GE
Measurement & Control

PACE

Pressure Automated Calibration Equipment

SCPI Remote Communications Manual - K0472 Revision A

PACE1000

PACE5000

PACE6000

Introduction

This technical manual provides SCPI protocol instructions for the remote control of the PACE Series indicators and controllers.

Safety

- The manufacturer has designed this product to be safe when operated using the procedures detailed in this manual. Do not use this product for any other purpose than that stated.
- This publication contains operating and safety instructions that must be followed to make sure of safe operation and to maintain the equipment in a safe condition. The safety instructions are either warnings or cautions issued to protect the user and the equipment from injury or damage.
- Use qualified* programming technicians and good engineering practice for all procedures in this publication.

Pressure

Do not apply pressure greater the maximum safe working pressure to the PACE Series.

Maintenance

The PACE Series must be maintained using the manufacturer's procedures and should be carried out by authorised service agents or the manufacturer's service departments.

Technical Advice

For technical advice contact the manufacturer or subsidiary.

***** A programming technician must have the necessary specialist knowledge of programming, technical knowledge and documentation to carry out the required work on the PACE Series.

Associated Documents:

A beginners Guide To SCPI by Barry Eppler, Published by Addison-Wesley Publishing Company Inc. for Hewlett Packard (ISBN 0-201-56350-9)

This issue of the manual

This manual details the SCPI commands and queries for the PACE1000, PACE5000 and PACE6000 pressure instruments. This issue removes erroneous code and clarifies the operation status group (3.3).

Table of Contents

List of Illustrations

List of Tables

Abbreviations

The following abbreviations are used in this manual; abbreviations are the same in the singular and plural.

a	Absolute	mV	millivolts
ALT	Altitude	MWP	Maximum working pressure
ASCII	American Standard Code for Information Interchange	No	Number
CAS	Calibrated airspeed	PACE	Pressure automated calibration equipment
e.g.	For example	PDCR	Pressure transducer
Fig.	Figure	PED	Pressure equipment directive
ft	Foot	PTX	Pressure transmitter
g	Gauge	ROC	Rate of climb
GPIB	General purpose interface bus	RS232	Serial communications standard
IDOS	Intelligent digital output sensor (GE product)	Rt CAS	Rate of Calibrated airspeed
i.e.	That is	Rt MACH	Rate of MACH
IEEE 488	Institute of Electrical and Electronic Engineers standard 488 (for programmable devices with a digital interface)	Rx	Receive data
kg	kilogram	SCPI	Standard commands for programmable instruments
kts	knots	SDS	Sales data sheet
m	Metre	Tx	Transmit data
mA	milliampere	UUT	Unit under test
max	Maximum	+ve	Positive
mbar	Millibar	-ve	Negative
min	Minute or minimum	°C	Degrees Celsius
mm	millimetre	°F	Degrees Fahrenheit

Pressure measurement units

The following units are used in this manual

ATM	atmosphere	KG/M2	kilogrammes per square metre
BAR	bar	KPA	kilo Pascals
CMH2O	centimetres of water at 20°C	LB/FT2	pounds per square foot
CMHG	centimetres of mercury	MH2O	metres of water
FTH2O	feet of water at 20°C	MHG	metres of mercury
FTH2O4	feet of water at 4°C	MMH2O	millimetres of water
HPA	hecto Pascals	MMHG	millimetres of mercury
INH2O	inches of water at 20°C	MPA	mega Pascals
INH2O4	inches of water at 4°C	PA	Pascals
INH2O60	inches of water at 60°F	PSI	pounds per square inch
INHG	inches of mercury	TORR	torr
KG/CM2	kilogrammes per square centimetre	MBAR	millibar

Code Definitions

The following codes are used in this manual.

ABOR	Abort
ADDR	Address
AVER	Average
ALT	Altitude
AMPL	Amplitude
ATOD	Analog to Digital
BAR	Barometer
BRID	Bridge
CAL	Calibration
CAT	Catalogue
CDIS	Cdisable (calibration disable)
CEN	Cenable (calibration enable)
CLS	Clear
COMM	Communicate
COMP	Compensate
COND	Condition
CONF	Configuration
CONT	Controller
CONV	Convert
CORR	Correction
COUN	Count
DEF	Define
DIAG	Diagnostic
DIOD	Diode
DISP	Display
EAV	Error in error queue
EFF	Effort
ENAB	Enable
EOI	End of input
ERR	Error
ESB	Summary bit from standard event
ESE	Event status enable
ESR	Event status register
EVEN	Event
FILT	Filter
FREQ	Frequency
FULL	Fullscale
GTL	Go to local
HEAD	Head
IDN	Identification
IMM	Immediate
INL	In limit
INP	Input
INST	Instrument
ISOL	Isolation
LEV	Level
LIM	Limit
LLO	Local lock out
LOG	Logical
LPAS	Low pass (filter)
MAV	Message available in output queue
MEAS	Measure
MSS	Summary bit after SRQ
NEGC	Negative Calibration
OFFS	Offset
OPC	Operational condition
OPER	Operation
OPT	Option
OSB	Summary bit from standard operations status register
OVER	Overshoot
OUTP	Output
PAR	Parity
PASS	Password
PERC	Percent
POIN	Points
PRES	Preset
PRES	Pressure
QUE	Queue
QUES	Questionable
RANG	Range
REF	Reference
RES	Resolution
RES	RESet
SAMP	Sample
SENS	Sense
SEPT	Set-point
SER	Serial
SOUR	Source
SPE	Speed
SRE	Service request enable
SRQ	Service request
STAR	Start
STB	Status register query
STAT	State
SYST	System
TIM	Time to set-point
UNIT	Unit of pressure
VAL	Value
VALV	Valve
VERS	Version
VOL	Volume

1 INTRODUCTION

1.1 General

The IEEE 488 and RS232 interfaces of the PACE Series provide remote control of the instrument from a suitable computer or controller. The SCPI protocol enables any instrument with a SCPI facility to be controlled using the same commands. The PACE Series use the full SCPI command set and the defined SCPI syntax.

The following sections describe and define each instrument command used by the PACE Series. The commands for the aeronautical option and the sensor calibration module option are described and defined in separate sections. Each section contains a quick reference structure of the relevant commands.

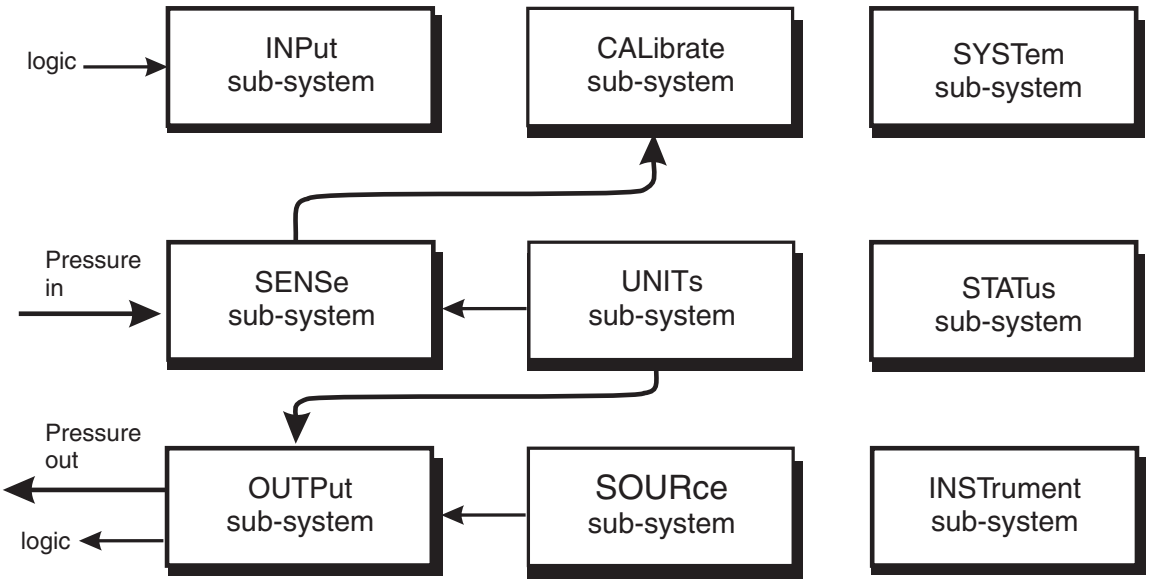


Figure 1-1 System Model

System Model

SCPI starts with a high-level block diagram of the measurement functions of the instrument. Each functional block is broken down into smaller block diagrams. SCPI contains a hierarchy of commands called a subsystem that maps directly to the hierarchy of the block diagram.

1.2 Remote/Local Operation

Most commands received over the SCPI interface automatically puts the PACE Series into remote control mode and disables the front panel touch-screen. Sending the LOC command returns the PACE Series to local control mode and enables the front panel touch-screen .

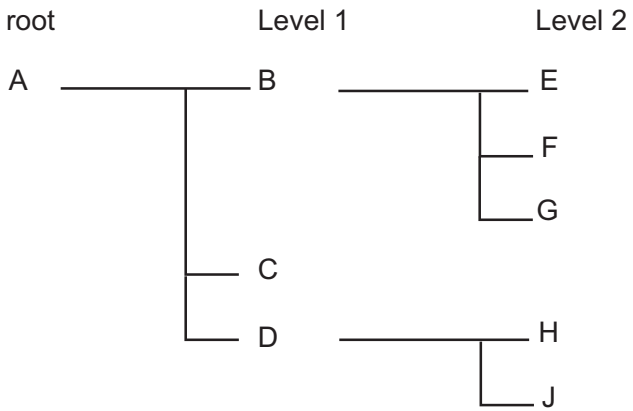
intentionally left blank

2 COMMAND STRUCTURE

This section describes the structure of the commands and data sent and received by a PACE Series Controller/Calibrator.

2.1 Notation

All SCPI commands are based on a hierarchical tree structure consisting of key words and parameters. Associated commands are grouped together under a common node in the hierachy.



In the command tree the command A is the root command. A tree pointer is used to decode the SCPI commands. At power-up the pointer goes to the root command.

2.2 Message Terminators

All SCPI commands are terminated by line feed i.e., either <newline> (ASCII character, decimal 10), EOI for IEEE. After receiving a termination character the tree pointer returns to the root command.

Colon

A colon moves the current path down one level in the command tree, (e.g., the colon in SOURCE:PRESSURE specifies PRESSURE the is one level below SOURCE). When the colon is the first character of the command, it specifies that the next command is a root level command (e.g., :SOURCE specifies that SOURCE is a root level command).

Semicolon

A semicolon separates two commands in the same message without changing the tree pointer.

(e.g., with reference to the tree):A:B:E;F;G

This equivalent to sending three messages:

:A:B:E
:A:B:F
:A:B:G

Commas
If a command requires more than one parameter, separate adjacent parameters by using a comma. A comma does not affect the tree pointer.
(e.g.) :SYSTEM:TIME 10,25,30

To execute a command the full path to the command must be specified:
(e.g.) :OUTPut:STATe ON

This turns the pressure controller on.

Note:
There must be a space between the command words and the parameter. In the above example there is a space between :STATe and ON.

SCPI commands are not case sensitive and may have a short form.
(e.g.) :OUTP is the short form of OUTPUT.

Some nodes can be the default node and these key words are optional when programming the command. The instrument processes the command, with the same effect, with or without the option node. In this manual [] enclose [default notes].

(e.g.) :SOURce[:PRESsure:][:LEVel][:IMMediate][:AMPlitude] 5.0

can be sent as

:SOURce:PRESsure:LEVel 5.0

or

:SOURce 5.0

This sets the set-point to 5.0

2.3 Program Headers

Program headers are keywords that identify a command, instruments accept both upper and lower case characters in a program header. There are two types of program header, common command headers and instrument control headers; each header can be a command or a query.

Common Command and Query Headers

The common command and query program header syntax, specified in IEEE 488.2, are defined as follows:

Command

*<PROGRAM MNEMONIC>

Query

*<PROGRAM MNEMONIC>?

Instrument Control Command and Query Headers

The instrument control command and query program header syntax controls and extracts data from the instrument as follows:

Command

:<MNEMONIC>

:<MNEMONIC> <PARAMETER>

Query

:<MNEMONIC>?

