <NR1> ASCII strings separated by “!” characters that specify a relay on a relay module. The format of a <channel_spec> field for each of the SurePath relay modules is:

- VX4320 RF Multiplexor: <NR1> ! <NR1>

  The range of the first <NRf> field is 1 to 4. This field specifies a relay within one of the sections of the VX4320. The range of the second <NRf> field is 1 to 8. This field specifies a section of the VX4320. A one-dimensional <channel_spec> may also be used to specify a channel on a VX4320 Module. The one-dimensional <channel_spec> is given by the formula:

  $$((\text{section} - 1) \times 4) + \text{relay}$$

  where variables “section” and “relay” are section and relay numbers specified in a two-dimensional <channel_spec>.

- VX4330 Scanner/Multiplexor: <NR1> ! <NR1>

  The first <NR1> field specifies a relay within the specified section. The range of this <NR1> field depends on the current configuration of the section of the VX4330 specified in the second <NR1> field. The range of the second <NR1> field is 1 to 6. This field specifies a section of the VX4330.

1 – 10	4-wire
1 – 20	4-wire independent
1 – 20	2-wire
1 – 40	1-wire

- VX4350 General Purpose Switching module: <NR1>

  The range of this field is 1 to 64. It specifies one of 64 relays on the VX4350.

- VX4351 40-Channel, 10 Amp, SPST Switch Module: <NRI>

  The range of this field is 1 to 40, specifying one of the 40 relays on the module. If the module is placed in two-wire mode, then the range of the field becomes 1 to 20, specifying one of the twenty relay pairs available on the module.

- VX4380 Matrix: <NR1> ! <NR1> ! <NR1>

  The range of the first <NR1> field is 1 to 4. It specifies the row of a relay in one of the sections of the VX4380. The range of the second <NR1> field is 1 to 16. It specifies the column of a relay in one of the sections of the VX4380. The range of the third <NR1> field is 1 to 4. It specifies a section of the VX4380. A one dimensional <channel_spec> may also be used to specify a channel on a VX4380 Module. The one dimensional <channel_spec> is given by the formula:

  $$((\text{section} - 1) \times 64) + ((\text{row} - 1) \times 16) + \text{column}$$

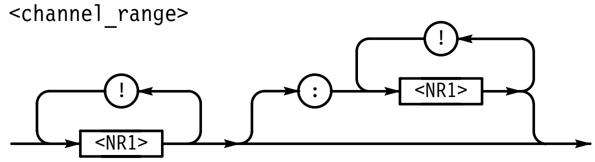
  where variables “section” and “row” and “column” are section, row, and column numbers specified in a three-dimensional <channel_spec>.

<channel_list>
A list of channel numbers on one or more relay modules.

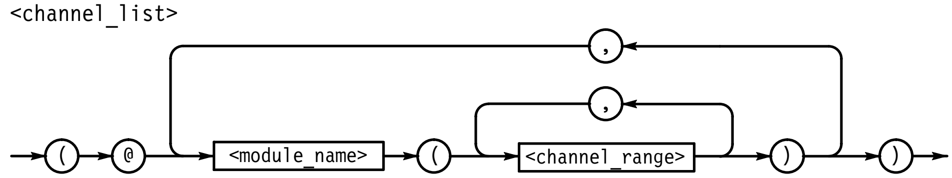
The limits on the channel numbers in a <channel_list> depend on the model number of the relay module(s) specified in the <channel_list>.

The syntax of a <channel_list> is shown in the following diagram:

<channel_range>



<channel_list>



A module_name is an ASCII string that has been associated with a relay module in a ROUTe:MODule:DEFine command. A channel_range may consist of a single <channel_spec> or a range of <channel_spec>s. A range of <channel_spec>s is indicated by two <channel_spec>s separated by a colon (:) character. A <channel_spec> may have one, two or three dimensions depending on the architecture of the relay module it applies to.

The following are examples of valid <channel_list>s for Tektronix VX4320, VX4330, VX4350, and VX4380 relay modules. In these examples it is assumed that the Option 01 is installed on a VX4320 Module. A VX4330, VX4350, and VX4380 are installed in consecutive slots to the right of the VX4320. The default module names for the VX4320, VX4330, VX4350, and VX4380 in this example are m1, m2, m3, and m4 respectively. These module names may be overridden by specifying new module names with the [ROUTe:]MODule[DEFine] command. It is also assumed in these examples that all sections on the VX4330 Module have been configured as 10-to-1 4-wire scanners.

<channel_list>	Channels Specified
(@m1(1!2))	Channel number 1 of section 2 on the VX4320 Module
(@m1(4!1,3!8))	Channel number 4 in section 1 and relay number 3 in section 8 of the VX4320 Module
(@m1(4!1:4!8))	Channel number 4 in sections 1 through 8 on the VX4320 Module
(@m1(4!1,4!2,4!3,4!4, 4!5,4!6,4!7,4!8))	Channel number 4 in sections 1 through 8 on the VX4320 Module
(@m2(1!6))	Channel 1 in section 6 of the VX4330 Module
(@m2(1!1:10!6))	All channels on the VX4330 Module in the following order: 1!1, 1!2, 1!3, 1!4, 1!5, 1!6, 2!1, 2!2, 2!3, 2!4, 2!5, 2!6, ... , 10!1, 10!2, 10!3, 10!4, 10!5, 10!6
(@m2(1!3:10!3))	All 10 channels in section 3 of the VX4330
(@m3(1:64))	All 64 channels on the VX4350
(@m3(1,2,3,10,11, 20:13))	Channels 1, 2, 3, 10, 11, and 20 through 13 on the VX4350
(@m4(1!13!3))	The channel that connects row 1 to column 13 in section 3 of the VX4380
(@m4(65))	The channel that connects row 1 to column 1 in section 2 of the VX4380
(@m4(1!1!2))	Same as the previous example
(@m4(1:16))	The channels that connect columns 1 through 16 to row 1 in section 1 of the VX4380

<channel_list>	Channels Specified
(@m4(1!1!1:1!16!1))	Same as the previous example
(@m4(1!1!1:2!3!4))	Channels 1!1!1, 1!1!2, 1!1!3, 1!1!4, 1!2!1, 1!2!2, 1!2!3, 1!2!4, 1!3!1, 1!3!2, 1!3!3, 1!3!4, 2!1!1, 2!1!2, 2!1!3, 2!1!4, 2!2!1, 2!2!2, 2!2!3, 2!2!4, 2!3!1, 2!3!2, 2!3!3, 2!3!4 on the VX4380 Module

As the <channel_list> syntax diagram shows, channels on more than one relay module may be specified in a <channel_list>. The next example specifies channels on three different relay modules:

<channel_list>	Channels Specified
(@m1(1!1), m2(4!6), m4(3!13!2))	Channel 1 of section 1 on the VX4320, Channel 4 of section 6 of the VX4330, and the channel on the VX4380 that connects row 3 to column 13 in section 2.

As mentioned above, the module names used in a <channel_list> may be specified with a [ROUTe:]MODule[:DEFine] command. The command

```
route:module:Define rfmux, 1
```

changes the module name assigned to the VX4320 to “rfmux”. The following <channel_list> can then be used to specify channels on the VX4320.

```
(@rfmux(3!1,2!2))
```

specifies channel 3 in section 1 and channel 2 in section 2 of the VX4320.

The order in which channels are specified is important in the [ROUTe:] CLOSE? <channel_list> and [ROUTe:]OPEN? <channel_list> queries. The states of the channels are returned in the same order that the channels are specified in the <channel_list>.

The order in which channels are specified is also important in the [ROUTe:] SCAN <channel_list> command. This determines the order in which the relays will be closed each time a trigger event is detected.

The order in which channels are specified in a <channel_list> is important in the [ROUTe:]CLOSe <channel_list> command when channels in the same section of a VX4320 or a VX4330 are specified. A VX4320 can close only one channel in a section. If a [ROUTe:]CLOSE <channel_list> command specifies more than one relay in a section of a VX4320, the last channel in the <channel_list> will be closed.

For example, the command

```
close (@m2(1!1,2!1))
```

will close channel 2 of section 1 of the VX4320.

A VX4330 can close only one channel in a group of joined sections that have been specified in a [ROUTe:]CLOSe:MODe SCAN,<module_name>,<section_list> command. If more than one channel in such a group of sections is specified in a [ROUTe:]CLOSe <channel_list> command, the last channel specified will be closed. For example, the commands

```
route:configure:join m2,(1:6)
```

```
route:close:mode scan,m2,(1:6)
```

join the commons of all six sections of the VX4330 Module and set the mode of the [ROUTe]:CLOSE <channel_list> to scan mode for all six sections of the VX4330.

The command

```
route:close (@m2(1!1,1!6))
```

will then result in channel 1 of section 6 being closed and all other channels on the module being opened.

<section_list>
One or more <nr1> fields separated by comma (,) or colon (:) characters and enclosed in left and right parentheses. A <section_list> is used to specify the sections of a relay module to be acted upon by a [ROUTe:]CONFigure or [ROUTe:]CLOSe:MODE command. The following are examples of valid <section_list>s.

For commands directed to a VX4330 which has six scanner sections:

<section_list>	Sections Specified
(1:6)	Sections 1 through 6
(1,2,3)	Sections 1, 2 and 3
(1:3,5:6)	Sections 1 through 3 and 5 and 6
(1:3,5,6)	Same as previous example
(3)	Section 3

Command Syntax

```
[ROUTe:]CLOSe <channel_list>
    :DWELl <module_name>, <nrf>
    :MODE <mode>,<module_name>,<section_list>
```

```
[ROUTe:]Configure <configuration>,<module_name>,<section_list>

   :DISJoin <module name>
   :JOIN <module_name>,<section_list>

[ROUTe:]MODule

   :DELete[:NAME] <module_name>
   :DELete:ALL
   [:DEFine] <module_name>, <nrf>

[ROUTe:]OPEN <channel_list>

   :ALL [module_name]
   :DWELl <module_name>, <nrf>

[ROUTe:]PFAil <action_at_powerfail>

[ROUTe:]SCAN <channel_list>

   :RATE <scan_rate>,<module_name>
```

Query Syntax

```
[ROUTe:]CLOSe? <channel_list>

[ROUTe:]ID?

[ROUTe:]MODule

   [:DEFine]? <module_name>
   :CATalog?
   :CATalog:SUPPorted?

[ROUTe:]OPEN? <channel_list>
```

SurePath™ Module

The following table shows which command is used for a specific SurePath™ module.

Command	Module
[ROUTe:]CLOSe <channel_list>	Valid for all SurePath™ modules.
[ROUTe:]:DWEL1 <module_name>,<nrf>	Valid for all SurePath™ modules.
[ROUTe:]CLOSe:MODE	Valid for VX4330 modules only.
[ROUTe:]CONFigure	Valid for VX4330, VX4351 only.
[ROUTe:]CONFigure:DISJoin	Valid for VX4330 modules only.
[ROUTe:]ID?	Valid for all SurePath™ modules.
[ROUTe:]CONFigure:JOIN	Valid for VX4330 modules only.

Command	Module
[ROUTe:]MODule:CATalog?	Valid for all SurePath™ modules.
[ROUTe:]MODule[:DEFine]	Valid for all SurePath™ modules.
[ROUTe:]MODule:DELete:ALL	Valid command for all SurePath™ modules.
[ROUTe:]MODule: DELete[:NAME]	Valid command for all SurePath™ modules.
[ROUTe:]OPEN<channel list>	Valid command for VX4330, VX4350, VX4351, and VX4380 SurePath™ modules
[ROUTe:]OPEN:ALL	Valid command for VX4330, VX4350, VX4351, and VX4380 SurePath™ modules
[ROUTe:]OPEN:DWELl	Valid command for VX4330, VX4350, VX4351, and VX4380, SurePath™ modules
[ROUTe:]PFAil <action_at_powerfail>	Valid command for VX4330, VX4350, and VX4380 SurePath™ modules
[ROUTe:]SCAN<channel_list>	Valid command for VX4330, VX4350, VX4351, and VX4380 SurePath™ modules
[ROUTe:]SCAN:RATE	Valid command for the VX4330 only

*RST Value

Parameters	Value
<channel_list>	Open
[ROUTe:]:DWEL1 <module_name>,<nrf>	0 seconds
[ROUTe:]CLOSe:MODE	All sections of the SurePath™ VX4330 Module are set to operate in the mux mode. One or more channels in a section may be closed at the same time.
[ROUTe:]CONFigure	The section numbers specified in the <section_list> portion of this command must be between 1 and 6. The configurations of each applicable instrument must be: VX4330: FWIRE VX4351: OWIRE
[ROUTe:]CONFigure:DISJoin	All sections on the VX4330 module are disjoined.
[ROUTe:]ID?	N/A
[ROUTe:]CONFigure:JOIN	All sections on all modules are disjoined.
[ROUTe:]MODule:CATalog?	N/A
[ROUTe:]MODule[:DEFine]	Default module names are assigned as follows: The module that is immediately to the right of the SurePath™ Master. Consecutive slots to the right of module “M1” are assigned module names “M2”, “M3”, ..., “M11”.

Parameters	Value
[ROUTe:]OPEN<channel list>	All relays on all modules are set to the open position.
[ROUTe:]OPEN:ALL	All relays on all modules are set to the open position.
[ROUTe:]OPEN:DWELl	The open dwell time of all modules is set to 0 seconds.
[ROUTe:]PFAil <action_at_powerfail>	All relays on all modules are opened when power is removed from the VXI chassis.
[ROUTe:]SCAN <channel_list>	No scan list is defined.
[ROUTe:]SCAN:RATE	The VX4330 scan rate is set to normal. The minimum time to sequence to the next channel in a scan list is approximately 16 ms. The SurePath™ master on the VX4101A runs on a multitasking system. The accuracy of the scan time depends on the load placed on the VX4101A. Each time a channel is opened or closed, front panel encode signals are generated, and the control signals applied to all relays on the module are verified.

Limits

For limits, look up the appropriate command in the user manual for your SurePath™ module.

***NOTE**. The limits of this command depend on the SurePath™ Module you are using. Consult the manual of your module to determine the limits.*

Related Commands

ABORt
OUTPut:TTLTrg<n>[:STATe]
TRIGger[:SEQuence]:SOURce
TRIGger[:SEQuence]:COUNt
TRIGger[:SEQuence]:DELay
TRIGger[:SEQuence]:[IMMediate]
INITiate:CONTinuous
INITiate[:IMMediate]

Command Description

The ROUTe commands perform the following operations:

[ROUTe:]CLOSe <channel_list>

The action taken when this command is received depends on the close mode that has been assigned to the sections specified in the <channel_list>. The close mode of a section is set to either “mux” or “scan” by the [ROUTe:]CLOSe:MODE command. At power-on or after an *RST command or SYSTem:PREset command, the close mode of all VX4330 sections is set to “mux”. If the close mode of a section has been set to “mux”, then a [ROUTe:]CLOSE command can be used to close one or more relays in that section.

[ROUTe:]CLOSe:DWELl <module_name>,<nrf>
Sets the time to wait after closing a relay before proceeding and pulsing any enabled VXI TTL trigger signals.

[ROUTe:]CLOSe:MODE <mode>,<module_name>,<section_list>
Sets the mode of operation of sections on a VX4330 to scan or mux. When this command is received, all relays in the sections specified in the <section_list> argument are opened. In the mux mode, one or more relays in a section may be closed at the same time. When a relay is closed in a section that has been set to scan mode, all relays in that section are opened before the relay is closed. If several sections are set to operate in scan mode and the commons of these sections are joined (see the [ROUTe:]CONFigure:JOIN command) then when a relay is closed in any of the sections, all relays in all of the sections are opened first.

[ROUTe:]CONFigure <configuration>,<module_name>,<section_list>
Each section of a SurePath™ VX4330 Module may be configured as a 40-to-1 1-wire scanner, or as a 20-to-1 2-wire scanner, or as a 10-to-1 4-wire scanner. Also, each section may be configured as a 10-to-1 4-wire scanner with independent control of the upper and lower halves of the 4-wire common. The <configuration> portion of this command indicates which of these configurations is selected according to the following table:

<configuration>	Configuration
OWIRE	40-to-1 1-wire scanner/mux
TWIRE	20-to-1 2-wire scanner/mux
FWIRE	10-to-1 4-wire scanner/mux
FWIRI	10-to-1 4-wire scanner/mux with independent control of the upper and lower halves of the 4-wire common

The <module_name> argument of the command indicates to which module the command is directed. The specified <module_name> is associated with a relay module with a [ROUTe:]MODule:DEFine command.

The <section_list> argument indicates which sections of the selected module are to be configured. All relays in the section(s) specified in this list are opened when the [ROUTe:]CONFigure command is received.

[ROUTe:]CONFigure:DISJoin <module_name>
Disconnect the commons of all sections of a module.

[ROUTe:]CONFigure:JOIN <module name>, <section_list>
This command connects the commons of adjacent sections on scanner modules. On VX4330 Modules, if a section that is configured as a 4-wire scanner is joined to a section that is configured as a 1-wire or 2-wire scanner, only the lower half of the 4-wire scan common is connected. If a 4-wire section is joined to another 4-wire section, both halves of the 4-wire scan common are joined. Also for the VX4330, if two or more sections that are set to scan mode are joined, when a [ROUTe:]CLOSe command closes a relay in one of these sections, all relays in the sections are opened before the specified relay is closed.

List of the model numbers of the modules controlled by the SurePath™ Master. The first model number returned is that of the module that the Scanner Master is installed on. Subsequent model numbers are those of modules in consecutive slots to the right of the first module.

[ROUTe:]MODule[:DEFine] <module_name>, <nrf>
This command assigns a module name to a relay switching module. This name is used to identify the module in channel lists in [ROUTe:]OPEN, [ROUTe:]CLOSe, and [ROUTe:]SCAN commands.

[ROUTe:]MODule:DELete:ALL
This command will delete all module name definitions.

[ROUTe:]MODule:DELete[:NAME] <module_name>
This command deletes a module name definition. After this command is executed, the specified module name is no longer associated with a relay module.

[ROUTe:]OPEN <channel list>
This command closes the relays specified in the <channel_list> portion of this command.

After these relays are opened, a delay specified in a previously issued [ROUTe:]OPEN:DWELl command is generated. This command is used to assign an open delay to each module controlled by an SurePath™ Master daughter board. If more than one module is specified in the <channel_list> argument of the [ROUTe:]OPEN command, the longest dwell time assigned to any of the specified modules is used.

[ROUTe:]OPEN:ALL <module_name>
If a module name is not specified in this command, open all relays on all modules controlled by the SurePath™ Master. If a module name is specified, open all relays on the specified module only. In either case, do not change the state of the configuration relays on the modules.

[ROUTe:]OPEN:DWELl <module_name>, <nrf>
This command sets the time to wait after opening a relay before proceeding.

[ROUTe:]PFAil <action_at_powerfail>
This command specifies the state that all latching relays on all modules controlled by the SurePath™ Master are to be placed in when power is removed from the VXI chassis containing the modules. <action_at_powerfail> must be OPEN or SAME. If OPEN is specified, all latching relays are opened at power fail. If SAME is specified, all latching relays are left in their current state at powerfail.

NOTE. *VXI chassis +5 V power is maintained for 4 ms after ACFAIL is asserted, in compliance with VXI Specifications. This allows for orderly system shutdown and implementation of the PFAil OPEN option.*

[ROUTe:]SCAN <channel_list>
This command defines a list of scan relay closures to sequence. When the ROUTe:SCAN <channel_list> command is received, all relays in this list are opened. In addition to defining a scan list, a trigger source must be specified using the TRIGger[:SEQuence]:SOURce command. Trigger events are not recognized until triggers are armed by a INITiate[:IMMediate] or INITiate:CONTinuous command.

When the first trigger event is detected, the first relay in the scan list is closed. When the second trigger event is detected, the first relay is opened and the second relay is closed. When the n^{th} trigger event is detected, the $(n–1)^{th}$ relay is opened and the n^{th} relay is closed. The act of opening the $(n–1)^{th}$ relay and closing the nth relay is called sequencing the scan list. At any given time after the first trigger event is detected, only one relay in the scan list is closed.

The TRIGger[:SEQuence]:COUNt command may be used to specify the number of times to sequence through the entire scan list. The TRIGger [:SEQuence]:DELay, [ROUTe:]CLOSe:DWELl, and [ROUTe:]OPEN:DWELl commands may optionally be used to specify the time to wait after a trigger event is detected, a relay is closed or a relay is opened. A TRIGger[:SEQuence] :IMMediate command causes the scan list to be sequenced without the delay specified by a previously issued TRIGger[:SEQuence]:DELay command.

[ROUTe:]SCAN:RATE <scan_rate>, <module_name>
<scan_rate> is either NORMal or FAST.

<module_name> is the module name assigned to the VX4330.

This command controls the maximum rate at which VX4330 channels in a scan list can be sequenced. If a <scan_rate> of NORMal is specified in the [ROUTe:]SCAN:RATE command, it takes approximately 16 ms plus the sum of the close dwell and open dwell times assigned to the VX4330 and the trigger delay time to:

open the VX4330 channel that is currently closed,

close the next VX4330 channel, and

pulse the front panel encode signal(s) corresponding to the closed channel.

The control signals applied to all relays on the VX4330 Module are verified after the current channel is opened and after the next channel in the scan list is closed.

If a <scan_rate> of FAST is specified in the [ROUTe:]SCAN:RATE command, it takes approximately 8 milliseconds plus the sum of the close dwell and open dwell times assigned to the VX4330 and the trigger delay time to:

open the channel that is currently closed, and

to close the next channel in the scan list.

The VX4330 front panel encode signals are disabled and the relay control signals are not verified after a channel is opened or closed.

NOTE. *The SurePath™ master on the VX4101A runs on a multitasking system. The accuracy of the scan time varies depending on the load placed on the VX4101A*

Query Response

Query	Value
[ROUTe:]CLOSe? <channel_list>	Indicates which relays are closed.
ROUTe:MODule: CATalog:SUPPorted?	VX4320, VX4330, VX4350, VX4351, and VX4380
[ROUTe:]MODule[:DEFine]?	The name of the module.
[ROUTe:]OPEN? <channel list>	Indicates which relays are opened.
[ROUTe:]ID?	List of the model numbers of the modules controlled by the SurePath Master. The first model number returned is that of the module that the Scanner Master is installed on. Subsequent model numbers are those of modules in consecutive slots to the right of the first module.

Examples

[ROUTe:]CLOSe <channel_list>

The following sequence of commands illustrates the operation of sections that have been assigned a close mode of mux. The VX4330 in these examples has been assigned a module name of m1.

Command	Response
output:ttltrg7:State on	Enable VXI TTL trigger 7
route:close:Dwell m1,.5	Assign a close dwell of 0.5 seconds to the scanner
route:close:mode mux,m1,(1:3)	Set the close mode of sections 1 through 3 of the scanner with module name "m1" to "mux". Open all channels in sections 1 through 3
route:close (@m1(1!1:10!1,1!2:5!2,5!3:10!3))	Close channels 1 through 10 in section 1, channels 1 through 5 in section 2 and channels 5 through 10 in section 3. After these relays are closed, wait 0.5 seconds then pulse VXI TTL trigger 7 and pulse the front panel encode signals corresponding to sections 1, 2, and 3

If the close mode of a section has been set to scan, then a [ROUTe:]CLOSE command will open all relays in that section before the channel specified in the <channel_list> is closed. Only one channel at a time is closed in a section that has been assigned a close mode of scan. Also, if two or more sections that have been assigned a close mode of scan are joined with the [ROUT:]CONFigure: JOIN command, only one channel in that group of sections will be closed at a time. When a channel in one of these sections is specified in a [ROUTe:] CLOSe command, all channels in all of the joined sections will be opened before the specified channel is closed.

The following sequence of commands illustrates the interaction between the [ROUTe:]CLOSe, [ROUTe:]CLOSe:MODE and [ROUTe:]CONFigure:JOIN commands. The SurePath™ Module in these commands has been assigned module name m1.

Command/Query	Response/Description
output:ttltrg7:State on	Enable VXI TTL trigger 7
close:Dwell m1, .5	Assign a close dwell time of 0.5 seconds to the scanner
open:dwell m1, .2	Assign an open dwell time of 0.2 seconds to the scanner
route:close:mode scan,m1,(1:6)	Set the close mode of all sections of the scanner to scan. Open all channels in all six sections of this module
route:configure:join m1,(1:6)	Join the commons of all six sections of the scanner

Command/Query	Response/Description
route:close (@m1(1!1))	Open all channels in sections 1 through 6; wait 0.2 seconds; close channel 1 of section 1; wait 0.5 seconds; then pulse VXI TTL trigger 7 and pulse the front panel encode signals corresponding to section 1
route:close (@m1(10!4))	Open all channels in sections 1 through 6; wait 0.2 seconds; close channel 10 of section 4; wait 0.5 seconds then pulse VXI TTL trigger 7 and pulse the front panel encode signals corresponding to section 4

Enabled VXI TTL triggers are pulsed low for 200 ns after the close dwell time has expired after a channel is closed.

The VX4330 has two front panel encode signals for each section. If the configuration of a section is set to 40-to-1 1-wire, 20-to-1 2-wire, or 10-to-1 4-wire, both encode signals corresponding to that section are pulsed low for 4 ms after a relay is closed in that section, 5 ms after the enabled VXI TTL triggers are pulsed.

If the section is set to the 10-to-1 4-wire independent configuration, the encode signal corresponding to the lower 2-wire common is pulsed when an odd numbered channel is closed in that section. The encode signal corresponding to the upper 2-wire common is pulsed when an even numbered channel is closed in that section.

