VXI Technology™

VM7004

PROGRAMMABLE RESISTOR MODULE

USER’S MANUAL

82-0029-000
Rev. April 7, 2003

VXI Technology, Inc.

2031 Main Street
Irvine, CA 92614-6509
(949) 955-1894

TABLE OF CONTENTS

CERTIFICATION

VXI Technology, Inc. (VTI) certifies that this product met its published specifications at the time of shipment from the factory. VTI further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

The product referred to herein is warranted against defects in material and workmanship for a period of three years from the receipt date of the product at customer's facility. The sole and exclusive remedy for breach of any warranty concerning these goods shall be repair or replacement of defective parts, or a refund of the purchase price, to be determined at the option of VTI.

For warranty service or repair, this product must be returned to a VXI Technology authorized service center. The product shall be shipped prepaid to VTI and VTI shall prepay all returns of the product to the buyer. However, the buyer shall pay all shipping charges, duties, and taxes for products returned to VTI from another country.

VTI warrants that its software and firmware designated by VTI for use with a product will execute its programming when properly installed on that product. VTI does not however warrant that the operation of the product, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The warranty shall not apply to defects resulting from improper or inadequate maintenance by the buyer, buyer-supplied products or interfacing, unauthorized modification or misuse, operation outside the environmental specifications for the product, or improper site preparation or maintenance.

VXI Technology, Inc. shall not be liable for injury to property other than the goods themselves. Other than the limited warranty stated above, VXI Technology, Inc. makes no other warranties, express or implied, with respect to the quality of product beyond the description of the goods on the face of the contract. VTI specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

VXI Technology, Inc.
2031 Main Street
Irvine, CA 92614-6509 U.S.A.

DECLARATION OF CONFORMITY

Declaration of Conformity According to ISO/IEC Guide 22 and EN 45014

MANUFACTURER’S NAME	VXI Technology, Inc.
MANUFACTURER’S ADDRESS	2031 Main Street Irvine, California 92614-6509
PRODUCT NAME	Programmable Resistor Module
MODEL NUMBER(S)	VM7004
PRODUCT OPTIONS	All
PRODUCT CONFIGURATIONS	All

VXI Technology, Inc. declares that the aforementioned product conforms to the requirements of the Low Voltage Directive 73/23/EEC *and the* EMC Directive 89/366/EEC *(inclusive* 93/68/EEC*) and carries the “CE” mark accordingly. The product has been designed and manufactured according to the following specifications:*

SAFETY	EN61010 (2001)
EMC	EN61326 (1997 w/A1:98) Class A CISPR 22 (1997) Class A VCCI (April 2000) Class A ICES-003 Class A (ANSI C63.4 1992) AS/NZS 3548 (w/A1 & A2:97) Class A FCC Part 15 Subpart B Class A EN 61010-1:2001

The product was installed into a C-size VXI mainframe chassis and tested in a typical configuration.

I hereby declare that the aforementioned product has been designed to be in compliance with the relevant sections of the specifications listed above as well as complying with all essential requirements of the Low Voltage Directive.

April 2003

Jerry Patton

Jerry Patton, QA Manager

GENERAL SAFETY INSTRUCTIONS

Review the following safety precautions to avoid bodily injury and/or damage to the product. These precautions must be observed during all phases of operation or service of this product. Failure to comply with these precautions, or with specific warnings elsewhere in this manual, violates safety standards of design, manufacture, and intended use of the product.

Service should only be performed by qualified personnel.

TERMS AND SYMBOLS

These terms may appear in this manual:

WARNING Indicates that a procedure or condition may cause bodily injury or death.

CAUTION Indicates that a procedure or condition could possibly cause damage to equipment or loss of data.

These symbols may appear on the product:

⚠	ATTENTION - Important safety instructions
⏚	Frame or chassis ground

WARNINGS

Follow these precautions to avoid injury or damage to the product:

Use Proper Power Cord To avoid hazard, only use the power cord specified for this product.

Use Proper Power Source To avoid electrical overload, electric shock, or fire hazard, do not use a power source that applies other than the specified voltage.

Use Proper Fuse To avoid fire hazard, only use the type and rating fuse specified for this product.

WARNINGS (CONT.)

Avoid Electric Shock

To avoid electric shock or fire hazard, do not operate this product with the covers removed. Do not connect or disconnect any cable, probes, test leads, etc. while they are connected to a voltage source. Remove all power and unplug unit before performing any service. ***Service should only be performed by qualified personnel.***

Ground the Product

This product is grounded through the grounding conductor of the power cord. To avoid electric shock, the grounding conductor must be connected to earth ground.

Operating Conditions

To avoid injury, electric shock or fire hazard:

- Do not operate in wet or damp conditions.
- Do not operate in an explosive atmosphere.
- Operate or store only in specified temperature range.
- Provide proper clearance for product ventilation to prevent overheating.
- DO NOT operate if any damage to this product is suspected. ***Product should be inspected or serviced only by qualified personnel.***

Improper Use

The operator of this instrument is advised that if the equipment is used in a manner not specified in this manual, the protection provided by the equipment may be impaired. Conformity is checked by inspection.

SUPPORT RESOURCES

Support resources for this product are available on the Internet and at VXI Technology customer support centers.

Internet Support

E-mail: support@vxitech.com
Web Address: http://www.vxitech.com

Telephone Support (U.S.)

Tel: (949) 955-1894 **West Coast**
(216) 447-8950 **East Coast**

Fax: (949) 955-3041 **West Coast**
(216) 447-8951 **East Coast**

VXI Technology Headquarters

Technical Support
VXI Technology, Inc.
2031 Main Street
Irvine, CA 92614-6509

Tel: (949) 955-1894
Fax: (949) 955-3041

SECTION 1

INTRODUCTION

INTRODUCTION

The VM7004 module provides four independent channels of programmable resistors, each with four decades per channel, and is designed for applications such as RTD or other sensor simulation, process control, and ATE calibration. The instrument uses the message-based word serial interface for programming and data movement, as well as supporting direct register access for very high-speed resistance setting. The command set conforms with the SCPI standard for consistency and ease of programming.

The VM7004 is a member of the VXI Technology VMIP™ (*VXI Modular Instrumentation Platform*) family and is available as a 4-, 8- or 12-channel, single-wide VXIbus instrument. In addition to these three standard configurations, the VM7004 may be combined with any of the other members of the VMIP family to form a customized and highly integrated instrument (see Figure 1-1). This allows the user to reduce system size and cost by combining the VM7004 with two other instrument functions in a single-wide C-size VXIbus module Figure 1-2 shows the 12-channel version of the VM7004. The 8-channel version would not have J200 and its associated LEDs and nomenclature while the 4-channel version would also eliminate J202.

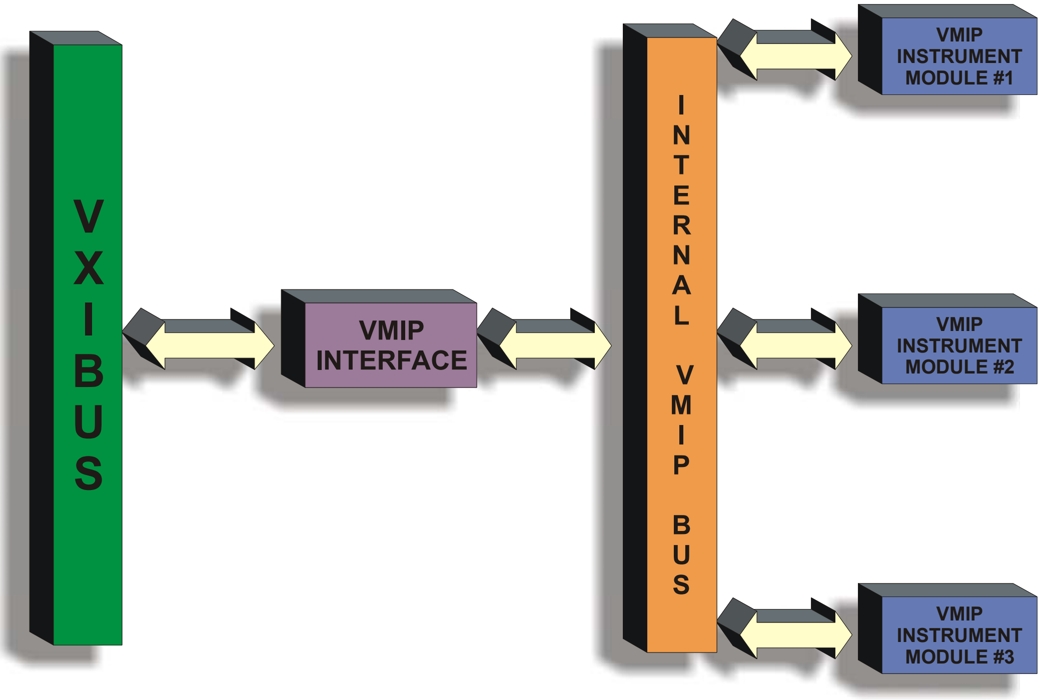

VXIBUS
VMIP INTERFACE
INTERNAL VMIP BUS
VMIP INSTRUMENT MODULE #1
VMIP INSTRUMENT MODULE #2
VMIP INSTRUMENT MODULE #3

FIGURE 1-1 VMIP™ PLATFORM

Regardless of whether the VM7004 is configured with other VM7004 modules or with other VMIP modules, each group of 4 channels is treated as an independent instrument in the VXIbus chassis and, as such, each group has its own FAIL and ACCESS light.

DESCRIPTION

The VM7004 is a high-density programmable resistor module with 4 channels per VMIP daughter module. Each channel is programmable from 0 to 16,383 Ω and has its own sense leads for feedback. Monitor points are also provided per group of 4 channels for ease of calibration.

The VM7004 can be software programmed to operate in four different modes to ease integration by the user:

Individual Channels Each channel is programmed individually for a given resistance between 0 to 16,383 Ω.

Potentiometer Two or four channels can be tied together and programmed to operate as a potentiometer.

Parallel Mode Two or more channels can be connected in parallel, increasing the accuracy and reducing the step size. With two channels in parallel the VM7004 can be programmed between 0 to 8192 Ω and for four channels in parallel, programmed between 0 and 4,096 Ω.

Series Mode Two or more channels can be connected in series, increasing the range. With two channels in series the VM7004 can be programmed between 0 to 32,766 Ω and for four channels in series, programmed between 0 and 65,533 Ω.

Figure 1-3 shows a block diagram of the VM7004.