Instrument command headers can have a numeric suffix to identify each of several cases of the same header; the numeric suffix applies to both the long and short forms. All commands headers without a numeric suffix assume the value 1.

e.g.,

:OUTPut:LOGic1?

is the same as

:OUTPut:LOGic:?

Queries

A query is a program header with an attached question mark character (?). On receiving a query, the current settings are loaded in the output buffer. A query does not affect the operation or set-up of the instrument.

When the parameter of a command contains enumerated character data, both long form and short form are recognised. A query causes the return of data in the short form.

Querying numeric parameters causes the resulting data to be returned in the units selected by the instrument unless specified otherwise.

2.4 SCPI Data Types

A variety of data types can be sent to the instrument as parameters or sent out from the instrument as response data.

Decimal Data

All normal decimal expressions are accepted including optional signs, decimal point and scientific notation.

Note:

This includes floating point data.

The following are valid:

123
45.67
-2.6
4.6e-10
.76

A suffix multiplier can be added to the numeric value.

:SOUR 100 m

would set the programmable output to 0.1 units (100m units).

The multipliers supported are:

Mnemonic	Multiplier
A	1e-18
G	1e+9
K	1e+3
M	1e-3
T	1e+12

If a real value is sent to the instrument when an integer is expected, it will be rounded to an integer.

Integer Data

Integer data are whole numbers (containing no decimal places). A query of an integer value returns numbers containing no decimal places.

Note:

Integer values can be specified in binary, octal or hexadecimal formats using the suffix letters (upper or lower case) B, Q and H respectively.

e.g., #B1010 binary representation of 10
#Q71 octal representation of 57
#HFA hexadecimal representation of 250

Hexadecimal digits A-F can be in upper or lower case.

Enumerated Character Data

Enumerated characters are used for data that has a finite number of values; enumerated parameters use mnemonics to represent each valid setting.
The mnemonics have long and short forms just like command mnemonics.

Example:

:SOURce:PRESsure:SLEW:MODE MAXimum

selects the maximum rate mode.

A query of an enumerated parameter always returns the short form data in upper case.

Example:

:SOURce:PRESsure:SLEW:MODE?

queries rate mode, reply:

:SOUR:PRES:SLEW:MODE MAX

Boolean Data

Boolean data can only be one of two conditions; the numbers 1 and 0. Boolean can be “on” or “off”, queries return 1 or 0.

Example:

:OUTPut:STATe 1

A query of boolean data always returns 1 or 0.

String Data

String data can contain any of the ASCII characters. A string must start with a double "quote" (ASCII 34) or a single `quote` (ASCII 39) and end with the same character.

Note:
Characters in a string in either double "quote" or single `quote` are case sensitive.

Example: 1

:SOURCe[:PRESsure]:RANGe ‘2.00barg’

Example: 2

:SOURCe[:PRESsure]:RANGe “2.00barg”

selects the 2 bar g range.

A query of a string parameter always returns the string in double "quotes".

Intentionally left blank

3 STATUS SYSTEM

The status reporting system informs the external controller that an event has occurred. This information is in the form of a service request (SRQ) using IEEE 488 or an SRQ message using RS232.

The PACE Series uses status reporting as defined in IEEE 488.2 with the implementation of status registers.

The OPERation status registers have been implemented to comply with the SCPI protocol. These are registers where the individual bits are summary bits of the status of the instrument. Since the SCPI protocol does not include pressure instruments, bit 10 of both these registers are used as a pressure summary bit. This pressure summary bit is expanded to two, 16 bit registers (Bit 15 is not used and is always zero).

The only bit implemented in the Operation status register is bit 10 (summary of the pressure operation status).

A summary bit is the final output of a data structure, it is a single bit that shows the status of one or more related events in the instrument. The basic structure of a summary bit

- Condition register
- Event register
- Enable register
- Logical ANDing of the Event and Enable registers
- Summary bit that summarises the result using OR logic

Condition Register
This register shows the current status of the device. The condition register is constantly updated - the bits in the register are set or reset showing the current condition.

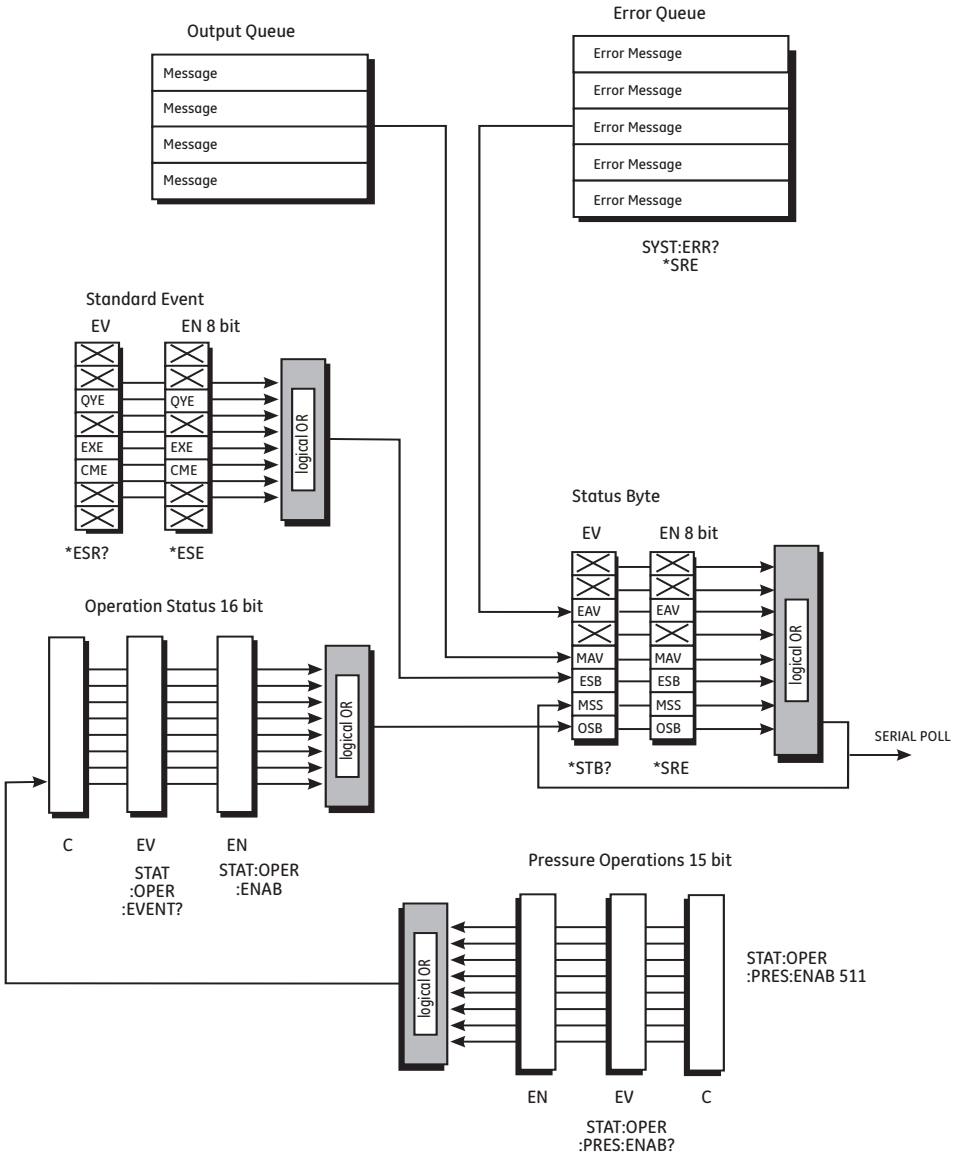
Event Register
The event register shows an event that occurs in the condition register (a condition bit goes from low to high). This condition change is stored and only reset when the event register is read or the *CLS command sent.

Enable Register
This register allows the results of the event register to pass through to the next cascaded register and enables the user to select the event that should generate the final SRQ event.

The status system implemented in the instrument is shown in the following diagram:

Note:

Initial values of registers are 0, with the queues empty.



Key:

C = Condition - variable values
EV = Latched values
EN = Bit mask

Figure 3-1 Status System

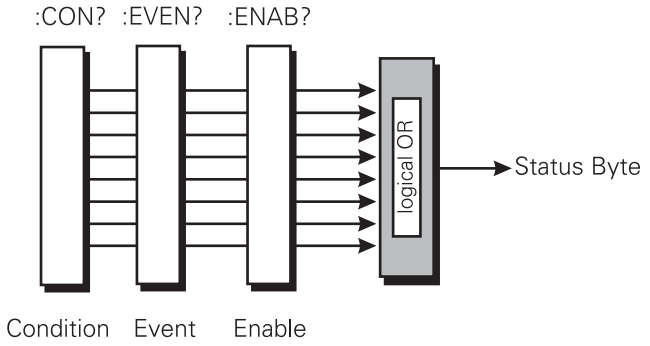
3.1 Output queue

The output queue is a text readable data queue that is read through the IEEE 488 talk command. The queue is cleared by reading all elements in it or by the *CLS command.

Every time a query has been successfully completed, the response, in a text readable format is placed at the end of the output queue. If the MAV bit in the "Status Byte" was previously cleared it will be set. The output queue can contain up to 256 characters. If there is not enough space in the output queue for a new message, the error -350, "Queue overflow" will be placed into the error queue and the most recent output message will be lost.

3.2 Standard event group

The standard event group are 8 bit registers that are read by the IEEE 488 standard commands. The event register is cleared by reading it; the event and enable registers are cleared by the *CLS command.



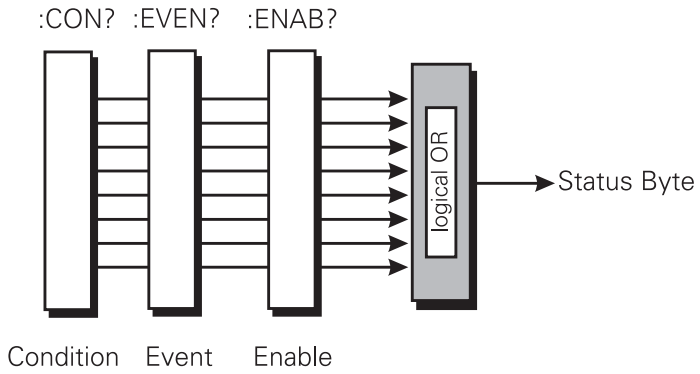
Bits within the standard event condition register are set by system errors and events. In addition to setting the status bits, a text message will be placed in the error queue. The ESB bit in the status byte sets if the associated bit in the event enable register is set. The enable register may be set through the *ESE command so that selected standard events cause the ESB bit to be set. The system events that set each bit are as follows:

Bit	Name	Description	Meaning/data
0	OPC	Not used	Reserved currently returns 0
1	RQC	Not used	Reserved currently returns 0
2	QYE	-400 to -499	Query errors
3	DDE	Not used	Reserved currently returns 0
4	EXE	-200 to -299	Execution errors
5	CME	-100 to -199	Command errors
6	URQ	Not used	Reserved currently returns 0
7	PON	Not used	Reserved currently returns 0

Table 3-1 Standard Event Register

3.3 Operation status group

The operation status group are 16 bit registers that are read by the STAT:OPER:PRES commands. The event register is cleared by reading it; the event and enable registers are cleared by the *CLS command.



When a standard operation condition occurs an appropriate bit is set in the condition register (this clears when the condition no longer exists). The bit is then latched in the event register. If the associated bit in the enable register is set, the OPR bit in the status byte sets. The enable register may be set through the STAT:OPER:PRES:ENAB command so that only selected standard operation events cause the OPR bit to set.

Problems can occur with some IEEE 488 controllers reading 16 bit unsigned numbers. All registers in this group do not use bit 15. The enable bit cannot be set and when read returns 0. The condition register is defined as follows:

Vent complete

This signal occurs when the controller has been requested to vent and the vent has completed or timed out.

Range change complete

This signal occurs when the controller has been requested to perform a range change and the range change is complete.

In-Limits reached

This signal is set every time the controlled pressure is within the specified limits. The signal is only generated if the pressure has been within limits for a user defined wait time period.

Zero complete

This signal is generated when a manual or timed zero is complete. If the zero times out then this signal is also generated.

Range compare alarm (PACE1000 only)

This signal is generated when the range compare alarm is triggered during the range compare process.