[ROUTe:]CLOSe:DWELl <module_name>,<nrf>

Command/Query	Response/Description
output:ttltrg1:state on	Enables VXI TTL trigger 1
route:close:dwell m1,.25	Sets the close dwell time for the VX4330 to 0.25 seconds
route:open:dwell m1,0.5	Sets the open dwell time for the VX4330 to 0.5 seconds
route:configure fwire,m1,(3)	Sets the configuration of section 3 to 10-to-1 4-wire. This command causes all channels in section 3 to be opened
route:close (@m1(1!3:10!3))	Closes relays 1 through 10 in section 3 of the VX4330, wait 0.25 seconds, then pulse VXI TTL trigger 1
route:configure twire,m1,(1:6)	Sets the configuration of all sections to 20-to-1 2-wire. This command causes all channels in all sections to be opened
route:scan (@m1(1!6:20!6))	Defines a scan list consisting of relays 1 through 20 in section 6 of the VX4330
trigger:Sequence: source:ttltrg2	Defines VXI TTL trigger 2 as the trigger source for the defined scan list
trigger:sequence: delay 1	Sets the trigger delay time to 1 second
initiate:immediate	Initiates the scan sequence

After this, each time the VXI TTL trigger 2 is pulsed low, the following sequence of events occurs:

1. One second delay. This is the delay specified in the trigger:Sequence:delay command.
2. The current relay in the scan list is opened.
3. 0.5 second delay. This is the delay specified in the route:open:dwell command.
4. Close the next relay in the scan list.
5. 0.25 second delay. This is the delay specified in the `route:close:dwell` command.
6. Pulse VXI TTL trigger 1 low for 200 ns.
7. Wait 5 ms then pulse the front panel encode signals corresponding to the section of the close relay. The encode signals are pulsed low for 4 ms.

[ROUTe:]CLOSe:MODE
Three additional VX4330 Modules are installed in consecutive slots to the right of this module.

Command	Response
route:close:mode mux m1,(1:6)	Set all six sections of the first VX4330 to mux mode
route:conf twire,m1,(1:6)	Configure all six sections of the first VX4330 to 20-to-1 2-wire scanner/mux
route:close (@m1(1!1:10!1))	Close the first 10 relays in section one of the first VX4330
route:close:mode scan,m3,(1:3)	Set the close mode of sections 1 through 3 of the third VX4330 to “scan”
route:conf:join m3,(1:3)	Join the commons of sections 1 through 3 of the third VX4330
route:conf owire,m3,(1:3)	Set the configuration of sections 1 through 3 of the third VX4330 to 40 to 1 one wire
NOTE: *The previous three commands configure sections 1 through 3 of the third VX4330 as a 120-to-1 1-wire VX4330. The 1-wire common of any of these sections can be used as the 120-to-1 1-wire common.*	
close (@m3(1!1))	Open all channels in sections 1 through 3 of the third VX4330, then close channel 1 of section 1.
close? (@m3(1!1:40!3))	1 0

Command	Response
close (@m3(40!3))	Open all channels in sections 1 through 3 of the third VX4330, then close channel 40 of section 3.
close? (@m3(1!1:40!3))	0 1

[ROUTe:]CONFigure

In this example, three additional VX4330 Modules are installed in consecutive slots to the right of the VX4101A.

Command	Response
route:configure fwire,m1,(1:6)	Configure all six sections of the first VX4330 as 10-to-1 4-wire scanners
route:close:mode scan,m1,(1:3)	Set the mode of operation of sections 1, 2, and 3 of the first VX4330 to scan. In this mode, only one channel in a section is closed at a time
route:close:mode mux,m1,(4:6)	Set the mode of operation of sections 4, 5 and 6 of the first VX4330 to mux. In this mode one or more channels in a section may be closed at the same time
rout:conf owire,m3,(1)	Configure section 1 of the third VX4330 as a 40-to-1 1-wire scanner
rout:conf twire,m2,(2,3)	Configure sections 2 and 3 of the second VX4330 as 20-to-1 2-wire scanners
route:configure fwire m3,(1:6)	Configure all six sections of the third VX4330 as 10-to-1 4-wire scanners with independent control of upper and lower halves of the 4-wire common
route:close (@m3(1!1))	Connect the lower half of channel 1, section 1, to the lower half of the 4-wire common. In this example, section 1 is assumed to be configured as a 10-to-1 4-wire scanner with independent control of the upper and lower halves of the 4-wire common
route:close (@m3(20!1))	Connect the upper half of the tenth channel of section 1 to the upper half of the 4-wire common. In this example, section 1 is assumed to be configured as a 10-to-1 4-wire scanner with independent control of the upper and lower halves of the 4-wire common

[ROUTe:]CONFigure:DISJoin
Two additional VX4330 modules are installed in consecutive slots to the right of the VX4101A.

Command	Response
route:configure:disjoin m1	Disconnect the commons of the first scanner

[ROUTe:]CONFigure:JOIN
Two additional VX4330 Modules are installed in consecutive slots to the right of the VX4101A.

Command	Response
route:configure:join m1,(1:3)	Connect the commons of sections 1, 2, and 3 on the first VX4330
route:close:mode scan,m2,(1:4)	Set the mode of operation of sections 1 through 4 on the second VX4330 to scan mode. In this mode, only one relay is a section is closed at a time
route:conf twire,m2,(1:4)	Configure sections 1 through 4 on the second VX4330 as 20-to-1 2-wire scanners
route:conf:join m2,(1:4)	Connect the commons of sections 1 through 4 on the second VX4330. Since these sections have been set to operate in the scan mode, and have been configured as 20-to-1 2-wire scanners, these sections now comprise a single 80-to-1 2-wire scanner
route:close (@m2(10!1))	Close channel 10 of section 1 of the second VX4330
route:close (@m2(2!4))	Close channel 2 of section 4 of the second VX4330. Since the first four sections of this module are joined and are set to operate in the scan mode, all relays in sections 1 through 4 of this module are opened before this relay is closed

[ROUTe:]ID?
A VX4330, VX4380, and VX4320 are installed in consecutive slots to the right of the slot containing the VX4350. The default module names for the VX4350, VX4380, VX4330, and VX4320 in this configuration are m1, m2, m3, and m4 respectively. These module names may be altered with the [ROUTe:]MODule:DEFine command.

Command	Response
route:id?	VX4350, VX4380, VX4330, VX4320
route:module:Catalog?	“M1”, “M2”, “M3”, “M4”
route:close (@m1(1))	Close relay number 1 on the VX4350

Command	Response
route:open:all m2	Open all relays on the VX4380
route:close (@m3(1!6))	Close relay number 1 in section 6 of the VX4330
route:close (@m4(3!1:3!8))	Close relay number 3 in all eight sections of the VX4320

[ROUTe:]MODule:CATalog?

A VX4380 and VX4330 are installed in consecutive slots to the right of the slot containing the VX4350.

Command	Response
route:module:catalog?	"M1", "M2", "M3"
route:module:define matrix_ 1,2	Assign module name "matrix_1" to the VX4380
route:module:catalog?	"M1", "MATRIX_1", "M3"
route:module:define? matrix_1	2
route:module:delete matrix_1	Delete module name "matrix_1"
route:module:catalog?	"M1", "M3"

[ROUTe:]MODule:CATalog:SUPPorted?

This query returns a list of SurePath™ modules that are supported in this version of firmware.

Command	Action	Returns
inst:sel SurePath	Selects the SurePath™ master	
rout:mod:cat:supp?	Returns SurePath™ module supported.	VX4320, VX4330, VX4350, VX4351, and VX4380

[ROUTe:]MODule[:DEFine]

A VX4380 and VX4330 are installed in consecutive slots to the right of the slot containing the VX4350.

Command	Response
route:module:define gp_switch,1	Assign module name "gp_switch" to the VX4350
route:close (@gp_switch(1:64))	Close all 64 relays on the VX4350

Command	Response
route:module:define matrix,2	Assign module name “matrix” to the VX4380
route:close (@matrix(4!16!3))	Close the relay at row 4, column 16 in section 3 of the VX4380
route:module:define? gp_switch	1
module:define scanner,3	Assign module name “scanner” to the VX4330
open:all scanner	Open all channels on the VX4330
route:conf owire, scanner,(1:6)	Set the configuration of all sections of the VX4330 to 40 to 1 one wire
close (@scanner(30!2))	Close channel 30 of section 2 on the VX4330
module? scanner	3

[ROUTe:]MODule:DELete:ALL

Command	Response
route:module:catalog?	“M1”, “M2”, “M3”
route:module: delete:all	Delete all module names
route:module:catalog?	“ ”

[ROUTe:]MODule:DELete[:NAME]

Command	Response
route:module:catalog?	“M1”, “M2”, “M3”
route:module:delete m1	Delete module name “M1”
route:module:catalog?	“M2”, “M3”

[ROUTe:]OPEN<channel list>

Three additional VX4330 Modules are installed in consecutive slots to the right of the first VX4101A. The default module names for these three modules are m1, m2, and m3. These module names may be altered with the [ROUTe:]MODule[:DEFine] command.

Command	Response
route:open:dwell m1,.1	Assign an open dwell time of 0.1 seconds to the first VX4330
route:open:dwell m2,.2	Assign an open dwell time of 0.2 seconds to the second VX4330
route:open:dwell m3,.5	Assign an open dwell time of 0.5 seconds to the third VX4330

Command	Response
configure owire,m1,(1:6)	Set the configuration of all six sections of the first VX4330 to 40-to-1 1-wire
configure twire,m2,(1:6)	Set the configuration of all six sections of the second VX4330 to 20-to-1 2-wire
configure fwire,m3,(1:6)	Set the configuration of all six sections of the third VX4330 to 10-to-1 4-wire
route:open (@m1(1:40))	Open channels all channels in section 1 of the first VX4330 then wait 0.1 seconds
route:open (@m2(1!2:10!2))	Open channels 1 through 10 in section 2 of the second VX4330 then wait 0.2 seconds
route:open (@m3(1!1:10!6))	Open all channels in all sections of the third VX4330 then wait 0.5 seconds. This command is equivalent to the command: route:open:all m3
open (@m1(1!1), m2(1!1),m3(1!1))	Open channel 1 in section 1 of all three VX4330 Modules then wait 0.5 seconds

[ROUTe:]OPEN:ALL

Command	Response
ROUTE:OPEN:ALL	Open all relays on all Modules controlled by the SurePath ™ Master. Do not change the state of the configuration relays on VX4330 Modules
rout:open:all	Same as the first example
route:open:all gp	Open all relays on the module that has been assigned module name “gp”. See the [ROUTe:]MODule:DEFine command

[ROUTe:]OPEN:DWELl

Command	Response
output:ttltrg1:state on	Enable VXI TTL trigger 1.
route:close:dwell m1,.25	Set the close dwell time for the VX4330 to 0.25 seconds.
route:open:dwell m1,0.5	Set the open dwell time for the VX4330 to 0.5 seconds.
route:configure twire,m1,(1:6)	Set the configuration of all sections to 20 to 1 2-wire. This command causes all channels in all sections to be opened.
route:scan (@m1(1!6:20!6))	Define a scan list consisting of relays 1 through 20 in section 6 of the VX4330.
trigger:Sequence: source ttltrg2	Define VXI TTL trigger 2 as the trigger source for the defined scan list.
trigger:sequence:delay 1	Set the trigger delay time to 1 second
initiate:immediate	Initiate the scan sequence

After this sequence, each time the VXI TTL trigger 2 is pulsed low, the following sequence of events occur:

1. One second delay. This is the delay specified in the trigger:Sequence:delay command.
2. The current relay in the scan list is opened.
3. 0.5 second delay. This is the delay specified in the route:open:dwell command.
4. Close the next relay in the scan list.
5. 0.25 second delay. This is the delay specified in the route:close:dwell command.
6. Pulse VXI TTL trigger 1 low for 3 ns.
7. Wait 5 ms then pulse the front panel encode signals corresponding to the section of the close relay. The encode signals are pulsed low for 4 ms.

[ROUTe:]PFAil <action_at_powerfail>

Command	Response
route:pfail same	Leave all latching relays in their current state at powerfail
route:pfail open	Open all latching relays at powerfail

[ROUTe:]SCAN <channel_list>

A VX4380, VX4330, and VX4350 are installed in consecutive slots to the right of the slot containing the VX4101A.

Command	Response
route:module:define gp,1	Assign module name “gp” to the VX4350
route:module:define matrix,2	Assign module name “matrix” to the VX4380
route:module:define scanner,3	Assign module name “scanner” to the VX4380
route:configure twire, scanner,(1:6)	Set the configuration of all sections of the VX4330 to 20-to-1 2-wire. This command causes all channels in all sections to be opened
route:scan (@gp(1:64), matrix(1!1!1, 2!10!3), scanner(1!1:20!1))	Define a scan list consisting of relays 1 through 64 on the VX4350, relays at row 1, column 1 of section 1 and row 2, column 10 of section 3 of the VX4380 and relays 1 through 20 of section 1 of the VX4330
trigger:sequence: source immediate	Define a trigger source of “immediate”. This means to sequence through the scan list without waiting for a trigger event

Command	Response
trigger:sequence:count 5	Sequence through the entire scan list five times
route:close:dwell gp,.5	Wait 0.5 seconds after closing a relay on the VX4350
initiate:immediate	Begin sequencing through the scan list
*OPC	Set the Operation Complete bit of the Standard Event Status register after sequencing through the scan list five times
*wai; init:cont	Wait until the scan list has been sequenced through five times, then begin sequencing through the list repeatedly until an ABORt command is received
abort	Quit sequencing through the scan list and place the trigger subsystem in the idle state

[ROUTe:]SCAN:RATE

Command	Response
scan (@m1(1!1:10!6))	Define a scan list consisting of channels 1 through 10 in each section of the VX4330 Module
trigger:count 1	Set the number of times to sequence through the scan list (after an INITiate[:IMMediate] command is received) to 1
trigger:source bus	Enable a VXI TRIGGER command as a trigger source
output:ttltrg1:State on	Enable VXI TTL trigger 1 to be pulsed each time a channel is closed
close:Dwell m1,0	Set the close dwell time associated with the VX4330 to 0
open:dwell m1,0	Set the open dwell time associated with the VX4330 to 0
trig:del 0	Set the trigger delay time to 0
scan:rate norm,m1	Set the scan rate of the VX4330 to normal
init	Initiate the scan list. Sequence through the entire scan list one time. Each time a VXI TRIGGER command is sent to the SurePath™ Master VXI Interface, it takes 16 ms to open the currently closed channel, close the next channel, pulse the corresponding VX4330 front panel encode signal and pulse VXI TTL trigger 1. The control signals applied to each relay on the scanner are verified each time a channel is opened or closed.
abort	Abort the scan list
scan:rate fast,m1	Set the scan rate of the VX4330 to FAST
trigger:sour immediate	Set the trigger source to immediate

Command	Response
initiate:continuous	Initiate the scan list. Sequence through the entire scan list repeatedly until an ABORt command is received. It takes 8 ms to open the currently closed channel, close the next channel and pulse VXI TTL trigger 1. The VX4330 front panel encode signals are set to a high logic level and the relay control signals applied to the scanner relays are not verified each time a channel is opened or closed
abort	Abort the scan list

STATus Subsystem

This subsystem lets you query the state of the status bits of the SurePath™ module.

Query Syntax `STATus:OPERation:CONDition?`

SurePath™ Module Valid command for all SurePath™ modules

***RST Value** 0

Limits N/A

Related Commands N/A

Query Response This query returns the contents of the SCPI Status Operation Condition register. For the SurePath™ Master, the value of this register is always equal to 0. The bit definitions are:

Bit	Definition	Function
0	Calibrating	Not used
1	Settling	Set when the instrument changes its function or range. Cleared when the all circuitry has settled
2	Ranging	Not used
3	Sweeping	Set when a scan list operation begins. Cleared when the list has been completely traversed or an abort occurs
4	Measuring	Not used
5	Triggering	Set when the instrument is waiting for an arm signal. Cleared when the arm is received

Bit	Definition	Function
6	Arming	Not used
7	Correcting	Not used
8	Testing (User 1)	Set when self-test is in progress
9	Aborting (User 2)	Set when the instrument is in the process of aborting an operation. Cleared when the abort is complete
10	User 3	Not used
11	User 4	Not used
12	User 5	Reserved
13	Instrument Summary	Not used
14	Program Running	Not used
15	Reserved	Always 0

Examples

Command	Response
status:operation: condition?	00000

TEST Subsystem

The TEST subsystem handles the self test operations of the instrument. The SurePath™ self test tests the control logic and data path for the SurePath™ module currently in use. The query returns pass/ fail information. In a failed situation, additional failure information can be obtained with the SYStem:ERRor? query.

Command Syntax N/A

Query Syntax `TEST:ALL?`

***RST Value** N/A

Limits N/A

Related Commands *TST?

Query Response

Initiates the Digital Input self test operation and returns one of two possible responses:

“SurePath: Self Test Passed”
“SurePath: Self Test Failed”

***NOTE**. If the self-test fails, you can obtain further information with a SYST:ERR? query.*

Examples

Command/Query	Response/Description
INSTRUMENT:SELECT SUREPATH	Selects the SurePath™ module.
TEST:ALL?	”SurePath: Self-Test Passed” Runs the self-test.

TRIGger Subsystem

This subsystem is used to control when a measurement should begin for more information on triggering, see *Instrument Functions.*

Command Syntax

```
TRIGger([:SEQuence]|:STARt)[:LAYer]

    :DELay <delay in seconds>
    :COUNt <number>
    :ECOunt <triggers to count>
    :IMMediate
    :SOURce <source>
```

Query Syntax

```
TRIGger([:SEQuence]|:STARt)[:LAYer]

    :COUNt?
    :DELay?
    :ECOunt?
    :SOURce?

        :CATalog[:ALL]?
        :CATalog:DELayable?
        :CATalog:FIXed?
```

Command Class

Instrument

SurePath™ Module

Valid command for VX4330, VX4350 and VX4380 SurePath™ modules

Query Syntax

N/A

***RST Value**

Parameter	Value
<count>	1
<delay in seconds>	0 seconds (pass-through)
<triggers to count>	0 triggers (pass-through)
<source>	IMMEDIATE

Limits

Parameter	Value
<count>	1–65535
<delay in seconds>	0=pass through, 1 ms–65.536 ms in 1 ms steps
<triggers to count>	0 = pass through 1 – 65,536 triggers
<source>	BUS,TTLGRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO

Related Commands

ABORt,
[ROUTe:]SCAN

Description

TRIGger([:SEQuence]|:STARt)[:LAYer]:COUNt <count>
Specifies the number of times the instrument will enter the waiting for trigger state prior to returning to the idle state. After an initiate, the instrument will enter the wait for trigger state until a trigger is received, at which time the trigger count will be decremented. The selected operation will take place and if the trigger count is not zero, the instrument will re-enter the wait for trigger state.

TRIGger([:SEQuence]|:STARt)[:LAYer]:DELay <delay in seconds>
Specifies a time delay to occur after receipt of a trigger command before actually triggering. If the selected trigger source is fixed, this command will have no effect on the instrument triggering. This command always zeros the event count delay, so specifying a delay of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger command.

TRIGger([:SEQuence]|:STARt)[:LAYer]:ECOunt <triggers to count>
Specifies the number of triggers to count prior to triggering. Upon receipt of trigger N (where N is the number specified in the command), the instrument will enter the triggered state. If the trigger source selected is fixed, this command will have no effect on the instrument triggering. This command always zeros the delay by time parameter, so specifying an event count of zero places the trigger subsystem in pass-through mode. In this mode, the instrument triggers immediately upon receipt of a trigger.

TRIGger([:SEQuence]|:STARt)[:LAYer]:IMMediate
Causes a one time entry into the triggered state without receiving the specified trigger. This command is often used to prepare for cases such as setting up a scan list measurement. The example below shows an example of an automated integrated trigger scan of four resistance values in autorange.

TRIGger([: SEQuence] |:STARt) [:LAYer]:SOURce <source>
Selects or queries the trigger source to be used when the instrument is initiated.

Query Response

Command	Response
TRIGger([:SEQuence]|: STARt) [:LAYer]:DELay	<delay in seconds>
TRIGger([:SEQuence]|: STARt)[:LAYer]:ECOunt	<triggers to count>
TRIGger([:SEQuence]|: STARt) [:LAYer]:SOURce	<current source>
TRIGger([:SEQuence] |: STARt)[:LAYer]:SOURce: CATalog[:ALL]?	Lists all trigger sources available for use with the TRIG:SOUR command. HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
TRIGger([:SEQuence]|: STARt)[:LAYer]:SOURce: CATalog:DELayable?	Lists all delayable trigger sources available for use with the TRIG:SOUR command. BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0 COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
TRIGger([:SEQuence]|: STARt)[:LAYer]:SOURce: CATalog:FIXed?	HOLD, IMMEDIATE, TIMER

Examples **TRIGger([:SEQuence]|STARt[LAYer]:COUNt <count>**

Two VX4330 Modules are installed in the slots next to the VX4101A. The default module names "m1" and "m2" have been assigned.

Command	Response
route:conf twire,m1,(1:6)	Set the configuration of all six sections of the first VX4330 to 20-to-1 2-wire
route:conf fwire,m2,(1:3)	Set the configuration of sections 1 through 3 of the second VX4330 to 10-to-1 4-wire
configure owire,m2,(4:6)	Set the configuration of sections 4 through 6 of the second VX4330 to 40-to-1 1-wire
scan(@m1(1!5:20!5),m2(1!3:10!3,1!6:40!6))	Define a scan list consisting of 2-wire channels 1 through 20 in section 5 of the first VX4330, 4-wire channels 1 through 10 in section 3 of the second VX4330 and 1-wire channels 1 through 40 in section 6 of the second VX4330
trigger:sequence:count 2	Define the number of times to sequence through the scan list before setting the state of the trigger subsystem back to the idle state
trigger:sequence: source immediate	Sequence through the scan list without waiting for a trigger event after an INITiate[:Immediate] command is received
initiate:immediate	Initiate the scan sequence. Sequence through the entire scan list two times

TRIGger([: SEQuence1] |:STARt) [:LAYer]:IMMediate

Command	Response
TRIG:SOUR:CAT?	HOLD, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM
INST:SEL DMM	Select DMM
CONF:ARR:RES 4	Configure for 4 resistance readings
TRIG:MODE:ONCE	Require a separate trigger for each
TRIG:SOUR SUREPATH	Select DMM Trigger source as SurePath™
INIT	Initiate and wait for trigger
INST:SEL SUREPATH	Select SurePath™ instrument
ROUT:SCAN (@m1 (1!1!1,1!2!1,1!3!1,1!4!1))a	Program a 4 channel scan list for the VX4380
ROUT:CLOS:DWEL m1, .3	0.3 second delay from channel close to trigger
TRIG:SOUR DMM	Select SurePath™ trigger source as DMM
INIT	Initiate and wait for trigger

Command	Response
TRIG:IMM	Self trigger first channel to get started
INST:SEL DMM	Selects DMM
FETC:COUN?	0 (no measurements available initially).Wait for measure completion. May take a few seconds.
FETC:COUN?	4
FETC?	#252+1.15123E+02, +1.15456E+03, +1.15789E+04, +1.16000E+05

TRIGger ([:SEQuence] \|:STARt) [:LAYer]:SOURce

Command	Response
TRIG:SOUR:CAT?	HOLD, EXTERNAL, IMMEDIATE, BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, TIMER, SUREPATH, DMM, COUNTER, CTR_EXTARM, DAC, DIGI, DIGO
CONF:VOLT:DC	
TRIG:SOUR BUS	
INIT	

TRIGger ([:SEQuence] \|:STARt)[:LAYer]:SOURce:CATalog:DELayable?

Command	Response
TRIG:SOUR:CAT:DEL?	BUS, TTLTRG0, TTLTRG1, TTLTRG2, TTLTRG3, TTLTRG4, TTLTRG5, TTLTRG6, TTLTRG7, COMMAND0, COMMAND1, COMMAND2, COMMAND3, COMMAND4, SUREPATH, DMM, COUNTER, CTR_EXTARM

TRIGger ([:SEQuence] \|:STARt)[:LAYer]:SOURce:CATalog:FIXed?

Command	Response
TRIG:SOUR:CAT:FIX?	HOLD,IMMEDIATE,TIMER

IEEE-488.2 Common Commands

This section includes IEEE-488.2 commmands. A summary of the commands are as follows:

```
*CLS
*ESE
*ESE?
*ESR
*ESR?
*IDN?
*OPC
*OPC?
*RST
*SRE
*SRE?
*STB?
*TRG
*TST?
*WAI
```

*CLS

Command Syntax `*CLS`

Query Syntax `N/A`

Command Class Global

***RST Value** N/A

Limits N/A

Related Commands N/A

Command Description Clears all event status registers and queues.

*ESE

Command Syntax `*ESE`

Query Syntax `*ESE?`

Command Class Global

***RST Value** N/A

Limits 0–255

Related Commands N/A

Command Description Sets the contents of the IEEE 488.2 Standard Event Status Enable Register.

Query Response Queries the contents of the IEEE 488.2 Standard Event Status Enable Register. The contents of this register are unaffected by a register read. The response is in NR1 format and ranges from 0 through 255.

Examples See *Status & Events* section for examples.

*ESR

Command Syntax `ESR`

Query Syntax `*ESR?`

Command Class Global

***RST Value** N/A

Limits 0-255

Related Commands N/A

Command Description Sets or queries the contents of the IEEE 488.2 Standard Event Status Register.

Query Response Queries the contents of the IEEE 488.2 Standard Event Status Register. The contents of this register are cleared after the read is complete. The response is in NR1 format and ranges from 0 through 255.

Examples See *Status & Events* section for examples

*IDN?

Command Syntax N/A

Query Syntax `*IDN?`

Command Class Global

***RST Value** N/A

Limits N/A

Related Commands
```
SYSTem:VERSion?
SYSTem:SNUMber?
SYSTem:OPTions?
```

Query Response Returns a unique instrument identification string which includes board serial number and firmware revision level in the following format:

Tektronix, VX4101A, <serial #>, Firmware v.<version>/SCPI:95.0

Examples

Command	Response
*IDN?	Tektronix, VX4101A, B000001, Firmware v.2.0.0/SCPI:95.0

*OPC

Command Syntax *OPC

Query Syntax *OPC?