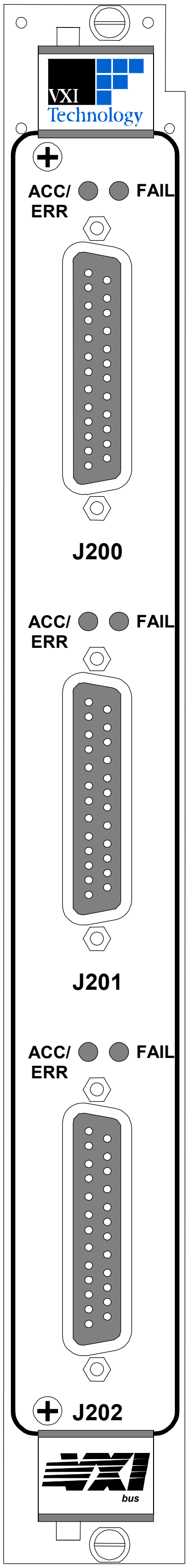


FIGURE 1-2 FRONT PANEL LAYOUT

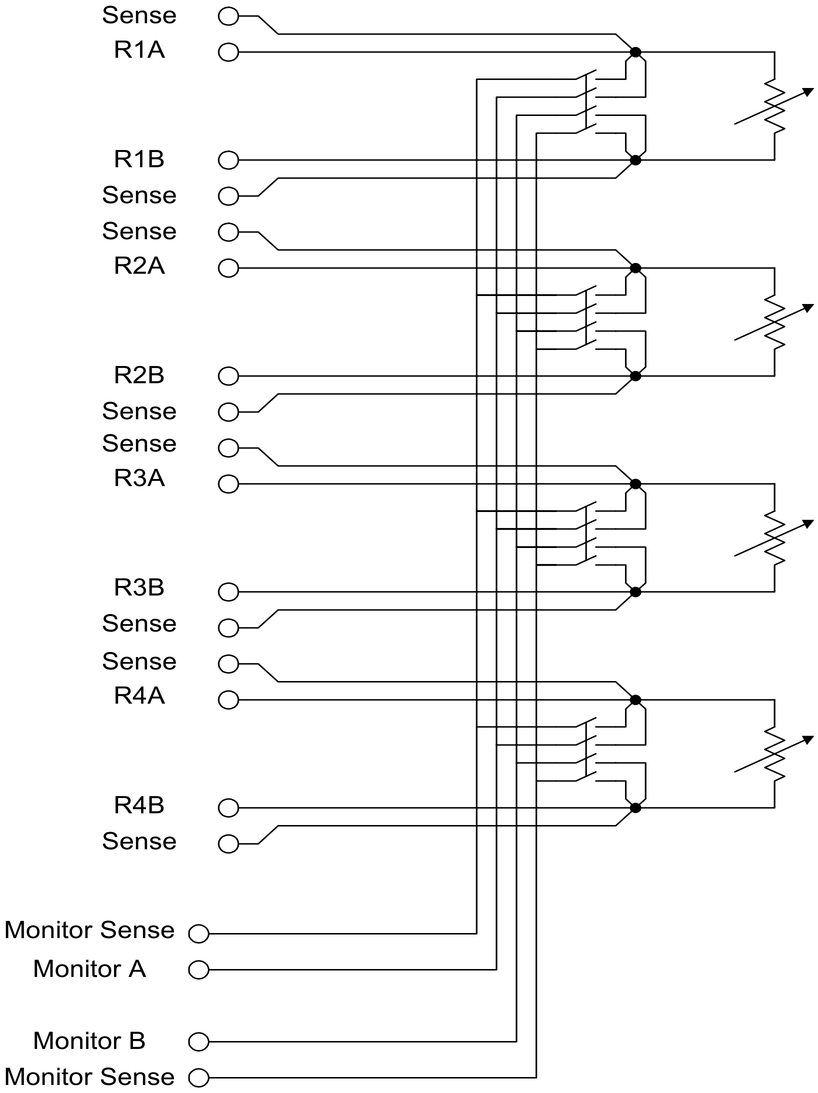


FIGURE 1-3 VM7004 BLOCK DIAGRAM

VM7004 Specifications

General Specifications	
Number of Channels	
VM7004-1	4 channels
VM7004-2	8 channels
VM7004-3	12 channels
Resistance	
	0 Ω to 16,383 Ω
Resolution	
	1 Ω steps
Accuracy	
Standard	±0.02% of programmed value ±0.5 Ω
Option 5	±0.02% of programmed value ±0.5 Ω
Option 6	±0.02% of programmed value ±0.5 Ω for values less than 200 k for Option 6 ±0.05% of programmed value ±0.5 Ω for values greater than or equal to 200 k
Option 7	±0.02% of programmed value ±0.5 Ω for values less than 200 k for Option 7 ±0.08% of programmed value ±0.5 Ω for values greater than or equal to 200 k
Maximum Power	
	0.5 W to 40°C Thermal
Thermal Offset	
	$\leq \pm 25\ \mu$V
VXI Interface	
	Message-based word serial interface Direct register access in the user defined area of the VXIbus register map
VXI Revision	
	Revision 1.3 and 1.4
Logical Addressing	
	Static or Dynamic Configuration
Power Requirements	
VM7004-1	+5 V @ 3.04 A
VM7004-2	+5 V @ 5.34 A
VM7004-3	+5 V @ 7.64 A
Manufacturer's ID	
	3915
Module Model Code	
	270

OPTIONS

Note that Options 5, 6 and 7 change the resistor value outputs x10, x100 and x1000 respectively. All information, software commands and software output will read as outlined in this manual; however, actual measured resistor values will reflect the VM7004 option installed.

Specifications

STANDARD	
Range	1 Ω to 16,388 Ω
Step Size	1 Ω
OPTION 5	
Range	10 Ω to 163,830 Ω
Step Size	10 Ω
OPTION 6	
Range	100 Ω to 1,638,300 Ω
Step Size	100 Ω
OPTION 7	
Range	1000 Ω to 16,383,000 Ω
Step Size	1000 Ω

Examples

STANDARD	
Program Input	10 Ω
Query Response	10 Ω
Measured Output	10 Ω
Program Input	4,096 Ω
Query Response	4,096 Ω
Measured Output	4,096 Ω
OPTION 5	
Program Input	10 Ω
Query Response	10 Ω
Measured Output	100 Ω
Program Input	4,096 Ω
Query Response	4,096 Ω
Measured Output	40,960 Ω
OPTION 6	
Program Input	10 Ω
Query Response	10 Ω
Measured Output	1000 Ω
Program Input	4096 Ω
Query Response	4096 Ω
Measured Output	409,600 Ω
OPTION 7	
Program Input	10 Ω
Query Response	10 Ω
Measured Output	10,000 Ω
Program Input	4096 Ω
Query Response	4096 Ω
Measured Output	4,096,000 Ω

Section 2

Preparation for Use

Installation

When the VM7004 is unpacked from its shipping carton, the contents should include the following items:

(1) VM7004 VXIbus module
(1) VM7004 Programmable Resistor Module User’s Manual (this manual)

All components should be immediately inspected for damage upon receipt of the unit.

Once the VM7004 is assessed to be in good condition, it may be installed into an appropriate C-size or D-size VXIbus chassis in any slot other than slot 0. The chassis should be checked to ensure that it is capable of providing adequate power and cooling for the VM7004. Once the chassis is found be adequate, the VM7004’s logical address and the chassis’ backplane jumpers should be configured prior to the VM7004’s installation.

Calculating System Power and Cooling Requirements

It is imperative that the chassis provide adequate power and cooling for this module. Referring to the chassis user’s manual, confirm that the power budget for the system (the chassis and all modules installed therein) is not exceeded and that the cooling system can provide adequate airflow at the specified backpressure.

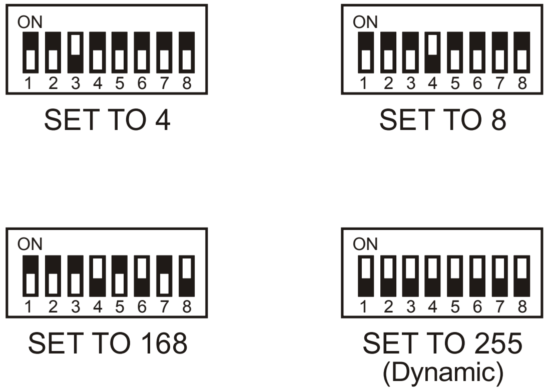
⚠ It should be noted that if the chassis cannot provide adequate power to the module, the instrument may not perform to specification or possibly not operate at all. In addition, if adequate cooling is not provided, the reliability of the instrument will be jeopardized and permanent damage may occur. Damage found to have occurred due to inadequate cooling would also void the warranty of the module.

SETTING THE CHASSIS BACKPLANE JUMPERS

Please refer to the chassis User's Manual for further details on setting the backplane jumpers.

SETTING THE LOGICAL ADDRESS

The logical address of the VM7004 is set by a single 8-position DIP switch located near the module's backplane connectors (this is the only switch on the module). The switch is labeled with positions 1 through 8 and with an ON position. A switch pushed toward the ON legend will signify a logic 1; switches pushed away from the ON legend will signify a logic 0. The switch located at position 1 is the least significant bit while the switch located at position 8 is the most significant bit. See Figure 2-1 for examples of setting the logical address switch.



SET TO 4

SET TO 8

SET TO 168

SET TO 255
(Dynamic)

Switch Position	Switch Value
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128

FIGURE 2-1 LOGICAL ADDRESS SWITCH SETTING EXAMPLES

The VMIP may contain three separate instruments and will allocate logical addresses as required by the VXIbus specification (revisions 1.3 and 1.4). The logical address of the instrument is set on the VMIP carrier. The VMIP logical addresses must be set to an even multiple of 4 *unless dynamic addressing is used*. Switch positions 1 and 2 must always be set to the OFF position. Therefore, only addresses of 4, 8, 12, 16, ... 252 are allowed. The address switch should be set for one of these legal addresses and the address for the second instrument (the instrument in the center position) will automatically be set to the switch set address plus one; while the third instrument (the instrument in the lowest position) will automatically be set to the switch set address plus two. If dynamic address configuration is desired, the address switch should be set for a value of 255 (All switches set to ON). Upon power-up, the slot 0 resource manager will assign the first available logical addresses to each instrument in the VMIP module.

If dynamic address configuration is desired, the address switch should be set for a value of 255. (All switches set to ON). Upon power-up, the slot 0 resource manager will assign the first available logical addresses to each instrument in the VMIP module.

FRONT PANEL INTERFACE WIRING

The VM7004's interface is made available on the front panel of the instrument. The 4- channel version (VM7004-1) will have J201 which contains all signals for this instrument. The 8-channel version (VM7004-2) will have J201 and J202 provided, while the 12-channel version (VM7004-3) will have J200, J201 and J202. The wiring for each of these connectors is identical and since each group of 4 channels is treated as a separate instrument, the module will have three Channel 1s, three Channel 2s, three Channel 3s, etc.

The connector used in the VM7004 is a commonly available 25-pin DSUB receptacle connector. The pin locations for J200, J201 and J202 are shown in Figure 2-2

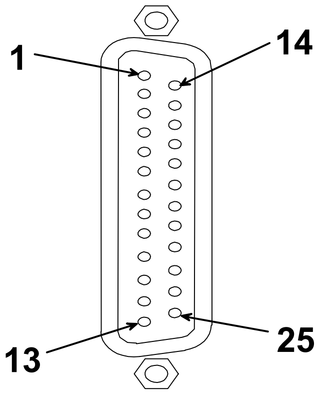


FIGURE 2-2 J200, J201 AND J202 PIN LOCATIONS

TABLE 2-1 PROGRAMMABLE RESISTOR PIN OUTS

SIGNAL	PIN NUMBER
CHANNEL 1A	1
CHANNEL 1B	2
	3
CHANNEL 2A	4
CHANNEL 2B	5
CHANNEL 3A	6
CHANNEL 3B	7
	8
CHANNEL 4A	9
CHANNEL 4B	10
MONITOR BUS A	11
MONITOR BUS B	12
TRIGIN	13
SENSE 1A	14
SENSE 1B	15
SENSE 2A	16
SENSE 2B	17
	18
SENSE 3A	19
SENSE 3B	20
SENSE 4A	21
SENSE 4B	22
SENSE BUS A	23
SENSE BUS B	24
GROUND	25

SECTION 3

PROGRAMMING

EXAMPLES OF SCPI COMMANDS

ABORT

This command disarms the VM7004 and stops latching data to select resistors (if active)

ABORt	***This command has no parameters.***

EXAMPLES

`ABOR`	*Disarms the VM7004 and stops latching data to select resistors (if active).*

ARM SEQUENCE

This command prepares the scan list data and enables the triggering subsystem for the scan list mode. It arms the module to be ready to latch the data to select resistors in the specified channel.

ARM[<p_chl>][:SEQuence][<offset>,<length>]	***Where <p_chl> is the physical channel to be armed*** ***Where <offset> is in the range 1 - 256 which the beginning location in the scan list data.*** ***Where <length> is in the range 1 - 256 which is the number of data words used in the scan list.***

EXAMPLES

`ARM1:SEQ 3,5`	*Arms Channel 1. Starts at location 3 and length 5 of the scan list.*
`SOUR1:LIST:RES 10,20,30,40,50,60,70,80`	*Sets the Channel 1's scan list to 10, 20, 30, 40, 50, 60, 70, 80*
`TRIG:SOUR:IMM`	*Sets Trigger source to Immediate*
`ARM1:SEQ 5,3`	*Arms Channel 1. Starts at location 5 and length 3 of the scan list*
`TRIG`	*Generates a trigger. It causes Channel 1 to go to a resistance of 50 Ω*
`TRIG`	*Generates a trigger. It causes Channel 1 to go to a resistance of 60 Ω*
`TRIG`	*Generates a trigger. It causes Channel 1 to go to a resistance of 70 Ω*
`TRIG`	*Generates a trigger. It causes Channel 1 to go to a resistance of 50 Ω*

ARM DELAY

This command sets the delay time for relay output to settle after a trigger occurs. It sets the time duration between the trigger occurrence and the arming of the next trigger.

ARM[:SEQuence][:LAYer]:DELay <delay>	***Where <delay> is a value in the range 0 to 0.65535 seconds which specifies the delay time for the relay outputs to settle after a trigger occurs.***

EXAMPLES

`ARM:DEL 0.005`	*Sets the delay to 5 ms.*

`ARM:DEL?`
`0.005`
	Returns the programmed delay as 5 ms.

OUTPUT TRIGGER SOURCE

This command configures the source of the output trigger.

OUTPut:TRIGger:SOURce <source>

Where <source> specifies the source of the output trigger. The source could be one of CH1 | CH2 | CH3 | CH4 | INTernal| EXTernal | IMMediate | TIMer

EXAMPLES

`OUTP:TRIG:SOUR EXT`	*Configures the front panel connector as the source of the output trigger*
`OUTP:TRIG:SOUR CH1`	*Configures Channel 1 as the source of the output trigger*
`OUTP:TRIG:SOUR?` `CH1`	*Returns Channel 1 as the source of the output trigger*

OUTPUT TRIGGER SLOPE

This command configures the active edge of the trigger driven onto the TTL trigger bus.