Bit (1)	Data (2)	Bit (3)	Data (4)
0	Vent complete	1	Range change complete
2	In-limits reached	3	Zero complete
4	Auto-zero started	5	Fill time, timed-out
6	Reserved - returns 0	7	Range compare alarm
8	Switch contacts changed state	9	Reserved - returns 0
10	Reserved - returns 0	11	Reserved - returns 0
12	Reserved - returns 0	13	Reserved - returns 0
14	Reserved - returns 0	15	Reserved - returns 0

Table 3-2 Operation Status Register

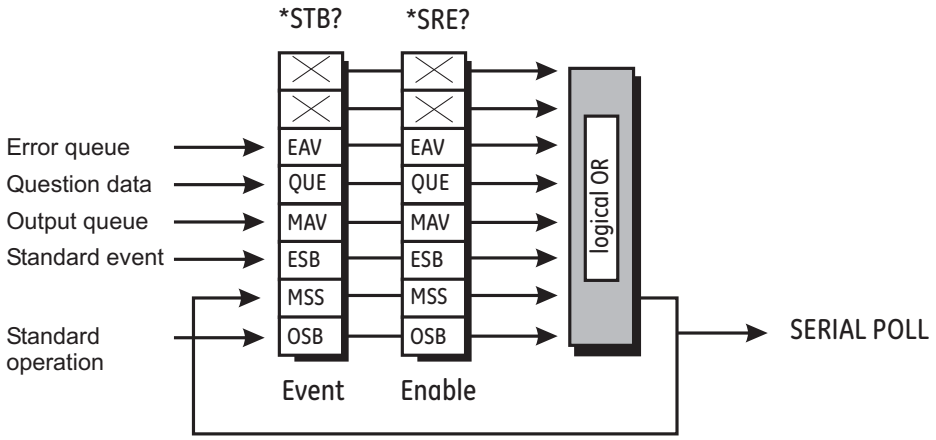
Auto zero started
When the controller is in the auto zero mode this signal indicates that the auto zero process has started. The zero complete signal indicates that the zero process has finished.

Fill timed out
If a set-point has been requested and the set-point cannot be achieved within the fill timeout time, the fill timed out signal is generated.

Switch contacts changed state
Every time the switch contacts used for performing a switch test change state this bit is set.

3.4 Status Byte group

The status byte group are 8 bit registers that are read by the IEEE 488 standard commands. The event register is cleared by reading it; the event and enable registers are cleared by the *CLS command.



Bits within the status byte are a summary of other data structures in the status system. These bits will become set if other parts of the status system indicates that they should do so (i.e., a message in the output queue or error queue or, a condition and enable set in a register pair).
If the associated bit in the status enable register is set, a serial poll is generated and bit 6 is set. The enable register may be set through the *SRE command so that only selected status bits cause a serial poll.

Note: *Bit 6 of the enable register is always set to 0.*

There are some small differences between * STB? and serial polling. Either method can be used to read the state of bits 0-5 and bit 7. The reading method is different for bit 6 when using *STB? and serial poll. In general, use serial polling inside interrupt service routines, not *STB?

Bit 2 - EAV sets when there is an error in the error queue. The :SYST:ERR? command has to be sent to retrieve the error. The error queue buffers a maximum of five errors. When no more errors are available the message “No Error” is returned.

Bit 4 - MAV sets when there is a message available in the output queue.

Bit 5 - ESB sets when a standard event has occurred in the Standard Event Register.

Bit 6 - MSS sets when an SRQ is generated - SRQ sets when both the Status byte and the Service Request Enable register are at logic 1 (AND function).

RS232 Specific
A service request (SRQ) produces the message::SRQ <value>

where:
<value> = the contents of the status summary byte.

The status system data structure sets each bit as follows:

Bit	Name	Description
0	-	Reserved currently returns 0
1	-	Reserved currently returns 0
2	EAV	Error in errror queque
3	-	Reserved currently returns 0
4	MAV	Messages available in output queque
5	ESB	Summary bit from standard event
6	MSS	Summary bit after service request - SRQ
7	OSB	Summary bit from standard operations status

Table 3-3 Status Byte Register

Example commands using the Status Byte and Status Byte Enable registers:

****SRE 16*** *Generate an SRQ interrupt when messages are available.*
****SRE?*** *Find out what events are enabled to generate SRQ interrupts.*
****STB?*** *Read and clear the Status Byte Enable register.*

IEEE 488 Specific
Bit 7 - OSB sets when the pressure operations register bit 10 changes state. The operations register is a 16 bit register only using bit 10. This bit is a summary of the pressure operations register.

Status reporting register structure

To set-up the status reporting system.

1 All status registers should be cleared by the command:
*CLS

2 The Pressure Operations Event register has to be set to enable the Pressure Operations Condition Register to send all the events to be reported; use the command:

:STAT:OPER:PRES:ENAB 511

The enabled events may also be read by the query:

:STAT:OPER:PRES:ENAB?

3 The Operation Status Event register must then be enabled to read bit 10 by the command:

:STAT:OPER:ENAB 1024

The enabled events may also be read by the query:

:STAT:OPER:ENAB?

4 The status request to enable the SRQ must then be set.

To enable only the Operation Status register (OSB) send the command:

*SRE 128

To enable the Operation Status register (OSB) and the Error Queue (EAV) send the command:

*SRE 132

This register may also be read by the query:

*SRE?

An event occurring generates an SRQ, the Status Byte should be queried to find the source of the event.

If bit 2 of the Status Byte Register is set the error queue can be read by the query:

:SYST:ERR?

Keep issuing this query until there are no more errors in the error queue. At this point, bit 2 of the Status Byte Register clears.

If bit 7 of the Status Byte Register is set the Pressure Operations event register can be read by the query:

:STAT:OPER:PRES?

returning the bits of events that have occurred. Reading this register clears it and the associated status bit (bit 7).

At any time the instantaneous status of the pressure system can be read by the query:

:STAT:OPER:PRES:COND?

3.5 Instrument Errors

Any instrument error that occurs, either programming errors or execution errors, is stored in an error queue which is separate from the main output queue. The errors can be read by issuing the following command query:

:SYST:ERR?

The error queue can hold up to five errors. Each time the error queue is queried the instrument responds with the next stored error in the queue. The response consists of an error number followed by a string describing the error. When the error queue is empty the instrument responds with:

:SYST:ERR 0, No error

Querying the error queue clears the storage location in the error buffer. If more than five errors occur, before being queried, the 'Queue overflow;Error queue overflow' message is placed into the error queue. All subsequent errors are lost until the error queue is cleared.

4 COMMAND AND QUERY SUMMARY

The following lists of all the SCPI commands and queries that apply to the instrument.

4.1 Command Structure

Some of the commands in the following summary are enabled at specific times and conditions, most can be enabled at any time. The command structure divides into subsystems as follows:

Command sub-system

:CALibration - calibration commands.

:DIAGnostic - instrument generated condition data.

INPut - switch input of the control module.

:INSTrument - instrument specific commands.

:OUTPut - controls the output pressure and logical outputs.

:SENSe - directs the instrument to measure selected parameters.

:SOURce - the commands that control the pressure outputs.

:STATus - instrument state.

:SYSTem - errors and SCPI version.

:UNIT - sets the units for the instrument.

Common SCPI commands - three letter commands, prefixed by *.

Instrument control commands - three letter commands, prefixed by **:**.

Command and Query Details

This section describes each command in detail including parameters passed to it and response data returned. The general short form command is shown at the top of each page.

The following information is then given:

Applicability	-	A list of instruments that accepts and responds to the command or query.
Command Syntax	-	The upper case represents the short form command.
Parameter	-	Type: DECIMAL, INTEGER, ENUMERATED CHARACTER, BOOLEAN or STRING.
Short form	-	The short alternative for the command to be effective.
Function	-	Basic function of the command.
Default	-	The default value or the maximum and minimum values where appropriate.
Query Syntax	-	The upper case represents the short form query command.
Parameter	-	Type: DECIMAL, INTEGER, ENUMERATED CHARACTER, BOOLEAN or STRING.
Short form	-	The short alternative for the query to be effective.
Function	-	Basic function of query command.
Response	-	Data returned by the instrument following the query command.

Description

Details of the command and query with any conditions of use and any related commands.

Note:

Many of the command descriptions contain an example code: sent (Tx) to the instrument and the data received (Rx) from the instrument.

Dual module instruments

Sending and receiving to a module must include either number 1 or 2 after the first part of the command or query. Without the module number, a dual module instrument defaults to module 1.

Example

Long form	Short form
Tx> :SOURce2:PRESsure:EFFort?	Tx> :SOUR2:PRES:EFF?
Rx> :SOUR2:PRES:EFF -0.2342882	Rx> :SOUR2:PRES:EFF -0.2342882

CALibration

The CALibration subsystem enables the calibration of the transducers and the rate control system, refer to the user manual for further details.

:CAL:PRES:POIN

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:
Defaults:

Query Syntax

:CALibration[x]:[PRESsure]:POINts?

where: x (module) = 1 or 2 (default - 1).
Short form: CAL:POIN?
Function: Gets the number of calibration points
Response: 3

Description

Valid only when calibration is enabled, this queries the number of calibration points.

Example

Tx> :CALibration2:PRESsure:POINts?
Rx> :CAL2:PRES:POIN 3

:CAL:PRES:ACC

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:CALibration[x]:[PRESsure]:ACCept

where: x (module) = 1 or 2 (default - 1)
Parameter: Integer 1

Short form: :CAL[x]:ACC
Function: Accepts calibration values
Defaults: no default value

Query Syntax

n/a

Short form:
Function:
Response:

Description

Valid only when calibration is enabled, this command accepts the calibration values entered in the calibration process.

:CAL:PRES:ABOR

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:CALibration:[PRESsure]:ABORt

Parameter: None

Short form: :CAL[x]:ABOR
Function: Aborts calibration values
Defaults: no default value

Query Syntax

n/a

Short form:
Function:
Response:

Description
Aborts calibration.

:CAL:PRES:VAL

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:CALibration[x]:[PRESsure]:VALue[y]

where: x (module) = 1 or 2 (default - 1). y = 1, 2 or 3 (pressure point) (default - 1).
Parameter: <decimal>

Short form: :CAL[x]:VAL[y]
Function: Enables calibration value to be entered.
Defaults:

Query Syntax

:CALibration[x]:[PRESsure]:VALue[y]?

where: x (module) = 1 or 2 (default - 1). y = 1, 2 or 3 (pressure point) (default - 1).
Short form: :CAL[x]:VAL[y]?
Function: Queries calibration point y value of module x.
Response: Returns pressure value for VALue[y] of module [x].

Description

Valid only when calibration is enabled. This command enables a calibration value to be entered during the calibration process. The query gets the calibration value.

y	3 Calibration points
1	lower pressure
2	middle pressure
3	higher pressure

Example for a 2 bar unit

Tx CAL:PRES:VAL1?
Rx CAL:PRES:VAL1 -0.9
Tx CAL:PRES:VAL2?
Rx CAL:PRES:VAL2 0
Tx CAL:PRES:VAL3?
Rx CAL:PRES:VAL3 2.0

:CAL:PRES:ZERO:VALV

Caution: Opening the zero valve with high pressure in the system can cause damage to the equipment. Reduce the system pressure and make sure the controller is OFF before opening the zero valve.

Applicability: PACE5000, PACE6000

Command Syntax

:CALibration:[PRESsure]:ZERO:VALVe[STATe]

where: x (module) = 1 or 2 (default - 1)
Parameter: <boolean>

Short form: :CAL:ZERO:VALV
Function: Opens and closes zero valve.
Default: 0

Query Syntax

:CALibration:[PRESsure]:ZERO:VALVe[STATe]?

where: x (module) = 1 or 2 (default - 1)
Short form: CAL:ZERO:VALV?
Function: Queries state of valve.
Response: 1 - open
0 - close

Description

This command is used to open and close the zero valve. The query gets the state of the zero valve - open or close.

Example

TX> :CAL:PRES:ZERO:VALV 1
TX> :CAL:PRES:ZERO:VALV:STAT 1
Either of above two commands opens the zero valve.

TX> :CAL:PRES:ZERO:VALV: 0
TX> :CAL:PRES:ZERO:VALV:STAT 0
Either of above two commands close the zero valve, switch OFF the controller.