Command Class Global

***RST Value** N/A

Limits N/A

Related Commands *WAI

Command Description This command delays the VX4101A parser from processing any further commands until all commands currently in progress have completed. After all commands have completed, the *OPC command sets the operation complete bit in the IEEE 488.2 Standard Event Status Register. Note that a command is considered complete as soon as its action has been initiated. See the description of *WAI for more on this concept.

Query Response The *OPC? query places an ASCII character 1 in the Output queue.

Examples

Command	Response
INST:SEL SUREPATH; ROUT:CLOS (@1); *OPC	
ESR?	0
ESR?	1
INST:SEL DMM; MEAS:VOLT:DC?	+2.39340E-1
INST:SEL SUREPATH; ROUT:CLOS (@1); *OPC?	1
INST:SEL DMM; MEAS:VOLT:DC?	+2.39340E-1

*RST

Command Syntax `*RST`

Query Syntax N/A

Command Class Global

Query Response N/A

***RST Value** The following are reset values for the components in the VX4101A MultiPaq™:

Table 3–11: VX4101A Reset Values

Characteristic	Description
Communication Protocol	Synchronous (IEEE 488.2)
Reference Oscillator Source	Internal 10 MHz
TTL Trigger Sources	HOLD
Periodic Trigger Period	0 (Off)
Query Timeout	Off
External Trigger Source	HOLD

Table 3–12: Counter Front Panel Arm Reset Values

Characteristic	Description
Threshold	1.4 V

Table 3–13: Counter Channels 1 & 2 Analog Front-End

Characteristic	Description
Coupling	AC
Impedance	1 MΩ
Attenuator	X1
Offset	0 V
Gain	1
Lowpass Filter	Off

Table 3–13: Counter Channels 1 & 2 Analog Front-End (Cont.)

Characteristic	Description
Lowpass Filter Frequency	20 MHz
Comparator Slope	Positive
Comparator Level	0 V
Hysteresis	.06 V

Table 3–14: Counter Measurement Settings

Characteristic	Description
Function	SCALar:FREQuency
Channel	1
Start Trigger Source	IMMediate
Stop Trigger Source	INTernal
Mode	Aperture
Events	1000
Auto Setup	On
Time Interval Delay	1.0e–6 Seconds

Table 3–15: DMM Calibration Settings

Characteristic	Description
Line Frequency	As Last Programmed (60 Hz Default)
Source	Internal
Auto-Zero	Off

Table 3–16: DMM Measurement Settings

Characteristic	Description
Function	SCALar:VOLT:DC
Range	MAX (300 V)
Trigger Source	Immediate
Aperture	200e–3 seconds
Auto-Ranging	On
Input Impedance	10e9 Ω

Table 3–16: DMM Measurement Settings (Cont.)

Characteristic	Description
50 Hz NPLC	10
60 Hz NPLC	12

Table 3–17: Digital Input Settings

Characteristic	Description
Format Output Data	Hexadecimal
Format Input Data	Formatted
Input Voltage Threshold	2.3V
Sample Rate	48 KHz
Prematch Mask	#h0
Prematch Pattern	#h0
Inverted/Normal Mode	Normal
Input Pins Active	#hFFFFFFFF or ALL
Number of Measurements	1
Trigger Count	1
Trigger Source	immediate
Trigger Timer	0

Table 3–18: Digital Output Settings

Characteristic	Description
Format Query Output Data	Hexadecimal
Sample Rate	48 KHz
Output State	High (Inactive)
Voltage Source	External
Trigger Mode	SEQ, 1
Trigger Count	1
Trigger Source	IMMediate
Trigger Timer	0
Output Trigger Source	SEQ

Table 3–19: Digital to Analog Converter (DAC) Settings

Characteristic	Description
Data Format	ASCII
Trace Voltages	0.0
Trace Buffer Points	1
Repeat Frequency	OFF
Repeat Period	OFF
Sample Rate	15000 Hz
Time Delay	0 seconds (pass-through)
Event Delay	0 triggers (pass-through)
Polarity	INVerted
Mode	SEQuence, 1
Trigger Source	IMMEDIATE
Output Trigger Source	SEQ

Table 3–20: SurePath Settings

Characteristic	Description
Trigger Source	Immediate
Scan Rate	16e–3 seconds
Default Module Names	M1,M2,M3,...,Mn
Open/Close Dwell Time	0 seconds
Power Fail Action	Open
VX4320	All sections closed to first relay
VX4330	All sections disjoined, set to 4-wire MUX mode, all relays opened
VX4350	All relays opened
VX4380	All relays opened

Limits N/A

Related Commands INSTrument:RESet

Command Description This command sets all instruments to known states independent of past use. All instruments are deselected by this command. The Error/Event Queue, Event Status Registers, and Event Enable Registers are not affected by this command.

Examples

Command	Response
INST:SEL DMM	
CONF:FRES	
CONF?	":SCAL:FRES 3e+08,180000"
*RST	
INST:SEL DMM	
CONF?	":SCAL:VOLT:DC 300,0.001"

*SRE

Command Syntax *SRE

Query Syntax *SRE?

Command Class Global

***RST Value** N/A

Limits 0–255

Related Commands N/A

Command Description Sets the contents of the IEEE 488.2 Service Request Enable Register. The contents of this register are unaffected by a register read.

Query Response Queries the contents of the IEEE 488.2 Service Request Enable Register. The contents of this register are unaffected by a register read. The response is in NR1 format and ranges from 0–255.

Examples See *Status & Events* section for examples.

*STB?

Command Syntax N/A

Query Syntax *STB?

Command Class Global

***RST Value** N/A

Limits 0-255

Related Commands N/A

Query Response Queries the contents of the IEEE 488.2 Status Byte Register. The contents of this register are cleared after the read is complete. Response is in NR1 format, 0–255.

*TRG

Command Syntax `*TRG`

Query Syntax N/A

Command Class Global

***RST Value** N/A

Limits N/A

Related Commands TRIGger:SOURce

Command Description

This command is equivalent to a Group Execute Trigger command. Upon its receipt, places any instrument which has selected BUS as its trigger source in the Device Trigger Active State as defined by the IEEE 488.2 standard.

Examples

Command	Response
INST:SEL DMM	
CONF:VOLT:DC	
TRIG:SOUR BUS	
INIT	
FETC:COUN?	0
*TRG	
FETC:COUN?	1
FETC?	+2.39340E-1

*TST?

Command Syntax

N/A

Query Syntax

`*TST?`

Command Class

Global

***RST Value**

N/A

Limits

N/A

Related Commands

TEST:ALL?

Query Response

This query is used to perform a system wide self-test of the VX4101A infrastructure and all associated instruments. At the end of the self-test the overall pass/fail status, as well as the pass/fail status for each instrument is returned as a response to the query. If a test fails, a descriptive error message is placed in the error queue and the error LED begins to blink. At the end of the self-test, the VX4101A is returned to the reset condition. The reponse format is as follows:

PASS,VX4101A: Self-Test Passed,SurePath: Self-Test Passed,DMM: Self-Test Passed,Counter: Self-Test Passed,DAC: Self-Test Passed,DIGI: Self-Test Passed,DIGO: Self-Test Passed

***NOTE**. PASS can also be FAIL, and Passed can be Failed, depending on the results of the test.*

Examples

Command	Response
*TST?	PASS,VX4101A: Self-Test Passed,SurePath: Self-Test Passed,DMM: Self-Test Passed,Counter: Self-Test Passed,DAC: Self-Test Passed,DIGI: Self-Test Passed,DIGO: Self-Test Passed
INST:SEL:COUNTER	
CONF:FREQ	
CONF?	“1:SCAL:FREQ”
INST:RES	
INST:SEL?	COUNTER
CONF?	“1:SCAL:VOLT:DC”

*WAI

Command Syntax `*WAI`

Query Syntax N/A

Command Class Global

***RST Value** N/A

Limits N/A

Related Commands *OPC
*OPC?

Command Description This command delays the VX4101A parser from processing any further commands until all commands currently in progress have completed. A command is considered complete as soon as its action has been initiated. The instrument may continue processing, collecting, or measuring after the command has been reported as complete. If, for instance, it is desired to wait for a measurement to complete, it is not sufficient to do a *WAI after sending an INIT command. Rather, one should set up and wait for a service request to be generated by a negative transition of the Measurement-In-Progress bit of the event register. In the following example, the DMM measurement is delayed until the relay is closed.

Examples

Command	Response
INST:SEL SUREPATH; ROUT:CLOS (@1); *WAI; INST:SEL DMM; MEAS:VOLT:DC?	+2.39340E-1

Status and Events

Status and Event Reporting System

The VX4101A Status system uses a hierarchical set of registers to provide status information on all instruments. The structure of each instrument register is composed of a set of three registers and two transition registers as defined in the SCPI standard. The Standard Event Status Register and Status Byte Register are defined by the IEEE 488.2 standard. For the purposes of this discussion, if a register bit is referred to as being set, its value is a positive logic one (1). A bit which is cleared has a value of positive logic zero (0).

In order to maintain conformance to the SCPI standard, the VX4101A has a SCPI Questionable status register. This register is typically used to convey information about the quality of an instrument operation. The VX4101A places warning messages in the error/event queue instead of using these registers. These registers are identical in structure to the (OSR) Operational Status Registers (simply substitute the keyword QUEStionable for OPERational), but are always in the cleared state.

The following pages contain an overview of the VX4101A status registers and a description of the associated command set.

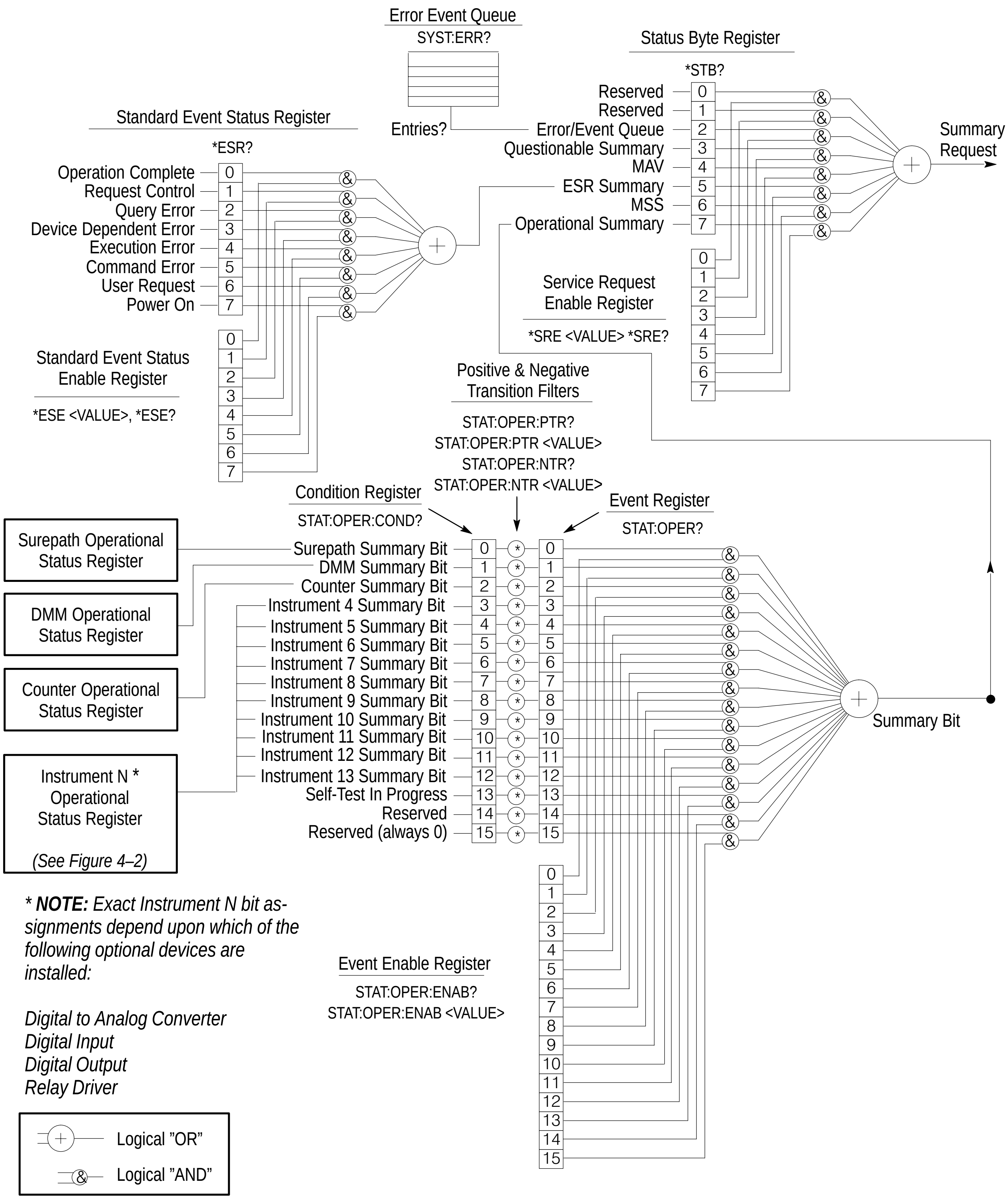


Figure 4–1: VX4101A Standard Registers

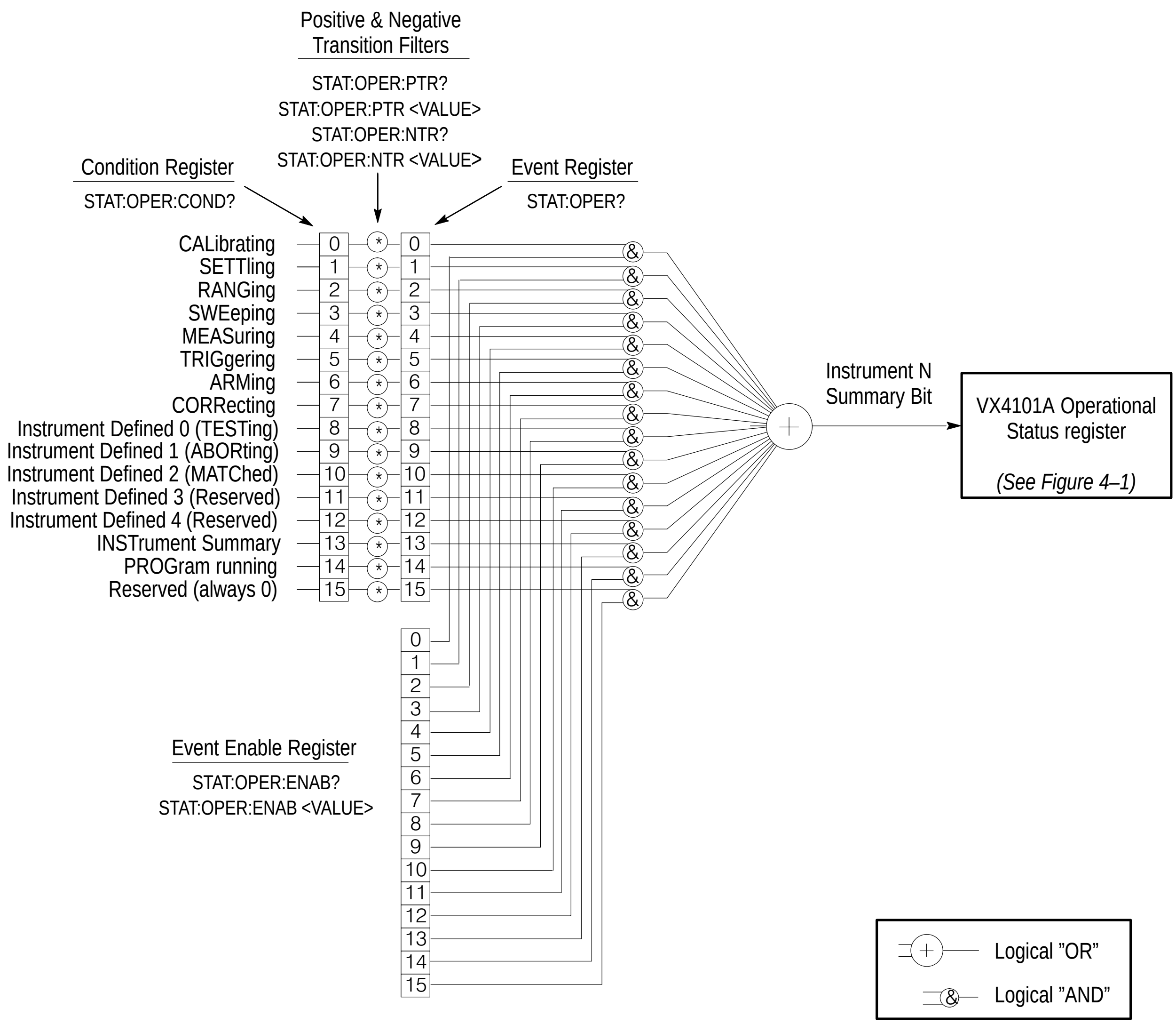


Figure 4–2: Instrument Operational Status Registers

Table 4–1: Instrument Operational Status Register

Bit #	Name	Description
0	CALibrating	Instrument performing calibration
1	SETTling	Waiting for controls to settle
2	RANGing	Instrument is changing range
3	SWEeping	Sweep is in progress
4	MEASuring	A triggered measurement is in progress
5	TRIGgering	Instrument is waiting for trigger condition

Table 4–1: Instrument Operational Status Register (Cont.)

Bit #	Name	Description
6	ARMing	Instrument is waiting for arm condition
7	CORRecting	Instrument is performing a correction
8	Instrument Defined 0	Self-Test in progress
9	Instrument Defined 1	Abort in progress
10	Instrument Defined 2	Reserved by VX4101A
11	Instrument Defined 3	Reserved by VX4101A
12	Instrument Defined 4	Reserved by VX4101A
13	INSTrument summary	Not used by VX4101A
14	PROGram running	User-defined program is running
15	Reserved	Always zero

The Instrument Operational Status Register structure appears in the above table. An instrument uses this register during normal operation to record its current state. As mentioned above, this register is composed of three subregisters and two transition filters.

The Condition Register is updated by the instrument firmware in real-time. Bits are set or cleared to identify what action the instrument is currently performing.

***NOTE**. The STATus:OPERational:CONDitional? query is described in the individual instrument sections.*

Note that some bits may not be used by some instruments and that there are five bits which are instrument defined. See the individual instrument documentation to determine which bits are used.

Following the condition register are two sets of transition filters. The Positive Transition Filter is used to specify which bits of the Condition Register are only of interest when they go from the Cleared to the Set state. The Negative Transition Filter is also used to specify which bits of the Condition Register are only of interest when they go from the Set to the Cleared state. For information on any transition of a bit, both filters would be set.

Any transition which successfully passes through the transition filters sets the corresponding bit in the event register. It is important to remember that for a bit with the Negative Transition Filter set, a event in the Condition Register would cause the same bit in the Event Register to be set, not cleared. Reading this register causes it to be cleared.

Table 4–2: VX4101 A Operational Status Register

Bit #	Name	Description
1	Surepath Summary	Summary bit from Surepath OSR
2	DMM Summary	Summary bit from DMM OSR
3	Counter Summary	Summary bit from Counter OSR
4	Instrument *4* Summary *	Summary bit for Instrument 4
5	Instrument *5* Summary *	Summary bit for Instrument 5
6	Instrument *6* Summary *	Summary bit for Instrument 6
7	Instrument *7* Summary *	Summary bit for Instrument 7
8	Instrument 8 Summary	Summary bit for Instrument 8
9	Instrument 9 Summary	Summary bit for Instrument 9
10	Instrument 10 Summary	Summary bit for Instrument 10
11	Instrument 11 Summary	Summary bit for Instrument 11
12	Instrument 12 Summary	Summary bit for Instrument 12
13	Instrument 13 Summary	Summary bit for Instrument 13
14	Instrument 14 Summary	Summary bit for Instrument 13
15	Reserved	Always zero

***NOTE**. Exact instrument N bit assignments will vary depending upon which of the following optional devices are installed:*

Digital to Analog Converter (DAC)
Digital Input
Digital Output
Relay Driver

The third register is the Event Enable Register. Bits are set or cleared in this register to indicate which bits of the Event Register should be propagated to the summary bit. The summary bit is the logical OR of each bit in the Event Register logically ANDed with the corresponding bit in the Event Enable Register. The summary bit becomes a single bit of another register in the next level of the register hierarchy.

In the case of the VX4101A, the next level in the hierarchy is the VX4101A Operational Status Register. This register has the same structure as the Instrument Operational Status Register with the following exception:

> The summary bit of the VX4101A Operational Status register propagates to a bit in the IEEE 488.2 Status Byte Register.

The IEEE 488.2 Standard Event Status Register provides general status on the VX4101 and all instruments and is at the same hierarchical level as the VX4101A Operational Status Register. Its structure is simpler than the operational status register, in that the Condition Register and transition filters do not exist. This register is cleared when read (by a *ESR? query) and when a *CLS command is received. The register is eight bits wide as shown in the table below.

Table 4–3: Status Byte Register

Bit #	Name	Description
0	Operation Complete	Set in response to *OPC?
1	Request Control	Not Used by VX4101A
2	Query Error	Error occurred during query
3	Device Specific Error	Any error besides query, command or execution
4	Execution Error	Error in command or query parameters
5	Command Error	Command or query syntax error
6	User Request	Not used by VX4101A
7	Power On	Indicates initial power on condition

Table 4–4: IEEE 488.2 Standard Event Status Register

Bit #	Name	Description
0	Reserved	Not Used by VX4101A
1	Reserved	Not Used by VX4101A
2	Error/Event Queue	Set when Error/Event Queue has one or more entries
3	Questionable Summary	Always zero
4	MAV	Set when a message is available for a VXIbus read
5	ESR Summary	Summary bit from IEEE 488.2 Standard Event Status Register
6	MSS	Master Summary Status
7	Operational Summary	Summary bit from VX4101A Operational Status Register

The summary bit from the IEEE 488.2 Standard Event Status Register propagates to the IEEE 488.2 Status Byte Register. The IEEE 488.2 Standard Event Status Enable Register is used to specify which events comprise the summary bit.

The IEEE 488.2 Status Byte Register is the top of the status hierarchy. Like the IEEE 488.2 Event Status Register, this register does not have a Condition Register or transition filters.

The IEEE 488.2 Service Request Enable Register is used to specify which bits of the IEEE 488.2 Status Byte Register propagate to the summary bit. This summary bit is a special case in that when it is set, the VX4101A generates a VXIbus Service Request.

STATus and Event Commands

The following commands control status and events in the VX4101A:

STATus Subsystem

```
STATus:PRESet
STATus:OPERation

    :ENABle
    :ENABle?
    [:EVENt]?
    :NTRansition
    :NTRansition?
    :PTRansition
    :PTRansition?

STATus:QUEue

    :ENABle <numeric list>
    :ENABle?
    [:NEXT]?

STATus:QUEStionable

    [:EVENt]?
    :CONDition?
    :ENABle?
    :ENABle
    :NTRansition
    :NTRansition?
    :PTRansition
    :PTRansition?
```

IEEE 488.2 Commands

The following IEEE 488.2 commands also control status and event reporting:

```
*SRE
*STB?
*CLS
```

```
*ESE
*ESR
```

Please consult the *IEEE 488.2 Commands* section elsewhere in this manual for a full explanation of these commands.

STATus

Command Syntax `STATus:PRESet`

Query Syntax N/A

Command Class Global

***RST Value** N/A

Limits N/A

Related Commands *CLS

Command Description This command clears the enable registers of all Operational Status Registers, sets all Positive Transition Filters, and clears all Negative Transition Filters. This command has no effect on the IEEE488.2 Standard Event Status Register, IEEE 488.2 Standard Event Status Enable Register, IEEE 488.2 Status Byte Register, or IEEE 488.2 Service Request Enable Register.

STATus : OPERation : ENABle

Command Syntax `STATus:OPERation:ENABle`

Query Syntax `STATus:OPERation:ENABle?`

Command Class Instrument

***RST Value** N/A

Limits 0-32767 (all enabled)

Related Commands N/A

Command Description Sets the Operational Enable Register for the currently selected instrument. Setting a bit in this register allows the corresponding bit in the Operational Event Register to propagate to the summary bit.

Query Response Queries the Operational Enable Register for the currently selected instrument. The response is in NR1 format and ranges from 0-32767

STATus : OPERation [:EVENt]?

Command Syntax N/A

Query Syntax `STATus:OPERation[:EVENt]?`

Command Class Instrument

***RST Value** N/A

Limits 0 - 32767

Related Commands N/A

Query Response Returns contents of Operational Event Register for currently selected instrument. Register contents are cleared after read completed. The response is in NR1 format with a range from 0-32767

STATus : OPERation : NTRansition

Command Syntax `STATus:OPERation:NTRansition`

Query Syntax `STATus:OPERation:NTRansition?`

Command Class Instrument

***RST Value** 0

Limits 0–32767

Related Commands N/A

Command Description Sets the Operational Negative Transition Filter for the currently selected instrument. Setting a bit in this filter latches one to zero transitions of the corresponding bit in the Operational Condition Register into the Operational Event Register.

Query Response Queries the Operational Negative Transition Filter for the currently selected instrument. The response is in NR1 format with a range from 0-32767.

STATus : OPERation : PTRansition

Command Syntax `STATus:OPERation:PTRansition`

Query Syntax `STATus:OPERation:PTRansition?`

Command Class Instrument

***RST Value** 32,767

Limits 0–32767

Related Commands N/A

Command Description Sets the Operational Positive Transition Filter for the currently selected instrument. Setting a bit in this filter latches zero to one transitions of the corresponding bit in the Operational Condition Register into in the Operational Event Register.