OUTPut:TRIGger:SLOPe <slope>	***Where <slope> is the active edge of the trigger driven onto the TTL trigger bus. The slope could be POSitive or NEGative***

EXAMPLES

`OUTP:TRIG:SLOP POS`	*Configures the positive edge as the active edge of the trigger driven onto the TTL trigger bus.*
`OUTP:TRIG:SLOP NEG`	*Configures the negative edge as the active edge of the trigger driven onto the TTL trigger bus.*
`OUTP:TRIG:SLOP?` `NEG`	*Returns the negative edge as the active edge of the trigger driven onto the TTL trigger bus.*

OUTPUT TRIGGER TTLTRIG

This command configures which VXIbus TTL trigger line the module will drive when the output is enabled.

OUTPut:TRIGger:TTLTrig <line>

Where <line> indicates which TTL trigger line is enabled as an output.

EXAMPLES

```
OUTP:TRIG:TTLT3
```

Configures the TTLT line 3 as the line that the module will drive when the output is enabled

```
OUTP:TRIG:TTLT?
3
```

Returns TTLT line 3 as the line that the module will drive when the output is enabled.

OUTPUT TTLTRIG STATE

This command enables or disables the trigger signal driven onto the VXIbus backplane TTL trigger line.

OUTPut:TTLTrig[:STATe] <boolean> ***Where <boolean> can be ON | OFF | 1 | 0.***

EXAMPLES

`OUTP:TTLT:STAT ON`	*Enables the driving of the trigger signal on the VXIbus trigger bus.*
`OUTP:TTLT OFF`	*Disables the driving of the trigger signal onto the VXIbus trigger bus.*

`OUTP:TTLT?`
`0`
	Returns that the driving of the trigger signal onto the VXIbus trigger bus has been disabled.

ROUTE CLOSE

This command connects one or more channel's output to the sense bus and the common bus.

ROUTe:CLOSe <channel_list>

Where <channel_list> specifies the channels whose outputs are to be connected to the sense bus and the common bus. For more details on the syntax of the channel list, see the beginning of Section 4.

EXAMPLES

```
ROUT:CLOS (@1:8)
```

Connects the outputs of Channels 1 through 8 to the sense bus and the common bus.

```
ROUT:CLOS? 1
1
```

Returns that the output of Channel 1 is connected to the sense bus and the common bus.

```
ROUT:OPEN (@2:8)
```

Disconnecting the outputs of Channels 2 through 8 from the sense bus and the common bus.

```
ROUT:CLOS? 2
0
```

Returns that the output of Channel 2 is disconnected from the sense bus and the common bus

ROUTE OPEN

This command disconnects one or more channel's output from the sense bus and the common bus.

ROUTe:OPEN <channel_list>

Where <channel_list> specifies the channels whose outputs are to be disconnected the sense bus and the common bus. For more details on the syntax of the channel list, see the beginning of Section 4.

EXAMPLES

```
ROUT:OPEN (@1:8)
```

Disconnects the outputs of Channels 1 through 8 from the sense bus and the common bus.

```
ROUT:OPEN? 2
1
```

Returns that the output of Channel 2 is disconnected from the sense bus and the common bus.

```
ROUT:CLOS (@1:4)
```

Connects the outputs of Channels 1 through 4 to the sense bus and the common bus.

```
ROUT:OPEN? 1
0
```

Returns that the output of Channel 1 is disconnected from the sense bus and the common bus.

ROUTE PROTECT

This command enables/disables the sense relay's protection circuit.

ROUTe:PROTect <booelan>	***Where <boolean> can be ON | OFF | 1 | 0***

EXAMPLES

`ROUT:PROT ON`	*Enables the sense relay's protection circuit.*
`ROUT:PROT OFF`	*Disables the sense relay's protection circuit.*
`ROUT:PROT?` 0	*Returns that the sense relay's protection circuit has been disabled.*

SOURCE ATTENUATION

This command sets the wiper position (ratio) of the logical channel in the voltage divider mode.

SOURce[<l_chl>]:ATTenuation <value>

Where <l_chl> is the logical channel operating the voltage divider mode whose wiper position is to be configured.

Where <value> is a value in the range 0.0 to 1.0 which specifies the wiper position to be configured.

EXAMPLES

```
SOUR3:ATT 0.75
```

Configures the wiper position for logical Channel 3 to 0.75 i.e. R3 = ¼(R3 + R4)

```
SOUR3:ATT?
0.75
```

Returns the wiper position of logical Channel 3 as 0.75

SOURCE COMBINE

This command sets the resistor mode for the VM7004. It defines how the resistor channels are combined externally.

SOURce:COMBine NONE
SOURce:COMBine <type> [<channel_list>]

Where <type> specifies the combination type for the specified channels. It can be one of NONE, DIV, PAR or SER.

Where <channel_list> specifies the channels that are to be configured. For more details on the syntax of the channel list, see the beginning of Section 4.

EXAMPLES

SOUR:COMB NONE	*Sets all the 4 channels to individual mode.*
SOUR:COMB DIV (@1,2)	*Sets Channels 1 and 2 to DIVider mode.*
SOUR:COMB PAR (@3,4)	*Sets Channels 3 and 4 to Parallel Mode*
SOUR:COMB? 1 DIV	*Returns that Channel 1 is operating in DIVider Mode*
SOUR:COMB? 3	*Returns that Channel 3 is operating in Parallel Mode*

SOURCE LIST RESISTANCE

This command loads the scan list for a resistor channel.

SOURce<p_chl>:LIST:RESistance [<value>] {,<value}	***Where <p_chl> specifies the channel for which the scan list is to be defined.*** ***Where <number> specifies the resistance value that is to be loaded into the scan list of the specified channel***

EXAMPLES

SOUR1:LIST:RES 10,20,30,40,50	*Loads the values 10 Ω, 20 Ω, 30 Ω, 40 Ω and 50 Ω into the scan list for Channel 1*
SOUR1:LIST:RES? 1 10	*Returns the scan list value at index 1 to be 10 Ω.*
SOUR1:LIST:RES? 5 50	*Returns the scan list value at index 5 to be 50 Ω*

SOURCE LIST RESISTANCE POINTS?

This query command returns the number of points that have been defined for the scan list of the specified channel.

SOURce<p_chl>:LIST:RESistance:POINts?	***Where <p_chl> specifies the channel whose scan list size is to be queried***

EXAMPLES

`SOUR1:LIST:RES:POIN?`
`1`
	Returns the scan list size of Channel 1 as 1 element

`SOUR1:LIST:RES 10,20,30,40,50`	*Loads the values 10 Ω, 20 Ω, 30 Ω, 40 Ω and 50 Ω into the scan list for Channel 1*

`SOUR1:LIST:RES:POIN?`
`5`
	Returns the scan list size of Channel 1 as 5 elements.

SOURCE RESISTANCE

This command sets the resistance of the specified logical channel immediately upon execution of the command.

SOURce[<l_chl>]:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>

Where <l_chl> is the logical channel whose resistance value is to be set

Where <value> specifies the resistance value that is to be set for the specified logical channel.

EXAMPLES

`SOUR1:RES 275`	*Sets Channel 1's resistance to 275 Ω immediately upon receipt of the command.*
`SOUR2:RES 550`	*Sets Channel 2's resistance to 550 Ω immediately upon receipt of the command*

`SOUR2:RES?`
`550.000000`
	Returns the resistance of Channel 2 as 550 Ω

SOURCE RESISTANCE TRIGGER

This command sets the resistance of the specified logical channel but waits for a trigger event to occur before transferring the new value to the output.

SOURce<l_chl>:RESistance[:LEVel]:TRIGger[:AMPLitude] <value>

Where <l_chl> is the logical channel whose resistance value is to be set where <value> specifies the resistance value to be set for the specified logical channel after a trigger event occurs

EXAMPLES

`TRIG:SOUR IMM`	*Sets the trigger source as Immediate*
`SOUR1:RES:TRIG 550`	*Configured the resistance of Channel 1 to be set to 550 Ω after receipt of a trigger event*
`TRIG`	*Generates a trigger event and sets the resistance of Channel 1 to 550 Ω*

`SOUR1:RES:TRIG?`
`550.000000`
	Returns the resistance of Channel 1 as 550 Ω

TRIGGER

This command generates a trigger event which causes the armed channels to update their resistance values.

TRIGger[:SEQuence][:IMMediate] ***There are no parameters for this command***

EXAMPLES

`TRIG`	*Causes a trigger event to occur*
`TRIG:SEQ`	*Causes a trigger event to occur*
`TRIG:IMM`	*Causes a trigger event to occur*
`TRIG:SEQ:IMM`	*Causes a trigger event to occur*
`TRIG:SOUR IMM`	*Sets the trigger source as Immediate*
`SOUR1:RES:TRIG 550`	*Configured the resistance of Channel 1 to be set to 550 Ω after receipt of a trigger event*
`TRIG`	*Generates a trigger event and sets the resistance of Channel 1 to 550 Ω*

`SOUR1:RES:TRIG?`
`550.000000`
	Returns the resistance of Channel 1 as 550 Ω

TRIGGER SLOPE

This command sets the active edge for triggering the VM7004.

TRIGger[:SEQuence]:SLOPe <slope>	***Where <slope> determines the active edge for triggering and can be POSitive or NEGative***

EXAMPLES

`TRIG:SLOP POS`	*Configures the positive edge as the active edge of the triggering signal*
`TRIG:SLOP NEG`	*Configures the negative edge as the active edge of the triggering signal*
`TRIG:SLOP?` `NEG`	*Returns the negative edge as the active edge of the triggering signal*

TRIGGER SOURCE

This command selects the source for triggering the VM7004.

TRIGger[:SEQuence]:SOURce <source>	***Where <source> specifies the triggering signal. This can be one of EXTernal, TTLTrig 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 or IMMediate***

EXAMPLES

`TRIG:SOUR EXT`	*Configures the front panel trigger input as the active triggering signal*
`TRIG:SOUR TTLT3`	*Configures TTLT line 3 as the active triggering signal.*
`TRIG:SOUR?` `TTLT3`	*Returns TTLT line 3 as the active triggering signal*

APPLICATION EXAMPLES

The following example shows how the SCPI commands can be used to setup the resistor channels parameters and to set their resistances.

EXAMPLE 1

`*RST`	*Resetting the module to its default state*
`SOUR:COMB DIV (@1,2)`	*Configuring Channels 1 and 2 to operate in the DIVIDER mode*
`SOUR:COMB PAR (@3,4)`	*Configuring Channels 3 and 4 to operate in the PARALLEL mode*
`ROUT:PROT OFF`	*Setting the sense relay protection off*
`ROUT:CLOS (@1:4)`	*Connecting the outputs of Channels 1 through 4 to the sense bus and the common bus*
`SOUR1:ATT 0.9`	*Setting the Wiper Position of Logical Channel 1 to 0.9*
`SOUR3:RES 2000`	*Setting the resistance of Logical Channel 3 to 2000 Ω*

EXAMPLE 2

The following example shows how the SCPI commands can be used to setup the scan lists for the resistor channels, configure the trigger parameters, to arm the module and to scan thru' the scan list.

`*RST`	*Resetting the module to its default state*
`TRIG:SOUR IMM`	*Configuring the software trigger as the trigger source*
`SOUR1:LIST:RES 10,20,30,40,50,60`	*Setting up the scan list for Channel 1*
`SOUR2:LIST:RES 100,200,300,400,500,600`	*Setting up the scan list for Channel 2*
`ROUT:CLOS (@1,2)`	*Connecting the outputs of Channels 1 and 2 to the sense bus and the common bus*
`ARM1:SEQ 1,5`	*Arming Channel 1 with the active scan list staring at location 1 and length 5*
`ARM2:SEQ 1,2`	*Arming Channel 2 with the active scan list staring at location 1 and length 2*
`TRIG`	*Generating a software trigger. This sets Channel 1's resistance to 10 Ω and Channel 2's resistance to 100 Ω*
`TRIG`	*Generating a software trigger. This sets Channel 1's resistance to 20 Ω and Channel 2's resistance to 200 Ω*
`TRIG`	*Generating a software trigger. This sets Channel 1's resistance to 30 Ω and Channel 2's resistance to 100 Ω*
`TRIG`	*Generating a software trigger. This sets Channel 1's resistance to 40 Ω and Channel 2's resistance to 200 Ω*
`ABOR`	*Disarms the VM7004 and aborts data latching*

REGISTER ACCESS EXAMPLES

REGISTER PROGRAMMING

This instrument can be programmed through the registers as well as through word serial commands. The programming registers are in the A16 address space. All registers are write only registers. Register offsets from the base address are shown in the following table. The base of the registers is set by the instrument's logical address:

```
BASE_ADDRESS = 49152 + logical_address * 64
```

For example a logical address of 8 means the base address is 49152 + 8*64 = 49664 (C200 in hex).