Either: TX> :CAL:PRES:ZERO:VALV?
RX> :CAL:PRES:ZERO:VALV 1
Or TX> :CAL:PRES:ZERO:VALV:STAT?
RX> :CAL:PRES:ZERO:VALV:STAT 1

Either: TX> :CAL:PRES:ZERO:VALV?
RX> :CAL:PRES:ZERO:VALV 0
Or TX> :CAL:PRES:ZERO:VALV:STAT?
RX> :CAL:PRES:ZERO:VALV:STAT 0
Gets the condition of the zero valve.

:CAL:PRES:ZERO:AUTO

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:CALibration[x]:[PRESsure]:ZERO:AUTO <Boolean>

where: x (module) = 1 or 2 (default - 1)

Parameter: <boolean>

0	-	aborts a zero process
1	-	starts a zero process

Short form: :CAL:ZERO:AUTO
Function: Pressure zeroing
Default: 0

Query Syntax

:CALibration[x]:[PRESsure]:ZERO:AUTO?

where: x (module) = 1 or 2 (default - 1)

Short form: :CAL:ZERO:AUTO?
Function: Query progress of zero
Response:

0 - Zero complete or not in progress.
1 - Zero in progress.

Description

This command starts or aborts a zero process. The progress of the zero can be monitored by using the query.

:CAL:PRES:ZERO:TIME

Applicability: PACE5000, PACE6000

Command Syntax

:CALibration[x]:[PRESsure]:ZERO:TIME

where: x (module) = 1 or 2 (default - 1)
Parameter: Numeric

Short form: :CAL:ZERO:TIME
Function: Sets timed zero in hours.
Default: -

Query Syntax

:CALibration[x]:[PRESsure]:ZERO:TIME?

where: x (module) = 1 or 2 (default - 1)
Short form: :CAL:ZERO:TIME?
Function: Queries timed zero.
Response: Integer number in hours.

Description

This command sets the time between zeroing function. The query sends the setting in hours.

:CAL:PRES:ZERO:TIME:STAT

Applicability: PACE5000, PACE6000

Command Syntax

:CALibration[x]:[PRESsure]:ZERO:TIME:STATe

where: x (module) = 1 or 2 (default - 1)
Parameter: <<boolean>>

Short form: :CAL:ZERO:TIME:STAT1
Function: Sets timed zero on/off.
Default: 0

0 - sets timed zero to OFF
1 - sets timed zero to ON

Query Syntax

:CALibration[x]:[PRESsure]:ZERO:TIME:STATe?

where: x (module) = 1 or 2 (default - 1)
Short form: :CAL:ZERO:TIME:STAT?
Function: Queries the status of the timed zero ON or OFF.
Response:

:CAL:ZERO:TIME:STAT 1
or
:CAL:ZERO:TIME:STAT 0

Description

This command sets the time period for zero on or off. This query gets the status of the timed zero (on or off).

DISPlay

The DISPlay subsystem shows the state of the display window.

:DISP:WIND

Applicability: PACE1000

Command Syntax

n/a

Parameter:

Short form:
Function:
Default:

Query Syntax

:DISP[x]:WIND?

where: x window index is 1 to 3 (top window is 1)
Short form: :DISP[x]:WIND?
Function: Asks for window for an allocated range.
Response: Disp 1, 2 or 3 and the value in the selected units.

Description

This query returns the allocated range to a particular display window.

Example:

TX> :Disp1:Wind?
RX> :DISP:WIND "979.44"

TX> :Disp2:Wind?
RX> :DISP2:WIND "993.55

INPut

The INPut subsystem shows the state of the logical inputs.

:INP:LOG

Applicability: PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:
Default:

Query Syntax

:INPut[x]:LOGic?

where: x (module) = 1 or 2 (default - 1),

Short form:	:INP:LOG?		
Function:	Asks for state of switch input within the controller module.		
Response:	first parameter	-	0 = OFF, 1 = ON
	second parameter	-	measured pressure at the time of switching (snapshot in current pressure units).

Description

This query returns the state of the switch input within the module and the pressure at time of switching operations.

Example
:INP:LOG?
:INP:LOG 0, 0.8321209
Current logic OFF, pressure was 0.8321209 in current pressure units when the module was switched to OFF condition.

:INP:LOG:STAT

Applicability: PACE1000, PACE5000, PACE6000

Note: *Only applies to the PACE1000 when the VFC option is fitted.*

Command Syntax

n/a

Parameter:

Short form:
Function:
Default:

Query Syntax

:INPut[x]:LOGic:STATe?

where: x (module) = 1 or 2 (default - 1),
Short form: :INP:LOG:STAT?
Function: Asks for state of switch input.
Response: 0 = OFF, 1 = ON

Description

This query returns the state of the switch input.

INSTrument

The INSTrument subsystem gets information about the configuration of the instrument .

:INST:CAT:ALL

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:INSTrument:CATalog[x]:[ALL]?

where: x (module) = 1 or 2 (default - 1)
Short form: :INST:CAT?
Function: Query all ranges fitted
Response: A list of comma separated strings of ranges fitted.

Description

This query returns all the ranges fitted to the instrument. The reply is a comma separated list of strings representing each range.

Example for a dual module instrument with a 7.00 barg module and a 3.50barg module both have a barometric option fitted.
TX> :INST:CAT1?
RX> :INST:CAT: "7.00barg","BAROMETER","8.00bara"
TX> :INST:CAT2?
RX> :INST:CAT: "3.50barg","BAROMETER","4.50bara"

:INST:CONT:SENS

Applicability: PACE5000 and PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:INSTrument(x):CONTroller(y]:SENse(z)

:Inst(x):Cont(y):Sens(z)?

INST index(x) where x must be 0 or 'blank'
Use the CONT index(y) to determine which control module 1 (or blank) or 2. Both :Inst:Cont and :Inst:Cont1 will address control module 1, whilst :Inst:Cont2 addresses only control module 2.
Use the SENS index(z) to determine which sensor as below.

Description

:INST:CONT:SENS or :INST:CONT:SENS1 "2.00barg" returns Control module 1 full-scale range sensor
:INST:CONT:SENS2 "7.00barg" returns the +ve port source pressure sensor full-scale
:INST:CONT:SENS3 "1.00barg" returns the –ve port vacuum pressure sensor full-scale
:INST:CONT:SENS4 "BAROMETER"
:INST:CONT:SENS5 "" Reference sensor range if fitted
:INST:CONT:SENS6 "" Not used
:INST:CONT:SENS7 "3.00bara" Returns the pseudo absolute range
:INST:CONT:SENS8 "" Not used

Example
TX> :Inst:Cont1:Sens2?
RX> :INST:CONT1:SENS2 "7.00barg" 1st control module vacuum sensor
TX> :Inst:Cont2:Sens4?
RX> :INST:CONT2:SENS4 "BAROMETER" 2nd control module barometric sensor

INST:DISP

Applicability: PACE6000

Command Syntax

:INSTrument:DISPlay

Parameter: Enumerated character single or dual

Short form: :INST:DISP
Function: Sets the single or dual display mode.

Query Syntax

:INSTrument:DISPlay?

Short form: :INST:DISP?
Function: Queries the display setting - single or dual.
Response: Enumerated character single or dual

Description

This command sets the single or dual display mode. The query returns display setting single or dual display mode.

Example:

```
TX> :INST:DISP?          //ask display mode
RX> :INST:DISP SING      //in single mode.
TX> :INST:DISP DUAL      //change to dual display mode
TX> :INST:DISP?          //ask display mode
RX> :INST:DISP DUAL      //in dual display mode
TX> :INST:DISP SINGLE    //change to single display mode
TX> :INST:DISP?          //ask display mode
RX> :INST:DISP SING      //change to single display mode.
```

In the single display mode:

```
TX> :INST:CAT?
RX> :INST:CAT "7.00barg","BAROMETER","8.00bara","3.50barg","4.50bara"
TX> :INST:CAT2?
RX> :INST:CAT2 "7.00barg","BAROMETER","8.00bara","3.50barg","4.50bara"
```

:INST:LIM

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INSTrument:[LIMits][x]?

where: x (sensor number) = 1, 2. . . 8 (default - 1).

Short form: :INST?

Function: Queries the upper and lower full-scale limits of the fitted sensor ranges. The index number x is used to index into the list of available ranges.

x	Sensor PACE5000, PACE6000	Sensor PACE1000
1	Module 1: control sensor	Internal sensor1
2	Module 1: source pressure +ve	Internal sensor 2
3	Module 1: source pressure -ve	Internal sensor 3
4	Module 1: barometric range (optional)	External IDOS-1
5	Module 2: control sensor	External IDOS-2
6	Module 2: source pressure +ve	External IDOS-3
7	Module 2: source pressure -ve	-
8	Module 2: barometric range (optional)	-

Response: A string representing the range, a number representing the upper full-scale and a number representing the lower full-scale.

Description

This query returns a string representing the range, a number representing the upper full-scale and lower full-scale.

Example for a dual module instrument, both modules with the barometric option fitted.

TX> :INST:LIM?
RX> :INST:LIM "7.00barg", 7350.0000000, -1100.0000000
TX> :INST:LIM1?
RX> :INST:LIM "7.00barg", 7350.0000000, -1100.0000000
TX> :INST:LIM2?
RX> :INST:LIM2 "20.00barg", 21000.0000000, -1100.0000000
TX> :INST:LIM3?
RX> :INST:LIM3 "2.00barg", 2100.0000000, -1100.0000000
TX> :INST:LIM4?
RX> :INST:LIM4 "BAROMETER", 1207.5000000, 825.0000000
TX> :INST:LIM5?
RX> :INST:LIM5 "3.50barg", 3675.0000000, -1100.0000000
TX> :INST:LIM6?
RX> :INST:LIM6 "20.00barg", 21000.0000000, -1100.0000000
TX> :INST:LIM7?
RX> :INST:LIM7 "10.00barg", 10500.0000000, -1100.0000000
TX> :INST:LIM8?
RX> :INST:LIM8 "BAROMETER", 1365.0000000, 38.5000000

INST:MAC

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:INSTrument:MACaddress?

Short form: :INST:MAC?
Function: Queries the MAC internet address.
Response: Address string

Description

This command returns the address string.
Example:
TX> :INSTrument:MACaddress?
RX> :INST:MAC “00-D0-1C-0B-1B-1A”

INST:SENS

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:INST:SENSor?

Short form: :INST:SENS?
Function: Queries the transducer range (sensor).
Response: Returns sensor range.

Description

This command returns the transducer range.

Example:
TX> :INST:SENS?
RX> :INST:SENS "7.00barg"
TX> :INST:SENS1?
RX> :INST:SENS "7.00barg"
TX> :INST:SENS2?
RX> :INST:SENS2 "20.00barg"
TX> :INST:SENS3?
RX> :INST:SENS3 "2.00barg"
TX> :INST:SENS4?
RX> :INST:SENS4 "BAROMETER"
TX> :INST:SENS5?
RX> :INST:SENS5 "3.50barg"
TX> :INST:SENS6?
RX> :INST:SENS6 "20.00barg"
TX> :INST:SENS7?
RX> :INST:SENS7 "10.00barg"
TX> :INST:SENS8?
RX> :INST:SENS8 "BAROMETER"

INST:SENS:CALD

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:INST:SENS[x]:CALD[y]?

where x (sensor number) = 1, 2. . . . . 8 (default -1) y = 1, 2, 3 . . . . .10 (y times last calibration date)
Short form: :INST:SENS:CALD?
Function: Queries sensor calibration dates.
Response: Returns sensor [x] with [y] calibration date:
where x is the sensor number.
x = 1, 2. 8

x	Sensor PACE5000, PACE6000	Sensor PACE1000
1	Module 1: control sensor	Internal sensor1
2	Module 1: source pressure +ve	Internal sensor 2
3	Module 1: source pressure -ve	Internal sensor 3
4	Module 1: barometric range (optional)	External IDOS-1
5	Module 2: control sensor	External IDOS-2
6	Module 2: source pressure +ve	External IDOS-3
7	Module 2: source pressure -ve	-
8	Module 2: barometric range (optional)	-

where y is the y^{th} calibration date for sensor x.
y = 1, 2, 3 . . . . .10 calibration dates stored for each sensor.

Description

This query returns the following:

Example

TX> :INST:SENS5:CALD1?
RX> :INST:SENS5:CALD 2009, 11, 21

TX> :INST:SENS5:CALD2?
RX> :INST:SENS5:CALD2 2009, 11, 17

This shows that the module 2 control sensor was last calibrated on 21st November 2009 and the previous calibration was on 17th November 2009.