Query Response Queries the Operational Positive Transition Filter for the currently selected instrument. Response is in NR1 format with a range of 0-32767.

STATus : QUEue : ENABle <numeric list>

Command Syntax `STATus:QUEue:ENABle <numeric list>`

Query Syntax `STATus:QUEue:ENABle?`

Command Class Global

***RST Value** (–499:–100)

Limits N/A

Related Commands N/A

Command Description Allows you to specify by error number which errors and events should be placed in the error/event queue.

Query Response Returns the currently enabled range as a numeric list in NR1 format.

STATus : QUEue [:NEXT]?

Command Syntax N/A

Query Syntax `STATus:QUEue [:NEXT]?`

Command Class Global

***RST Value** N/A

Limits N/A

Related Commands SYST:ERR?

Query Response Returns next item from error/event queue in FIFO order. The possible responses are as follows:

If errors have ocurred:

```
<error #>,<error message>
```

If no errors:

```
0,"No error"
```

STATus : QUEStionable [:EVENt]?

Command Syntax N/A

Query Syntax `STATus:QUEStionable [:EVENt]?`

Command Class Instrument

***RST Value** N/A

Limits 0-32767

Related Commands N/A

Query Response Returns contents of Questionable Event Register for currently selected instrument. Register contents are cleared after read completed. The response is in NR1 format with a range of 0-32767.

STATus : QUEStionable : CONDition?

Command Syntax	N/A
Query Syntax	`STATus:QUEStionable:CONDition?`
Command Class	Instrument
***RST Value**	32767 (all enabled)
Limits	0 - 32767
Related Commands	N/A
Query Response	Returns contents of Questionable Condition Register for currently selected instrument. Register contents are unaffected by this query. The response is in NR1 format with a range of 0-32767.

STATus : QUEStionable : ENABle

Command Syntax `STATus:QUEStionable:ENABle`

Query Syntax `STATus:QUEStionable:ENABle?`

Command Class Instrument

***RST Value** 32767 (all enabled)

Limits 0–32767

Related Commands N/A

Command Description Sets the Questionable Enable Register for the currently selected instrument. Setting a bit in this register allows the corresponding bit in the Questionable Event Register to propagate to the summary bit.

Query Response Queries the Questionable Enable Register for the currently selected instrument. The response is in NR1 format with a range of 0-32767

STATus : QUEStionable : NTRansition

Command Syntax `STATus:QUEStionable:NTRansition`

Query Syntax `STATus:QUEStionable:NTRansition?`

Command Class Instrument

***RST Value** 0

Limits 0–32767

Related Commands N/A

Command Description Sets the Questionable Negative Transition Filter for the currently selected instrument. Setting a bit in this filter latches one to zero transitions of the corresponding bit in the Questionable Condition Register into the Questionable Event Register.

Query Response Queries the Questionable Negative Transition Filter for the currently selected instrument. Response is in NR1 format with a range of 0-32767.

STATus : QUEStionable : PTRansition

Command Syntax `STATus:QUEStionable:PTRansition`

Query Syntax `STATus:QUEStionable:PTRansition?`

Command Class Instrument

***RST Value** 32767

Limits 0-32767

Related Commands N/A

Command Description Sets the Questionable Positive Transition Filter for the currently selected instrument. Setting a bit in this filter latches zero to one transitions of the corresponding bit in the Questionable Condition Register into in the Questionable Event Register.

Query Response Queries the Questionable Positive Transition Filter for the currently selected instrument. Response is in NR1 format with a range of 0–32767.

Status Subsystem Example

The following is an example of using the Status Subsystem to receive a service request at the end of a DMM array measurement. The DMM sets bit 4 of the Operational Condition Register while a measurement is in progress and clears the bit when the measurement has completed (or aborted). The service request will therefore depend upon a negative transition of this bit.

When using the Status subsystem commands for the VX4101A, it must be selected using "INST:NSEL 0" commands. It is unnecessary when using all other global commands to actually select the VX4101A.

***NOTE**. Successful interrupt generation is dependent upon following the sequence outlined in this example.*

Table 4–5: Status Subsystem and Service Requests

Command/Query	Response	Comments
*RST		Reset
*CLS		Clear all event status registers and queues
INST:SEL DMM		Select DMM
STAT:OPER:PTR 0		Don't report any positive transitions
STAT:OPER:NTR 16		Report negative transitions of bit 4
STAT:OPER:ENAB 16		Propagate bit 4 into summary
STAT:OPER?	0	Clear event register by reading
CONF:ARR:VOLT:DC 150		Take approximately 30 seconds of measurements
INST:NSEL 0		Select VX4101A
STAT:OPER:PTR 2		Report positive transitions of bit 1
STAT:OPER:NTR 0		Don't report any negative transitions
STAT:OPER:ENAB 2		propagate bit 1 into summary
STAT:OPER?	0	Clear event register by reading
*SRE 128		Generate SRQ on Bit 8
*STB?	16	Clear status register by reading (except MAV)
INST:SEL DMM		Select DMM
INIT		Initiate measurement
<WAIT FOR SRQ>		Approximately 30 seconds
FETC:COUN?	150	All measurements taken
INST:NSEL 0		Select VX4101A

Table 4–5: Status Subsystem and Service Requests (Cont.)

Command/Query	Response	Comments
*STB?	208	Operational Summary bit, MSS bit, MAV bit
STAT:OPER?	2	Bit 1 (DMM Summary)
INST:SEL DMM		Select DMM
STAT:OPER?	16	Bit 4 (Measuring)
FETC?	<DATA IN IEEE 488.2 BLOCK FORMAT>	

Appendices

Appendix A: Specifications

VX4101A General Characteristics

Table A–1: VXI Instrument Characteristics

Characteristics	Description
VXI General Characteristics	The instrument provides a VXI interface that complies with Revision 1.4. The VXI interface is defined by the VXI Consortium, Inc.
Interface Type	Message Based (1.4)
Other Protocols	Word Serial (WSP), FDC 2.0
Firmware Revision	2.0
Hardware Revision	671-3328-00/671–3532-00 671-3533-00/671-3537-00
Manufacturer ID	9FFD
Device ID	779A
Power Dissipated	42W
Cooling Required	For 15°C Rise, 0.08 mm H^2O@2.3 L/s
Operating Ambient Temperature	0–50°C
DMM Input Ratings	300 V CAT II 1A DC
Counter Input Ratings	300 V CAT II
TTL Outputs	VXI TTLTRG* Lines TTLTRG0* through TTLTRG7* under program control

Table A–2: Power Supply Voltage and Current

DC Volts	Current
+24 V	325 mA
+12 V	130 mA
+5 V	3.5 A
–2 V	60 mA
–5.2 V	1.2 A
–12 V	72 mA
–24 V	175 mA

Table A–3: Environmental/Reliability Characteristics

Characteristics	Description
Temperature	
Operating	Meets or exceeds MIL-T-28800E for Type III, 0 to 50° C external ambient, when operated in a mainframe providing Class 3 equipment
Non-operating	–40° C to + 71° C
Relative Humidity	
Operating	Up to 95% at up to 30° C, and up to 45%, at up to 50° C
Nonoperating	Up to 95%, at up to 50° C
Altitude (1) Operating	10,000 ft. altitude
Altitude (2)	Meets or exceeds MIL-T-28800E for Type III, (operating to 10,000 ft., nonoperating to 15,000 ft.)

Table A–4: VX4101A-Specific Characteristics

Characteristics	Description
VXI Compliance	VXI message-based, fully compliant with revision 1.4 of the VXI specification
VXI Device Classification	Message based device
VME Interrupter Level	Dynamically configured
VXI Logical Address	Switch selectable to a value between 0 and 254
Contents of device/manufacturer dependent VXI registers.	ID Register: 9FFD hexadecimal. Device Type: 779A hexadecimal.
VXI TTL Trigger Outputs	One or more of the VXI TTLTRG* signals may be driven. All TTLTRG* outputs may be disabled
VXI TTL Trigger Inputs	One of the VXI TTLTRG* signals may be selected to be polled or to act as an interrupt source to the modules microprocessor
Module Size	“C” size, one slot wide
VXI Protocol	Word Serial (WSP) with Fast Data Channel for DMM and Digital to Analog Converter (DAC) data
Command Set	SCPI, IEEE-488.2

Table A–5: VX4101A-Specific Physical Characteristics

Characteristics	Description
Weight	1.8 kg (4 lbs)
Mounting location	Installs in an instrument module slot (1-12) of a “C” size VXIbus mainframe
Front panel signal connectors	BNC for Counter channels 1 and 2 SMA for Counter channel 3 SMB for Counter external arm DB-9 (male) for DMM 160-pin connector for Digital to Analog Converter (DAC), Digital Input, Digital Output, and Relay Drivers

Over Voltage Indication

Bench top DMMs typically display some maximum value and an over-voltage indication when an over-voltage or over-range condition is measured. These features are intended to indicate a possible unsafe condition.

For modules (plug in or VXI), a display and an easily noticeable over-voltage indication light are not feasible and over-voltage becomes the responsibility of the user. The VX4101 DMM returns a value of 9.9E + 37 or –9.9E + 37 to detect this situation.

CAUTION. *The user should provide some visible indication of a possible unsafe condition if a value greater that 9.89E + 37 or less than –9.89E + 37 is returned. Due to rounding errors inherent in floating point storage, it is recommended that the user not test for explicit equality with the over-range values.*

Universal Counter Specifications

This section contains specification tables for the counter. The specifications contain a variety of equations and mathematical operators. A table at the end of the counter specifications contains definitions for the operators.

Table A–6: Universal Counter General Specifications

Characteristics	Description
Number of Channels	One input channel for frequency, period, pulse width, time interval, time interval after delay, rise time, fall time, an ratio. A second input channel can be used independently for frequency, period, pulse width, rise time, and fall time; or in conjunction with the first input for time interval, time interval after delay, and phase angle. The second input can also be used for very-high-speed gating. A third input channel (OPTIONAL) uses a divide-by-4 prescaler for extremely high speed signals. It measures frequency or period, or measures ratio in conjunction with either of the first two inputs.
Number of Digits	Up to 17 digits will be returned. Engineering notation is used for positioning the decimal point except for totalizing event measurements.
Gating	Front panel Arm signal, VXI command gating, VXI backplane triggers, and fixed interval timing. Gating can be: level, pulse on from one source and off from a second source, or on from the edge of one pulse then off from the corresponding edge of a second pulse.

Table A–7: Channel 1 and 2 Frequency

Characteristics	Description
Range	0.1 μHz to 250 MHz (500MHz with Option 1C)
Least Significant Digit	$10exp\{Int(log((P_{VCO}/4) * F^2/(N \pm F(P_{VCO}/4)))) + 1\}$.
Resolution	UnGated Resolution: $\pm LSD \pm 1.414(TJE)\ F^2/N$. Gated Resolution: $\pm((P_{VCO})\ F^2/n) \pm (1.414(TJE) * F^2/n)$.
Accuracy	$\pm$Resolution$\pm$F(TBE).

Table A–8: Channel 1 and 2 Period

Characteristics	Description
Range	4 ns to 9 million seconds (2ns min. with Option 1C).
Least Significant Digit	250 ps or 250 ps/N rounded downward to the nearest decade.
Resolution	UnGated Resolution: ±LSD±1.414(TJE)/N±2 ps Gated Resolution: ±(P_{VCO}/n)±(1.414(TJE)/n).±2 ps
Accuracy	±Resolution±TBE_{VCO}. ***NOTE:*** *Detection of signals crossing prescribed thresholds is subject to timing uncertainty (trigger jitter error) due to system noise and threshold inaccuracies. This error affects all measurements but is more pronounced for single-shot measurements and for signals with slow rise and fall time. The timing trigger jitter error (TJE) equals:* $\frac{[(RMSnoiseofinputsignal)^2 + (RMSnoiseofinputChannel)^2]^{1/2}}{InputSlewRateattheTriggerPoint}$

Table A–9: Channel 3 Frequency (Option 2C)

Characteristics	Description
Range	200 MHz to 3000 MHz
Least Significant Digit	10exp{Int(log((P_{VCO}/4) * 4 * F^2/(N±4 * F(P_{VCO}/4)))) + 1}.
Resolution	Ungated Resolution: ±LSD±1.414(TJE) F^2/N. Gated Resolution: ±((P_{VCO}) * F^2/n)±(1.414(TJE) * F^2/n).
Accuracy	±Resolution±F(TBE)

Table A–10: Channel 3 Period (Option 2C)

Characteristics	Description
Range	333 ps to 5 ns
Least Significant Digit	250 ps or 250 ps/N rounded downward to the nearest decade.
Resolution	Ungated Resolution: ±LSD±1.414(TJE)/N±2 ps Gated Resolution: ±(P_{VCO}/n)±(1.414(TJE)/n)±2 ps
Accuracy	±Resolution±TBE_{VCO}

Table A–11: Time Interval

Characteristics	Description
Frequency Range	Up to 250 MHz (500 MHz with option 1C) Up to 150 MHz for Time Interval with Delay
Range	–1 ns to 9 million seconds
Least Significant Digit	250 ps for N = 1, single-shot (1 ns/$\sqrt{N}$) for N > 1 rounded downward to the nearest decade.
Resolution	$\pm$LSD$\pm$1.414 ((TJE$_1$ + TJE$_2$)/$\sqrt{N}$)$\pm$2 ps
Accuracy	$\pm$TBE (Time Interval) $\pm$ Resolution + (TLE$_1$/Slew$_S$ – TLE$_2$/Slew$_E$)$\pm$200 ps

Table A–12: Frequency Ratio

Characteristics	Description
Ch 1/Ch 2 and Ch 2/Ch 1	10^{-14} to 10^{14}
Ch 1/Ch 3 and Ch 2/Ch 3 with Option 2C	10^{-15} to 1
Ch 3/Ch 1 and Ch 3/Ch 2 with Option 2C	1 to 10^{15}

Table A–13: Channels 1 and 2 Totalizer

Characteristics	Description
Range	1 to 2^{52} (4.5×10^{15})
Least Significant Digit	One count.

Table A–14: Channels 1 and 2 Rise/Fall Time

Characteristics	Description
Frequency Range	Up to 250 MHz (0.5 V_{p-p} minimum signal for specified accuracy).
Range	250 ps single-shot 1 ns to 9 million seconds
Least Significant Digit	250 ps for N = 1, single-shot (1 ns/$\sqrt{N}$) for N > 1 rounded downward to the nearest decade.
Resolution	LSD$\pm$1.414 ((TJE$_1$ + TJE$_2$)/$\sqrt{N}$)$\pm$2 ps RMS
Accuracy	$\pm$TBE (Time Interval) $\pm$ Resolution + (TLE$_2$/Slew$_S$ – TLE$_1$/Slew$_E$)$\pm$200 ps

Table A–15: Channels 1 and 2 Positive/Negative Pulse Width

Characteristics	Description
Frequency Range	Up to 250 MHz (500 MHz with Option 1C)
Range	1 ns to 9 million seconds
Least Significant Digit	250 ps (single-shot)
Resolution	$\pm$LSD$\pm$1.414 ((TJE$_{1}$ + TJE$_{2}$)/$\sqrt{}$N)$\pm$2ps (for 1 $\leq$ N $\leq$9 million)
Accuracy	$\pm$Resolution$\pm$Width (TBE)$\pm$TLE$_{1}$$\pm$TLE$_{2}$$\pm$1 ns.

Table A–16: Frequency Ratio 1/2,2/1,3/1,3/2,1/3,2/3

Characteristics	Description
Range	10^{-12} to 10^{12}
Least Significant Digit	(F_n/ F_d) * Gate time (divide F by 4 for channel 3 prescaler)
Resolution	Same as LSD
Accuracy	Same as LSD

Table A–17: Voltages

Characteristics	Description
Offset Range	±100 V DC
Maximum Input Signal	± 125 VPP + offset
Resolution	200 µV × Attenuation/Gain
Accuracy	± 7.5% of reading ± 25 mV × Attenuation

Table A–18: Channels 1 and 2 Input Characteristics

Characteristics	Description
Hysteresis (@ 1 MHz)	Programmable in 10 mV_{p-p} steps from: Minimum (10 mV_{p-p} on-to-off) to Maximum (60 mV_{p-p} on-to-off) × Attenuation/Gain
Coupling	AC,DC
Trigger Slope	Independent selection of slope polarity
Attenuator	×1, ×10, ×100 Nominal
Impedance	1 MΩ shunted by 15 pF or 50 Ω

Table A–18: Channels 1 and 2 Input Characteristics (Cont.)

Characteristics	Description
Damage Level	Input voltage at 1 MΩ ×1: 300 V (DC or peak AC) derate to 5 V (DC + peak AC) at 20 dB/decade above 10 kHz ×10: 300 V (DC or peak AC) derate to 5 V (DC + peak AC) at 20 dB/decade above 10 MHz ×100: 300 V (DC or peak AC) derate to 5 V (DC + peak AC) at 20 dB/decade above 10 MHz Input Voltage at 50 Ω = 5 V (0.5 Ω) maximum (Overload protection auto-switch to 1M Ω over 840 mW (6.5 VDC) in less than 5 seconds.)
Common Input Limitations	None
Sensitivity	–20 dBm to 200 MHz, then 6 db per octave to –12 dBm at 500 MHz –13 dBm at 1000 MHz typical 3dB Low Frequency limit with AC coupling: 1 e6 Ω input impedance = 10 Hz 50 Ω input impedance = 200 kHz
Input Channel Noise	$\pm 2mV_{rms}$ typ $< 500\ \mu V_{rms}$
Filters	None, 20 MHz, or 100 MHz

Table A–19: Channel 3 Input Characteristics (Option 2C)

Characteristic	Description
Coupling	AC
Trigger Slope 3	Positive slope (threshold at mean)
Impedance 3	50 Ω
Damage Level 3	5 VPTP (AC)
Common Input Limitations	< 5 VPP
Sensitivity 3	200 MHz to 1000 MHz–(–30 dBm) 1000 MHz to 3000 MHz–(–25 dBm) 4000 MHz –(–13 dBm typical)

Table A–20: Arm Characteristics

Characteristic	Description
Arm Sources	VXI TTL Trigger lines, front panel arm, Channel 2, DMM, SurePath, Software, and an internal timer
Arm Start/Stop	Except for channel 2, the arm can be a level or a pulse to start and a second pulse to stop. The start and stop pulses may be different sources. Channel 2 is high speed arm that is a level
Front Panel Arm Polarity	Positive or Negative
Threshold	Programmable
Measurement Time-out	Programmable time-out to abort measurements in case no input signal is present

Table A–21: Channel 1 and 2 Trigger Level

Characteristic	Description
Trigger Level Range	±4 mV to±225 V
Trigger Level Accuracy	For DC or 1 kHz 100 MHz: Resolution: 0.2 mV * (Gain/Attenuation) Error =±7.5% of setting ±25mV *Attenuation

Table A–22: Channels 1 and 2 Autotrigger

Characteristic	Description
Auto Trigger Range	Same as user selected for continuous waveforms ≤ 1 kHz

Table A–23: Front Panel Connectors

Characteristic	Description
Connector Types	BNC (Ch1, Ch2), SMA (Ch3), SMB (Arm)

Table A–24: TimeBase Characteristics (Option 1T)

Characteristic	Description
Time Base Characteristics	The time base can be drawn from four different sources: the on-board 10 MHz source, the optional on-board high accuracy 10 MHz source, the VXI Slot 0 10 MHz source, and the Slot 0 External 10 MHz source. The user can select, by programmed command, the source and what degree of accuracy is required. The Slot 0 External 10 MHz source can be synchronized to other systems.
Time Base Reference Clock	On-board 25 ppm VXI Slot 0 required accuracy 100 ppm VXI Slot 0 External As required
Time Base Temperature Stability	Standard 25 ppm High Accuracy (optional) 1.5 ppm (TXCO) Option 1T
External Time Base	User provided
Frequency Range	VXI cards are available with tightly controlled specifications, including a Rubidium Vapor Oscillator with a claimed accuracy of $< 5 \times 10^{-11}$ per month stability.
Interface	10 MHz connected into 50 Ω Slot 0 connector. The Slot 0 card must be programmed to source the VXI backplane 10 MHz from the front-panel input.

Table A–25: Gate Input Trigger

Characteristic	Description
Gate Inputs	There are four sources of gate inputs: VXI TTL backplane triggers, on-board processor generated triggers (either by VXI Command or timer algorithm), front panel Gate/Arm input, or Channel 2 input as gate. All gate inputs will start and stop counting synchronously with the measured input on its next transition (Trigger) for all measurements except totalizing. ***NOTE:*** *Tektronix terminology uses the measured signals input (Channel 1 or Channel 2) as Triggers, processor and backplane on/off signals as Gates, and the front-panel control signal as the Arm signal. When the Channel 2 input signal controls the starting and stopping of the Channel 1 signal, the Channel 2 signal will be referred to as a Gate.*
Front Panel Gate Input Interface	SMB connector
Polarity	Programmable
Gate Type	Programmable: level, edge, pulse-on/pulse-off
Edge Triggering Amplitude	±500 mV minimum, ±20 V Maximum
Pulse Width Requirements	40 ns minimum pulse width
Gate Start/End Uncertainty	$<$ 20 ns

Table A–26: VXIBus TTLTRG Gate Input

Characteristic	Description
TTLTRG Timing	Synchronous with VXI backplane 10 MHz Asynchronous
TTLTRG Type	Programmable: level, edge, pulse-on/pulse-off
Channel-B Counter Input Gate Polarity	Programmable
Gate Type	Programmable: level, edge
Edge Triggering Amplitude/Pulse Width Requirements	±5 mV minimum, ±200 V Maximum 2.86 ns minimum

Table A–27: Software Gate/Trigger

Characteristic	Description
Software Gate Type	Level
Software Gate Duration Time Range	Command based: User defined Timer based: 10 ns to 9 million seconds, 1 ns resolution

Table A–28: Gate Duration Modes

Characteristic	Description
Gate Duration Time Range	2.86 ns minimum (Channel 2 Gate) 1000 seconds maximum
Gate Duration Indefinite	Software command or hardware based
Gate Output	At the conclusion of the measurement, defined time, or defined number or events.

Table A–29: Measurement Throughput

Characteristic	Description
Throughput Rate	30 measurements/second for array measurements.
Single Shot Array Capture Rate	As fast as 4 1/3 μs per measurement. (6.5 μs for phase or duty cycle) As fast as 4 1/3 ms for a 1000 measurement array (6.5 ms for a phase or duty cycle 1000 measurement array)

Table A–30: Counter Specifications Terms

Term	Description	
Frequency Terms	F	Frequency of the signal being measured (F = 1/P).
	F_L	Frequency of the leading (first) signal being measured.
	F_T	Frequency of the trailing (second) signal being measured.
	F_1	Frequency of the Channel 1 signal being measured.
	F_2	Frequency of the Channel 2 signal being measured.
	F_n	Frequency of the Numerator (Ratio Measurement).
	F_d	Frequency of the Denominator (Ratio Measurement).
	F_{VCO}	Frequency of the Universal Counter Voltage Controlled Oscillator time base. (F_{VCO} has been chosen so that it is not evenly divisible by 10 kHz, thereby reducing the likelihood of aliasing.) It is calculated as follows: ($F_{VCO} = 10e6 \times 304.0/3.0$)
Numeric Terms	N	Number of events in a measurement interval. For Auto-Averaging, N is the greater of one or the number of events in a 100 ms interval.
	N_1	Number of events in a leading (first) measurement interval.
	N_2	Number of events in a trailing (second) measurement interval.
	N_1	Number of events in a Channel 1 measurement interval.
	N_2	Number of events in a Channel 2 measurement interval.
	n	For gated measurements, n is the number of events within the gated interval.

Table A–30: Counter Specifications Terms (Cont.)

Term	Description	
Timing Terms	DJ	Delay Jitter
	P	The period of the signal being measured (P = 1/F).
	P_{VCO}	(1/ F_{VCO}). Approximately 0.987 ns.
	$Slew_E$	Slew at the ending edge of a measurement. For sine waves, the slew rate at the 50% level is as follows: $2 \times \pi \times F \times VPP/2$
	For sine waves at 10% (or 90%) the slew rate is as follows:	
		$\cos(\text{arcsine}(-1.0 + (10\% \text{ of } 2.0))) \times 2 \times \pi \times F \times VPP/2$
	$Slew_S$	Slew at the starting edge of a measurement.
	TBE	Time Base Error of the 10 MHz root time base provided by the Slot 0 board, a frequency plugged into the Slot 0 board, or the optional on-board oscillator.
	TBE_{VCO}	Time Base Error of the root time base applied to the frequency of the Universal Counter Voltage Controlled Oscillator time base.
	TJE_1	Trigger Jitter Error of the Channel 1 event.
	TJE_2	Trigger Jitter Error of the Channel 2 event.
	NOTE: *Detection of signals crossing prescribed thresholds is subject to timing uncertainty (trigger jitter error) due to system noise and threshold inaccuracies. This error affects all measurements but is more pronounced for single-shot measurements and for signals with slow rise and fall time. The timing trigger jitter error (TJE) equals:* $\frac{[(RMSnoiseofinputsignal)^2 + (RMSnoiseofinputChannel)^2]^{1/2}}{InputSlewRateattheTriggerPoint}$	
	TLE_1	Trigger Level Error of the leading (first) event.
	TLE_2	Trigger Level Error of the trailing (second) event.
	TJE	Trigger Jitter Error which is the input signal RMS noise voltage.
	TJE_1	Trigger Jitter Error of the leading (first) event.
	TJE_2	Trigger Jitter Error of the trailing (second) event.
Mathematical Terms	exp{}	Exponentiates to the power given by the expression within the bracket.
	Int()	Converts the expression within the parenthesis to the nearest integer of lesser or equal value. For example: Int(1.7) = 1 and Int(-1.7) = -2 (Use floor (x) in the C programming language.)
	log()	Computes the base 10 (common) logarithm of the expression within the parenthesis.