To set a resistance value for Channel 1, store the 16 bit binary value of the resistance at offset 20 hex. A value of 123 (007B hex) will cause Channel 1 to go to a value of 123 Ω. To set a resistance value for Channel 4, store the 16 bit binary value of the resistance at offset 26 hex.

The instrument can be configured to load new values when a trigger occurs. Word serial commands are required to configure the instrument to do this. Once the instrument is configured, values to be loaded at the next trigger can be set in the trigger registers.

For example, configure Channel 2 to load values on trigger. Write a 16 bit resistance value at offset 2A hex. The next trigger will cause Channel 2 to assume the resistance value.

Even though the registers are 16 bits, only the lower 14 bits are used. The upper 2 bits should be set to 0.

One register is available to control the sense relays. If PROTECT is on then only one set of relays can be closed at a time. Attempting to close more than one set of relays while PROTECT is on will cause all sense relays to open. The LSB (bit 0) controls the connection to Channel 1. Bit 1 controls the connection to Channel 2, bit 2 controls Channel 3, and bit 3 controls Channel 4. Setting a bit makes the connection.

See the following table for the register layout.

TABLE 3-1 A16 MEMORY

Offset	Description
3E	
3C	
3A	
38	
36	
34	
32	
30	Sense Relay Control
2E	Trigger Register Channel 4
2C	Trigger Register Channel 3
2A	Trigger Register Channel 2
28	Trigger Register Channel 1
26	Direct Register Channel 4
24	Direct Register Channel 3
22	Direct Register Channel 2
20	Direct Register Channel 1
1E	
1C	
1A	
18	
16	[A32 Pointer Low]
14	[A32 Pointer High]
12	[A24 Pointer Low]
10	[A24 Pointer High]
E	Data Low
C	Data High
A	Response [/Data Extended]
8	Protocol [/Signal] Register
6	[Offset Register]
4	Status / Control Register
2	Device Type
0	ID Register

VXI*PLUG&PLAY* EXAMPLES

```
/****************************************************************************
 *                                                                          *
 *                       APPLICATION FUNCTIONS                              *
 *                                                                          *
 ****************************************************************************/
/****************************************************************************
  Function:              vtvm7004_configChansAndSetResis

  Formal Parameters      ViSession instrHndl,
                         - A valid session handle to the instrument

                         ViInt16  channelArray[]
                         - This parameter specifies the physical channels which are to be setup.

                         Valid Range
                         --------------
                         vtvm7004_CHANNEL_NUM_MIN (1) to
                         vtvm7004_CHANNEL_NUM_MAX (4).

                         ViInt16  numOfChannels
                         - This parameter specifies the number of valid channels that are
                         specified in the 'channelArray[]' parameter.

                         Valid Range
                         --------------
                         vtvm7004_CHANNEL_NUM_MIN (1) to
                         vtvm7004_CHANNEL_NUM_MAX (4).

                         ViInt16  combinationType
                         - This parameter specifies the type of combination desired for the specified
                         channels.  It must be ensured that the channel outputs are wired externally
                         according to the combination configured by this function.

                         Valid Values                    Interpretation
                         ------------                    ----------------
                         vtvm7004_COMBO_NONE             Individual channels
                         vtvm7004_COMBO_DIV              Divider Mode
                         vtvm7004_COMBO_PAR              Parallel Mode
                         vtvm7004_COMBO_SER              Serial Mode

                         ViReal32 resistance,
                         - This parameter specifies the resistance value that is to be programmed   for the
                         specified logical channel.  The valid range for this parameter should satisfy the
                         below mentioned ranges else an instrument error is generated.
```

Channel Type	Valid Range
------------	-----------
Single Channel	0 to 16383 ohms (1 ohm steps)
2 Channels (series)	0 to 32766 ohms(1 ohm steps)
3 Channels (series)	0 to 49149 ohms(1 ohm steps)
4 Channels (series)	0 to 65532 ohms(1 ohm steps)
2 Channels (parallel)	0 to 8191.5 ohms(0.5 steps)
3 Channels (parallel)	0 to 5461.0 ohms(0.333 ohm steps)
4 Channels (parallel)	0 to 4095.75 ohms(0.25 ohm steps)
DIVider Mode	0.0 to 1.0 (Wiper Position)

Return Values: Returns VI_SUCCESS if successful.
else returns error value of the error encountered.

Description This function demonstrates how the core driver functions can be used to build an application function. This function sets up the specified channels by configuring up the combination type, connects their outputs' to their front panel connectors and sets the channel resistances.

Note that this function resets the module to its default state before performing the rest of its operations.
***/

```
ViStatus   _VI_FUNC vtvm7004_configChansAndSetResis
                                        (ViSession instrHndl,
                                        ViInt16 channelArray[],
                                        ViInt16 numOfChannels,
                                        ViInt16 combinationType,
                                        ViReal32 resistance)
{
     /*
      * Variable used to store the return status of the function.
      */

     ViStatus status = VI_NULL;

     ViInt16 relayProtectFlag = NULL;

     ViInt16 logicalChannel = NULL,

     i = NULL;


     /*
      * Check if input session handle is valid.
      */

     status = vtvm7004_invalidSession(instrHndl);
           if (status < VI_SUCCESS)
     return status;
```

```
 /*
  * Setting the module to its default state
  */

 status = vtvm7004_reset(instrHndl);
       if (status < VI_SUCCESS)
 return status;

 /*
  * Configuring the channel combination of the specified channels
  */

 status = vtvm7004_configChanCombination (instrHndl, combinationType,
                                           channelArray, numOfChannels);

       if (status < VI_SUCCESS)
 return vtvm7004_ERROR_CONFIG_CHAN_COMB;

 /*
  * Connecting the specified channels outputs' to their front
  * panel connectors
  */

       if (combinationType == vtvm7004_COMBO_PAR)
             relayProtectFlag = vtvm7004_DISAB_PROTECTION;
       else
             relayProtectFlag = vtvm7004_ENAB_PROTECTION;

 status = vtvm7004_connectDisconnectChans (instrHndl,
                                           vtvm7004_CONNECT_CHANS,
                                           channelArray, numOfChannels,
                                           relayProtectFlag);
       if (status < VI_SUCCESS)
       return vtvm7004_ERROR_CONNECTING_CHANS;

 /*
  * Determining the logical channel from the specified channel set
  */

status = vtvm7004_findLowestChan (channelArray, numOfChannels,
                                           &logicalChannel);
       if (status < VI_SUCCESS)
 return vtvm7004_ERROR_FINDING_LOG_CHAN;
```

```
    /*
     * Setting the channel resistance or the wiper position depending
     * on the combination type
     */

          if (combinationType == vtvm7004_COMBO_DIV)
    {
          status = vtvm7004_setWiperPosition  (instrHndl, logicalChannel,
                                              resistance);
                if (status < VI_SUCCESS)
          return vtvm7004_ERROR_SETTING_WIPER_POS;
    }
    else
    {
          if (combinationType != vtvm7004_COMBO_NONE)

          {
          status = vtvm7004_setChannelResistance (instrHndl,
                                              logicalChannel,
                                              resistance,
                                              vm7004_SET_RESIS_IMM);
                if (status < VI_SUCCESS)
          return vtvm7004_ERROR_SETTING_RES;
          }
          else
          {
              for (i = 0; i < 4; i++)
                {
                status = vtvm7004_setChannelResistance (instrHndl,
                                              channelArray[i],
                                              resistance,
                                              vtvm7004_SET_RESIS_IMM);
                      if (status < VI_SUCCESS)
                return vtvm7004_ERROR_SETTING_RES;
                }
          }
    }

    return VI_SUCCESS;
}
```

SECTION 4

COMMAND DICTIONARY

INTRODUCTION

This section presents the instrument command set. It begins with an alphabetical list of all the commands supported by the VM7004 divided into three sections: IEEE 488.2 common commands, the instrument specific SCPI commands and the required SCPI commands. With each command is a brief description of its function, whether the command's value is affected by the *RST command and its default value.

The remainder of this section is devoted to describing each command, one per page, in detail. The description is presented in a regular and orthogonal way assisting the user in the use of each command. Every command entry describes the exact command and query syntax, the use and range of parameters and a complete description of the command's purpose.

ALPHABETICAL COMMAND LISTING

The following tables provide an alphabetical listing of each command supported by the VM7004 along with a brief description. If an X is found in the column titled *RST, then the value or setting controlled by this command is possibly changed by the execution of the *RST command. If no X is found, then *RST has no effect. The default column gives the value of each command's setting when the unit is powered up or when a *RST command is executed.

TERMINOLOGY

Description of <channel_list>

EXAMPLES

(@1)	Channel 1
(@1,4)	Channels 1 and 4
(@ 1,2,3,4)	Channels 1, 2, 3 and 4
(@ 1:4)	Channels 1, 2, 3 and 4

Description of logical and physical channels

The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel (<l_chl>). The logical channel number is assigned to the lowest channel number in the channel's group.

For example: If R2 and R4 are in series, <l_chl> = 2
If R1 and R4 are in parallel, <l_chl> = 1

TABLE 4-1 IEEE 488.2 COMMON COMMANDS

Command	Description	*RST	*RST Value
*CLS	Clears the Status Register.	X	
*ESE	Sets the Event Status Enable Register.		N/A
*ESR?	Query the Standard Event Status Register		N/A
*IDN?	Query the Module Identification String.		N/A
*OPC	Sets the OPC bit in the Event Status Register	X	0
*RST	Resets the module to a known state		N/A
*SRE	Sets the Service Request Enable Register		N/A
*STB?	Query the Status Byte Register.		N?A
*TRG	Causes a trigger event to occur.		N/A
*TST?	Starts and reports a self-test procedure.		N/A
*WAI	Halts execution and queries		N/A

TABLE 4-2 INSTRUMENT SPECIFIC SCPI COMMANDS

Command	Description	*RST	*RST Value
ABORt	Disarms the VM7004 and stops data triggering (if active).		N/A
ARM[:SEQuence]	Prepares the scan list data and enables the triggering subsystem for scan list mode.	X	1,0
ARM[:SEQuence][:LAYer]:DELay	Programs the timer delay value. Sets the ARM mode to timer.	X	0.005
OUTPut:TRIGger:SOURce	Sets the source for the output trigger.	X	IMM
OUTPut:TRIGger:SLOPe	Sets the active slope of the trigger driven onto the TTL trigger bus.	X	POS
OUTPut:TRIGger:TTLTrig	Selects which VXIbus TTL trigger line the module will drive if the output is enabled.	X	TTLT0
OUTPut:TTLTrig [:STATe]	Enables or disables driving the trigger signal onto the VXIbus backplane TTL trigger lines.	X	OFF
ROUTe:CLOSe	Connects one or more channel's output in the channel list to the sense bus.	X	All open
ROUTe:OPEN	Disconnects one or more channel's output in the channel list from the sense bus.	X	All open
ROUTe:PROTect	Turns ON/OFF the sense bus relay protection.	X	ON
SOURce:ATTenuation	Sets the ratio (wiper position) in the voltage divider mode.		N/A
SOURce:COMBine	Sets multiple resistors to be treated as one.	X	NONE
SOURce:LIST:RESistance	Loads a scan list.	X	EMPTY
SOURce:LIST:RESistance:POINts?	Queries the number of points in the selected physical channel's scan list.		N/A
SOURce:RESistance[:LEVel][:IMMediate][:AMPLitude]	Sets resistance of logical channel <l_chl> to <value> ohms.	X	16383
SOURce:RESistance[:LEVel]:TRIGger[:AMPLitude]	Waits for trigger event before setting resistance of logical channel <l_chl> to <value> ohms.		
TRIGger[:SEQuence][:IMMediate]	Generate a trigger.		N/A
TRIGger:[SEQuence]:SLOPe	Selects the active edge for triggering the VM7004.	X	NEG
TRIGger:[SEQuence]:SOURce	Selects the source for the trigger	X	CH1

TABLE 4-3 SCPI REQUIRED COMMANDS

Command	Description	*RST	*RST Value
STATus:OPERation:CONDition?	Queries the Operation Status Condition Register.	X	
STATus:OPERation:ENABle	Sets the Operation Status Enable Register.	X	
STATus:OPERation[:EVENt]?	Queries the Operation Status Event Register.	X	
STATus:PRESet	Presets the Status Register.	X	
STATus:QUEStionable:CONDition?	Queries the Questionable Status Condition Register	X	
STATus:QUEStionable:ENABle	Sets the Questionable Status Enable Register.	X	
STATus:QUEStionable[:EVENt]?	Queries the Questionable Status Event Register	X	
SYSTem:ERRor?	Queries the Error Queue	X	Clears queue
SYSTem:VERsion?	Queries which version of the SCPI standard the module complies with.		N/A

COMMAND DICTIONARY

The remainder of this section is devoted to the actual command dictionary. Each command is fully described on its own page. In defining how each command is used, the following items are described:

Purpose	Describes the purpose of the command.
Type	Describes the type of command such as an event or setting.
Command Syntax	Details the exact command format.
Command Parameters	Describes the parameters sent with the command and their legal range.
Reset Value	Describes the values assumed when the *RST command is sent.
Query Syntax	Details the exact query form of the command.
Query Parameters	Describes the parameters sent with the command and their legal range. The default parameter values are assumed the same as in the command form unless described otherwise.
Query Response	Describes the format of the query response and the valid range of output.
Description	Describes in detail what the command does and refers to additional sources.
Examples	Present the proper use of each command and its query (when available).
Related Commands	Lists commands that affect the use of this command or commands that are affected by this command.