:INST:SENS:FULL

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INSTrument:SENSe[x]:FULLscale?

where: x = 1, 2...8 is the sensor number. It defaults to 1.

Short form: :INST:SENS:FULL?
Function: Queries sensor full-scale value.
Response: Returns the full-scale value of the selected sensor.

x	SensorPACE5000, PACE6000	PACE1000	x	SensorPACE5000, PACE6000	PACE1000
1	Module 1: control sensor	Internal sensor1	5	Module 2: control sensor	External IDOS-2
2	Module 1: source pressure +ve	Internal sensor 2	6	Module 2: source pressure +ve	External IDOS-3
3	Module 1: source pressure -ve	Internal sensor 3	7	Module 2: source pressure -ve	-
4	Module 1: barometric range (optional)	External IDOS-1	8	Module 2: barometric range (optional)	-

Description

This query returns the following:

:INST:SENS:FULL?
:INST:SENS:FULL 7.0000000
Default - module 1 control sensor full-scale.

:INST:SENS1:FULL?
:INST:SENS:FULL 7.0000000
Returns module 1 control sensor full-scale.

:INST:SENS2:FULL?
:INST:SENS2:FULL 20.0000000
Returns module 1 +ve source sensor full-scale.

:INST:SENS3:FULL?
:INST:SENS3:FULL 2.0000000
Returns module 1 -ve source sensor full-scale.

:INST:SENS4:FULL?
:INST:SENS4:FULL 1.1500000
Returns module 1 barometric sensor full-scale.

:INST:SENS5:FULL?
:INST:SENS5:FULL 3.5000000
Returns module 2 control sensor full-scale.

:INST:SENS6:FULL?
:INST:SENS2:FULL 20.0000000
Returns module 2 +ve source sensor full-scale.

:INST:SENS7:FULL?
:INST:SENS7:FULL 10.0000000
Returns module 2 -ve source sensor full-scale.

:INST:SENS8:FULL?
:INST:SENS8:FULL 1.1500000
Returns module 2 barometric sensor full-scale.

INST:SENS:NEG

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INSTrument:SENSor[x]:NEGCal?

where: x = 1, 2...8 is the sensor number. It defaults to 1.

Short form: :INST:SENS:NEG?
Function: Queries sensor negative calibration.
Response: <boolean>: 0 - negative calibration not supported
1 - negative calibration supported.

x	SensorPACE5000, PACE6000	PACE1000	x	SensorPACE5000, PACE6000	PACE1000
1	Module 1: control sensor	Internal sensor1	5	Module 2: control sensor	External IDOS-2
2	Module 1: source pressure +ve	Internal sensor 2	6	Module 2: source pressure +ve	External IDOS-3
3	Module 1: source pressure -ve	Internal sensor 3	7	Module 2: source pressure -ve	-
4	Module 1: barometric range (optional)	External IDOS-1	8	Module 2: barometric range (optional)	-

Description

This command query returns sensor negative calibration supported or not supported. Used in DPI515 emulation; for PACE5000/6000 this query always returns '1'.

:INST:SENS:NEGC?
:INST:SENS:NEGC 1
Default - module 1 control sensor NEGcal.

:INST:SENS1:NEGC?
:INST:SENS:NEGC 1
Returns module 1 control sensor NEGcal.

:INST:SENS2:NEGC?
:INST:SENS2:NEGC 1
Returns module 1 +ve source sensor NEGcal.

:INST:SENS3:NEGC?
:INST:SENS3:NEGC 1
Returns module 1 -ve source sensor NEGcal.

:INST:SENS4:NEGC?
:INST:SENS4:NEGC 1
Returns module 1 barometric sensor NEGcal.

:INST:SENS5:NEGC?
:INST:SENS5:NEGC 1
Returns module 2 control sensor NEGcal.

:INST:SENS6:NEGC?
:INST:SENS2:NEGC 1
Returns module 2 +ve source sensor NEGcal.

:INST:SENS7:NEGC?
:INST:SENS7:NEGC 1
Returns module 2 -ve source sensor NEGcal.

:INST:SENS8:NEGC?
:INST:SENS8:NEGC 1
Returns module 2 barometric sensor NEGcal.

INST:SENS:READ

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INSTrument:SENSor[x]:READing?

Short form: :INST:SENS:READ?
Function: Queries sensor zero reading.
Response: Gets reading of selected sensor [x].
where x is the sensor number.
x = 1, 2. 8

Description

This query gets the filtered reading followed by the instantaneous reading from the selected sensor.

INST:SENS:SN

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INSTrument:SENSor[x]:Serial Number?

Short form: :INST:SENS:SN?
Function: Queries the sensor serial number.
Response: Gets the serial number of selected sensor [x].
where x is the sensor number.
x = 1, 2. 8

Description

This query gets the serial number of selected sensor.

:INST:SENS:ZERO

Applicability: PACE1000

Command Syntax

INSTrument:SENSor:ZERO

where: x = 1, 2...8 is the sensor number.
Parameter: <integer> 1, 2...8
Short form: INST:SENS:ZERO
Function: Performs a zero on the selected sensor specified in the sensor index.

Query Syntax

:INSTrument:SENSor[x]:ZERO?

where: x = 1, 2...8 is the sensor number. It defaults to 1.
Short form: :INST:SENS:ZERO?
Function: Queries sensor zero condition.
Response: Gets zero condition [y] of selected sensor [x].
where x is the sensor number.

x = 1, 2. 8

y =

0	Zero ok
1	Zero in progress
2	Zero timed out
3	Zero outside limits
4	Zero aborted

Description

The command selects a sensor and performs a zero. The query gets the zero condition after a zero has been performed on the selected sensor.

:INST:SN

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INSTrument:SN[x]?

where: x = 1, 2…7 is the hardware circuit number. It defaults to 1.
Short form: :INST:SN?
Function: Used to query the serial numbers of the instrument hardware and options.
Response: Integers representing serial numbers.

Description

This query returns the serial numbers of the hardware installed.

x	Hardware circuit	x	Hardware circuit
1	User interface (instrument main board)	5	Analogue Output 2 †
2	Controller 1 *	6	VFC 1 †
3	Controller 2 *	7	VFC 2 †
4	Analogue Output 1 †		

Notes:

1. *Depending on future instrument development, additional numbers may be included.*
2. ** not applicable to PACE indicators.*
3. *† option must be fitted.*

Example

TX> :INST:SN?
RX> :INST:SN 68795

TX> :INST:SN1?
RX> :INST:SN 68795

TX> :INST:SN2?
RX> :INST:SN2 2803347

TX> :INST:SN3?
RX> :INST:SN3 65795

TX> :INST:SN4?
RX> :INST:SN4 68884

TX> :INST:SN5?
RX> :INST:SN5 0

TX> :INST:SN6?
RX> :INST:SN6 88704

TX> :INST:SN7?
RX> :INST:SN7 0

In this example, 'RX> :INST:SN5 0' and 'RX> :INST:SN7 0' mean the 2nd Analogue Output and 2nd VFC board are not installed.

:INST:TASK

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:INSTrument:TASK

Parameter: Enumerated

Short form: :INST:TASK
Function: Sets the task

Query Syntax

:INSTrument:TASK?

Short form: :INST:TASK?
Function: Used to query the task set.
Response: Enumerated character representing task.

Description

This command sets the task and the query returns the task setting.

	PACE1000	PACE5000	PACE6000
AERONAUTICAL	¸		
AIRFIELD	¸		
AIRSPEEDLEAK	¸		
ALTLEAK	¸		
BAROGRAPH	¸		
BASIC	¸	¸	¸
BURSTTEST		¸	¸
DIVIDER		¸	¸
LEAKTEST	¸	¸	¸
PRESET		¸	¸
SWITCHTEST		¸	¸
TESTPROGRAM		¸	¸

:INST:TASK:AERO

Applicability: PACE1000, PACE6000

Command Syntax

:INSTrument:TASK:AERO:[RANGe]

Parameter:

Short form: :INST:TASK:AERO
Function: Sets the aero ranges: pressure, altitude, speed and MACH.

Query Syntax

:INSTrument:TASK:AERO:[RANGe]?

Short form: :INST:TASK:AERO?
Function: Used to query the task set.
Response: Integers representing task numbers.

Description

The command sets the aero ranges: pressure, altitude, speed and MACH. This query returns the aero range setting.

:INST:UNIT

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INStrument:UNIT?

where: x = 1, 2...32 is the unit number. It defaults to 1.
Short form: :INST:UNIT?
Function: Used to query the units of measurement available. This does not query the current unit in use (refer to the 'UNIT' family of commands/queries).
Response: Enumerated character string.

Description

This command sets the units of measurement available to the instrument.

TX> :INST:UNIT1? RX> :INST:UNIT MBAR	TX> :INST:UNIT12? RX> :INST:UNIT12 KG/M2	TX> :INST:UNIT23? RX> :INST:UNIT23 INH2O_4
TX> :INST:UNIT2? RX> :INST:UNIT2 BAR	TX> :INST:UNIT13? RX> :INST:UNIT13 MMH2O_4	TX> :INST:UNIT24? RX> :INST:UNIT24 INH2O_20
TX> :INST:UNIT3? RX> :INST:UNIT3 PA	TX> :INST:UNIT14? RX> :INST:UNIT14 CMH2O_4	TX> :INST:UNIT25? RX> :INST:UNIT25 INH2O_60
TX> :INST:UNIT4? RX> :INST:UNIT4 HPA	TX> :INST:UNIT15? RX> :INST:UNIT15 MH2O_4	TX> :INST:UNIT26? RX> :INST:UNIT26 FTH2O_4
TX> :INST:UNIT5? RX> :INST:UNIT5 KPA	TX> :INST:UNIT16? RX> :INST:UNIT16 MMH2O_20	TX> :INST:UNIT27? RX> :INST:UNIT27 FTH2O_20
TX> :INST:UNIT6? RX> :INST:UNIT6 MPA	TX> :INST:UNIT17? RX> :INST:UNIT17 CMH2O_20	TX> :INST:UNIT28? RX> :INST:UNIT28 FTH2O_60
TX> :INST:UNIT7? RX> :INST:UNIT7 MMHG	TX> :INST:UNIT18? RX> :INST:UNIT18 MH2O_20	TX> :INST:UNIT29? RX> :INST:UNIT29 USER1
TX> :INST:UNIT8? RX> :INST:UNIT8 CMHG	TX> :INST:UNIT19? RX> :INST:UNIT19 TORR	TX> :INST:UNIT30? RX> :INST:UNIT30 USER2
TX> :INST:UNIT9? RX> :INST:UNIT9 MHG	TX> :INST:UNIT20? RX> :INST:UNIT20 ATM	TX> :INST:UNIT31? RX> :INST:UNIT31 USER3
TX> :INST:UNIT10?RX> :INST:UNIT10 INHG	TX> :INST:UNIT21? RX> :INST:UNIT21 PSI	TX> :INST:UNIT32? RX> :INST:UNIT32 USER4
TX> :INST:UNIT11? RX> :INST:UNIT11 KG/CM2	TX> :INST:UNIT22? RX> :INST:UNIT22 LB/FT2	

:INST:VERS

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:INSTrument:VERSion[x]?

where: x = 1, 2...15 is the software version. It defaults to 1.
Short form: :INST:VERS[x]?
Function: Queries the software versions of the instrument and options.
Response: Returns strings representing the software version:

Description

x	Software item	x	Software item
1	Instrument main code	9	Analogue output board 1 boot code
2	Instrument OS build	10	Analogue output board 2 main code
3	Instrument boot ROM	11	Analogue output board 2 boot code
4	Controller 1 main code	12	VFC board 1 main code
5	Controller 1 Boot code	13	VFC board 1 boot code
6	Controller 2 main code	14	VFC board 2 main code
7	Controller 2 Boot code	15	VFC board 2 boot code
8	Analogue output board 1 main code		

Example

TX> :Inst:Vers? RX> :INST:VERS "00.01.09"	TX> :Inst:Vers6? RX> :INST:VERS6 "02.00.29"	TX> :Inst:Vers12? RX> :INST:VERS12 "00.01.53"
TX> :Inst:Vers1? RX> :INST:VERS "00.01.09"	TX> :Inst:Vers7? RX> :INST:VERS7 "01.00.00"	TX> :Inst:Vers13? RX> :INST:VERS13 "00.00.40"
TX> :Inst:Vers2? RX> :INST:VERS2 "01.06.16"	TX> :Inst:Vers8? RX> :INST:VERS8 "00.01.53"	TX> :Inst:Vers14? RX> :INST:VERS14 ""
TX> :Inst:Vers3? RX> :INST:VERS3 "01.01.04"	TX> :Inst:Vers9? RX> :INST:VERS9 "00.00.40"	TX> :Inst:Vers15? RX> :INST:VERS15 ""
TX> :Inst:Vers4? RX> :INST:VERS4 "02.00.29"	TX> :Inst:Vers10? RX> :INST:VERS10 ""	
TX> :Inst:Vers5? RX> :INST:VERS5 "01.00.00"	TX> :Inst:Vers11? RX> :INST:VERS11 ""	

Note: *A return of an empty string means the corresponding option is not installed.*

OUTPut

The OUTPut subsystem turns the pressure controller on/off and controls the state of the logical outputs.