Digital Multimeter (DMM) Specifications

Table A–31: Aperture Specifications

Characteristics	Description
Aperture (50 Hz)	1 ms to 2 sec in 1 ms to 10 ms steps (total of 560 apertures)
Aperture (60 Hz)	833 μs to 2 sec in 833 μs to 8.33 ms steps (total of 680 apertures)

Table A–32: Digits vs. Aperture

Digits	Aperture	Readings/Second
5.5	≥16.67 ms (20.0 ms)	≤60 (50)
4.5	<16.67 ms	>60 (50)

Table A–33: Memory Capacity

Characteristics	Description
On-Board Memory	4096 Measurements

Table A–34: DC Voltage

Range	Maximum Reading	Resolution
30 mV	± 30.0000 mV	100 nV
300 mV	± 300.000 mV	1 μV
3 V	± 3.00000 V	10 μV
30 V	± 30.0000 V	100 μV
300 V	± 300.000 V	1 mV

Table A–35: Accuracy Specifications for 2-Second Aperture

Range	24 Hour	90 Day	1 Year	Temp Coefficient
30 mV	0.006% + 8 μV (9 μV)	0.012% + 8 μV (9 μV)	0.018% + 8 μV (9 μV)	0.0010% + 1.5 μV
300 mV	0.005% + 14 μV (15 μV)	0.010% + 14 μV (15 μV)	0.015% + 14 (15 μV)	0.0010% + 2.5 μV
3 V	0.004% + 70 μV (80 μV)	0.008% + 70 μV (80 μV)	0.012% + 70 μV (80 μV)	0.0010% + 5 μV

Table A–35: Accuracy Specifications for 2-Second Aperture (Cont.)

Range	24 Hour	90 Day	1 Year	Temp Coefficient
30 V	0.009% + 900 µV (1.0 mV)	0.016% + 900 µV (1.0 mV)	0.023% + 900 µV (1.0 mV)	0.0010% + 50 µV
300 V	0.007% + 8 mV (9 mV)	0.014% + 8 mV (9 mV)	0.020% + 8 mV (9 mV)	0.0010% + 500 µV

***NOTE**. Values shown above in parenthesis are for a 200 ms aperture.*

Table A–36: Accuracy Specification for ≤ 1 Millisecond Aperture

Range	24 Hour	90 Day	1 Year	Temp Coefficient
30 mV	0.007% + 15 µV (12 µV)	0.012% + 15 µV (12 µV)	0.018% + 15 µV (12 µV)	0.0015% + 1.5 µV
300 mV	0.005% + 30 µV (20 µV)	0.010% + 30 µV (20 µV)	0.015% + 30 µV (20 µV)	0.0015% + 2.5 µV
3 V	0.004% + 150 µV (100µV)	0.008% + 150 µV (100µV)	0.012% + 150 µV (100µV)	0.0015% + 5 µV
30 V	0.009% + 1.7 mV (1.2 mV)	0.016% + 1.7 mV (1.2 mV)	0.023% + 1.7 mV (1.2 mV)	0.0015% + 50 µV
300 V	0.008% + 15 mV (10 mV)	0.014% + 15 mV (10 mV)	0.020% + 15 mV (10 mV)	0.0015% + 500 µV

***NOTE**. Values shown above in parenthesis are for a 16.67 ms or 20 ms aperture.*

***NOTE**. Percents listed above are percents of reading. The instrument setup is as follows:*

Autozero On following twenty minute warm-up period.

Tcal is the calibration temperature (18° to 28° C). Specifications are for Tcal ±3°C. Multiply the total temperature coefficient by the difference between the actual operating temperature and Tcal ±3°C.

Add 0.001% of the range setting to the first reading following a range change.

For Autozero Off, do the following:

For daily and long term drift, add 24 µV per day for the 30 mV and 300 mV ranges, and 0.0027% of range for the other ranges.

For temperature drift, add .4 µV per °C for the 30 mV range, .001% of range per °C for the other ranges.

Table A–37: DC Input Resistance

Range	Resistance
30 mV, 300 mV, 3 V Ranges	10 MΩ ±5%, or > 10 GΩ, programmable
30 V, 300 V Ranges	10 MΩ ± 5%

Table A–38: DC Input Protection

Range	Protection
V + V–	350 VDC or VRMS, or 450 V peak AC, all ranges
V + to Chassis	
V– to Chassis	

Table A–39: CMRR

Aperture	CMRR (DC)
All	140 dB

Table A–40: DC CMRR (0 to 400 Hz)

Aperture	CMRR 0–60 Hz	CMRR 0–400 Hz
2 seconds	115 dB	115 dB
200 ms	95 dB	95 dB
20 ms	75 dB	75 dB
16.7 ms	73 dB	73 dB
1 ms	63 dB	50 dB

Table A–41: DC Normal Mode Rejection (50/60/400 Hz)

Aperture	Frequency	NMR
All	50 Hz	83 dB
	60 Hz	83 dB
	400 Hz	79 dB

Table A–42: DC ECMR (50/60/400 Hz)

Aperture	Frequency	ECMR
All	50 Hz	122 dB
	60 Hz	120 dB
	400 Hz	102 dB

Table A–43: TRMS AC Voltage (DC Coupled and AC Coupled)

Range	Maximum Reading	Resolution
30 mV	30.0000 mV	100 nV
300 mV	300.000 mV	1 μV
3 V	3.00000	10 μV
30 V	30.0000 V	100 μV
300 V	300.000	1 mV

Table A–44: TRMS Accuracy Specifications –24-Hour

Range	Frequency	DC Coupled	AC Coupled	Temp Coefficient / °C
30 mV	20–45 Hz	0.65% + 180 μV	1.10% + 180 μV	.03% + 15.0 μV
	45–100 Hz	0.40% + 180 μV	0.40% + 180 μV	.03% + 15.0 μV
	100 Hz–10 kHz	0.30% + 180 μV	0.30% + 180 μV	.03% + 15.0 μV
	10–20 kHz	0.60% + 180 μV	0.60% + 180 μV	.03% + 15.0 μV
	20–50 kHz	1.50% + 180 μV	1.50% + 180 μV	.03% + 15.0 μV
300 mV	20–45 Hz	0.65% + 300 μV	1.10% + 300 μV	.03% + 15.0 μV
	45–100 Hz	0.40% + 300 μV	0.40% + 300 μV	.03% + 15.0 μV
	0.1–10 kHz	0.30% + 300 μV	0.30% + 300 μV	.03% + 15.0 μV
	10–20 kHz	0.60% + 300 μV	0.60% + 300 μV	.03% + 15.0 μV
	20–50 kHz	1.00% + 300 μV	1.00% + 300 μV	.03% + 15.0 μV
3 V	20–45 Hz	0.65% + 3 mV	1.10% + 3 mV	.03% + 250 μV
	45–100 Hz	0.40% + 3 mV	0.40% + 3 mV	.03% + 250 μV
	0.1–10 kHz	0.30% + 3 mV	0.30% + 3 mV	.03% + 250 μV
	10–20 kHz	0.90% + 3 mV	0.90% + 3 mV	.03% + 250 μV
	20–50 kHz	2.50% + 3 mV	2.50% + 3 mV	.03% + 250 μV
30 V	20–45 Hz	0.65% + 30 mV	1.10% + 30 mV	.03% + 250 μV
	45–100 Hz	0.40% + 30 mV	0.40% + 30 mV	.03% + 2.5 mV

Table A–44: TRMS Accuracy Specifications –24-Hour (Cont.)

Range	Frequency	DC Coupled	AC Coupled	Temp Coefficient / °C
30 V (Cont.)	0.1–10 kHz	0.30% + 30 mV	0.30% + 30 mV	.03% + 2.5 μV
	10–20 kHz	0.90% + 30 mV	0.90% + 30 mV	.03% + 2.5 mV
	20–50 kHz	1.50% + 30 mV	1.50% + 30 mV	.03% + 2.5 mV
300 V	20–45 Hz	0.65% + 300 mV	1.10% + 300 mV	.03% + 25 mV
	45–100 Hz	0.60% + 300 mV	0.50% + 300 mV	.03% + 25 mV
	0.1–10 kHz	0.50% + 300 mV	0.40% + 300 mV	.03% + 25 mV
	10–20 kHz	1.30% + 300 mV	1.30% + 300 mV	.03% + 25 mV
	20–50 kHz	2.50% + 300 mV	2.50% + 300 mV	.03% + 25 mV

***NOTE**. Specifications are for > 7% of range and are valid for all apertures.*

Table A–45: TRMS Accuracy–90 Day and 1 Year

Characteristic	Description
90-Day Accuracy	Add 0.08% of reading to 24-Hour Specifications above
1 Year Accuracy	Add 0.15% of reading to 24-Hour Specifications above

Table A–46: TRMS Crest Factor

Characteristic	Description
100% Full Scale	1.9:1
10% Full Scale	7:1

Table A–47: TRMS Input Impedance

Characteristics	Description
30 mV and 300 mV, 3V Ranges	1.01 MΩ ± 5%, <120 pF
3 V, 30 V, and 300 V Ranges	1.02 MΩ ± 5%, <120 pF

Table A–48: TRMS Input Protection – V+ to V–, V+ to Chassis, and V– to Chassis

Characteristics	Description
DC & AC/RMS	350 V on all ranges
AC Peak	450 V on all ranges

Table A–49: TRMS CMRR (0 to 400 Hz)

Range	Range	Description
30 mV	0 to 60 Hz	>75 dB
300 mV	0 to 60 Hz	>67 dB
3 V	0 to 60 Hz	>67 dB
30 V	0 to 60 Hz	>70 dB
300 V	0 to 60 Hz	>63 dB
30 mV	0 to 400 Hz	>59 dB
300 mV	0 to 400 Hz	>51 dB
3 V	0 to 400 Hz	>51 dB
30 V	0 to 400 Hz	>54 dB
300 V	0 to 400 Hz	>47 dB

Table A–50: Resistance (2-Wire and 4-Wire)

Range	Maximum Reading	Resolution
30 Ω	30.0000 Ω	100 μΩ
300 Ω	300.000 Ω	1 mΩ
3 kΩ	3.00000 kΩ	10 mΩ
30 kΩ	30.0000 kΩ	100 mΩ
300 kΩ	300.000 kΩ	1 Ω
3 MΩ	3.00000 MΩ	10 Ω
30 MΩ	30.0000 MΩ	100 Ω
300 MΩ	300.000 MΩ	1 kΩ (nom)

Table A–51: Resistance Accuracy Specifications for 2-Second Aperture

Range	Source Current	24 Hour	90 Day	1 Year	Temp Coefficient / °C
30 Ω	1 mA	0.017% + 8 mΩ (9 mΩ)	0.023% + 8 mΩ (9 mΩ)	0.037% + 8 mΩ (9 mΩ)	0.007% + 1.5 mΩ
300 Ω	1 mA	0.010% + 14 mΩ (15 mΩ)	0.015% + 14 mΩ (15 mΩ)	0.020% + 14 mΩ (15 mΩ)	0.007% + 2.5 mΩ
3 kΩ	1 mA	0.010% + 70 mΩ (80 mΩ)	0.015% + 70 mΩ (80 mΩ)	0.020% + 70 mΩ (80 mΩ)	0.007% + 5 mΩ
30 kΩ	100 µA	0.010% + 700 mΩ (800 mΩ)	0.015% + 700 mΩ (800 mΩ)	0.020% + 700 mΩ (800 mΩ)	0.007% + 50 mΩ
300 kΩ	10 µA	0.010% + 7 Ω (8 Ω)	0.015% + 7 Ω (8 Ω)	0.020% + 7 Ω (8 Ω)	0.007% + 500 mΩ
3 MΩ	1 µA	0.040% + 70 Ω (80 Ω)	0.060% + 70 Ω (80 Ω)	0.080% + 70 Ω (80 Ω)	0.012% + 20 Ω
30 MΩ	100 nA	0.300% + 700 Ω (800 Ω)	0.400% + 700 Ω (800 Ω)	0.500% + 700 Ω (800 Ω)	0.050% + 200 Ω
300 MΩ	100 nA	(.3 + .03R)% + 7 kΩ (8 kΩ)	(.4 + .04R)% + 7 kΩ (8k Ω)	(.5 + .05R)% + 7 kΩ (8 kΩ)	(.05 + .005R)% + 2000 Ω

***NOTE**. Values shown in parenthesis above are for a 200 ms aperture.*

***NOTE**. The value R in the 300 MΩ range specification is the measured resistance in MΩ.*

Table A–52: Resistance Accuracy Specifications for 1 Millisecond Aperture

Range	Source Current	24 Hour	90 Day	1 Year	Temp Coefficient / °C
30 Ω	1 mA	0.037% + 15 mΩ (12 mΩ)	0.037% + 15 m Ω (12 mΩ)	0.037% + 15 mΩ (12 mΩ)	0.007% + 1.5 mΩ
300 Ω	1 mA	0.015% + 30 mΩ (20 mΩ)	0.020% + 30 mΩ (20 mΩ)	0.025% + 30 mΩ (20 mΩ)	0.007% + 2.5 mΩ
3 kΩ	1 mA	0.015% + 150 mΩ (100 mΩ)	0.020% + 150 mΩ (100 mΩ)	0.025% + 150 mΩ (100 mΩ)	0.007% + 5.0 mΩ
30 kΩ	100 µA	0.015% + 1.5 Ω (1.0 Ω)	0.020% + 1.5 Ω (1.0 Ω)	0.025% + 1.5 Ω (1.0 Ω)	0.007% + 50 mΩ
300 kΩ	10 µA	0.030% + 15 Ω (10 Ω)	0.030% + 15 Ω (10 Ω)	0. 040% + 15 Ω (10 Ω)	0.007% + 500 mΩ

Table A–52: Resistance Accuracy Specifications for 1 Millisecond Aperture (Cont.)

Range	Source Current	24 Hour	90 Day	1 Year	Temp Coefficient / °C
3 MΩ	1 μA	0.080% + 150 Ω (100 Ω)	0.100% + 150 Ω (100 Ω)	0.120% + 150 Ω (100 Ω)	0.012% + 20 Ω
30 MΩ	100 nA	0.300% + 1.5 kΩ (1.0 kΩ)	0.400% + 1.5 kΩ (1.0 kΩ)	0.500% + 1.5 kΩ (1.0 kΩ)	0.050% + 200 Ω
300 MΩ	100 nA	(.3 + .03R)% + 15 kΩ (10 kΩ)	(.4 + .04R)% + 15 kΩ (10 kΩ)	(.5 + .05R)% + 15 kΩ (10 kΩ)	(.05 + .005R)% + 2000 Ω

***NOTE**. Values shown in parenthesis above are for a 16.67 ms or 20 ms aperture.*

***NOTE**. The value R in the 300 MΩ range specification is the measured resistance in MΩ.*

***NOTE**. Percents listed above are percents of reading. Instrument set up is as follows:*

Autozero On, after 20 minute warm-up.

The above specifications are for 4-wire Ω. Add 50 MΩ to all specifications for 2-wire ohms.

Tcal is the calibration temperature (18 to 28° C). Specifications are for Tcal ±3° C. Multiply the total temperature coefficient by the difference between the actual operating temperature and Tcal ±3° C.

For the 300 MΩ range, the measured resistance in MΩ multiplied by the temperature difference in °C is limited to 1000, for example, at 100 MΩ the temperature drift specification is only valid for a temperature difference of 10 °C.

Add 0.002% of the range setting to the first reading following a range change.

For Autozero Off, do the following:

For daily and long term drift, add 500 μΩ/Day for the 30 Ω range, 1 mΩ/ Day for the 300 Ω range, 100 kΩ/ Day for the 300 mΩ range, and 0.0001% of range for the other ranges.

For temperature drift, increase the second part of the temperature coefficient by a factor of 4, for 300 kΩ range or lower (increase 1.5 mΩ to 6 mΩ, 2.5 mΩ to 10 mΩ, etc.) .

Table A–53: Resistance Input Protection-All Ranges

Terminals	DC and AC/RMS	AC Peak
R+ to R–	250 V	250 V
R+ to chassis	350 V	450 V
R– to chassis	350 V	450 V

***NOTE**. For Common Mode Rejection, see DCV Specifications.*

Table A–54: DC Current

Range	Maximum Reading	Resolution
150 mA	±150.000 mA	0.5 μA
1 A	±1.00000 A	5 μA

Table A–55: DC Current Sense Resistance

Characteristic	Description
Sense Resistance	0.19 Ω

Table A–56: DC Current Accuracy Specifications for 2-Second and 0.2-Second Aperture

Range	Burden	24 Hour	90 Day	1 Year	Temp Coefficient / °C
150 mA	±0.1 V	0.10% + 40 µA (50 µA)	0.12% + 40 µA (50 µA)	0.15% + 40 µA (50 µA)	0.05% + 8 µA
1A	±0.4 V	0.15% + 200 µA (220 µA)	0.15% + 240 µA (260 µA)	0.18% + 270 µA (300 µA)	0.05% + 50 µA

***NOTE**. Values shown in parenthesis above are for 16.67 or 20 ms aperture.*

Table A–57: DC Current Accuracy Specifications for 1 Millisecond Aperture

Range	Burden	24 Hour	90 Day	1 Year	Temp Coefficient / °C
150 mA	±0.1 V	0.10% + 80 µA	0.12% + 100 µA	0.15% + 120 µA	0.05% + 8 µA
1 A	±0.4 V	0.15% + 240 µA	0.15% + 280 µA	0.18% + 320 µA	0.05% + 50 µA

***NOTE**. Percents listed above are percent of reading. Instrument set up is as follows:*

Autozero On, after 20 minute warm-up.

Tcal is the calibration temperature (18 to 28° C). Specifications are for Tcal ±3° C. Multiply the total temperature coefficient by the difference between the actual operating temperature and Tcal ±3° C.

Add 0.02% of the range setting to the first reading following a range change.

Burden voltage is the maximum voltage drop caused by the current measurement for the maximum reading in the range.

For Autozero Off, do the following:

For daily and long term drift, add 110 µA/Day for the 150 mA range and 120 µA/Day for the 1A range.

For temperature drift, add 2 µA per °C for 150 mA range, and 15 µA per °C for the 1A range.

Digital Input and Output (Option 1D)

Table A–58: Digital Input Characteristics

Characteristic	Description
Number Of Pins	32 (shared with Digital Output)
Input/Output Selectability	Each pin in individually useable as an input pin, as and output pin, or as a bidirectional open collector pin
Threshold Level *	0.01 - 20.00 V, programmable in 5 mV steps
Accuracy (typical)	±(40 mV + 0.8% of setting), 1 year
Input Loading	8.25 K pullup to selected Digital Output excitation (5 VDC, 12 VDC, 24 VDC or external) additional nominal 23.0 K load to 5 VDC or 0 VDC depending on input level (all excitation selections)
Hysteresis	.12 VDC nominal
External Handshake Characteristics	Request TTL compatible (with 1 KΩ pullup), minimum pulse width = 50 nsec, programmable active high or low (default). Request input must be in the inactive state prior to selecting Handshake as the trigger source, input data setup 0, hold time, 5.3 µsec of the Request pulse leading edge. Acknowledge output pulse, TTL compatible, 1.04 µsec pulse width, programmable, high or low (default), the Acknowledge pulse leading edge may be used to indicate that the input hold time has been met.
Clock *	External handshake (48 kHz maximum) or internal (3.662 Hz to 48 kHz)
Memory *	4095 maximum prematch, 4096 maximum postmatch

*** Must be identical for all 32 bits**

Table A–59: Digital Output Characteristics

Characteristic	Description
Number Of Pins	32 (shared with Digital Input)
Input/Output Selectability	Each pin in individually useable as an input pin, as and output pin, or as a bidirectional open collector pin
Output Level *	+5 V, +12 V, +24 V, or user-supplied (+5 V to +32 V)
Output Sink Current	50 mA maximum at V_{CESAT} of 0.4 VDC maximum.
Source Current	The following assumes the Digital Input instrument threshold is set to 50% (or less) of the Digital Output excitation level: 5 VDC excitation, 240 µA at 3.5 VDC 12 VDC excitation, 350 µA at 8.0 VDC 24 VDC excitation, 500 µA at 16.0 VDC
Rise/Fall Time	5.21 µsec, 10 % to 90%, 90% to 10%
Memory *	4096 samples
Clock*	External handshake (48 kHz maximum) or internal (3.662 Hz to 48 kHz)

Table A–59: Digital Output Characteristics (Cont.)

Characteristic	Description
External Handshake Characteristics	Request TTL compatible (with 1 kΩ pullup), minimum pulse width = 50 nsec, programmable active high or low (default). Request input must be in the inactive state prior to selecting Handshake or the trigger source, update will occur within 5.21 µsec of the Request Leading edge pulse, TTL compatible 1.04 µsec nominal pulse width, programmable high or low pulse, leading edge may be used as a handshake acknowledge, trailing edge as a settled indication.
Segment Repeat Count *	1-4096 or continuous
Segment Repeat Period *	20.83 µs to 16 seconds

***	Must be identical for all 32 bits**

Digital to Analog Converter (DAC) (Option 1A)

Table A–60: Digital to Analog Converter Characteristics

Characteristic	Description
Number Of Channels	8
Vertical Resolution	112 bits (3501 levels)
Output	± 14 V with 8 mV resolution
Accuracy	±(20.0 mV + 0.5% of reading)
Drift	±(1 mV + 0.025% of reading)/month
Protection	Short circuit to ground, indefinite. Do not short to other DAC outputs (channel high output pins are placed non-adjacent to help prevent this from happening).
Setting Time	33.3 µsec to published accuracy
Memory *	1024 samples per channel
Clock	External handshake (15 kHz maximum) or internal (3.6662 Hz - 15 kHz)
External Handshake Characteristics	Request TTL compatible (with 1 kΩ pullup), minimum pulse width = 50 ns, programmable, active, high or low (default), Request input must be in inactive state prior to selecting handshake as the trigger source, update will occur within 5.21 µsec of the Request Leading edge. Strobe, Output pulse, TTL compatible, 2.0 pulse width, programmable high or low (default) pulse, leading edge may be used as a handshake acknowledge, trailing edge as a settled indication.

Table A–60: Digital to Analog Converter Characteristics (Cont.)

Characteristic	Description
Maximum Output Load	<30°C: >800 Ω (600 Ω to ±12 VDC) <55°C: >1545 Ω (1345 Ω to ±12 VDC)
Segment Repeat Count *	1-4096 or continuous
Segment Repeat Period *	66.7 µS to 16 s

*** Must be identical for all 32 bits**

Relay Drivers (Option 1D)

Table A–61: Relay Drivers Characteristics

Characteristic	Description
Number Of Bits	8
Configuration	Open collector
Maximum sink current	100 mA each channel
Maximum Output Level at 100 mA	1.2 V
Maximum Switching Voltage	35 V

SurePath Specifications

The VX4101A controls the following currently available SurePath™ switching modules as slaves:

- VX4320 1.3 GHz RF Multiplexer
- VX4330 120 Channel Scanner/Multiplexer
- VX4350 64 Channel SPDT
- VX4380 256 Cross Point Relay Matrix Module
- VX4351 40-Channel, 10 Amp, SPST Switch Module

Maximum number of slave modules: 11

Certifications and Compliances

Table A–62: Certifications and compliances

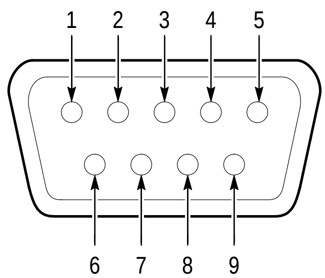
Characteristics	Description
EC Declaration of Conformity - EMC	Meets intent of EMC Directive 89/336/EEC for Electromagnetic Compatibility. Compliance was demonstrated to the following specifications as listed in the Official Journal of the European Communities: EN 55011 — Class A Radiated and Conducted Emissions EN 50081-1 Emissions: EN 60555-2 — AC Power Line Harmonic Emissions EN 50082-1 Immunity: IEC 801-2 — Electrostatic Discharge Immunity IEC 801-3 — RF Electromagnetic Field Immunity IEC 801-4 — Electrical Fast Transient/Burst Immunity IEC 801-5 — Power Line Surge Immunity To ensure compliance with EMC requirements, this module must be installed in a mainframe which has backplane shields installed complying with Rule B.7.45 of the VXIbus Specification. Also, only high quality shielded cables having a reliable, continuous outer shield with low impedance connections to shielded connector housings at both ends should be connected to this product.
Australian Declaration of Conformity - EMC	Conforms with the following standards in accordance with the Electromagnetic Compatibility Framework: AS/NZS 2064.1/2 — Class A Radiated and Conducted Emissions To ensure compliance with EMC requirements, this module must be installed in a mainframe which has backplane shields installed which comply with Rule B.7.45 of the VXIbus Specification. Additionally, only high quality shielded cables having a reliable, continuous outer shield which has low impedance connections to shielded connector housings at both ends should be connected to this product.
EC Declaration of Conformity - Low Voltage	Meets intent of Low Voltage Directive 73/23/EEC as amended by 93/68/EEC, for Product Safety. Compliance was demonstrated to the following specification as listed in the Official Journal of the European Communities: EN 61010-1/A2:1995 — Safety requirements for electrical equipment for measurement, control, and laboratory use.
Approvals	UL3111-1 - Standard for electrical measuring and test equipment CAN/CSA C22.2 No. 1010.1 - Safety requirements for electrical equipment for measurement, control and laboratory use.
Safety Certification of Plug-in or VXI Modules	For modules (plug-in or VXI) that are safety certified by Underwriters Laboratories, UL Listing applies only when the module is installed in a UL Listed product. For modules (plug-in or VXI) that have cUL or CSA approval, the approval applies only when the module is installed in a cUL or CSA approved product.
Installation Category Descriptions	Terminals on this product may have different installation category designations. The installation categories are:

Table A–62: Certifications and compliances (cont.)