IEEE 488.2 Common Commands

*CLS

Purpose	Clears the Status Register	
Type	IEEE488.2 Common Command	
Command Syntax	*CLS	
Command Parameters	None	
***RST Value**	*RST performs all the functions of *CLS	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	This command clears all event registers, clears the OPC flag and clears all queues (except the output queue).	
Examples	**Command / Query**	**Response *(Description)***
	`*CLS`	
Related Commands	None	

*ESE

Purpose	Sets the bits of the Event Status Enable Register
Type	IEEE488.2 Common Command
Command Syntax	*ESE <mask>
Command Parameters	<mask> = numeric ASCII value in the range of 0 to 255
***RST Value**	N/A
Query Syntax	*ESE?
Query Parameters	None
Query Response	Numeric ASCII value, from 0 to 255
Description	The Event Status Enable command is used to set the bits of the Event Status Enable Register. See ANSI/IEEE488.2-1987 section 11.5.1 for a complete description of the ESE register. A value of 1 in a bit position of the ESE register enables generation of the ESB (Event Status Bit) in the Status Byte by the corresponding bit in the ESR. If the ESB is set in the SRE register then an interrupt will be generated. See the ESR? command for details regarding the individual bits. The ESE register layout is: Bit 0 - Operation Complete Bit 1 - Request Control (not used in the VM7004) Bit 2 - Query Error Bit 3 - Device Dependent Error (not used in the VM7004) Bit 4 - Execution Error Bit 5 - Command Error Bit 6 - User Request (not used in the VM7004) Bit 7 - Power On The Event Status Enable query reports the current contents of the Event Status Enable Register.
Examples	
Related Commands	*ESR

Examples

Command / Query	**Response *(Description)***
`*ESE 36`	
`*ESE?`	36

*ESR?

Purpose	Queries and clears the Standard Event Status Register	
Type	IEEE488.2 Common Command	
Command Syntax	None - Query Only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*ESR?	
Query Parameters	None	
Query Response	Numeric ASCII value from 0 to 255	
Description	The Event Status Register query - queries and clears the contents of the Standard Event Status Register. This register is used in conjunction with the ESE register to generate the ESB (Event Status Bit) in the Status Byte. The layout of the ESR is: Bit 0 - Operation Complete Bit 1 - Request Control (not used in the VM7004, always 0) Bit 2 - Query Error Bit 3 - Device Dependent Error (not used in the VM7004, always 0) Bit 4 - Execution Error Bit 5 - Command Error Bit 6 - User Request (not used in the VM7004, always 0) Bit 7 - Power On The Operation Complete bit is set by the VM7004 when it receives an *OPC command. The Query Error bit is set when data is over-written in the output queue. This could occur if one query is followed by another without reading the data from the first query. The Execution Error bit is set when an execution error is detected. See the section in the manual covering Error Messages for a list of execution errors. Errors which range from -200 to -299 are execution errors. The Command Error bit is set when a command error is detected. See the section in this manual covering Error Messages for a list of command errors. Errors which range from -100 to -199 are command errors. The Power On bit is set when the module is first powered on or after it receives a reset via the VXI Control Register. Once the bit is cleared (by executing the *ESR? command) it will remain cleared.	
Examples	**Command / Query**	**Response *(Description)***
	*ESR?	4
Related Commands	*ESE	

*IDN?

Purpose	Queries the module for its identification string	
Type	IEEE488.2 Common Command	
Command Syntax	None - Query Only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*IDN?	
Query Parameters	None	
Query Response	ASCII character string	
Description	The identification query returns the identification string of the VM7004 module. The response is divided into four fields separated by commas. The first field is the manufacturer's name, the second field is the model number, the third field is an optional serial number and the fourth field is the firmware revision number. If a serial number is not supplied, the third field is set to 0 (zero).	
Examples	**Command / Query**	**Response *(Description)***
	`*IDN?`	VXI Technology Inc.,VM7004,0,1.00 *(The revision listed here is for reference only; the response will always be the current revision of the instrument.)*
Related Commands	None	

*OPC

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Sets the OPC bit in the Event Status Register</td></tr>
<tr><td><b>Type</b></td><td colspan="2">IEEE488.2 Common Command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">*OPC</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">*RST moves any pending *OPC request</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">*OPC?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">1</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Operation Complete command sets the OPC bit in the Event Status Register when all pending operations have completed. The Operation Complete query will return a 1 to the output queue when all pending operations have completed.</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>*OPC</code>
<code>*OPC?</code></td><td>
1</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">*WAI</td></tr>
</table>

*RST

Purpose	Resets the module's hardware and software to a known state
Type	IEEE488.2 Common Command
Command Syntax	*RST
Command Parameters	None
***RST Value**	N/A
Query Syntax	None
Query Parameters	N/A
Query Response	N/A
Description	The Reset command resets the module's hardware and software to a known state. See the command index at the beginning of this section for the default parameter values set by this command.
Examples	**Command / Query**: *RST **Response *(Description)***:
Related Commands	None

*SRE

<table>
<tr><td><b>Purpose</b></td><td>Sets the service request enable register</td></tr>
<tr><td><b>Type</b></td><td>IEEE488.2 Common Command</td></tr>
<tr><td><b>Command Syntax</b></td><td>*SRE <mask></td></tr>
<tr><td><b>Command Parameters</b></td><td><mask> = numeric ASCII value in the range of 0 to 255</td></tr>
<tr><td><b>*RST Value</b></td><td>N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td>*SRE?</td></tr>
<tr><td><b>Query Parameters</b></td><td>None</td></tr>
<tr><td><b>Query Response</b></td><td>Numeric ASCII value from 0 to 255</td></tr>
<tr><td><b>Description</b></td><td>The service request enable mask is used to control which bits in the status byte generate backplane interrupts. If a bit is set in the mask that newly enables a bit set in the status byte and interrupts are enabled, the module will generate a REQUEST TRUE event via an interrupt. See the *STB? Command for the layout of bits. Note: Bit 6 is always internally cleared to zero as required by IEEE 488.2 section 11.3.2.3.</td></tr>
<tr><td><b>Examples</b></td><td><table>
<tr><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>*SRE 4</code></td><td></td></tr>
<tr><td><code>*SRE?</code></td><td>4</td></tr>
</table></td></tr>
<tr><td><b>Related Commands</b></td><td>*STB?</td></tr>
</table>

*STB?

Purpose	Queries the Status Byte Register
Type	IEEE 488.2 Common Command
Command Syntax	None - Query Only
Command Parameters	N/A
***RST Value**	N/A
Query Syntax	*STB?
Query Parameters	None
Query Response	Numeric ASCII value from 0 to 255
Description	The Read Status Byte query fetches the current contents of the Status Byte Register. See the IEEE 488.2 specification for additional information regarding the Status Byte Register and its use. The layout of the Status Byte Register is: Bit 0 - Unused Bit 1 - Unused Bit 2 - Error Queue Has Data Bit 3 - Questionable Status Summary (not used) Bit 4- Message Available Bit 5- Event Status Bit (ESB) Bit 6 - Master Summary Status Bit 7 - Operation Status Summary
Examples	**Command / Query**: `*STB?` — **Response *(Description)***: 16
Related Commands	None

*TRG

Purpose	Causes a trigger event to occur	
Type	IEEE 488.2 Common Command	
Command Syntax	*TRG	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	The Trigger command causes a trigger event to occur. In the VM7004 this is used to initiate the SOURce<_ch1>:RESistance[:LEVel]:TRIGger[:AMPLitude]<value> command.	
Examples	**Command / Query**	**Response *(Description)***
	`*TRG?`	
Related Commands	TRIGger	

*TST?

<table>
<tr><td><b>Purpose</b></td><td>Causes a self-test procedure to occur and queries the results</td></tr>
<tr><td><b>Type</b></td><td>IEEE488.2 Common Command</td></tr>
<tr><td><b>Command Syntax</b></td><td>None - Query Only</td></tr>
<tr><td><b>Command Parameters</b></td><td>N/A</td></tr>
<tr><td><b>*RST Value</b></td><td>N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td>*TST?</td></tr>
<tr><td><b>Query Parameters</b></td><td>None.</td></tr>
<tr><td><b>Query Response</b></td><td>Numeric ASCII value from 0 to 255</td></tr>
<tr><td><b>Description</b></td><td>The Self-Test query causes the VM7004 to run its self-test procedures and report on the results. The following tests are performed:

1. Non-volatile memory test
2. Timer test
3. Data path test
4. Trigger test

The *TST? query returns a numeric ASCII value which has the following meaning:

Bit 0 - Non-volatile memory failed
Bit 1 - Timer failed
Bit 2 - Data Path
Bit 4 - Trigger
Bit 5 - Unused
Bit 6 - Unused
Bit 7 - Unused

A bit value of 1 in any location indicates a failure while a 0 value indicates that the test passed.</td></tr>
<tr><td><b>Examples</b></td><td><table><tr><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr><tr><td><code>*TST?</code></td><td>0</td></tr></table></td></tr>
<tr><td><b>Related Commands</b></td><td>None</td></tr>
</table>

*WAI

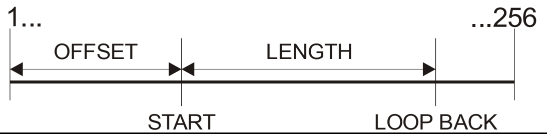
Purpose	Halts execution of additional commands and queries until the No Operation Pending message is true	
Type	IEEE488.2 Common Command	
Command Syntax	*WAI	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	The Wait to Continue command halts the execution of additional commands and queries until the No Operation Pending message is true. This command makes sure that all previous commands have been executed before continuing processing. It provides a way of synchronizing the module with its master.	
Examples	**Command / Query**	**Response *(Description)***
	`*WAI`	
Related Commands	*OPC	

INSTRUMENT SPECIFIC SCPI COMMANDS

ABORt

Purpose	Disarms the VM7004 and stops latching data to select resistors (if active)	
Type	Instrument specific SCPI	
Command Syntax	ABORt	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	The Abort command disarms the VM7004 and stops latching data to select resistors (if active). It also stops any ARM:DELay in progress.	
Examples	**Command / Query**	**Response *(Description)***
	ABOR	
Related Commands	None	

ARM[:SEQuence]

Purpose	Prepares the scan list data and enables the triggering subsystem for scan list mode
Type	Instrument specific SCPI
Command Syntax	ARM[<p_chl>][:SEQuence][<offset>,<length>]
Command Parameters	<p_chl> = physical channel and range from 1 to 4, if <p_ch1> is not specified, it defaults to Channel 1 <offset> = 1 to 256 <length> = 1 to 256
***RST Value**	*RST sets to disarm the scan list mode. All <offsets> = 1 All <lengths> = 0
Query Syntax	ARM[<p_chl>][:SEQuence]?
Query Parameters	<p_chl>= channel, defaults to Channel 1
Query Response	Returns the currently set value of the <offset> and length <parameters> in the following format: <offset>,<length>
Description	Arms the VM7004 to be ready to latch the data to select the resistors in channel <p_chl>. This command enables the trigger subsystem. If the parameters <offset> and <length> are not provided, the Scan List mode for physical channel <p_chl> is turned off. Setting the resistance values by the SOURce<l_chl>:RESistance[:LEVel]:TRIGger[:AMPLitude] command also turns off the Scan List mode for all channels. If the parameters <offset> and <length> are provided, then triggering will cause the resistance values defined for each channel's Scan List to be transferred to each active channel. The <offset> is the beginning location in the scan list data. The first location is 1. The <length> is the number of data words used in the scan list with a minimum length of 1 and a maximum length of 256. When the end of length is reached, the list wraps back around to start again at offset. If the parameter <p_chl> is supplied, then arming occurs for the channel corresponding to <p_chl> only. If <p_chl> is not supplied, then all channels are armed with a single command. Use of <p_ch1> allows some channels to run a scan list while other channels remain static. 

When the ARM<p_chl> is received, the values of the Scan List (which is set by the SOUR:LIST:RESistance command) are written to the internal trigger registers and wait for the next trigger. When a trigger occurs, the trigger registers are written to the relay registers to set the output resistance, and the CPU writes the next values in the Scan List to trigger registers. A second trigger will write the second set of values to the relay registers.

The query form of the command returns the offset and length of the specified channel. If <p_chl> is not provided, it then defaults to <p_chl> = 1.