:OUTP:STAT

Applicability: PACE5000, PACE6000

Command Syntax

:OUTPut[x]:STATe <Boolean>

where: x (module) = 1 or 2 (default - 1)

Parameter: <boolean>

0	-	turn controller off
1	-	turns controller on

Short form: :OUTP
Function: Turn the pressure controller on/off
Default: 0

Query Syntax

:OUTPut:STATe?

where: x (module) = 1 or 2 (default - 1)

Short form: :OUTP?
Function: Asks for state of pressure controller
Response: 0 - controller off
1 - controller on

Description

Sets or queries the state of the pressure controller.

Example

TX> :OUTP:STAT?
RX> :OUTP:STAT 0
The module 1 controller currently turned off.

To turn the module 1 controller off:
TX> :OUTP:STAT OFF

To turn the module 1 controller on:
TX> :OUTP:STAT ON
TX> :OUTP:STAT?
RX> :OUTP:STAT 1

:OUTP:LOG

Applicability: PACE1000, PACE5000, PACE6000

Note: *The VFC option must be fitted.*

Command Syntax

:OUTP[x]:LOGic[y]

where: x (module) = 1 or 2 (default - 1) and y = 1, 2 or 3 represents the relay number (default - 1).

Parameter: <boolean>

OFF - turn relay [y] OFF

ON - turn relay [y] ON

[y] = 1, 2 or 3

Short form: :OUTP:LOG

Function: Turns relay ON and OFF

Default: Relay 1 of module 1

Query Syntax

where: x (module) = 1 or 2 (default - 1) and y = 1, 2 or 3 represents the relay number (default - 1).

:OUTP[x]:LOGic[y]?

Short form: :OUTP:LOG?

Function: Asks for relay ON/OFF status.

Response: 0 - relay OFF

1 - relay ON

Description

With the volt-free contact option installed and each relay set to 'communication' through the front panel control. The three relays can be queried and switched on and off. Omitting [x] or [y], the default of 1 is used.

These commands return an error message: "The option board is not configured correctly", if the volt-free contact option is installed but the relay not set to 'communication'.

Example

TX> :OUTP:LOG?
RX>:OUTP:LOG0 //relay1 of VFC 1 is currently OFF.

TX> :OUTP:LOG2 ON
TX>:OUTP:LOG2?
RX>:OUTP:LOG2 1 //relay2 of VFC 1 is now turned ON.

TX> :OUTP:LOG3?
RX>:OUTP:LOG3 0 // relay3 currently OFF

TX> :OUTP:LOG3 ON
TX> :OUTP:LOG3?
RX>:OUTP:LOG3 1 // relay3 ON

TX> :OUTP:LOG2 ON
TX>:OUTP:LOG2?
RX>:OUTP:LOG2 1 //relay2 of VFC 1 is currently ON.

TX> :OUTP:LOG ON
TX>:OUTP:LOG?
RX>:OUTP:LOG 1 //relay1 ON

TX> :OUTP:LOG2?
RX>:OUTP:LOG2 0 // relay2 currently OFF

If the relay is not set to 'communication' through front panel, it will not be switchable and an error message will be produced:
TX> :OUTP:LOG1?
TX>:SYST:ERR?
RX>:SYST:ERR -240, "The option board is not configured correctly"
If the VFC option is not installed, an error message will be produced:
TX> :OUTP2:LOG1?
TX>:SYST:ERR?
RX>:SYST:ERR -241, "Hardware missing", i.e. there is no 2nd VFC board installed in the PACE6000 instrument.

:OUTP:ISOL:STAT

Applicability: PACE5000, PACE6000

Command Syntax

:OUTPut[x]:ISOLation:STATe

where x = 1 or 2 is the module number (default - 1)

Parameter: <boolean>

0	OFF	-	turn isolation valve of the module [x] OFF
1	ON	-	turn isolation valve of the module [x] ON

Short form: :OUTP:ISOL:STAT
Function: Turns isolation valve ON and OFF
Default: -

Query Syntax

:OUTPut[x]:ISOLation:STATe?

where x = 1 or 2 is the module number (default - 1)

Short form: :OUTP:ISOL:STAT?
Function: Asks for isolation valve ON/OFF status.
Response: 0 - isolation valve OFF
1 - isolation valve ON

Description

The isolation valve of the module [x] can be queried and switched on and off.

SENSe

The SENSe subsystem selects, configures and queries the measurement functions of the instrument.

SENS:ALT

Applicability: PACE1000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:ALTitude?

Short form: SENS:ALT?
Function: This query reads the aeronautical sensor.
Response: A decimal, altitude reading in the current aeronautical units.

Description

Queries the altitude reading for the aeronautical sensor in the selected aeronautical units.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SENS:ALT:INL

Applicability: PACE6000

Command Syntax

:SENSe[x][:ALTitude]:INLimits <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> in limits value as % full-scale

Short form: :SENS:INL <number>
Function: The instrument has an in-limits indicator. This can generate a service request when the pressure is within limits for a set time period.

Default: 0.01 % full-scale
minimum 0 % full-scale
maximum 100 % full-scale

Query Syntax

:SENSe[x][:ALTitude]:INLimits?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:INL?
Function: Query in-limits value
Response: Decimal representing in-limits value as % full-scale.

Description

Sets the in-limits value.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SENS:ALT:INL:TIME

Applicability: PACE6000

Command Syntax

:SENSe[x][:ALTitude]:INLimits:TIME <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <integer> in-limits time in seconds

Short form: :SENS:INL:TIME <number>
Function: Sets the time that the pressure has to be within limits before generating a service request.

Default: 2 seconds
minimum 2 seconds
maximum 999 seconds

Query Syntax

:SENSe[x][:ALTitude]:INLimits:TIME?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:INL:TIME?
Function: Query in-limits timers.
Response: Number representing in-limits time in seconds.

Description

Sets the in-limits timer value.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SENS:ALT:RANG

Applicability: PACE1000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:ALTitude:RANGe?

Short form: SENS:ALT:RANG?
Function: This queries the range of the aeronautical sensor.
Response: A string (number), representing altitude range in pressure units.
example: 1.30 bara

Description

Queries the range of the aeronautical sensor in pressure units.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SENS:ALT:SLEW

Applicability: PACE1000, PACE6000

Command Syntax

SENSE:ALTitude:SLEW

Parameter: <string>

Short form: SENS:ALT:SLEW
Function: Sets the rate set-point for altitude slew.

Default: -

Query Syntax

SENSe:ALTitude:SLEW?

Short form: SENS:ALT:SLEW?
Function: Queries the rate set-point for altitude slew.
Response: Decimal number representing the rate set-point for altitude slew (rate of climb) or (vertical speed) in current units.

Description

This command sets the rate set-point for altitude slew. The query gets the rate set-point for altitude (rate of climb) or (vertical speed) per second or minute.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SENS:AIRF:QFE

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:AIRField:QFE?

Short form: SENS:AIRF:QFE?
Function: This query reads the QFE value when the instrument is in airfield mode.
Response: A decimal reading in selected units of pressure.

Description

Queries the QFE reading, returns the QFE reading in selected units of pressure when the instrument is in airfield mode.

Note: *QFE is the mean sea level pressure corrected for temperature and adjusted for a specific site (i.e. a datum such as an airfield).*

SENS:AIRF:QFF

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:AIRField:QFF?

Short form: SENS:AIRF:QFF?
Function: This query reads the QFF value when the instrument is in airfield mode.
Response: A decimal reading in selected units of pressure.

Description

Queries the QFF reading, returns the QFF reading in selected units of pressure when the instrument is in airfield mode.

Note: *QFF is the mean sea level pressure derived from the barometric pressure calculated, assuming mean long term values of temperature and relative humidity, for a specific site. (i.e. the QFF is the location value plotted on the surface synoptic chart).*

SENS:AIRF:QNH

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:AIRField:QNH?

Short form: SENS:AIRF:QNH?
Function: This query reads the QNH value when the instrument is in airfield mode.
Response: A decimal reading in selected units of pressure.

Description

Queries the QNH reading, returns the QNH reading in selected units of pressure when the instrument is in airfield mode.

Note: *QNH is the pressure measured at a specific site then reduced to mean sea level pressure.*

SENS:MACH

Applicability: PACE1000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:MACH?

Short form: SENS:MACH?
Function: This query reads the aeronautical sensors.
Response: A decimal reading in MACH number.

Description
Queries the MACH reading.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SENS:MACH:INL

Applicability: PACE6000

Command Syntax

:SENSe[x][:MACH]:INLimits <number>

where x = 1 or 2 is the module number (default - 1)

Parameter: <decimal> MACH in limits value as % full-scale

Short form: :SENS:MACH:INL <number>

Function: The instrument has a MACH in-limits indicator. This can generate a service request when the MACH value is within limits for a set time period.

Default: 0.01 % full-scale
minimum 0 % full-scale
maximum 100 % full-scale

Query Syntax

:SENSe[x][:MACH]:INLimits?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:MACH:INL?

Function: Query MACH in-limits value

Response: Decimal representing MACH in-limits value as % full-scale.

Description

Sets the in-limits value.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SENS:MACH:INL:TIME

Applicability: PACE6000

Command Syntax

:SENSe[x][:MACH]:INLimits:TIME <number>

where x = 1 or 2 is the module number (default - 1)

Parameter: <integer> MACH in-limits time in seconds

Short form: :SENS:MACH:INL:TIME <number>

Function: Sets the time that the MACH value has to be within limits before generating a service request.

Default: 2 seconds
minimum 2 seconds
maximum 999 seconds

Query Syntax

:SENSe[x][:MACH]:INLimits:TIME?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:MACH:INL:TIME?

Function: Query MACH in-limits timers.

Response: Number representing MACH in-limits time in seconds.

Description

Sets the MACH in-limits timer value.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SENS:MACH:RANG

Applicability: PACE1000, PACE6000

Command Syntax

SENSe[2]:MACH:RANG

where x = 1 or 2 is the module number (default - 1)
Parameter:
Short form: SENS:MACH:RANG
Function: Sets the MACH range

Query Syntax

SENSe:MACH:SLEW?

Short form: SENS:MACH:RANG?
Function: This queries the MACH range.
Response: A string (number), repesenting MACH range.

Description
Queries the MACH range reading.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SENS:MACH:SLEW

Applicability: PACE1000, PACE6000

Command Syntax

SENSe[2]:MACH:SLEW

where x = 1 or 2 is the module number (default - 1)
Parameter:
Short form: SENS:MACH:SLEW
Function: Sets the rate set-point for MACH slew.

Query Syntax

SENSe:MACH:SLEW?

Short form: SENS:MACH:SLEW?
Function: Queries the rate set-point for MACH slew.
Response: A decimal, pressure reading/second in MACH.

Description

This command sets the rate set-point for MACH slew. The query gets the rate set-point for MACH.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SENS:PRES

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:SENSe[:PRESsure]?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS?
Function: This query reads the sensor which has been selected by the RANGE command.
Response: A decimal, pressure reading in the current units.

Description

Queries the pressure reading for the selected sensor in the selected units. The sensor can be changed see, :SENSe[:PRESsure]:RANGe.

Example

TX> :SENS1:PRES?
RX> :SENS:PRES 3616.9282227
TX> :Sens2:Pres?
RX> :SENS2:PRES 3617.1921387

TX> :SENS2:PRES:RANG?
RX> :SENS2:PRES:RANG "3.50barg"
Currently module 2 sensing range is in gauge mode.
TX> :SENS2:PRES:RANG "4.50bara"
TX> :SENS2:PRES:RANG?
RX> :SENS2:PRES:RANG "4.50bara"
It has been changed to pseudo-absolute mode.