Characteristics	Description
	CAT III Distribution-level mains (usually permanently connected). Equipment at this level is typically in a fixed industrial location. CAT II Local-level mains (wall sockets). Equipment at this level includes appliances, portable tools, and similar products. Equipment is usually cord-connected. CAT I Secondary (signal level) or battery operated circuits of electronic equipment.

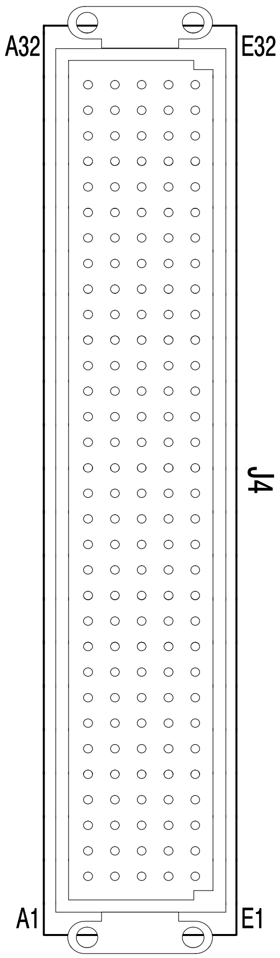
Appendix B: Input/Output Connections

Table B–1: Digital Multimeter (DMM) Input/Output Connections



Pin Number	Signal
6	Voltage/2-wire Ω +, 4-wire Ω +
1	Voltage/2-wire Ω –, 4-wire Ω –
9	4-wire Ω +
5	4-wire Ω –
7	Current +
3	Current –
2	Not connected
4	Not connected
8	Not connected

Table B–2: 160-Pin Connector Pinouts



Pin number	Signal description	Input or Output	Maximum voltage limits
23B	DAC High Channel 1	Output	± 14 VDC
23A	DAC Low Channel 1	Output	Ground
22E	DAC High Channel 2	Output	± 14 VDC
23E	DAC Low Channel 2	Output	Ground
22C	DAC High Channel 3	Output	± 14 VDC
22D	DAC Low Channel 3	Output	Ground
22A	DAC High Channel 4	Output	± 14 VDC
22B	DAC Low Channel 4	Output	Ground
21D	DAC High Channel 5	Output	± 14 VDC
21C	DAC Low Channel 5	Output	Ground
21B	DAC High Channel 6	Output	± 14 VDC
21A	DAC Low Channel 6	Output	Ground
20E	DAC High Channel 7	Output	± 14 VDC
21E	DAC Low Channel 7	Output	Ground
20C	DAC High Channel 8	Output	± 14 VDC
20D	DAC Low Channel 8	Output	Ground
20A	DAC Request *	Input	TTL
20B	DAC Strobe*	Output	TTL
17D	Digital I/O Pin 1	Input/Output	+0 - 32 VDC
17C	Digital I/O Pin 2	Input/Output	+0 - 32 VDC
17B	Digital I/O Pin 3	Input/Output	+0 - 32 VDC
17A	Digital I/O Pin 4	Input/Output	+0 - 32 VDC
16D	Digital I/O Pin 5	Input/Output	+0 - 32 VDC
16C	Digital I/O Pin 6	Input/Output	+0 - 32 VDC
16B	Digital I/O Pin 7	Input/Output	+0 - 32 VDC
16A	Digital I/O Pin 8	Input/Output	+0 - 32 VDC
15D	Digital I/O Pin 9	Input/Output	+0 - 32 VDC
15C	Digital I/O Pin 10	Input/Output	+0 - 32 VDC
15B	Digital I/O Pin 11	Input/Output	+0 - 32 VDC
15A	Digital I/O Pin 12	Input/Output	+0 - 32 VDC
14D	Digital I/O Pin 13	Input/Output	+0 - 32 VDC
14C	Digital I/O Pin 14	Input/Output	+0 - 32 VDC
14B	Digital I/O Pin 15	Input/Output	+0 - 32 VDC
14A	Digital I/O Pin 16	Input/Output	+0 - 32 VDC

Table B–2: 160-Pin Connector Pinouts (Cont.)

	Pin number	Signal description	Input or Output	Maximum voltage limits
	13D	Digital I/O Pin 17	Input/Output	+0 - 32 VDC
	13C	Digital I/O Pin 18	Input/Output	+0 - 32 VDC
	13B	Digital I/O Pin 19	Input/Output	+0 - 32 VDC
	13A	Digital I/O Pin 20	Input/Output	+0 - 32 VDC
	12D	Digital I/O Pin 21	Input/Output	+0 - 32 VDC
	12C	Digital I/O Pin 22	Input/Output	+0 - 32 VDC
	12B	Digital I/O Pin 23	Input/Output	+0 - 32 VDC
	12A	Digital I/O Pin 24	Input/Output	+0 - 32 VDC
	11D	Digital I/O Pin 25	Input/Output	+0 - 32 VDC
	11C	Digital I/O Pin 26	Input/Output	+0 - 32 VDC
	11B	Digital I/O Pin 27	Input/Output	+0 - 32 VDC
	11A	Digital I/O Pin 28	Input/Output	+0 - 32 VDC
	10D	Digital I/O Pin 29	Input/Output	+0 - 32 VDC
	10C	Digital I/O Pin 30	Input/Output	+0 - 32 VDC
	10B	Digital I/O Pin 31	Input/Output	+0 - 32 VDC
	10A	Digital I/O Pin 32	Input/Output	+0 - 32 VDC
	18B	Digital I/O External Excitation	Input	+0 - 32 VDC
	17E	Digital I/O Return	Input/Output	Ground
	18C	Digital I/O Return	Input/Output	Ground
	18A	Digital In Acknowledge*	Output	TTL
	13E	Digital In Request*	Input	TTL
	11E	Digital Out Request *	Input	TTL
	15E	Digital Out Acknowledge *	Output	TTL
	9E	Relay Driver 1	Output	+0 - 35 VDC
	9D	Relay Driver 2	Output	+0 - 35 VDC
	9C	Relay Driver 3	Output	+0 - 35 VDC
	9B	Relay Driver 4	Output	+0 - 35 VDC
	8E	Relay Driver 5	Output	+0 - 35 VDC
	8D	Relay Driver 6	Output	+0 - 35 VDC
	8C	Relay Driver 7	Output	+0 - 35 VDC
	8B	Relay Driver 8	Output	+0 - 35 VDC
	9A	Relay External Excitation	Input	+0 - 35 VDC
	8A	Relay Excitation Return	Input/Output	Ground
	24A	DMM Trigger *	Input	TTL

Table B–2: 160-Pin Connector Pinouts (Cont.)

	Pin number	Signal description	Input or Output	Maximum voltage limits
	19E	DMM Acknowledge *	Output	TTL
	24B	VX4101A MultiPaq™ Instrument External Trigger In	Input	TTL
	24E	VX4101A MultiPaq™ Instrument External Trigger Out	Output	TTL
	25E	Daughter Common Bus Signal 1	Input/Output	Reserved for daughter cards
	25D	Daughter Common Bus Signal 2	Input/Output	Reserved for daughter cards
	23C	Digital Ground	Input/Output	Ground
	23D	Digital Ground	Input/Output	Ground
	24C	Digital Ground	Input/Output	Ground
	24D	Digital Ground	Input/Output	Ground

Appendix C: Instrument I/O Operation

CAUTION. *If the user's mainframe has other manufacturers' computer boards operating in the role of VXIbus foreign devices, the assertion of BERR* (as defined by the VXIbus Specification) may cause operating problems on these boards.*

This section describes the input and output operations of the VX4101A MultiPaq™ Instrument. The VX4101A supports the Normal Transfer Mode of the VXIbus using the Write Ready, Read Ready, Data In Ready (DIR), and Data Out Ready (DOR) bits of the module Response register. The read and write operations in Normal Transfer Mode are as follows:

Normal Transfer Mode Read Operation

A Normal Transfer Mode read of the VX4101A Module proceeds as follows:

1. The commander reads the VX4101A Response register and checks if the Write Ready and DOR bits are true. If they are, the commander proceeds to the next step. If not, the commander continues to poll these bits until they become true.
2. The commander writes the Byte Request command (hexadecimal 0DEFF) to the Data Low register of the VX4101A.
3. The commander reads the VX4101A Response register and checks if the Read Ready bit is true. If it is, the commander proceeds to the next step. If not, the commander continues to poll this bit until it becomes true.
4. The commander reads the VX4101A Data Low register.

Normal Transfer Mode Write Operation

A Normal Transfer Mode write to the VX4101A Module proceeds as follows:

1. The commander reads the VX4101A Response register and checks if the Write Ready and DIR bits are true. If they are, the commander proceeds to the next step. If not, the commander continues to poll the Write Ready and DIR bits until they are true.
2. The commander writes the Byte Available command which contains the data (hexadecimal 0BCXX or 0BDXX, depending on the End bit) to the Data Low register of the VX4101A.

The module has no registers beyond those defined for VXIbus message based devices. All communications with the module are through the Data Low register, the Response register, or the VXIbus interrupt cycle. Any attempt by another module to read or write to any undefined location of the VX4101A address space may cause incorrect operation of the module.

As with all VXIbus devices, the VX4101A Module has registers located within a 64 byte block in the A16 address space. The base address of the VX4101A device registers is determined by the device unique logical address and can be calculated as follows:

$$\text{Base Address} = V_{16} * 40_{16} + C000_{16}$$

where V is the logical address of the device as set by the Logical Address switches.

Configuration Registers

Table C–1 contains a list of the VX4101A Configuration registers and a complete description of each register. The offset is relative to the module base address.

Table C–1: Register Definitions

Register	Address (hexadecimal)	Type	Value (Bits 15-0)
ID Register	0000	RO	1001 1111 1111 1101 (hexadecimal 9FFD)
Device Type	0002	RO	0111 0111 1001 1010 (hexadecimal 779A)
Status	0004	R	Defined by state of interface
Control	0004	W	Defined by state of interface
Offset	0006	WO	Assigned by Resource Manager
Protocol	0008	RO	1011 0111 1111 1111 (hexadecimal B7FF)
Response	000A	RO	Defined by state of the interface
Data High	000C		Not used
Data Low	000E	W	Not fixed; command-dependent
Data Low	000E	R	Not fixed; command-dependent

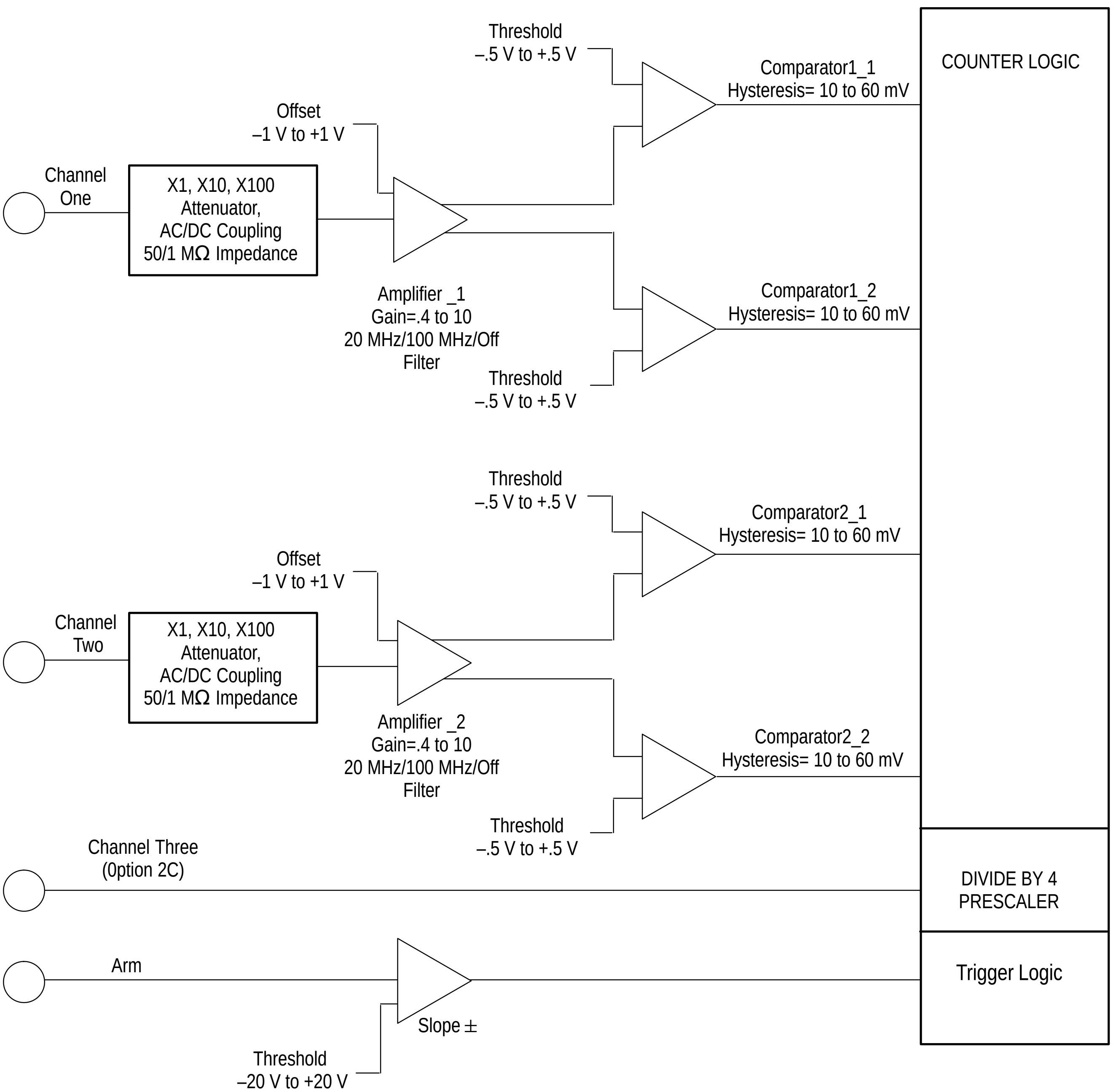
RO is Read Only
WO is Write Only
R is Read
W is Write

VMEbus Interrupt Level Selection

Each function module in a VXIbus System can generate an interrupt at a specified level on the VMEbus to request service from the interrupt handler located on its commander. The VX4101 supports programmable interrupt selection for setting the interrupt level for the device.

Interrupts are used by the module to return VXIbus Protocol Events to the module commander.

Appendix D: Counter Architecture



Appendix E: Obsolete Commands

The following SENSe subsystem commands provide backward compatibility with the previously released Counter included with the VX4101 DMM/Counter. For complete information about these commands, consult the VX4101 DMM/Counter User Guide.

Counter Commands

The following commands are backwards compatible with the VX4101 DMM/Counter:

SENSe Subsystem

```
SENSe[:...]:COUNt
SENSe[:...]:COUNt?
SENSe[:...]:APERture
SENSe[:...]:APERture?
SENSe[:...]:EVENts
SENSe[:...]:EVENts?
SENSe[:...]:MODe
SENSe[:...]:MODe?
```

On the VX4101, you were required to replace [:...] with one of the functions below. This is no longer necessary on the VX4101A. For example, the VX4101A command to set the aperture was SENSe:FREQuency:APERture. The new VX4101A command is SENSe:APERture.

```
VOLTage:AC
VOLTage:DC
VOLTage:MINimum
VOLTage:MAXimum
VOLTage:PTP
FREQuency
PERiod
FREQuency:RATio
RISE:TIME
RTIME
FALL:TIMe
FTIMe
PWIDth
NWIDTH
TINTerval
```

***NOTE**. The SENSe:TINTerval:DELay[STATe] commands were used by the VX4101 to switch off the delay by time capability. With the VX4101A, you can now program time interval with delay by events.*

The SENSe:FUNCtion commands included with the VX4101A enable a variety of time interval programing options, including time interval, time interval with delay by time, or time interval with delay by events using SENSe:FUNCtion.

This means that the command SENSe:TINTerval:DELay[:STATe] and the associated SENSe:TINTerval:DELay[:STATe] query are no longer necessary and are not active in the VX4101A.

WARNING

The following servicing instructions are for use only by qualified personnel. To avoid injury, do not perform any servicing other than that stated in the operating instructions unless you are qualified to do so. Refer to all Safety Summaries before performing any service.

Appendix F: Performance Verification Procedure

Semi-Automated PVP Procedures

The VX4101A MultiPaq™ Instrument is designed to run semi-automated performance verification procedures (PVPs) using the VXI*plug&play* soft front panels.

There is a menu choice on the pull-down menus of each soft front panel/

Manual PVP Procedures

Manual PVP procedures are included on the media accompanying your instrument. You can use them if you do not want to use the semi-automated procedures. The files are in .PDF format for viewing with the Adobe Acrobat viewer. The file names are as follows:

CNTRPVP.PDF
DMMPVP.PDF
DACPVP.PDF

***NOTE**. The manual PVPs for the DMM and Counter are very long and complex. Tektronix recommends that you use the semi-automated PVPs available through the soft front panels.*

Appendix G: Calibration

This section contains calibration procedures for the following modules in the VX4101A MultiPaq™ Instrument:

- Digital Multimeter (DMM)
- Counter
- Digital to Analog Converter (DAC)

Calibration for the DMM

The DMM calibration procedure consists of the following:

- Null and gain calibration in all DC ranges
- All AC TRMS (DC Coupled) ranges
- All 4-wire resistance ranges and the two current ranges.

An additional 2-wire Ω null calibration in the 30 Ω range is required and may be used to compensate for 2-wire Ω field wiring.

Nonzero values specified below may be ±10% of the value specified, with a maximum value equal the value of the range. Enter the exact value as displayed on the calibrator (or as measured by an accurate system DMM if a DMM is used as a transfer standard. The tolerances shown following the applied values below are the accuracy to which the `cal:val` command argument must be known. A calibration standard meeting those accuracies must be used (or an accurate system DMM used as a transfer standard).

NOTE. *See the CALibration:VALue command description in the SCPI Command section of this user manual for a description of the calibration commands and additional information about the calibration of the* VX4101A *DMM.*

Between applying an external voltage and sending the cal:val command in the procedures below, sufficient time should be allowed for both the calibration source and the VX4101A input circuit to settle to the tolerance shown. A good way to assure this in an automated program is to take continuous measurements with the VX4101A DMM after the voltage, resistance or current is applied, make sure that the reading is stable within a value equal to 25% of the accuracy of the measurement, and that the VX4101A measurement is reasonable.

Recommended Calibration Interval

1 year

Before You Begin

Connect calibrator voltage outputs to VX4101A voltage inputs.
Program the VX4101A as follows:

```
inst:sel dmm
cal:sour ext
```

After completion, program the VX4101A as follows:

```
cal:sour int
```

DC Mode Calibration Procedure

Perform this procedure to calibrate the DMM through its entire range of DC measurements. This procedure is separated into phases to permit calibration of all ranges, but should be run in its entirety. Perform the following steps to calibrate the DMM in DC mode:

DC Range .03 VDC with 10 MΩ Impedance

This part of the procedure calibrates the DMM for 0.03 VDC with 10 MΩ impedance.

1. Program the VX4101A as follows:

   ```
   conf:dc .03
   ```

2. Program the VX4101A as follows:

   ```
   inp:imp 10e6; init
   ```

3. Apply 0.0 VDC ±100 nVDC
4. Program the VX4101A as follows:

   ```
   cal:val 0
   ```

5. Apply 0.029 VDC±1.9 μVDC
6. Program the VX4101A as follows:

   ```
   cal:val .029
   ```

DC Range .3 VDC with 10 MΩ Impedance

This part of the procedure calibrates the DMM for 0.3 VDC with 10 MΩ impedance.

7. Program the VX4101A as follows:

```
conf:dc .3;init
```

8. Apply 0.0 VDC ±1 μVDC

9. Program the VX4101A as follows:

```
cal:val 0
```

10. Apply 0.290 VDC ±5.3 μVDC

11. Program the VX4101A as follows:

`cal:val .29` (or any other desired value)

DC Range 3 VDC with 10 MΩ Impedance

This part of the procedure calibrates the DMM for 3 VDC with 10 MΩ impedance.

12. Program the VX4101A as follows:

```
conf:dc 3;init
```

13. Apply 0.0 VDC ±4 μVDC

14. Program the VX4101A as follows:

```
cal:val 0
```

15. Apply 2.90 VDC ±35 μVDC

16. Program the VX4101A as follows:

`cal:val 2.9` (or any other desired value)

DC Range 30 VDC with 10 MΩ Impedance

This part of the procedure calibrates the DMM for 30 VDC with 10 MΩ impedance.

17. Program the VX4101A as follows:

```
conf:dc 30;init
```

18. Apply 0.0 VDC ±40 μVDC

19. Program the VX4101A as follows:

```
cal:val 0
```

20. Apply 29.0 VDC ±450 μVDC

21. Program the VX4101A as follows:

`cal:val 29` (or any other desired value)

DC Range 300 VDC with 10 MΩ Impedance

This part of the procedure calibrates the DMM for 300 VDC with 10 MΩ impedance.

22. Program the VX4101A as follows:

```
conf:dc 300;init
```

23. Apply 0.0 VDC ±400 μVDC

24. Program the VX4101A as follows:

```
cal:val 0
```

25. Apply 290 VDC ±4.5 mVDC

26. Program the VX4101A as follows:

`cal:val 290` (or any other desired value)

DC Range .03 VDC with 10 GΩ Impedance

This part of the procedure calibrates the DMM for 0.03 VDC with 10 GΩ impedance.

27. Program the VX4101A as follows:

```
conf:dc .03
```

28. Program the VX4101A as follows:

```
inp:imp 10e9;init
```

29. Apply 0.0 VDC ±100 nVDC

30. Program the VX4101A as follows:

```
cal:val 0
```

31. Apply 0.029 VDC ±1.9 μVDC

32. Program the VX4101A as follows:

`cal:val .029` (or any other desired value)

DC Range 0.3 VDC with 10 GΩ Impedance

This part of the procedure calibrates the DMM for 0.3 VDC with 10 GΩ impedance.

33. Program the VX4101A as follows:

```
conf:dc .3;init
```

34. Apply 0.0 VDC ±1 μVDC

35. Program the VX4101A as follows:

```
cal:val 0
```

36. Apply 0.290 VDC ±5.3 μVDC

37. Program the VX4101A as follows:

`cal:val .29` (or any other desired value)

DC Range 3 VDC with 10 GΩ Impedance

This part of the procedure calibrates the DMM for 3 VDC with 10 GΩ impedance.

38. Program the VX4101A as follows:

```
conf:dc 3;init
```

39. Apply 0.0 VDC ±4 μVDC

40. Program the VX4101A as follows:

```
cal:val 0
```

41. Apply 2.90 VDC ±35 μVDC

42. Program the VX4101A as follows:

`cal:val 2.9` (or any other desired value)

TRMS AC (DC Coupled) Mode Calibrations

This procedure will calibrate the DMM throughout its range of TRMS AC (DC Coupled) measurements. This procedure is separated into phases to permit calibration of all ranges, but should be run in its entirety. The procedure is as follows:

AC/DC Coupled Range .03 VDC or RMS

This part of the procedure calibrates the DMM for AC/DC Coupled Range 0.03 VDC or RMS.

1. Program the VX4101A as follows:

```
conf:acdc .03;init
```

2. Apply 0.009 VRMS ±1.9 μVRMS at 1 kHz

3. Program the VX4101A as follows:

```
cal:val .009
```

4. Apply .029 VRMS ±1.9 μVRMS at 1 kHz

5. Program the VX4101A as follows:

`cal:val .029` (or any other desired value)

AC/DC Coupled Range 0.3 VDC or RMS

This part of the procedure calibrates the DMM for AC/DC Coupled Range 0.3 VDC or RMS.

6. Program the VX4101A as follows:

```
conf:acdc .3;init
```

7. Apply .03 VRMS ±1.9 μVRMS at 1 kHz

8. Program the VX4101A as follows:

`cal:val .03` (or any other desired value)

9. Apply .290 VRMS ±5.3 μRMS at 1 kHz

10. Program the VX4101A as follows:

`cal:val .29` (or any other desired value)

AC/DC Coupled Range 3 VDC or RMS

This part of the procedure calibrates the DMM for AC/DC Coupled Range 3 VDC or RMS.

11. Program the VX4101A as follows:

`conf:acdc 3;init`

12. Apply .300 VRMS ±5.3 μVRMS at 1 kHz

13. Program the VX4101A as follows:

`cal:val .3` (or any other desired value)

14. Apply 2.90 VRMS ±35 μVRMS at 1 kHz

15. Program the VX4101A as follows:

`cal:val 2.9` (or any other desired value)

AC/DC Coupled Range 30 VDC or RMS

This part of the procedure calibrates the DMM for AC/DC Coupled Range 30 VDC or RMS.

16. Program the VX4101A as follows:

`conf:acdc 30;init`

17. Apply 3.0 VRMS ±35 μVRMS at 1 kHz

18. Program the VX4101A as follows:

`cal:val 3` (or any other desired value)

19. Apply 29.0 VRMS ±450 μVRMS at 1 kHz

20. Program the VX4101A as follows:

`cal:val 29` (or any other desired value)

AC/DC Coupled Range 300 VDC or RMS

This part of the procedure calibrates the DMM for AC/DC Coupled Range 300 VDC or RMS.