The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel <l_chl>. The logical channel number is assigned to the lowest channel number in the channel's group.

For example:
If R2 and R4 are in series, <l_chl> = 2
If R1 and R4 are in parallel, <l_chl> = 1

Examples	**Command / Query**	**Response *(Description)***
	`SOUR1:LIST:RES 10,20,30,40,50,60,70, 80,90,100`	*(Sets the Channel 1 scan list to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100)*
	`SOUR2:LIST:RES 435,123`	*(Sets the Channel 2 scan list to 435 and 123.)*
	`TRIG:SOUR:IMM`	*(Sets Trigger Source to Immediate.)*
	`ARM 1 5,5`	*(Arms Channel 1. Starts at location 5 and length 5 of the scan list.)*
	`ARM 2 1,2`	*(Arms Channel 2. Starts at location 1 and length 2 of the scan list.)*
	`TRIG`	*(Generates a trigger. It causes Channel 1 to go to a resistance of 50 Ω. It also causes Channel 2 to go to a resistance of 435 Ω.)*
	`TRIG`	*Generates a trigger. It causes Channel 1 to go to a resistance of 60 Ω. It also causes Channel 2 to go to a resistance of 123 Ω.*
	`TRIG`	*Generates a trigger. It causes Channel 1 to go to a resistance of 70 Ω. It also causes Channel 2 to go to a resistance of 435 Ω.*
	`ARM 1?`	5,5 (*Queries the ARM physical Channel 1.)*
Related Commands	ARM:DELay	

ARM[:SEQuence][:LAYer]:DELay

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Set the delay time for relay output settle after a trigger occurs. This delay command sets the time duration between trigger occurrence to the arming of the next trigger.</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">ARM[:SEQuence][:LAYer]:DELay <delay></td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2"><delay> = 0.000000 to 0.655350 s</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">5 ms</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">ARM[:SEQuence][:LAYer]:DELay?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">Returns the currently set value of the <delay> parameter</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The VM7004 provides one count down programmable timer to set the time duration between the trigger occurrence to the arming of the next trigger. When the external trigger, VXI TTL trigger or s/w(TRIG:IMM) are selected as the source to generate the trigger, the count down timer must finish the count. When trigger occurs, all four trigger registers are written to the relay registers and the timer is reloaded with the value which is set by this command. The timer must be counted down to equal 0 (to arm for next trigger) before another trigger can occur. The trigger which occurs before the timer reaches 0 will not be recognized and must be resent. The timer could be disabled by setting command ARM[:SEQuence][:LAYer]:IMMediate.</td></tr>
<tr><td rowspan="3"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>ARM:DEL 0.002</code></td><td><i>(Program delay 2 ms)</i></td></tr>
<tr><td><code>ARM:DEL?</code></td><td>0.000200</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">ARM[:SEQuence][:LAYer]:IMMediate
TRIGger:SOURce</td></tr>
</table>

OUTPut:TRIGger:SOURce

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Select the source of the output trigger</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">OUTPut:TRIGger:SOURce <source></td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2"><source> = CH1 | CH2 | CH3 | CH4 | INTernal | EXTernal | IMMediate | TIMer</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">IMM</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">OUTPut:TRIGger:SOURce?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">CH1 | CH2 | CH3 | CH4 | INT | EXT | IMM | TIM</td></tr>
<tr><td><b>Description</b></td><td colspan="2">This command selects the source of the output trigger.</td></tr>
<tr><td rowspan="3"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>OUTP:TRIG:SOUR CH1</code></td><td></td></tr>
<tr><td><code>OUTP:TRIG:SOUR?</code></td><td>CH1</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">OUTPut:TRIGger:SLOPe</td></tr>
</table>

OUTPut:TRIGger:SLOPe

Purpose	Sets the active slope of the trigger driven onto the TTL trigger bus	
Type	Instrument specific SCPI	
Command Syntax	OUTPut:TRIGger:SLOPe <slope>	
Command Parameters	<slope> = POSitive or NEGative	
***RST Value**	POS	
Query Syntax	OUTPut:TRIGger:SLOPe?	
Query Parameters	None	
Query Response	POS | NEG	
Description	The output trigger slope command sets the active slope of the trigger driven onto the TTL trigger bus.	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP:TRIG:SLOP NEG`	
	`OUT:TRIG:SLOP?`	NEG
Related Commands	OUTPut:TRIGger:SOURce	

OUTPut:TRIGger:TTLTrig

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Selects which VXIbus TTL trigger line the module will drive when the output is enabled.</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">OUTPut:TRIGger:TTLTrig <line></td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2"><line> = 0 - 7</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">TTLT0</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">OUTPut:TRIGger:TTLTrig?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">0 | 1 | 2 | 3 | 4 | 5 | 6 | 7</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The output trigger command selects which VXIbus TTL trigger line the module will drive when the output is enabled. The parameter may have the value of 0 to 7. Note that only one output line may be selected.</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>OUTP:TRIG:TTLT 3</code>
<code>OUTP:TRIG:TTLT?</code></td><td>
3</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">OUTPut:TTLTrig[:STATe]</td></tr>
</table>

OUTPut:TTLTrig [:STATe]

<table>
<tr><td><b>Purpose</b></td><td>Enables or disables driving the trigger signal onto the VXIbus backplane TTL trigger lines.</td></tr>
<tr><td><b>Type</b></td><td>Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td>OUTPut:TTLTrig[:STATe] <boolean></td></tr>
<tr><td><b>Command Parameters</b></td><td><boolean> = ON | OFF | 0 | 1</td></tr>
<tr><td><b>*RST Value</b></td><td>OFF</td></tr>
<tr><td><b>Query Syntax</b></td><td>OUTPut:TTLTrig[:STATe]?</td></tr>
<tr><td><b>Query Parameters</b></td><td>None</td></tr>
<tr><td><b>Query Response</b></td><td>1 | 0</td></tr>
<tr><td><b>Description</b></td><td>The output TTL trigger state command enables or disables driving the trigger signal onto the VXIbus backplane TTL trigger lines. ON enables the drivers, while OFF disables them.

This command causes the VM7004 to source a trigger event to the VXI TTL trigger bus when a trigger is received. The source of the trigger is selected by the OUTPut:TRIGger:SOURce command.</td></tr>
<tr><td><b>Examples</b></td><td><table>
<tr><th>Command / Query</th><th>Response <i>(Description)</i></th></tr>
<tr><td><code>OUTP:TTLT ON</code></td><td></td></tr>
<tr><td><code>OUTP:TTLT?</code></td><td>1</td></tr>
</table></td></tr>
<tr><td><b>Related Commands</b></td><td>OUTPut:TRIGger:TTLTrig</td></tr>
</table>

ROUTe:CLOSe

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Connect one or more channel’s output in the channel list to the sense bus and the common bus.</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">ROUTe:CLOSe <channel_list></td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2"><channel_list> = Standard channel list syntax supporting Channels 1 through 4
(See the beginning of Section 4 for correct channel list syntax and terminology)</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">*RST sets Channel 1 to close and all other sense relays are open</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">ROUTe:CLOSe ? <p_chl></td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2"><p_chl> = 1 | 2 | 3 | 4</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">1 | 0 (1 = Close, 0 = Open)</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Route Close command connects the sense bus and common bus to the <channel_list>. The sense bus and common bus typically are connected to an ohmmeter or voltmeter. This Route Close command enables <channel_list> to be connected to a sensor device for measuring. Only the first channel in the <channel_list> is connected when ROUTe:PROTect is ON. When ROUTe:PROTect is OFF, more than one channel can be connected to the sense bus. NOTE: This has the effect of internally paralleling the channels in the <channel_list>.
The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel <l_chl>. The logical channel number is assigned to the lowest channel number in the channel’s group.
For example: If R2 and R4 are in series, <l_chl> = 2
If R1 and R4 are in parallel, <l_chl> = 1</td></tr>
<tr><td rowspan="3"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>ROUT:CLOS (@2)</code></td><td><i>(Connects Channel two to sense bus and common bus to prepare for measuring.)</i></td></tr>
<tr><td><code>ROUT:CLOS? 3</code></td><td>0 <i>(Ask if Channel 3 is disconnected to the sense bus.)</i></td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">ROUTe:PROTect<boolean>
ROUTe:OPEN</td></tr>
</table>

ROUTe:OPEN

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Disconnect one or more channel’s output in the channel_list from the sense bus and the common bus</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">ROUTe:OPEN <channel_list></td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2"><channel_list> = Standard channel list syntax supporting Channels 1 through 4
(See the beginning of Section 4 for correct channel list syntax and terminology)</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">*RST sets Channel 1 to close and all other sense relays are open</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">ROUTe:OPEN? <p_chl></td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2"><p_chl> = 1 | 2 | 3 | 4</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">1 | 0 (1 = Open, 0 = Close)</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Route Open command disconnects the sense bus and common bus from <channel_list>. The sense bus and common bus typically are connected to an ohmmeter or voltmeter. All the channels are disconnected when ROUTe:PROTect is ON. When ROUTe:PROTect is OFF, one or more than one channels can be disconnected from the sense bus.

The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel (<l_chl>). The logical channel number is assigned to the lowest channel number in the channel’s group.

For example: If R2 and R4 are in series, <l_chl> = 2.
If R1 and R4 are in parallel, <l_chl> = 1.</td></tr>
<tr><td rowspan="3"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>ROUT:OPEN (@2)</code></td><td><i>(Disconnects Channel 2 from the sense bus and common bus.)</i></td></tr>
<tr><td><code>ROUT:OPEN? 3</code></td><td>0 (<i>Indicates that Channel 3 is not connected to the sense bus)</i></td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">ROUTe:PROTect
ROUTe:CLOSe</td></tr>
</table>

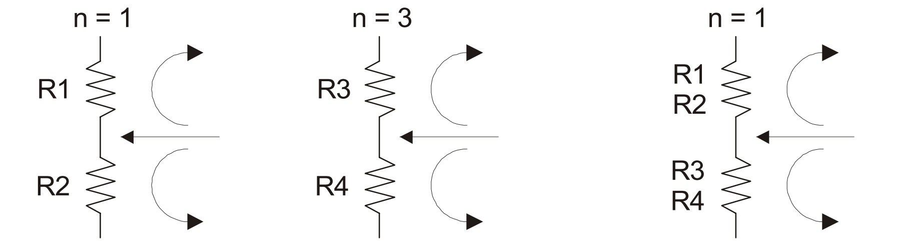
ROUTe:PROTect

Purpose	This command turns ON/OFF the sense relay’s protection circuit	
Type	Instrument specific SCPI	
Command Syntax	ROUTe:PROTect <boolean>	
Command Parameters	<boolean> = ON | OFF | 0 | 1	
***RST Value**	*RST sets PROTect to ON	
Query Syntax	ROUTe:PROTect?	
Query Parameters	None	
Query Response	1 | 0 (1 = ON, 0 = OFF)	
Description	When Protect is ON, only one channel can be connected to the sense bus and common bus. When Protect is OFF, multiple channels may be connected to the sense bus and common bus at the same time. This allows multiple channels to be configured in parallel. The channels should be programmed to Parallel mode before using this command. **Note: Do not use this command unless you know exactly how the sense relays are connected.**	
Examples	**Command / Query**	**Response *(Description)***

	`ROUT:PROT OFF`
	`ROUT:PROT?`
		
		0

Related Commands	ROUTe:CLOSe ROUTe:OPEN	

SOURce:ATTenuation

Purpose	Sets the wiper position (ratio) for logical channel in the voltage **DIVider mode**
Type	Instrument specific SCPI
Command Syntax	SOURce[<l_chl>]:ATTenuation <numeric_value>
Command Parameters	Mode DIV (divider) only. <l_chl> = 1 (R1, R2), <l_chl> = 3 (R3,R4), <l_chl> = 1 (R1, R2, R3, R4) <l_chl> = logical channel. 0 ≤ <numeric_value> ≤ 1 The parameter <l_chl> may have the value of 1 or 3; defaults to 1 when not specified.
***RST Value**	N/A
Query Syntax	SOURce[<l_chl>]:ATTenuation?
Query Parameters	<l_chl> = 1 | 3
Query Response	Numeric value from 0 to 1
Description	This command sets the wiper position of a voltage divider combination and should only be set after the logical channel has been established by the command SOUR:COMB DIV. This command has the same effect as the command (does not require trigger) SOUR[<l_chl>]:RES[:LEV][:IMM][:AMPL] <value> when the logical channel is in divider mode. The parameter <numeric_value> has a range of 0 to 1 and represents the fraction of the input voltage at the wiper. The lowest channel number (1 or 3) is considered to be the clockwise leg of the potentiometer, the common connection between two channels (or the common connection between R2 and R3 for a four resistor combination) is considered the wiper and the highest channel number (2 or 4) is considered the counter-clockwise leg of the potentiometer.  n = 1 R1 R2 n = 3 R3 R4 n = 1 R1 R2 R3 R4 R1 + R2 = 16,383 Ω R3 + R4 = 16,383 Ω R1 + R2 + R3 + R4 = 32,766 Ω The user must wire the output externally. The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel (<l_chl>). The logical channel number is assigned to the lowest channel number in the channel's group. For example: If R2 and R4 are in series, <l_chl> = 2 If R1 and R4 are in parallel, <l_chl> = 1