TX> :SENS2:PRES?
RX> :SENS2:PRES 4615.0807447
The reading is also changed to pseudo-absolute mode.

Note:
:SOURe[:PRESsure]:RANGe command does not change 'SENS?' reading.
It changes the front-panel display.

:SENS:PRES:AVER

Applicability: PACE1000

Command Syntax

:SENSe[:PRESsure]:AVERage

Parameter:
Short form: :SENS:AVE
Function: Enables current average pressure value to be sent.

Query Syntax

:SENSe[:PRESsure]:AVERage?

Short form: :SENS:AVE?
Function: Query average pressure value
Response: Current average pressure value.

Description
This command query gets the average pressure value and the current pressure units.

:SENS:PRES:AVER:RES

Applicability: PACE1000

Command Syntax

:SENSe[:PRESsure]:AVERage:RESet

Parameter:
Short form: :SENS:AVE:RES
Function: Resets average, minimum and maximum pressure readings.

Query Syntax

n/a

Short form:
Function:
Response:

Description
This command resets the maximum, average and minimum pressure readings.

:SENS:PRES:AVER:TIME

Applicability: PACE1000

Command Syntax

:SENSe[:PRESsure]:AVERage:TIME

Parameter:
Short form: :SENS:AVE:TIME
Function: Sets the averaging function time period.

Query Syntax

n/a

Short form:
Function:
Response:

Description
This command sets the averaging function time period.

:SENS:PRES:INL

Applicability: PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:SENSe[x][:PRESsure]:INLimits?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:INL?
Function: Query in-limits value
Response: First parameter - current pressure.
Second parameter - in limit:

0	=	not in limits
1	=	in limits

Description

This command query gets the in-limits value and the in-limits status:

Example 1

TX> :SENS:PRES:INL?
RX> :SENS:PRES:INL 990.0527344, 0

Example 2

TX> :SENS:PRES:INL?
RX> :SENS:PRES:INL 990.0527344, 1

:SENS:PRES:INL:TIME

Applicability: PACE6000

Command Syntax

:SENSe[x][:PRESsure]:INLimits:TIME <number>

where x = 1 or 2 is the module number (default - 1)

Parameter: <integer> pressure in-limits time in seconds

Short form: :SENS:PRES:INL:TIME <number>

Function: Sets the time that the pressure value has to be within limits before generating a service request.

Default: 2 seconds
minimum 2 seconds
maximum 999 seconds

Query Syntax

:SENSe[x][:PRESsure]:INLimits:TIME?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:PRES:INL:TIME?

Function: Query pressure in-limits timers.

Response: Number representing pressure in-limits time in seconds.

Description

Sets the pressure in-limits timer value.

:SENS:PRES:SLEW

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:SENSe[x][:PRESsure]:SLEW?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:SLEW?
Function: Asks for current slew rate.
Response: Decimal number representing slew rate always in current pressure units per second.

Description

This query gets the slew rate of the input pressure always in current pressure units per second. A constant input pressure gives a slew rate of: 0.0.

Example 1

TX> :SENS:PRES:SLEW?
RX> :SENS:PRES:SLEW -20.1089802

Example 2

TX> :SENS2:PRES:SLEW?
RX> :SENS2:PRES:SLEW -20.0536633

:SENS:PRES:BAR

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:SENSe[x][:PRESsure]:BARometer?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:BAR?
Function: Queries the barometric pressure value.
Response: <number> in the selected units of pressure measurement.

Description

Returns the barometric pressure measured by the optional barometric transducer. If the optional barometric transducer is not fitted the response is zero pressure.

Example
:SENS:PRES:BAR?
:SENS:PRES:BAR 982.8430904 (this value in mbar).

:SENS:PRES:RANG

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:RANGe <string>

where x = 1 or 2 is the module number (default - 1)
Parameter: <string> range information

Short form: :SENS:RANG
Function: Used to select a range to be sensed.

Query Syntax

:SENSe[x][:PRESsure]:RANGe?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:RANG?
Function: Asks for currently sensed range.
Response: A string (number), representing the selected pressure range in pressure units.
example: 3.50 barg

Description

This command selects the pressure range to be used for returning the pressure reading, also see the SOURCe:RANGe command.

Example 1
Ranges of module2 can be found by the query:
TX> :INST:CAT2:ALL?
RX> :INST:CAT2:ALL "3.50barg","BAROMETER","4.50bara"
This shows that module2 has three ranges.

Example 2
The current range selected by module2 can be found by the query:
TX> :SENS2:PRES:RANG?
RX> :SENS2:PRES:RANG "3.50barg"

Example 3
To select a different range of module 2 use the command:
TX> :SENS2:PRES:RANG "4.50bara"
Selection confirmed by query:
TX> :SENS2:PRES:RANG?
RX> :SENS2:PRES:RANG "4.50bara"

Note:

The command parameter, "4.50bara", has to be typed exactly. It is case-sensitive and not typo-error tolerant. The command does not affect on instrument front-panel display. Use :SOUR:PRES:RANG to change front-panel range display. (See instrument user's manual).

:SENS:PRES:RES

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:RESolution <string>

where x = 1 or 2 is the module number (default - 1)
Parameter: <integer> resolution information

Short form: :SENS:RES
Function: Used to select a resolution to be used.

Query Syntax

:SENSe[x][:PRESsure]:RESolution?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:RES?
Function: Asks for current resolution.
Response: An integer from 4 to 7 representing selected resolution.

Description

This command selects the resolution to be used for showing the pressure reading in the front panel display.

Example 1
TX> :SENS:PRES:RES?
RX> :SENS:PRES:RES 6
The resolution can be changed by the command, such as:
TX> :SENS:PRES:RES 4

Example 2
Query again confirms that the resolution has been changed to '4' :
TX> :SENS:PRES:RES?
RX> :SENS:PRES:RES 4

Example 3
Any resolution number outside the 4 to 7 boundary generates error messages, such as:
TX> :SENS2:PRES:RES 8
TX> :SYST:ERR?
RX> :SYST:ERR -222,"Data out of range; Parameter 1"
TX> :SENS2:PRES:RES 3
TX> :SYST:ERR?
RX> :SYST:ERR -222,"Data out of range; Parameter 1"

:SENS:PRES:CORR:HEAD

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:CORRection:HEAD <enumerated>,<decimal>

where x = 1 or 2 is the module number (default - 1)

Parameters:	<enumerated>	AIR	-	Air used as gas
		NITRogen	-	Nitrogen used as gas
	<numeric>	Height of gas in metres.		
Short form::	SENS:CORR:HEAD <enumerated>,<numeric>			
Function:	Head correction parameters.			
Default:	Enumerated	AIR		
	decimal	0		

Query Syntax

:SENSe[x][:PRESsure]:CORRection:HEAD?

where x = 1 or 2 is the module number (default - 1)

Short form:	SENS:CORR:HEAD?
Function:	Query gas and height of head correction.
Response:	AIR/NITRogen and height in metres (+100 to -100).

Description

A correction must be made if the unit under test is at a different height from the instrument. This command programs the gas used and the height difference.

Example 1	TX> :SENS:PRES:CORR:HEAD? RX> :SENS:PRES:CORR:HEAD AIR, -0.7500000
Example 2	Head correction can be set to a new height by the command: TX> :SENS:PRES:CORR:HEAD AIR, 1.2
Example 3	Another query will confirm the height change: TX> :SENS:PRES:CORR:HEAD? RX> :SENS:PRES:CORR:HEAD AIR, 1.2000000
Example 4	Head correction can be set for another gas by: TX> :SENS:PRES:CORR:HEAD NITROGEN, 1.2
Example 5	Another query will confirm the height change: TX> :SENS:PRES:CORR:HEAD? RX> :SENS:PRES:CORR:HEAD NITR, 1.2000000

Note: *NITROGEN or NITR is an enumerated data type (not in punctuation marks), not a string and is not case-sensitive and can have a short form.*

:SENS:PRES:CORR:HEAD:STAT

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:CORRection:HEAD:STATe <Boolean>

where x = 1 or 2 is the module number (default - 1)

Parameter <boolean>

0	-	Disables head correction
1	-	Enables head correction

Short form:: SENS:CORR:HEAD:STAT <Boolean>
Function: Enables/disables head correction.
Default: 0

Query Syntax

:SENSe[x][:PRESsure]:CORRection:HEAD:STATe?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:CORR:HEAD:STAT?
Function: Query head correction state
Response: 0 - head correction off
1 - head correction on

Description

This command enables or disables the head correction compensation.

Example

TX> :SENS:PRES:CORR:HEAD:STATe?
RX> :SENS:PRES:CORR:HEAD:STAT 0
Head correction off

It can be turned on and off by:
TX> :SENS:PRES:CORR:HEAD:STATe on
TX> :SENS:PRES:CORR:HEAD:STATe off

Or by:
TX> :SENS:PRES:CORR:HEAD:STATe 1
TX> :SENS:PRES:CORR:HEAD:STATe 0

:SENS:PRES:CORR:OFFS

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:CORRection:OFFSet

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> tare offset value in current pressure units.

Short form: :SENS:OFFS
Function: Subtracts the offset value from the processed reading.
Default: 0

Query Syntax

:SENSe[x][:PRESsure]:CORRection:OFFSet?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:OFFS?
Function: Asks for the tare value.
Response: Number corresponding to the tare offset value.

Description

Example

TX> :SENS:PRES:CORR:OFFS?
RX> :SENS:PRES:CORR:OFFS 0.0
Offset is zero.

Offset can be changed to 100 mbar by sending, (instrument must be in mbar):
TX> :SENS:PRES:CORR:OFFS 100

Send query again to confirm offset has changed:
TX> :SENS:PRES:CORR:OFFS?
RX> :SENS:PRES:CORR:OFFS 100.0000000

Note:
The offset depends on the units of measurement used by the instrument. If changed from mbar to bar, the same query will be returned:

TX> :SENS:PRES:CORR:OFFS?
RX> :SENS:PRES:CORR:OFFS 0.1000000

:SENS:PRES:CORR:OFFS:STAT

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:CORRection:OFFSet:STATe

where x = 1 or 2 is the module number (default - 1)

Parameter: <boolean>

0	-	disables offset
1	-	enables offset

Short form: :SENS:OFFS:STAT
Function: Enables and disables the offset function.

Query Syntax

:SENSe[x][:PRESsure]:CORRection:HEAD:OFFSet:STATe?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:OFFS:STAT?
Function: Asks if offset function is on or off.
Response: 1 (on)
0 (off)

Description

This command enables and disables the offset function. The query gets the state of the offset (or tare) on or off.

Example

:SENS:PRES:CORR:OFFS:STATe
TX> :SENS:PRES:CORR:OFFS:STATe?
RX> :SENS:PRES:CORR:OFFS:STAT 0
The offset correction disabled.
Offset can be enabled and disabled by sending:
TX> :SENS:PRES:CORR:OFFS:STATe on
TX> :SENS:PRES:CORR:OFFS:STATe off
Or by:
TX> :SENS:PRES:CORR:OFFS:STATe 1
TX> :SENS:PRES:CORR:OFFS:STATe 0

:SENS:PRES:CORR:VOL

Applicability: PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:
Default:

Query Syntax

:SENSe[x][:PRESsure]:CORRection:VOLume?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:PRES:CORR:VOL?
Function: Ask for the estimated volume of the system connected to the instrument.
Response: Decimal number in litres corresponding to volume.

Description

The instrument calculates and reports the volume of the system connected.

Example

TX> :SENS:PRES:CORR:VOL?
RX> :SENS:PRES:CORR:VOL 0.0150000

:SENS:PRES:FILT:LPAS:BAND

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:FILTer:[LPASs]:BAND <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> filter band response value in % full-scale.

Short form: :SENS:FILT:BAND
Function: Used to set-up the response band component of the filter.
Default: 0
minimum 0
maximum 100.0

Query Syntax

:SENSe[x][:PRESsure]:FILTer:[LPASs]:BAND?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:FILT:BAND?
Function: Ask for filter step response band parameter.
Response: Number corresponding to filter step response value in % full-scale.

Description

The digital low pass filter has a response band configured as percentage of full-scale. e.g., defaults to 0.05 %FS. If the reading has changed by more than the configured band response value then the filtering is ignored for that conversion and the pressure goes instantly to the new value.