21. Program the VX4101A as follows:

`conf:acdc 300;init`

22. Apply 30.0 VRMS ±450 μVRMS at 1 kHz

23. Program the VX4101A as follows:

`cal:val 30` (or any other desired value)

24. Apply 290 VRMS ±4.5 mVRMS at 1 kHz

25. Program the VX4101A as follows:

`cal:val 290` (or any other desired value)

***NOTE**. TRMS AC (AC Coupled) ranges use the above calibration information. A separate TRMS AC (AC Coupled) calibration is not required.*

Resistance Mode Calibration Procedure

This procedure will calibrate the DMM through its entire range of resistance measurement settings. The procedure is separated into several phases, but should be run in its entirety. The procedure is as follows:

Resistance 30 Ω 4-Wire

This part of the procedure calibrates the DMM for 30 Ω Resistance, 4-wire.

1. Program the VX4101A as follows:

`conf:fres 30;init`

2. Apply 0.0 Ω ±1.0 mΩ

3. Program the VX4101A as follows:

`cal:val 0`

4. Apply 10.0 Ω ±1.3 mΩ

5. Program the VX4101A as follows:

`cal:val 10` (or any other desired value)

Resistance 30 Ω 2-Wire

This part of the procedure calibrates the DMM for 30 Ω Resistance, 2-wire.

6. Program the VX4101A as follows:

`conf:res 30;init`

7. Apply 0.0 Ω ±1.0 mΩ

8. Program the VX4101A as follows:

`cal:val 0`

9. Apply 10.0 Ω ±1.3 mΩ

10. Program the VX4101A as follows:

`cal:val 10` (or any other desired value)

Resistance 300 Ω

This part of the procedure calibrates the DMM for 300 Ω resistance.

11. Program the VX4101A as follows:

`conf:fres 300;init`

12. Apply 0.0 Ω ±1.0 mΩ

13. Program the VX4101A as follows:

`cal:val 0`

14. Apply 100.0 Ω ±3 mΩ

15. Program the VX4101A as follows:

`cal:val 10` (or any other desired value)

Resistance 3 kΩ

This part of the procedure calibrates the DMM for 300 kΩ resistance.

16. Program the VX4101A as follows:

`⊂onf:fres 3000;init`

17. Apply 0.0 Ω ±4.0 mΩ

18. Program the VX4101A as follows:

`cal:val 0`

19. Apply 1000.0 Ω ±16 mΩ

20. Program the VX4101A as follows:

`cal:val 100` (or any other desired value)

Resistance 30 kΩ

This part of the procedure calibrates the DMM for 30 kΩ resistance.

21. Program the VX4101A as follows:

`conf:fres 30000;init`

22. Apply 0.0 Ω ±40 mΩ

23. Program the VX4101A as follows:

`cal:val 0`

24. Apply 10 kΩ ±.16 Ω
25. Program the VX4101A as follows:

    `cal:val 10e3` (or any other desired value)

Resistance 300 kΩ

This part of the procedure calibrates the DMM for 300 kΩ resistance.

26. Program the VX4101A as follows:

    `conf:fres 300e3;init`

27. Apply 0.0 Ω ±.4 Ω
28. Program the VX4101A as follows:

    `cal:val 0`

29. Apply 100 kΩ ±1.8 Ω
30. Program the VX4101A as follows:

    `cal:val 100e3` (or any other desired value)

Resistance 3 MΩ

This part of the procedure calibrates the DMM for 3 MΩ resistance.

31. Program the VX4101A as follows:

    `conf:fres 3e6;init`

32. Apply 0.0 Ω ±8 Ω
33. Program the VX4101A as follows:

    `cal:val 0`

34. Apply 1.0 MΩ ± 48 Ω
35. Program the VX4101A as follows:

    `cal:val 1e6` (or any other desired value)

Resistance 30 MΩ

This part of the procedure calibrates the DMM for 30 MΩ resistance.

36. Program the VX4101A as follows:

    `conf:fres 30e6;init`

37. Apply 0.0 Ω ±160 Ω

38. Program the VX4101A as follows:

```
cal:val 0
```

39. Apply 10 MΩ ±2000 Ω

40. Program the VX4101A as follows:

`cal:val 10e6` (or any other desired value)

Resistance 300 MΩ

This part of the procedure calibrates the DMM for 300 MΩ resistance.

41. Program the VX4101A as follows:

```
conf:fres 300e6;init
```

42. Apply 0.0 Ω ±1600 Ω

43. Program the VX4101A as follows:

```
cal:val 0
```

44. Apply 100 MΩ ±200 kΩ

45. Program the VX4101A as follows:

`cal:val 100e6` (or any other desired value)

***NOTE**. 2-wire resistance Ω ranges use the above calibration information. A separate 2-wire Ω calibration (other than at 30 Ω) is not required.*

Current Mode Calibration Procedure

This procedure calibrates the DMM in current mode throughout its range of current measurements. This procedure is separated into phases to permit calibration at all ranges, but is intended to be run in its entirety. The procedure is as follows:

Current .15A

This part of the procedure calibrates for 0.15A current.

1. Program the VX4101A as follows:

```
conf:curr .15;init
```

2. Apply 0.0 mA ±8 μA

3. Program the VX4101A as follows:

```
cal:val 0
```

4. Apply 140 mA ±35 μA

5. Program the VX4101A as follows:

`cal:val .14` (or any other desired value)

Current 1A

This part of the procedure calibrates for 1A current.

6. Program the VX4101A as follows:

```
conf:curr 1;init
```

7. Apply 0.0 mA ±14 μA

8. Program the VX4101A as follows:

```
cal:val 0
```

9. Apply 900 mA ±0.28 mA

***NOTE**. This measurement takes extra time to settle*

10. Program the VX4101A as follows:

`cal:val 1` (or any other desired value)

Calibration for the Counter

This procedure shows you how to calibrate the Counter for the VX4101A MultiPaq™ Instrument.

Prerequisites

It is assumed the module has completed its power-on self test. For information on specific commands or syntax, please review the *Syntax and Commands* section.

Equipment Required

You need the following equipment to calibrate the Counter:

- Precision DC calibrator
- Function generator (1 kHz square wave at ±2.5 V)
  (10 MHz square wave at ±1.0 V)
- High frequency signal generator (Required for factory calibration only)
  (1000 MHz Sine @–21 dBm, 10 MHz Sine @ 0 dBm)
- High accuracy clock standard, such as a rubidium clock source (10 MHz @ 0 dBm)

Recommended Calibration Interval

1 year

What You Should Know About

Before you begin the calibration procedures for the Counter, you should understand the following:

Channel Specifications. For the calibration procedures referenced below, [<channel>] is either 1 or 2, depending on the channel being calibrated. For example, the command CALibrate[<channel>]:ZERO could be specified as one of the following:

cal1:zero	to zero channel 1, or
cal2:zero	to zero channel 2

If [<channel>] is not specified (e.g. cal:zero), channel 1 is the default.

For channel 1, the input signal should be connected to the CH 1 input of the Counter.

For channel 2, the input signal should be connected to the CH 2 input of the Counter.

***NOTE**. For the ARM calibrations, the input should be connected to the SMB connector.*

For the channel 3 (factory) calibrations, the input should be connected to the SMA Connector, if this option is included.

Determining Status. To determine the status of a calibration command, do the following:

1. Issue the following query:

```
STATus:OPERation:CONDition?
```

***NOTE**. Bit 0 in the response data indicates the following:*

Bit 0 = 0	*Indicates that the calibration command is complete*
Bit 0 = 1	*Indicates that the calibration is in progress*

2. You can issue the following query to determine if there was a failure in the calibration:

```
SYSTem:ERRor?
```

***NOTE**. When polling the cards for calibration complete, it is recommended that the polling period be greater than 1 ms to minimize the overhead incurred for processing the query while the calibration is in progress. An alternative method to eliminate the polling overhead is to program the card to generate an SRQ (service request) interrupt when the calibration bit (described above) changes from a 1 to a 0*

About the Adjustment Procedures. The individual channel adjustments detailed in Step Two below should be performed in the sequence in which they appear for the channel 1 and channel 2 inputs of the Counter. Once these steps have been performed, the command function and cross channel calibrations steps in Step Three should be executed.

Step One: Initialize the instrument

To initialize the instrument and prepare it for adjustment, do the following:

1. Send the following command to select the counter function of the VX4101A:

```
INSTrument:SELect COUNTER
```

2. Connect the calibrator to the channel being calibrated.

Step Two: Individual Channel Adjustments

Adjust each channel as follows:

Offset Adjustment. To adjust offset, do the following:

1. Set the calibrator to 0.0 ±0.001 V.
2. For the channel being calibrated, send the command

```
CALibrate[<channel>]:ZERO
```

   This command will take approximately 110 seconds to execute.

Preamp Linearization. To adjust preamp linearization:

1. Set the calibrator to 0.5 V ±0.1%.
2. For the channel being calibrated, send the command

```
CALibrate[<channel>]:LINearity
```

This command takes approximately 30 seconds to execute.

Gain and Offset Gain Correction. To adjust gain and offset gain correction, perform the following steps for each of the voltages specified below:

1. For each of the following voltages, set the calibrator to the voltage value ±0.1%:

<voltage> Value	<voltage> Value	<voltage> Value	<voltage> Value
50	0.5	–10	–0.1
20	0.2	–5.0	0.05
10	0.1	–2.0	
5.0	0.05	–1.0	
2.0	–50	–0.5	
1.0	–20	–0.2	

2. For the channel being calibrated, send the following command:

```
CALibrate[<channel>]:VALue <voltage>
```

   where <voltage> is the value in the table above (e.g. cal1:val 50.0).

This command takes approximately 2 seconds to execute for each voltage value.

Hysteresis Calibration. Since this function uses an internal reference, it requires no external inputs. The calibrator should either be disconnected or set to 0 V for this step.

To calibrate hysteresis, do the following:

1. For the channel being calibrated, send the following command:

```
CALibrate[<channel>]:HYSTeresis
```

This command takes approximately 90 seconds to execute.

Low Frequency Compensation Adjustment. To adjust low frequency compensation:

1. Connect the function generator to the channel being calibrated, and set it to output a 1 kHz square wave at ±2.5 ±0.1 V (1 MΩ load impedance).
2. For the channel being calibrated, send the command

   ```
   CALibrate[<channel>]:LFCOmp
   ```

   This command takes approximately 6 seconds to execute.

Step Three: Common Function and Cross-Channel Calibration Adjustments

After performing all of the previous procedures steps for both channels, perform the following required additional steps as required:

ARM Input Zero and Gain Correction. To adjust ARM Input zero and gain Correction:

1. Set the calibrator to 0.00 ±0.001 V, and connect it to the ARM input of the card.
2. Send the command

   ```
   CALibrate:ARM:VALue 0.0
   ```

   This command takes approximately 0.5 second to execute.
3. Set the calibrator to 20.0 V ±0.1%.
4. Send the command

   ```
   CALibrate:ARM:VALue 20.0
   ```

   This command takes approximately 0.5 seconds to execute.

Digital Time Interpolation. To calibrate the Digital Time Interpolation, do as follows:

1. Connect the function generator for a square wave at ±0.5 V ±0.1 V @10 MHz to the channel 1 input (50 Ω load impedance).

2. Send the command:

   ```
   CALibrate:DTI
   ```

   This command takes approximately 16 seconds to execute.

Cross Channel Delays. To calibrate the cross channel delays, do as follows:

1. Using a 50 Ω RF splitter and equal length cables, connect the function generator to both the Channel 1 and Channel 2 inputs. Set up the function generator for a 10 MHz square wave at ±2.0 V ±0.1 V (±0.5 V at each 50 Ω input).

2. For determining the channel 1 cable delay, send the command:

   ```
   CALibrate1:DELay
   ```

   This command takes approximately 3 seconds to execute.

3. For determining the channel 2 cable delay, send the command:

   ```
   CALibrate2:DELay
   ```

   This command takes approximately 3 seconds to execute.

4. For determining the channel 1 to 2 cross channel delay, send the command:

   ```
   CALibrate:DELay 12
   ```

   This command takes approximately 3 seconds to execute.

5. For determining the channel 2 to 1 cross channel delay, send the command:

   ```
   CALibrate:DELay 21
   ```

   This command takes approximately 3 seconds to execute.

***NOTE**. If an invalid input signal is present, these commands will timeout and generate an error after approximately 5 seconds.*

Factory Calibration

The following additional steps are performed during initial factory calibration, when the channel 3 option is available. This step is not part of the normal calibration sequence, and is included here for completeness only.

Channel 3 Pre-Scaler BIAS Adjustment.

1. Set up the high frequency source for a sine wave at 1000 MHz ±100 kHz at –21 dBm, and connect it to the prescaler (channel 3) input.
2. Send the command

   ```
   CALibrate3:BIAS
   ```

   This command takes approximately 2 seconds to execute.

System Clock Calibration

There are two clock sources available which the Counter can use as its reference oscillator. These are the internal 10 MHz clock located on the VX4101A CPU card, and the VXI backplane 10 MHz clock.

Either or both of these sources could be used in the actual operation of the card. It is recommended that both sources be calibrated.

Selecting a noncalibrated source during operation will generate an error. For best results, the VXI clock should be calibrated in the actual system in which it is used.

Calibrate the clock sources as follows:

1. Connect the High Accuracy Clock Standard into the Channel 1 input, and set it to 10 MHz @ 0 dBm. To do this, send the following command:

   ```
   INP1:IMP50
   ```

2. Select the source of the clock to be calibrated, as follows:

   ```
   SOURce:COSCillator ROSC
   ```

3. Calibrate the internal clock with the following command:

   ```
   SOURce:ROSCillator INTernal
   ```

4. Calibrate the VXI backplane clock with the following command:

   ```
   SOURce:ROSCillator CLOCk10
   ```

5. Calibrate the Option 1T, TXC01 with the following command:

   ```
   SOURce:COSCillator TXCO1
   ```

6. To complete the calibration, issue the following command:

```
CALibrate:ROSCillator
```

This command takes approximately 11 seconds to execute.

***NOTE**. If an invalid input signal is present, this command will time out and generate an error after approximately 12 seconds.*

Calibration for the Digital to Analog Converter

This procedure shows you how to calibrate the Digital to Analog Converter (DAC) of the VX4101A MultiPaq™ Instrument. Each of the eight channels of the DAC requires a separate offset and gain calibration. In this procedure, you will initialize the instrument, and then calibrate each channel in succession for offset, and then for gain.

Prerequisites

It is assumed the module has completed its power-on self test. For information on specific commands or syntax, please review the *Syntax and Commands* section.

Equipment Required

You need the following equipment to calibrate the DAC:

- 4 1/2 digit or better digital voltmeter

Recommended Calibration Interval

1 Year

What You Should Know About

Before you begin the calibration procedure, you should understand the following:

DAC <channel> Values For the commands referenced below, [<channel>] can be 1 through 8, depending on the DAC channel being calibrated. For example, the command CALibrate[<channel>]:OUTput 1.0 could be specified as

cal1:out 8 for outputting 8 volts on channel one

cal8:out 0 for outputting 0 volts on channel eight

***NOTE**. If you do not enter a specific value for <channel>, the instrument will use channel one as the default.*

DAC Pin Assignments The Pin connections for the eight channels are shown in the table below.

Table G–1: DAC Pin Connections

Channel	Positive (+) Output	Negative (–) Output
1	23B	23A
2	22E	23E
3	22C	22D
4	22A	22B
5	21D	21C
6	21B	21A
7	20E	21E
8	20C	20D

Querying Status To determine the status of a calibration command, issue the following query:

STATus:OPERation:CONDition? query.

Bit 0 (LSB) in the response indicates the following:

Bit 0 = 0 Calibration command complete

Bit 0 = 1 Calibration in progress.

You can send a SYSTem:ERRor? query to determine if there was a failure in the calibration.

***NOTE**. When polling the device to see if the calibration is complete, it is recommended that the polling period be greater than 1 mS to minimize the overhead incurred for processing the query while the calibration is in progress. An alternative method to eliminate the polling overhead is to program the card to generate an SRQ (service request) interrupt when the calibration bit (described above) transitions from a 1 to a 0.*

Before You Begin

To prepare instrument for calibration, do the following:

1. Connect the digital voltmeter to the channel being calibrated.
2. Send the following command to select the DAC as the active instrument:

```
INSTrument:SELect DAC
```

DAC Calibration Procedure

The following procedure shows you how to program a single DAC channel. You should perform the procedure for each DAC channel in succesion for each of the eight DAC channels.

***NOTE**. Until you complete the calibration procedure for all eight DAC channels, the instrument will return a DAC calibration error.*

Adjusting Offset

This part of the procedure adjusts offset for a selected DAC channel. You will select the DAC channel that you want to calibrate, and then adjust the channel total of six times.

1. For the channel being calibrated, set the DAC output to nominal zero by sending the command

```
CALibrate[<channel>]:OUTput 0.0
```

2. Send the following command:

```
CALibrate[<channel>]:VALue 0.0,<DVM reading>
```

***NOTE**. <DVM reading> is the value measured by the digital voltmeter*

---

3. Change the value in <DVM reading> to the most recent reading from the digital voltmeter.
4. Repeat steps two and three a total of six times, each time changing the value in <DVM reading> to the most recent reading from the digital voltmeter.

Adjusting Gain

This part of the procedure calibrates the gain for a single DAC channel. You will select the DAC channel that you want to calibrate, and then make a total of twenty adjustments to the channel. The first six iterations are a coarse adjustment. The next 14 find the optimal value centered around the best value found in the first six tries.

1. Set the DAC output for the channel being calibrated to nominal 8 V with the following command:

```
CALibrate[<channel>]:OUTput 8.0
```

2. Send the following command:

```
CALibrate[<channel>]:VALue 8.0,<DVM reading>
```

***NOTE**. <DVM reading> is the value measured by the digital voltmeter.*

3. Change the value in <DVM reading> to the most recent reading from the digital voltmeter.
4. Repeat steps two and three a total of twenty times, each time changing the value in <DVM reading> to the most recent reading from the digital voltmeter.

Repeating the Proceedure

Repeat the calibration procedure from the beginning for the next DAC channel.

Saving the Results in EEPROM

After you have calibrated all channels, you use this procedure to save the best values found in the offset and gain adjustment procedures in the EEPROM. The calibration channel of the DAC are then loaded with the values.

1. Send the following command once:

```
CALibrate[<channel>]:SAVE
```

Calibration for the Digital Input

This procedure shows you how to calibrate the Digital Input for the VX4101A MultiPaq™ Instrument. You must calibrate the instrument for both a 2.5 VDC and 12 VDC threshold.

Prerequisites

It is assumed the module has completed its power-on self test. For information on specific commands or syntax, please review the *Syntax and Commands* section.

You must hook up the external source to Discrete Input 1 and program the Digital Output excitation to other than external and make sure that Digital Output 1 is programmed for an inactive (high) output.

Equipment Required

You need the following equipment to calibrate the Digital Input:

- External voltage sources of 2.500 ± .001 VDC and 12.00 ± .001 VDC

Recommended Calibration Interval

1 Year

What You Should Know About

Before you begin the calibration procedure, you should understand the following:

- The Digital Output instrument uses the same pins as the Digital Input, so you must reset Digital Output 1 used by the Digital Input to ensure that it is inactive.

Before You Begin

To prepare instrument for calibration, do the following:

1. Program the Digital Output excitation for other than external and program Digital Output 1 inactive (high).

2. Connect the external voltage source to Digital Input 1.

3. Send the following command to select the Digital Input as the active instrument:

```
INSTrument:SELect DIGI
```

Digital Input Calibration Procedure

This procedure calibrates the Digital Input threshold. This is a 2-point calibration using voltages of 2.5 VDC and 12 VDC respectively.

Setting the 2.5 Volt Calibration Factor

In this procedure, you will calibrate the Digital Input for a 2.5 VDC threshold.

1. Set the external voltage source to 2.500 VDC.
2. Enter the following command to calibrate for 2.5 V:

```
CALibration:VALue 2.5
```

Setting the 12 V Calibration Factor

In this procedure, you will calibrate the Digital Input for a 12 VDC threshold.

1. Set the external voltage source to 12.00 VDC.
2. Enter the following command to calibrate for 12 V:

```
CALibration:VALue 12
```

Appendix H: User Service

This appendix contains service-related information for the VX4101A MultiPaq™ that covers the following topics:

- Performance Verification
- Preventive maintenance
- Troubleshooting

Performance Verification

See Appendix F.

Preventive Maintenance

You should perform inspection and cleaning as preventive maintenance. Preventive maintenance, when done regularly, may prevent VX4101A malfunction and enhance reliability. Inspect and clean the VX4101A as often as conditions require by following these steps:

1. Turn off power and remove the VX4101A from the VXIbus mainframe.
2. Remove loose dust on the outside of the instrument with a lint-free cloth.
3. Remove any remaining dirt with a lint-free cloth dampened with water or a 75% isopropyl alcohol solution. Do not use abrasive cleaners.

Troubleshooting

If you suspect a malfunction, first double check connections to and from the VX4101A. If the trouble persists, perform a self test.

If the self test indicates a failure, contact your Tektronix field office or representative for assistance.

Appendix I: Replaceable Parts

This section contains a list of the replaceable modules for the VX4101A MultiPaq™ Instrument. Use this list to identify and order replacement parts.

Parts Ordering Information

Replacement parts are available through your local Tektronix field office or representative.

Changes to Tektronix products are sometimes made to accommodate improved components as they become available and to give you the benefit of the latest improvements. Therefore, when ordering parts, it is important to include the following information in your order:

- Part number
- Instrument type or model number
- Instrument serial number
- Instrument modification number, if applicable

If you order a part that has been replaced with a different or improved part, your local Tektronix field office or representative will contact you concerning any change in part number.

Change information, if any, is located at the rear of this manual.

Module Servicing

Modules can be serviced by selecting one of the following three options. Contact your local Tektronix service center or representative for repair assistance.

Module Exchange. In some cases you may exchange your module for a remanufactured module. These modules cost significantly less than new modules and meet the same factory specifications. For more information about the module exchange program, call 1-800-TEK-WIDE, extension 6630.

Module Repair and Return. You may ship your module to us for repair, after which we will return it to you.

New Modules. You may purchase replacement modules in the same way as other replacement parts.

Using the Replaceable Parts List

This section contains a list of the mechanical and/or electrical components that are replaceable for the VX4101A MultiPaq™ Instrument. Use this list to identify and order replacement parts. The following table describes each column in the parts list.

Parts list column descriptions

Column	Column name	Description
1	Figure & index number	Items in this section are referenced by figure and index numbers to the exploded view illustrations that follow.
2	Tektronix part number	Use this part number when ordering replacement parts from Tektronix.
3 and 4	Serial number	Column three indicates the serial number at which the part was first effective. Column four indicates the serial number at which the part was discontinued. No entry indicates the part is good for all serial numbers.
5	Qty	This indicates the quantity of parts used.
6	Name & description	An item name is separated from the description by a colon (:). Because of space limitations, an item name may sometimes appear as incomplete. Use the U.S. Federal Catalog handbook H6-1 for further item name identification.
7	Mfr. code	This indicates the code of the actual manufacturer of the part.
8	Mfr. part number	This indicates the actual manufacturer's or vendor's part number.

Abbreviations

Abbreviations conform to American National Standard ANSI Y1.1–1972.

Mfr. Code to Manufacturer Cross Index

The table titled Manufacturers Cross Index shows codes, names, and addresses of manufacturers or vendors of components listed in the parts list.