Examples	Command / Query	Response *(Description)*
	`SOUR3:ATT 0.5`	*(Sets the voltage divider ratio to 0.5. R3=R4)*
	`SOUR:ATT 0.75`	*(Sets the voltage divider ratio of R1 & R2 to 0.75. R1=1/3 R2 (2 resistors in series, R1=4096, R2=12,287).* *Sets the voltage divider ratio of R1, R2 & R3, R4 to 0.75 R1+R2=1/3 (R3+R4) (4 resistors in series).*
	`SOUR1:ATT?`	0.500000
Related Commands	SOURce:COMBine	

SOURce:COMBine

Purpose	Sets the resistor mode for the VM7004. This defines how resistor channels are **combined externally.**
Type	Instrument specific SCPI
Command Syntax	SOURce:COMBine NONE SOURce:COMBine <type> [<channel_list>]
Command Parameters	<type> = NONE | DIV | PAR | SER <channel_list> = Standard channel list syntax supporting Channels 1 through 4 (See the beginning of Section 4 for correct channel list syntax and terminology)
***RST Value**	<type> = NONE
Query Syntax	SOURce:COMBine? <p_chl>
Query Parameters	<p_chl> = 1 | 2 | 3 | 4
Query Response	NONE | DIV | PAR | SER
Description	This command is used to define how resistor channels are combined externally so that further programming might be simplified. When the <type> is NONE, then all channels are treated as individual resistors. When <type> is SER, then a list of resistors are treated as though they are wired in *series*. When <type> is PAR, the a list of resistors are treated as though they are wired in *parallel*. When <type> is DIV, the resistors in the list are treated as though they are wired up as a *divider*. Valid lists for each of the types are as follows: **NONE** : Sets VM7004 to 4 individual resistor channels **DIV** : <channel_list> = (@1,2) or (@3,4) or (@1:4) **PAR** : <channel_list> = any combination of 1,2,3,4 **SER** : <channel_list> = any combination of 1,2,3,4 When multiple resistor channels are connected in series, the accuracy of the resistor series will decrease by the number of channels combined. **Note**: The user must wire the output externally according to the configuration programmed by this command. The instrument does **not** make the connections between resistors! If the <p_chl> is programmed to a combine mode (DIV or PAR or SER), it must be removed from the combination with mode = NONE before that <p_chl> can be reprogrammed with another mode. The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel (<l_chl>). The logical channel number is assigned to the lowest channel number in the channel's group. For example : If R2 and R4 are in series, <l_chl> = 2. If R1 and R4 are in parallel, <l_chl> = 1.

Examples	**Command / Query**	**Response *(Description)***
	`SOUR:COMB NONE`	*(Sets the VM7004 to four individual resistor channels.)*
	`SOUR:COMB DIV (@1,2)`	*(Set R1 & R2 to voltage divider with two resistors in series. R2 is at the bottom).*
	`SOUR:COMB NONE`	
	`SOUR:COMB DIV (@1:4)`	*(Sets the VM7004 to voltage divider with four resistors in series. R4 is at the bottom.)*
	`SOUR:COMB NONE`	
	`SOUR:COMB PAR (@2,3)`	*(Sets two resistors R2, R3 to parallel mode with logical channel = 2.)*
	`SOUR:COMB SER (@1,4)`	*(Sets two resistors R1 & R4 to series mode with logical Channel = 1. While R2 & R3 are still in parallel mode.)*
	`SOUR:COMB? 3`	PAR (*Queries the physical Channel 3 for its current mode.)*
	`SOUR:COMB? 4`	SER (*Queries the physical Channel 4 for its current mode.)*
Related Commands	SOURce:ATTenuation SOURce:RESistance[:LEVel][:IMMediate][:AMPLitude]	

SOURce:LIST:RESistance

Purpose	Loads the scan list for a the resistor channel
Type	Instrument specific SCPI
Command Syntax	SOURce<p_chl>:LIST:RESistance[<numeric_value>]{,<numeric_value>}
Command Parameters	<p_chl> = 1 | 2 | 3 | 4, if not specified, it defaults to Channel 1 <numeric_value> = 0 to 16383, if not specified, it defaults to maximum resistance
***RST Value**	*RST will set all scan lists empty
Query Syntax	SOURce<p_chl>:LIST:RESistance? <index>
Query Parameters	<p_chl> = 1 | 2 | 3 | 4 <index> = the location in scan list. Range is from 1 to 256
Query Response	Numeric ASCII value
Description	This command saves the scan list (<numeric_value> = resistance setting) for channel <p_chl> into an internal memory array. When the ARM command is received, the values of the scan list are written to the internal trigger registers and the module waits for a trigger. When a trigger occurs, the trigger registers are written to the relay registers to set the output resistances and a new values are loaded into the trigger registers. A new trigger will write the new values etc. The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel (<l_chl>). The logical channel number is assigned to the lowest channel number in the channel's group. For example: If R2 and R4 are in series, <l_chl> = 2. If R1 and R4 are in parallel, <l_chl> = 1 If no numeric_value is given with the command, the scan list is deleted. Numeric_value may be any valid value for a single resistor channel or combined resistor function including the divider function. Each numeric_value represents the resistance value setting for each channel. For combined resistor functions, the lowest channel in the function identifies the function (R1 in series with R2 is identified as Channel 1, R3 in parallel with R4 is identified as Channel 3). A maximum of 256 points may be loaded per channel. Each time the scan list is loaded for a given channel, it replaces the previously loaded scan list in its entirety. ARM command must be used to set up data and TRIG:SOURce and select the source of the trigger. The query form of the command returns the first data in the scan list if the parameter <index> is not provided. If <index> is provided, then it returns a single value for the selected point in the scan list (if index is 3, then the third value in the scan list is returned for the selected channel). An <index> value of 0 is illegal and causes an error.

Examples	**Command / Query**	**Response *(Description)***
	`SOUR4:LIST:RES`	*(Deletes the scan list for Channel 4.)*
	`SOUR2:LIST:RES 435,123,456,789`	*(Loads the Channel 2scan list with 4 data values: 435, 123, 456 and 789.)*
	`ARM 2 1,4`	*(Arms Channel 2. Start with the first data in the list and use 4 items before wrapping back to the first.)*
	`TRIG`	*(Generates the trigger. Set the resistance of Channel 2 to 435).*
	`SOUR2:LIST:RES?`	435 *(Queries the first data in the scan list for Channel 2.)*
	`SOUR2:LIST:RES? 3`	456 *(Queries the third data in the scan list for Channel 2.)*
Related Commands	SOURce:LIST:RESistance:POINt? ARM:SEQuence	

SOURce:LIST:RESistance:POINts?

<table>
<tr><td><b>Purpose</b></td><td colspan="2">This query returns the number of points in the selected channel’s scan list</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">None</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">*RST sets to all lists to 0 points</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">SOURce<p_chl>:LIST:RESistance:POINts?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2"><p_chl> = 1 | 2 | 3 | 4</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">Numeric ASCII value</td></tr>
<tr><td><b>Description</b></td><td colspan="2">This query returns the number of points in the selected channel’s scan list.</td></tr>
<tr><td rowspan="3"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>SOUR2:LIST:RES 5,7,9,12</code></td><td></td></tr>
<tr><td><code>SOUR2:LIST:RES:POIN?</code></td><td>4</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">SOURce:LIST:RESistance</td></tr>
</table>

SOURce:RESistance[:LEVel][:IMMediate][:AMPLitude]

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Sets the resistance of channel <l_chl> to <value> ohms upon receipt of the command.</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">SOURce[<l_chl>]:RESistance[:LEVel][:IMMediate][:AMPLitude]<value></td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2"><value> = See description for allowed values
<l_chl> = 1 | 2 | 3 | 4</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">*RST sets all channels to maximum resistance value (16,383 Ω)</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">SOURce[<l_chl>]:RESistance[:LEVel][:IMMediate][:AMPLitude]?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2"><l_chl> = 1 | 2 | 3 | 4</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2"><value></td></tr>
<tr><td><b>Description</b></td><td colspan="2">Sets the resistance of logical channel <l_chl> to <value> ohms upon receipt of the command. <l_chl> may have a value of 1 to 4. If <l_chl> is not specified, the channel defaults to Channel 1.

The VM7004 has four channels from 1 to 4. When multiple channels are grouped together by the SOURce:COMBine command, it is called a logical channel (<l_chl>). The logical channel number is assigned to the lowest channel number in the channel’s group.

For example: If R2 and R4 are in series, <l_chl> = 2.
If R1 and R4 are in parallel, <l_chl> = 1.

The parameter <value> sets the programmed resistance for the channel. The valid range of <value> depends upon how channels are combined. <b>When logical channels are combined to form a divider, this command is not used. Instead the SOURce:ATTenuation command is used.</b> Valid ranges for the parameter <value> are as follows:

Single channel : 0 to 16,383 Ω in 1 Ω steps
Two channels in series : 0 to 32,766 Ω in 1 Ω steps
Three channels in series : 0 to 49,149 Ω in 1 Ω steps
Four channels in series : 0 to 65,532 Ω in 1 Ω steps
Two channels in parallel : 0 to 8,191.5 Ω in 0.5 Ω steps
Three channels in parallel : 0 to 5,461.0 Ω in 0.333 Ω steps
Four channels in parallel : 0 to 4,095.75 Ω in 0.25 Ω steps

This command does not require a trigger.</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td>SOUR3:RES 500
SOUR3:RES?</td><td><i>(Sets Channel 3 to 500 Ω.)</i>
500.000000</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">SOURce:COMBine</td></tr>
</table>

SOURce:RESistance[:LEVel]:TRIGger[:AMPLitude]

<table>
<tr><td><b>Purpose</b></td><td>Set the resistance of channel <n> to <value> ohms but wait for a trigger event to occur before transferring the new value to the output.</td></tr>
<tr><td><b>Type</b></td><td>Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td>SOURce<l_chl>:RESistance[:LEVel]:TRIGger[:AMPLitude] <value></td></tr>
<tr><td><b>Command Parameters</b></td><td><value> = See description for allowed values
<l_chl> = 1 | 2 | 3 | 4</td></tr>
<tr><td><b>*RST Value</b></td><td>*RST sets all channels to maximum resistance value (16,383 Ω)</td></tr>
<tr><td><b>Query Syntax</b></td><td>SOURce<l_chl>:RESistance[:LEVel]:TRIGger[:AMPLitude]?</td></tr>
<tr><td><b>Query Parameters</b></td><td><l_chl> = 1 | 2 | 3 | 4</td></tr>
<tr><td><b>Query Response</b></td><td><value></td></tr>
<tr><td><b>Description</b></td><td>Sets the resistance of logical channel <l_chl> to <value> ohms upon receipt of the command. <l_chl> may have a value of 1 to 4. If <l_chl> is not specified, the channel defaults to Channel 1.
The VM7004 has four physical channels (<p_chl>) from 1 to 4. When one or multiple physical channels are grouped together, it is called the logical channel (<l_chl>). The logical channel number is assigned by the lowest channel number in the physical channel’s group.
For example : If R2 and R4 are in series, <l_chl> = 2.
If R3 and R4 are in parallel, <l_chl> = 3.
The parameter <value> sets the programmed resistance for the channel. The valid range of <value> depends upon how channels are combined. <b>When logical channels are combined to form a divider, this command is not used.</b> Valid ranges for the parameter <value> are as follows:
<table>
<tr><td>Single channel</td><td>: 0 to 16,383 Ω in 1 Ω steps</td></tr>
<tr><td>Two channels in series</td><td>: 0 to 32,766 Ω in 1 Ω steps</td></tr>
<tr><td>Three channels in series</td><td>: 0 to 49,149 Ω in 1 Ω steps</td></tr>
<tr><td>Four channels in series</td><td>: 0 to 65,532 Ω in 1 Ω steps</td></tr>
<tr><td>Two channels in parallel</td><td>: 0 to 8,191.5 Ω in 0.5 Ω steps</td></tr>
<tr><td>Three channels in parallel</td><td>: 0 to 5,461.0 Ω in 0.333 Ω steps</td></tr>
<tr><td>Four channels in parallel</td><td>: 0 to 4,095.75 Ω in 0.25 Ω steps</td></tr>
</table>
This command will turn off the Scan List mode for channel in which is set by command ARM<l_chl>[:SEQuence]<offset>,<length>.</td></tr>
<tr><td><b>Examples</b></td><td>
<table>
<tr><th>Command / Query</th><th>Response <i>(Description)</i></th></tr>
<tr><td><code>SOUR3:RES:TRIG 500</code></td><td><i>(Sets logical Channel 3 to 500 Ω).</i></td></tr>
<tr><td><code>TRIG:SOUR:IMM</code></td><td>(Sets Trigger SOURce mode wait for command TRIG)</td></tr>
<tr><td><code>TRIG</code></td><td><i>Arms trigger and sets the Channel 3 to 500 Ω.</i></td></tr>
<tr><td><code>SOUR3:RES:TRIG?</code></td><td>500.000000</td></tr>
</table>
</td></tr>
<tr><td><b>Related Commands</b></td><td>SOURce:COMBine
TRIGger[:SEQuence]:SOURce
TRIGger[:SEQuence][:IMMediate]</td></tr>
</table>

TRIGger[:SEQuence][:IMMediate]

<table>
<tr><td><b>Purpose</b></td><td colspan="2">This command causes armed channels to update their resistance values</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">TRIGger[:SEQuence][:IMMediate]</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Description</b></td><td colspan="2">Performs the same function as *TRG. This command causes armed channels to set their output to programmed resistance values. Values are set by the command SOURce<n>:RESistance[:LEVel]:TRIGger[:AMPLitude]<value>. It is the equivalent command to the IEEE 488.2 *TRG command. This command is active and available regardless of which trigger source is selected.</td></tr>
<tr><td rowspan="5"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td>TRIG</td><td></td></tr>
<tr><td>TRIG:SEQ</td><td></td></tr>
<tr><td>TRIG:IMM</td><td></td></tr>
<tr><td>TRIG:SEQ:IMM</td><td></td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">*TRG
TRIGger[:SEQuence]:SOURce</td></tr>
</table>

TRIGger[:SEQuence]:SLOPe

<table>
<tr><td><b>Purpose</b></td><td>Selects the active edge for triggering the VM7004</td></tr>
<tr><td><b>Type</b></td><td>Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td>TRIGger[:SEQuence]:SLOPe <slope></td></tr>
<tr><td><b>Command Parameters</b></td><td><slope> = POSitive | NEGative</td></tr>
<tr><td><b>*RST Value</b></td><td>POS</td></tr>
<tr><td><b>Query Syntax</b></td><td>TRIGger:SLOPe?</td></tr>
<tr><td><b>Query Parameters</b></td><td>None</td></tr>
<tr><td><b>Query Response</b></td><td>POS | NEG</td></tr>
<tr><td><b>Description</b></td><td>The trigger slope command selects the active edge for triggering the VM7004. Selecting the positive slope will require that the trigger input make a negative to positive transition through the trigger level while a negative slope requires the trigger input to make a positive to negative transition.</td></tr>
<tr><td><b>Examples</b></td><td><table>
<tr><th>Command / Query</th><th>Response <i>(Description)</i></th></tr>
<tr><td>`TRIG:SLOP NEG`</td><td></td></tr>
<tr><td>`TRIG:SLOP?`</td><td>NEG</td></tr>
</table></td></tr>
<tr><td><b>Related Commands</b></td><td>TRIGger[:SEQuence]:SOURce
SOURce:RESistance[:LEVel]:TRIGger[:AMPLitude]</td></tr>
</table>

TRIGger[:SEQuence]:SOURce

<table>
<tr><td><b>Purpose</b></td><td colspan="2">This command selects the source for triggering</td></tr>
<tr><td><b>Type</b></td><td colspan="2">Instrument specific SCPI</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">TRIGger:[:SEQuence]:SOURce <source></td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2"><source> = EXTernal | TTLTrig {<[0 | 1 | 2 | 3 | 4 | 5 | 6 | 7]>} | IMMediate</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">IMMediate</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">TRIGger[:SEQuence]:SOURce?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">EXT | TTLT <[0 | 1 | 2 | 3 | 4 | 5 | 6 | 7]> | IMM</td></tr>
<tr><td><b>Description</b></td><td colspan="2">This command selects the sources for triggering. The following sources are valid:
EXTernal : Trigger input provided by the user from the front panel connector. This input is TTL compatible.
IMMediate : No source selected. Triggering is done by the TRIG:IMM command or *TRG.
TTLTrig<n> : Backplane TTL trigger bus where <n> = 0 to 7.</td></tr>
<tr><td rowspan="4"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>TRIG:SOUR TTLT2</code></td><td></td></tr>
<tr><td><code>TRIG:SOUR EXT</code></td><td></td></tr>
<tr><td><code>TRIG:SOUR?</code></td><td>EXT</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">SOURce<n>:RESistance[:LEVel]:TRIGger[:AMPLitude]
TRIGger[:SEQuence]:SLOPe</td></tr>
</table>

SCPI REQUIRED COMMANDS

STATus:OPERation:CONDition?

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Queries the Operation Status Condition Register</td></tr>
<tr><td><b>Type</b></td><td colspan="2">SCPI required command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">None - Query Only</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">STATus:OPERation:CONDition?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">0</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Operation Status Condition Register query is provided for SCPI compliance only. The VM7004 does not alter the state of any of the bits in this register and always reports a 0.</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>STAT:OPER:COND?</code></td><td>0</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">None</td></tr>
</table>

STATus:OPERation:ENABle

Purpose	Sets the Operation Status Enable Register
Type	SCPI required command
Command Syntax	STATus:OPERation:ENABle <NRf>
Command Parameters	<NRf> = numeric ASCII value from 0 to 32767
***RST Value**	NRf must be specified
Query Syntax	STATus:OPERation:ENABle?
Query Parameters	None
Query Response	Numeric ASCII value from 0 to 32767
Description	The Operation Status Enable Register is included for SCPI compatibility. The register layout is as follows: Bit 0 - Calibrating (not used on VM7004) Bit 1 - Settling Bit 2 - Ranging (not used on VM7004) Bit 3 - Sweeping (not used on VM7004) Bit 4 - Measuring (not used on VM7004) Bit 5 - Waiting for trigger (not used on VM7004) Bit 6 - Waiting for arm (not used on VM7004) Bit 7 - Correcting (not used on VM7004)
Examples	**Command / Query** | **Response *(Description)*** `STAT:OPER:ENAB 0` | `STAT:OPER:ENAB?` | 0
Related Commands	None

STATus:OPERation[:EVENt]?

Purpose	Queries the Operation Status Event Register	
Type	SCPI required command	
Command Syntax	None - Query Only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:OPERation[:EVENt]?	
Query Parameters	None	
Query Response	0	
Description	The Status Operation Event Register query is included for SCPI compliance. The register layout is as follows: Bit 0 - Calibrating (not used on VM7004) Bit 1 - Settling Bit 2 - Ranging (not used on VM7004) Bit 3 - Sweeping (not used on VM7004) Bit 4 - Measuring (not used on VM7004) Bit 5 - Waiting for trigger (not used on VM7004) Bit 6 - Waiting for arm (not used on VM7004) Bit 7 - Correcting (not used on VM7004)	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:OPER?`	0
Related Commands	None	

STATus:PRESet

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Presets the Status Registers</td></tr>
<tr><td><b>Type</b></td><td colspan="2">SCPI required command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">STATus:PRESet</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">None - Command Only</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Status Preset command presets the Status Registers. The Operational Status Enable Register is set to 0 and the Questionable Status Enable Register is set to 0. This command is provided for SCPI compliance only.</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td>STAT:PRES</td><td></td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">None</td></tr>
</table>

STATus:QUEStionable:CONDition?

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Queries the Questionable Status Condition Register</td></tr>
<tr><td><b>Type</b></td><td colspan="2">SCPI required command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">None - Query Only</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">STATus:QUEStionable:CONDition?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">0</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Questionable Status Condition Register query is provided for SCPI compliance only. The VM7004 does not alter any of the bits in this register and a query always reports a 0.</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>STAT:QUES:COND?</code></td><td>0</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">None</td></tr>
</table>

STATus:QUEStionable:ENABle

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Sets the Questionable Status Enable Register</td></tr>
<tr><td><b>Type</b></td><td colspan="2">SCPI required command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">STATus:QUEStionable:ENABle &lt;NRf&gt;</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">&lt;NRf&gt; = numeric ASCII value from 0 to 32767</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">NRf must be supplied</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">STATus:QUEStionable:ENABle?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">Numeric ASCII value from 0 to 32767</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Status Questionable Enable command sets the bits in the Questionable Status Enable Register. This command is provided only to comply with the SCPI standard.

The Status Questionable Enable query reports the contents of the Questionable Status Enable Register. The VM7004 does not alter the bit settings of this register and will report the last programmed value.</td></tr>
<tr><td rowspan="3"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>STAT:QUES:ENAB 64</code></td><td></td></tr>
<tr><td><code>STAT:QUES:ENAB?</code></td><td>64</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">None</td></tr>
</table>

STATus:QUEStionable:EVENt

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Queries the Questionable Status Event Register</td></tr>
<tr><td><b>Type</b></td><td colspan="2">SCPI required command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">None - Query Only</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">STATus:QUEStionable [:EVENt]?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">0</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The Questionable Status Event Register is provided for SCPI compliance only. The VM7004 does not alter the bits in this register and queries always report a 0</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>STAT:QUES?</code></td><td>0</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">None</td></tr>
</table>

SYSTem:ERRor?

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Queries the Error Queue</td></tr>
<tr><td><b>Type</b></td><td colspan="2">SCPI required command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">None - Query Only</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">SYSTem:ERRor?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">ASCII string</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The System Error query is used to retrieve error messages from the error queue. The error queue will maintain the two error messages. If additional errors occur, the queue will overflow and the second and subsequent error messages will be lost. In the case of an overflow, an overflow message will replace the second error message. See the SCPI standard Volume 2: Command Reference for details on errors and reporting them. Refer to the “Error Messages” section of this manual for specific details regarding the reported errors (at the end of section 4).</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td><code>SYST:ERR?</code></td><td><i>0, “No error”</i></td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">None</td></tr>
</table>

SYSTem:VERSion?

<table>
<tr><td><b>Purpose</b></td><td colspan="2">Queries the SCPI version number the VM7004 complies with</td></tr>
<tr><td><b>Type</b></td><td colspan="2">SCPI required command</td></tr>
<tr><td><b>Command Syntax</b></td><td colspan="2">None - Query Only</td></tr>
<tr><td><b>Command Parameters</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>*RST Value</b></td><td colspan="2">N/A</td></tr>
<tr><td><b>Query Syntax</b></td><td colspan="2">SYSTem:VERSion?</td></tr>
<tr><td><b>Query Parameters</b></td><td colspan="2">None</td></tr>
<tr><td><b>Query Response</b></td><td colspan="2">Numeric ASCII value</td></tr>
<tr><td><b>Description</b></td><td colspan="2">The System Version query reports version of the SCPI standard with which the VM7004 complies.</td></tr>
<tr><td rowspan="2"><b>Examples</b></td><td><b>Command / Query</b></td><td><b>Response <i>(Description)</i></b></td></tr>
<tr><td>SYST:VERS?</td><td>1996.0</td></tr>
<tr><td><b>Related Commands</b></td><td colspan="2">None</td></tr>
</table>

ERROR MESSAGES

The following is a list of error messages that may be generated by the instrument:

-100, "Command error; A maximum of 256 resistance value may be specified"
-101, "Invalid character; Internal error, please contact factory"
-108, "Parameter not allowed; No channel is allowed"
-120, "Numeric data error; Valid attenuation values are 0.0 to 1.0"
-120, "Numeric data error; Valid channels are 1 to 4"
-120, "Numeric data error; Valid delay values are 0.00000 to 0.65535"
-120, "Numeric data error; Valid index values are 1 to 256"
-120, "Numeric data error; Valid length values are 1 to 256"
-120, "Numeric data error; Valid offset values are 1 to 256"
-120, "Numeric data error; Valid resistance values are 0 to 65532"
-120, "Numeric data error; Valid trigger lines are 0 to 7"
-120, "Numeric data error; Valid trigger lines are 0 to 7"
-211, "Trigger ignored; Trigger before delay finished"
-211, "Trigger ignored; Trigger source must be IMMediate"
-220, "Parameter error; Missing channel in ATTenuation command"
-220, "Parameter error; Valid channels are 1 or 3"
-220, "Parameter error; Valid channels are 1 to 4"
-221, "Settings conflict; Channel is already COMBined"
-221, "Settings conflict; Channel not in DIV mode"
-221, "Settings conflict; DOWNLOAD can only be used by the top module"
-221, "Settings conflict; When protect is on, only one channel is allowed"
-222, "Data out of range; Index is out of range"
-222, "Data out of range; Invalid channel"
-222, "Data out of range; Offset + length must be inside resistance list"
-222, "Data out of range; Resistance is outside valid range"
-222, "Data out of range; Valid attenuation values 0 to 1"
-222, "Data out of range; Valid sections are 0 to 3"
-223, "Too much data; Channel buffer full"
-223, "Too much data; Too many channels in channel list"
-350, "Queue overflow"

INDEX