Manufacturers cross index

Mfr. code	Manufacturer	Address	City, state, zip code
00779	AMP INC.	CUSTOMER SERVICE DEPT PO BOX 3608	HARRISBURG, PA 17105–3608
060D9	UNITREK CORPORATION	3000 COLUMBIA HOUSE BLVD, SUITE 1 20	VANCOUVER, WA 98661
0KB01	STAUFFER SUPPLY CO	810 SE SHERMAN	PORTLAND, OR 97214–4657
0KB05	NORTH STAR NAMEPLATE INC	5750 NE MOORE COURT	HILLSBORO, OR 97124–6474
0KM03	INSTRUMENT SPECIALTIES CO INC.	505 PORTER WAY	PLACENTIA, CA 92670
30817	INSTRUMENT SPECIALTIES CO INC	EXIT 53, RT 80 BOX A	DELAWARE WATER GAP, PA 18327
57997	EMI PRECISION	20350 71ST AVE NE SUITE C	ARLINGTON, WA 98223
62559	SCHROFF INC	170 COMMERCE DRIVE	WARWICK, RI 02886–2430
6V439	ERNI COMPONENTS INC	12701 NORTH KINGSTON AVENUE	CHESTER, VA 23831
75915	LITTELFUSE INC	800 E NORTHWEST HWY	DES PLAINES, IL 60016–3049
80009	TEKTRONIX INC	14150 SW KARL BRAUN DR PO BOX 500	BEAVERTON, OR 97077–0001
93907	CAMCAR DIV OF TEXTRON INC	ATTN: ALICIA SANFORD 516 18TH AVE	ROCKFORD, IL 611045181
TK0198	HAMILTON HALLMARK	9750 SW NIMBUS AVE	BEAVERTON, OR 97005
TK0588	UNIVERSAL PRECISION PRODUCT	1775 NW CORNELIUS PASS RD	HILLSBORO, OR 97124
TK1943	NEILSEN MANUFACTURING INC	3501 PORTLAND RD NE	SALEM, OR 97303
TK2548	XEROX CORPORATION	14181 SW MILLIKAN WAY	BEAVERTON, OR 97005
TK2626	OPTIMA ELECTRONIC PACKAGING SYSTEMS	2166 MOUNTAIN INDUSTRIAL BLVD.	TUCKER, GA 30084–5088
06383	PANDUIT CORP	17303 RIDGELAND AVE	TINLEY PARK, IL 60477–3048
22526	BERG ELECTRONICS INC	825 OLD TRAIL ROAD	ETTERS, PA 17319
2K262	BOYD CORPORATION	6136 NE 87TH AVENUE	PORTLAND, OR 97220

Replaceable parts list

Fig. & index number	Tektronix part number	Serial no. effective	Serial no. discont'd	Qty	Name & description	Mfr. code	Mfr. part number
1	174–2675–00			2	CA ASSY,SP,ELEC:100 OHM,1.9NS TWISTED PAIR,SHEILDED (REFERENCE DESIGNATOR, J322 to J323, and J222 to J42)	00779	620273–1
2	671–4153–XX			1	CIRCUIT BD ASSY:COUNTER BD	80009	671–4153–00
3	131–3199–00			1	CONN,SHUNT:SHUNT,FEMALE JUMPER	22526	68786–202
4	211–0914–00			2	JACKSCREW:JACKSCREW,4–40 X 0.394,STL	TK0588	211091400
5	671–3877–XX			1	CIRCUIT BD ASSY:CLOCK DAUGHTER CARD	80009	671–3877–XX
6	348–1434–00			4	GASKET,EMI:2.912 L,CLIP ON	30817	97–613–17–029
7	211–0373–00			12	SCREW,MACHINE:4–40 X 0.250,PNH, T–10 TORX DR	93907	ORDER BY DESCRIPTION
8	671–3328–XX			1	CIRCUIT BD ASSY:CPU LOWER ASSY	80009	671–3328–XX
9	159–5014–00			5	FUSE:2.0A,125V,FAST BLOW (REFERENCE DESIGNATOR, F1141, F1352, F1451, F651, F652)	75915	154 002
10	159–5015–00			1	FUSE,SMD:10.0A,125V,FAST BLOW (REFERENCE DESIGNATOR, F75)	75915	R451 010
11	163–0599–00			1	IC,DIGITAL:CMOS (REFERENCE DESIGNATOR, U931)	TK0198	163059900
12	671–3532–XX			1	CIRCUIT BD ASSY:CPU UPPER BOARD	80009	671–3532–XX
13	211–0894–00			5	SCREW,MACHINE:4–40 X 0.625,PNH,T–10,TORX DR	0KB01	211–0894–00
14	200–4231–00			1	COVER:VXI APPLICATION,SAFETY CONTROLLED	TK1943	200–4231–00
15	159–5010–00			1	FUSE,SMD:7A,125V,FAST BLOW (REFERENCE DESIGNATOR, F1351)	75915	451007
16	160–9691–01			1	IC, MEMORY:CMOS,EPROM (REFERENCE DESIGNATOR, U1121)	TK0198	160969101
17	337–4139–00			1	SHIELD,ELEC:DMM,W/MYLAR ATTACHED	TK1943	337–4139–00
18	211–0897–00			6	SCREW,MACHINE:4–40 X 0.188,PNH,T–10,TORX	0KB01	211–0897–00
19	351–0983–00			1	GUIDE:TOP GUIDE	TK1943	351–0983–00
20	671–4011–XX			1	CIRCUIT BD ASSY:DMM PLUS	80009	671–4011–XX
21	351–0984–00			1	GUIDE:BOTTOM GUIDE	TK1943	351–0984–00
22	950–3794–00			2	WASHER WAVY	80009	950–3794–00
23	950–4448–00			2	SCREW M2.5X10 CHEESEHEAD	TK2626	409013905
24	211–0391–00			7	SCR,ASSEM,WSHR:2–56 X 0.437,PNH,STL,T–8 TORX DR	93907	ORDER BY DESCRIPTION
25	441–2110–00			1	CHASSIS:CHASSIS VXI	80009	441–2110–00
26	950–4827–00			2	SCREW:PHIL,M,2.5 X 8,CSK	0KB01	950–4827–00
27	367–0410–00			1	HANDLE,EJECTOR:BOTTOM,SINGLE WIDE MODULE	62559	20817–327
28	334–9091–00			1	MARKER,IDENT:LABEL MARKED MULTIPAQ	80009	334–9091–00
29	348–1365–01			1	SHLD GSKT,ELEC:SYMETRICAL SLOTTED FINGER,0.350 W X 7.5 L	0KM03	0493–0070–00
30	131–0890–01			2	CONN,HARDWARE:DSUB,JACK SCREW,4–40 X 0.312 L HEX HD	00779	205818–2

Replaceable parts list (cont.)

Fig. & index number	Tektronix part number	Serial no. effective	Serial no. discont'd	Qty	Name & description	Mfr. code	Mfr. part number
31	214–4709–00			1	KEY:KEY TTL RIGHT,ALUM	57997	214470900
32	367–0411–00			1	HANDLE,EJECTOR:TOP,SINGLE WIDE	62559	20817–328
33	334–9468–00			1	MARKER,IDENT:LABEL,MKD VX4101A	0KB05	334–9468–00
34	174–3653–00			1	CABLE ASSY:COAX,RFP,50 OHM,6.0L,SMA (REFERENCE DESIGNATOR, J232 TO FRONT PANEL)	060D9	501–1429–05
35	174–3549–00			1	CABLE ASSY:COAX,RFP,50 OHM,6.0L,SMB (REFERENCE DESIGNATOR, J231 TO FRONT PANEL)	060D9	174–3549–00
36	131–0955–00			2	CONN,RF JACK; BNC,;50 OHM,FEMALE	00779	87–3334–017
37	343–0549–00			3	STRAP,TIEDOWN	06383	PLT1M
	The following parts are Optional Accessories						
	VX1630			1	CABLE, ANALOG; 160 PIN CONN W/160 COND CBL		
	VX1630M			1	MOD KIT; VX4101 DMM MOD KIT FOR VX1630 CABLE, 3 TWISTED PAIRS, DE–9P MALE SUBMINIATURE, 24L		
	VX1630S			1	160 PIN CON KIT; BAK SHL, PIN HOUSNG, 200 PINS		
	VX1729			1	CABLE,INTCON; COAX,VX1729;RFD,50 OHM,RG188, 10 FT,BNC,MALE X SMB,STR		
	VX1730			1	CABLE, ADAPTER; SMB SNAP ON PLUG TO PLUG		
	VX1760			1	SMA–BNC MALE; LE COAX CABLE, RFD50–OHM, 1784S RG188, 10FT, BNC MALE–SMA, STR		
	VX1784S			1	CONN HOODED; DE–9 FEMALE SOCKET		
	174–1428–00			1	CABLE ASSY:COAX,RFD,50 OHM,60.0L,SMA,STR,BOTH ENDS,MALE	060D9	174–1428–00
	012–0057–01			1	CA ASSY,RF:COAXIAL,RFD,50 OHM,43 L,BNC,MALE	060D9	012–0057–01
	003–1493–00			1	HAND TOOL:DISASSEMBLY TOOL FOR CRIMP & POKE CONTACTS	6V439	471 555
	003–1494–00			1	HAND TOOL:HAND TOOL FOR LOOSE CRIMP & POKE CONTACTS	6V439	014 374
	The following parts are Standard Accessories						
	VX1784S				CONN HOODED; DE–9 FEMALE SOCKET		
	071–0049–XX			1	MANUAL,TECH:USERS,VX4101A	TK2548	071–0049–XX/
	071–0050–XX			1	MANUAL,TECH:REFERENCE,VX4101A	TK2548	071–0050–XX

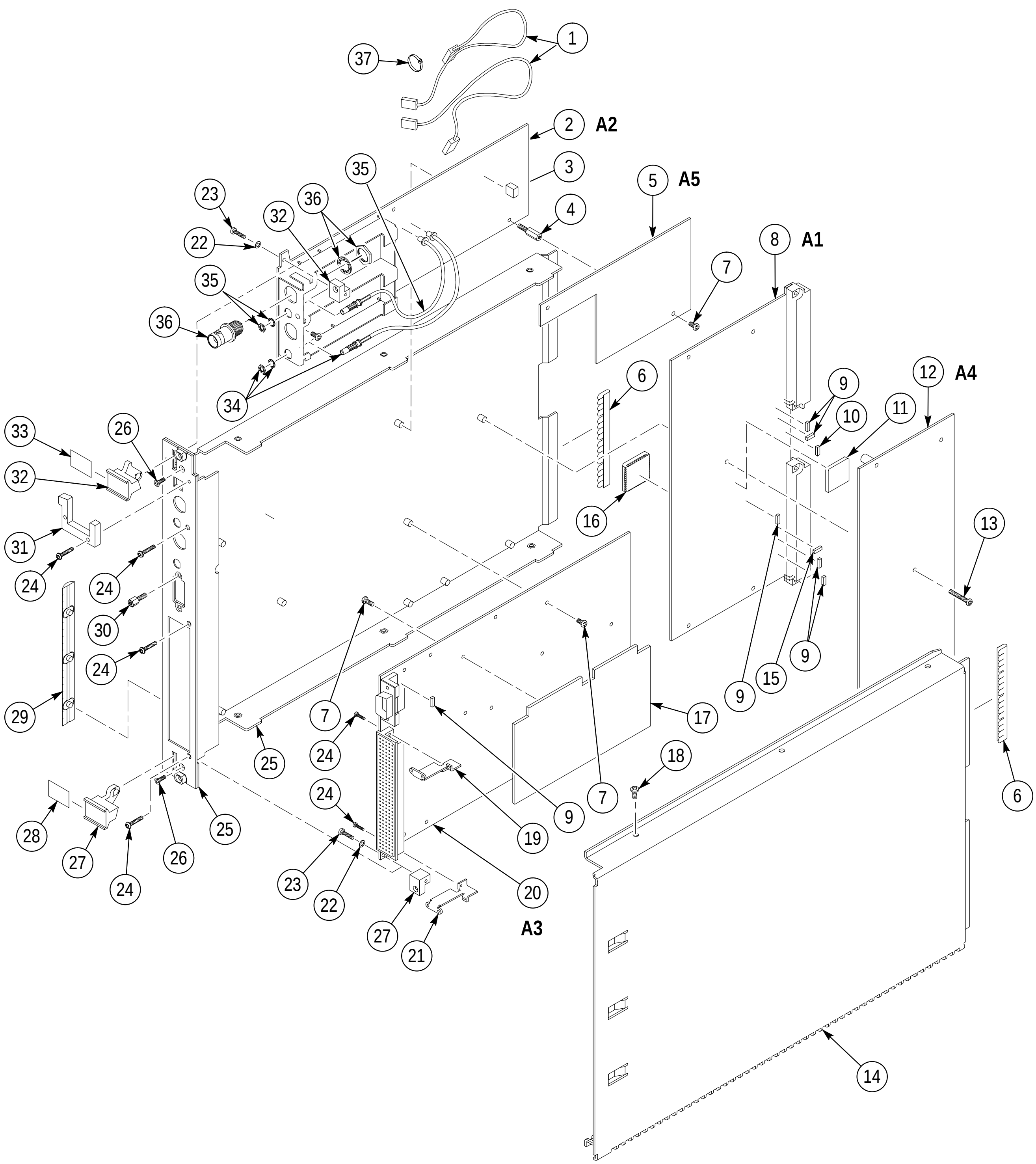


Figure I–1: VX4101A Exploded View

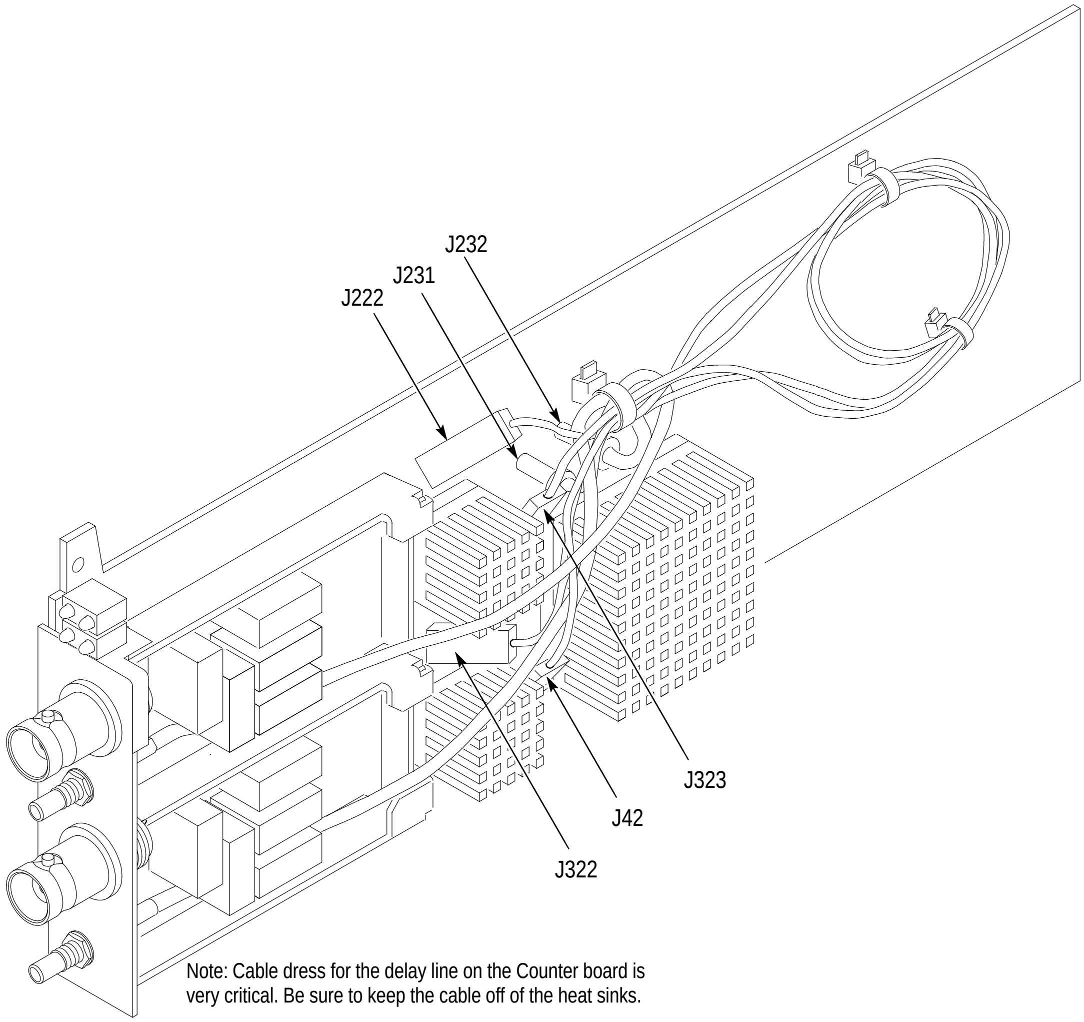


Figure I–2: VX4101A Delay Line Cable Dress

Glossary and Index

Glossary

The terms in this glossary are defined as used in the VXIbus System. Although some of these terms may have different meanings in other systems, it is important to use these definitions in VXIbus applications. Terms which apply only to a particular instrument module are noted. Not all terms appear in every manual.

Accessed Indicator
An amber LED indicator that lights when the module identity is selected by the Resource Manager module, and flashes during any I/O operation for the module.

ACFAIL*
A VMEbus backplane line that is asserted under these conditions: 1) by the mainframe Power Supply when a power failure has occurred (either ac line source or power supply malfunction), or 2) by the front panel ON/STANDBY switch when switched to STANDBY.

A-Size Card
A VXIbus instrument module that is 100.0 × 160 mm × 20.32 mm (3.9 × 6.3 in × 0.8 in), the same size as a VMEbus single-height short module.

Asynchronous Communication
Communications that occur outside the normal “command-response” cycle. Such communications have higher priority than synchronous communication.

Backplane
The printed circuit board that is mounted in a VXIbus mainframe to provide the interface between VXIbus modules and between those modules and the external system.

B-Size Card
A VXIbus instrument module that is 233.4 × 160 mm × 20.32 mm (9.2 × 6.3 in × 0.8 in), the same size as a VMEbus double-height short module.

Bus Arbitration
In the VMEbus interface, a system for resolving contention for service among VMEbus Master devices on the VMEbus.

Bus Timer
A functional module that measures the duration of each data transfer on the Data Transfer Bus (DTB) and terminates the DTB cycle if the duration is excessive. Without the termination capability of this module, a Bus Master attempt to transfer data to or from a non-existent Slave location could result in an infinitely long wait for the Slave response.

Client

In shared memory protocol (SMP), that half of an SMP channel that does not control the shared memory buffers.

CLK10

A 10 MHz, ±100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 and distributed to Slots 1–12 on P2. It is distributed to each module slot as a single source, single destination signal with a matched delay of under 8 ns.

CLK100

A 100 MHz, ±100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 and distributed to Slots 1–12 on P3. It is distributed to each module slot in synchronous with CLK10 as a single source, single destination signal with a maximum system timing skew of 2 ns, and a maximum total delay of 8 ns.

Commander

In the VXIbus interface, a device that controls another device (a servant). A commander may be a servant of another commander.

Command

A directive to a device. There are three types of commands:

In Word Serial Protocol, a 16-bit imperative to a servant from its commander.

In Shared Memory Protocol, a 16-bit imperative from a client to a server, or vice versa.

In a Message, an ASCII-coded, multi-byte directive to any receiving device.

Communication Registers

In word serial protocol, a set of device registers that are accessible to the commander of the device. Such registers are used for inter-device communications, and are required on all VXIbus message-based devices.

Configuration Registers

A set of registers that allow the system to identify a (module) device type, model, manufacturer, address space, and memory requirements. In order to support automatic system and memory configuration, the VXIbus standard specifies that all VXIbus devices have a set of such registers, all accessible from P1 on the VMEbus.

C-Size Card

A VXIbus instrument module that is 340.0 mm × 233.4 mm × 30.48 mm (13.4 in. × 9.2 in × 1.2 in).

Custom Device
A special-purpose VXIbus device that has configuration registers so as to be identified by the system and to allow for definition of future device types to support further levels of compatibility.

Data Transfer Bus
One of four buses on the VMEbus backplane. The Data Transfer Bus allows Bus Masters to direct the transfer of binary data between Masters and Slaves.

DC SUPPLIES Indicator
A red LED indicator that illuminates when a DC power fault is detected on the backplane.

Device Specific Protocol
A protocol for communication with a device that is not defined in the VXIbus specification.

D-Size Card
A VXIbus instrument module that is 340.0 × 366.7 mm × 30.48 mm (13.4 × 14.4 in × 1.2 in).

DTB
See Data Transfer Bus.

DTB Arbiter
A functional module that accepts bus requests from Requester modules and grants control of the DTB to one Requester at a time.

DUT
Device Under Test.

ECLTRG
Six single-ended ECL trigger lines (two on P2 and four on P3) that function as inter-module timing resources, and that are bussed across the VXIbus subsystem backplane. Any module, including the Slot 0 module, may drive and receive information from these lines. These lines have an impedance of 50 Ω; the asserted state is logical High.

Embedded Address
An address in a communications protocol in which the destination of the message is included in the message.

ESTST
Extended STart/STop protocol; used to synchronize VXIbus modules.

Extended Self Test
Any self test or diagnostic power-on routine that executes after the initial kernel self test program.

External System Controller
The host computer or other external controller that exerts overall control over VXIbus operations.

FDC
See *Fast Data Channel*

Fast Data Channel
A communication protocol that uses a block of memory that is accessible to both client and server. The memory block operates as a message buffer for either data or command communication.

FAILED Indicator
A red LED indicator that lights when a device on the VXIbus has detected an internal fault. This might result in the assertion of the SYSFAIL* line.

IACK Daisy Chain Driver
The circuit that drives the VMEbus Interrupt Acknowledge daisy chain line that runs continuously through all installed modules or through jumpers across the backplane.

ID-ROM
An NVRAM storage area that provides for non-volatile storage of diagnostic data.

Instrument Module
A plug-in printed circuit board, with associated components and shields, that may be installed in a VXIbus mainframe. An instrument module may contain more than one device. Also, one device may require more than one instrument module.

Interface Device
A VXIbus device that provides one or more interfaces to external equipment.

Interrupt Handler
A functional module that detects interrupt requests generated by Interrupters and responds to those requests by requesting status and identity information.

Interrupter
A device capable of asserting VMEbus interrupts and performing the interrupt acknowledge sequence.

IRQ
The Interrupt ReQuest signal, which is the VMEbus interrupt line that is asserted by an Interrupter to signify to the controller that a device on the bus requires service by the controller.

Local Bus
A daisy-chained bus that connects adjacent VXIbus slots.

Local Controller
The instrument module that performs system control and external interface functions for the instrument modules in a VXIbus mainframe or several mainframes. See Resource Manager.

Local Processor
The processor on an instrument module.

Logical Address
The smallest functional unit recognized by a VXIbus system. It is often used to identify a particular module.

Mainframe
Card Cage. For example, the Tektronix VX1410 Mainframe, an operable housing that includes 13 C-size VXIbus instrument module slots.

Memory Device
A storage element (such as bubble memory, RAM, and ROM) that has configuration registers and memory attributes (such as type and access time).

Message
A series of data bytes that are treated as a single communication, with a well defined terminator and message body.

Message Based Device
A VXIbus device that supports VXI configuration and communication registers. Such devices support the word serial protocol, and possibly other message-based protocols.

MODID Lines
Module/system identity lines.

Physical Address
The address assigned to a backplane slot during an access.

Power Monitor
A device that monitors backplane power and reports fault conditions.

P1
The top-most backplane connector for a given module slot in a vertical mainframe such as the Tektronix VX1410. The left-most backplane connector for a given slot in a horizontal mainframe.

P2
The bottom backplane connector for a given module slot in a vertical C-size mainframe such as the VX1410; or the middle backplane connector for a given module slot in a vertical D-size mainframe such as the VX1500.

Query
A form of command that allows for inquiry to obtain status or data.

READY Indicator
A green LED indicator that lights when the power-on diagnostic routines have been completed successfully. An internal failure or failure of +5 V power will extinguish this indicator.

Register Based Device
A VXIbus device that supports VXI register maps, but not high level VXIbus communication protocols; includes devices that are register-based servant elements.

Requester
A functional module that resides on the same module as a Master or Interrupt Handler and requests use of the DTB whenever its Master or Interrupt Handler requires it.

Resource Manager
A VXIbus device that provides configuration management services such as address map configuration, determining system hierarchy, allocating shared system resources, performing system self test diagnostics, and initializing system commanders.

Sample
A single DAC output point.

Self Calibration
A routine that verifies the basic calibration of the instrument module circuits, and adjusts this calibration to compensate for short- and long-term variables.

Self Test
A set of routines that determine if the instrument module circuits will perform according to a given set of standards. A self test routine is performed upon power-on.

Segment
A group of Digital to Analog Converter (DAC) points that generate a waveform.

Sequence
A programmed repetition of segments. The maximum number of segments is 4096. At the end of a sequence, the DAC output amplitude remains at the last sequence value until a new sequence is triggered. You use the TRIG:MODE SEQ,<sequence length> command to define the number of segments. You can initiate a new sequence either by a trigger, or continuously.

Servant
A VXIbus message-based device that is controlled by a commander.

Server
A shared memory device that controls the shared memory buffers used in a given Shared Memory Protocol channel.

Shared Memory Protocol
A communications protocol that uses a block of memory that is accessible to both client and server. The memory block operates as a message buffer for communications.

Slot 0 Controller
See Slot 0 Module. Also see Resource Manager.

Slot 0 Module
A VXIbus device that provides the minimum VXIbus slot 0 services to slots 1 through 12 (CLK10 and the module identity lines), but that may provide other services such as CLK100, SYNC100, STARBUS, and trigger control.

SMP
See Shared Memory Protocol.

STARX
Two (2) bi-directional, 50 Ω, differential ECL lines that provide for inter-module asynchronous communication. These pairs of timed and matched delay lines connect slot 0 and each of slots 1 through 12 in a mainframe. The delay between slots is less than 5 ns, and the lines are well matched for timing skew.

STARY
Two (2) bi-directional, 50 Ω, differential ECL lines that provide for inter-module asynchronous communication. These pairs of timed and matched delay lines connect slot 0 and each of slots 1 through 12 in a mainframe. The delay between slots is less than 5 ns, and the lines are well matched for timing skew.

STST
STart/STop protocol; used to synchronize modules.

SYNC100
A Slot 0 signal that is used to synchronize multiple devices with respect to a given rising edge of CLK100. These signals are individually buffered and matched to less than 2 ns of skew.

Synchronous Communications
A communications system that follows the “command-response” cycle model. In this model, a device issues a command to another device; the second device executes the command; then returns a response. Synchronous commands are executed in the order received.

SYSFAIL*
A signal line on the VMEbus that is used to indicate a failure by a device. The device that fails asserts this line.

System Clock Driver
A functional module that provides a 16 MHz timing signal on the Utility Bus.

System Hierarchy
The tree structure of the commander/servant relationships of all devices in the system at a given time. In the VXIbus structure, each servant has a commander. A commander may also have a commander.

Test Monitor
An executive routine that is responsible for executing the self tests, storing any errors in the ID-ROM, and reporting such errors to the Resource Manager.

Test Program
A program, executed on the system controller, that controls the execution of tests within the test system.

Test System
A collection of hardware and software modules that operate in concert to test a target DUT.

TTLTRG
Open collector TTL lines used for inter-module timing and communication.

VXIbus Subsystem
One mainframe with modules installed. The installed modules include one module that performs slot 0 functions and a given complement of instrument modules. The subsystem may also include a Resource Manager.

Waveform Period
The waveform period is defined as the number of points in the waveform times the sample period.

Word Serial Protocol
A VXIbus word oriented, bi-directional, serial protocol for communications between message-based devices (that is, devices that include communication registers in addition to configuration registers).

Word Serial Communications
Inter-device communications using the Word Serial Protocol.

WSP
See Word Serial Protocol.

10-MHz Clock
A 10 MHz, ±100 ppm timing reference. Also see CLK10.

100-MHz Clock

A 100 MHz, ±100 ppm clock synchronized with CLK10. Also see CLK100.

488-To-VXIbus Interface

A message based device that provides for communication between the IEEE-488 bus and VXIbus instrument modules.

Index

A

C

D

M

N

O

P

R

S

T

U

V

W