Example

:SENS:PRES:FILT:LPAS:BAND

TX> :SENS:PRES:FILT:LPAS:BAND?
RX> :SENS:PRES:FILT:LPAS:BAND 50.0000000
The current setting for the filter band is 50% of full-scale, i.e., the filter applies only when the change of pressure is within this band.

It can be changed to another number, (for example: 12% of full-scale), by sending:
TX> :SENS:PRES:FILT:LPAS:BAND 12
TX> :SENS:PRES:FILT:LPAS:BAND?
RX> :SENS:PRES:FILT:LPAS:BAND 12.0000000

:SENS:PRES:FILT:LPAS:FREQ

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:FILTer:[LPASs]:FREQuency <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> filter averaging time in seconds.

Short form: :SENS:FILT:FREQ
Function: Used to set up the averaging component of the filter.
Default: 0
minimum 0
maximum 20

Query Syntax

:SENSe[x][:PRESsure]:FILTer:[LPASs]:FREQuency?

where x = 1 or 2 is the module number (default - 1)
Short form: :SENS:FILT:FREQ?
Function: Ask for filter average parameter.
Response: Decimal number corresponding to filter average time in seconds.

Description

A digital low pass filter can be applied to the pressure reading. This is a first order low pass filter, the time constant depends on the value set by this command.

Note:
The decimal number used by the command and query does not represent frequency even though 'FREQ' is used.

Example

:SENS:PRES:FILT? TX> :SENS:PRES:FILT? RX> :SENS:PRES:FILT:LPAS:STAT 0	It can be set to another value, such as: TX> :SENS:PRES:FILT:LPAS:FREQ 1.76 TX> :SENS:PRES:FILT:LPAS:FREQ? RX> :SENS:PRES:FILT:LPAS:FREQ 1.7600000
:SENS:PRES:FILT:LPAS:FREQ TX> :SENS:PRES:FILT:LPAS:FREQ? RX> :SENS:PRES:FILT:LPAS:FREQ 2.0000000 The current setting for low-pass filter's time constant is 2 seconds.	

:SENS:PRES:FILT:LPAS:STAT

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SENSe[x][:PRESsure]:FILTer[:LPASs]:[STATE] <Boolean>

where x = 1 or 2 is the module number (default - 1)

Parameter: <Boolean>

0	-	Disables low pass filter
1	-	Enables low pass filter

Short form: :SENS:FILT <Boolean>
Function: Sets low pass filter ON or OFF.
Default: OFF

Query Syntax

:SENSe[x][:PRESsure]:FILTer[:LPASs]:[STATe]?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:FILT?
Function: Query state (on or off) for the low pass filter
Response: 1 (ON) 0 (OFF)

Description

This command is used to enable or disable the low pass filter for producing a more stable reading. An 'intelligent' filter is implemented so that any noise in the system is filtered while step changes pass straight through the filter.

Example
:SENS:PRES:FILT:LPAS:STAT
TX> :SENS:PRES:FILT:LPAS:STAT?
RX> :SENS:PRES:FILT:LPAS:STAT 0
The filter is currently off.

It can be set on and off by
TX> :SENS:PRES:FILT:LPAS:STAT on
TX> :SENS:PRES:FILT:LPAS:STAT off

Or by
TX> :SENS:PRES:FILT:LPAS:STAT 1
TX> :SENS:PRES:FILT:LPAS:STAT 0

:SENS:PRES:MAX

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

:SENSe:[PRESsure]:MAXimum?

Short form: :SENS:MAX?
Function: This query reads the maximum pressure.
Response: A reading of the current set maximum pressure in selected pressure units.

Description

Queries the current maximum setting for pressure.

:SENS:PRES:MAX:RES

Applicability: PACE1000

Command Syntax

SENSe:[PRESsure]:MAXimum:RESet

Parameter:
Short form: SENS:MAX:RES
Function: Resets maximum, average and minimum pressure readings.

Query Syntax

n/a

Short form:
Function:
Response:

Description

Command resets the maximum, average and minimum pressure readings.

:SENS:PRES:MIN

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:PRESsure:MINimum?

Short form: SENS:MIN?
Function: This query reads the minimum pressure.
Response: A reading of the current set minimum pressure in selected pressure units.

Description

Queries the current minimum setting for pressure.

:SENS:PRES:MIN:RES

Applicability: PACE1000

Command Syntax

SENSe:[PRESsure]:MINimum:RESet

Parameter:
Short form: SENS:MIN:RES
Function: Resets minimum, average and maximum pressure readings.

Query Syntax

n/a

Short form:
Function:
Response:

Description

This command resets minimum, average and maximum pressure readings.

:SENS:PRES:PERC

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:[PRESsure]:PERCent?

Short form: SENS:PERC?
Function: This query reads the percentage pressure.
Response: A reading of the current set percentage pressure.

Description

Queries the current percentage pressure reading.

:SENS:PRES:PERC:SPAN

Applicability: PACE1000

Command Syntax

SENSe:[PRESsure]:PERCent:SPAN

Parameter: NONE|FS|SPAN
Short form: SENS:PERC:SPAN
Function: Sets the percentage span.

Query Syntax

SENSe:[PRESsure]:PERCent:SPAN?

Short form: SENS:PERC:SPAN?
Function: This query reads the percentage span.
Response: A reading of the current set percentage span.

Description

Sets the percentage span: NONE|FS|SPAN. Queries the current percentage span setting.

:SENS:PRES:PERC:STAT

Applicability: PACE1000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:[PRESsure]:PERCent:STATe?

Short form: SENS:PERC:STAT?
Function: This query reads the percentage state.
Response: NONE|FS|SPAN

Description

Queries the current percentage state instrument returns NONE or FS or SPAN.

SENS:SPE

Applicability: PACE1000, PACE6000

Command Syntax

n/a

Parameter:
Short form:
Function:

Query Syntax

SENSe:SPEed?

Short form: SENS:SPE?
Function: This query reads the aeronautical sensors.
Response: A decimal reading in currently selected airspeed units.

Description
Queries the airspeed reading.

Note: If the instrument is not in aero mode it records an error: -221. Settings conflict.

:SENS:SPE:INL

Applicability: PACE6000

Command Syntax

:SENSe[x][:SPEed]:INLimits <number>

where x = 1 or 2 is the module number (default - 1)

Parameter: <decimal> airspeed in limits value as % full-scale

Short form: :SENS:SPE:INL <number>

Function: The instrument has a airspeed in-limits indicator. This can generate a service request when the airspeed value is within limits for a set time period.

Default: 0.01 % full-scale
minimum 0 % full-scale
maximum 100 % full-scale

Query Syntax

:SENSe[x][:SPEed]:INLimits?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:SPE:INL?

Function: Query airspeed in-limits value

Response: Decimal representing airspeed in-limits value as % full-scale.

Description

Sets the in-limits value.

Note: If the instrument is not in aero mode it records an error: -221, Settings conflict.

:SENS:SPE:INL:TIME

Applicability: PACE6000

Command Syntax

:SENSe[x][:SPEed]:INLimits:TIME <number>

where x = 1 or 2 is the module number (default - 1)

Parameter: <integer> airspeed in-limits time in seconds

Short form: :SENS:SPE:INL:TIME <number>

Function: Sets the time that the airspeed value has to be within limits before generating a service request.

Default: 2 seconds
minimum 2 seconds
maximum 999 seconds

Query Syntax

:SENSe[x][:SPEed]:INLimits:TIME?

where x = 1 or 2 is the module number (default - 1)

Short form: :SENS:SPE:INL:TIME?

Function: Query airspeed in-limits timers.

Response: Number representing airspeed in-limits time in seconds.

Description

Sets the airspeed in-limits timer value.

Note: If the instrument is not in aero mode it records an error: -221, Settings conflict.

SENS:SPE:RANG

Applicability: PACE1000, PACE6000

Command Syntax

SENSe[2]:SPEed:RANGe

where x = 1 or 2 is the module number (default - 1)
Parameter:
Short form: SENS:SPE:RANG
Function: Sets the airspeed range.

Query Syntax

SENSe:SPEed:SLEW?

Short form: SENS:SPE:RANG?
Function: This query reads the airspeed range.
Response: A string (number), representing the airspeed range in pressure units.
example: 3.50 barg

Description
Queries the airspeed range.

Note: If the instrument is not in aero mode it records an error: -221. Settings conflict.

SENS:SPE:SLEW

Applicability: PACE1000, PACE6000

Command Syntax

SENSe[2]:SPEed:SLEW

where x = 1 or 2 is the module number (default - 1)
Parameter:
Short form: SENS:SPE:SLEW
Function: Sets the airspeed rate of change.

Query Syntax

SENSe:SPE:SLEW?

Short form: SENS:SPE:SLEW?
Function: This query reads the airspeed rate of change.
Response: A decimal, airspeed reading/second.

Description
Queries the airspeed rate of change reading.

Note: If the instrument is not in aero mode it records an error: -221, Settings conflict.

SOURce

The SOURce subsystem controls the pressure output of the instrument.

SOUR:ALT

Applicability: PACE6000

Command Syntax

SOURce:ALTitude:LEVel:IMMediate:AMPLtude

Parameter: <string>

Short form: SOUR:ALT
Function: Sets the altitude set-point.

Default: -

Query Syntax

SOURce:ALTitude:LEVel:IMMediate:AMPLtude?

Short form: SOUR:ALT?
Function: Queries altitude set-point.
Response: Decimal number representing altitude in current units.

Description

This command sets the altitude set-point. This query gets the decimal number representing altitude in current units.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SOUR:MACH:REF

Applicability: PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Default:

Query Syntax

SOURce:MACH:REFerence?

Short form: SOUR:MACH:REF
Function: Queries the MACH reference value from either Ps reference or barometric.
Response: Decimal number representing the MACH reference value in currently selected pressure units.

Description
The query gets the MACH reference value in currently selected pressure units.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SOUR:MACH:REF:MODE

Applicability: PACE6000

Command Syntax

SOURce:MACH:REFerence:MODE

Parameter: <number>

Short form: SOUR:MACH:REF:MODE
Function: Enables a MACH reference mode to be set.
Default: -

Query Syntax

SOURce:MACH:REFerence:MODE?

Short form: SOUR:MACH:REF:MODE?
Function: Queries the MACH reference mode.
Response: Numeric value representing the MACH reference mode.

Description

This command enables a MACH reference mode to be set. The query gets the MACH reference entered mode.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

SOUR:MACH:REF:VAL

Applicability: PACE6000

Command Syntax

SOURce:MACH:REFerence:VALue

Parameter: <string>

Short form: SOUR:MACH:REF:VAL
Function: Sets the MACH reference entered value.
Default: -

Query Syntax

SOURce:MACH:REFerence:VALue?

Short form: SOUR:MACH:REF:VAL?
Function: Queries the MACH reference entered value from either Ps reference.
Response: Numeric value of the MACH reference.

Description

This command sets the MACH reference entered value from either Ps reference. The query gets the MACH reference entered value.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SOUR:MACH:LEV:IMM:AMPL

Applicability: PACE6000

Command Syntax

:SOURce[x]:MACH[:LEVel][:IMMediate][:AMPLitude] <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> in MACH number.

Short form: SOURce:MACH <number>
Function: Sets the MACH set-point

Default: 0.0

Query Syntax

:SOURce[x]:MACH[:LEVel][:IMMediate][:AMPLitude] <number>?

Short form SOURce:MACH?
Function: Programmable set-point value
Response: Decimal number representing the set-point in MACH number.

Description

This command sets the MACH set-point. This query gets the decimal number representing the set-point in MACH number.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SOUR:PRES:COMP

Applicability: PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Default:

Query Syntax

:SOURce[x][:PRESsure]:COMPensate[y]?

where: x (module) = 1 or 2 (default - 1) y (+ve or -ve source pressure) = 1 or 2.
Short form: :SOUR:COMP?
Function: Queries +ve and -ve source pressures.
Response: Pressure value in current pressure units.

Description

This query gets the +ve and -ve source pressures in the current pressure units.

Example
:SOUR:PRES:COMP[x]?
TX> :SOUR:PRES:COMP?
RX> :SOUR:PRES:COMP 3165.9526002
TX> :SOUR:PRES:COMP1?
RX> :SOUR:PRES:COMP 3165.9484591
TX> :SOUR:PRES:COMP2?
RX> :SOUR:PRES:COMP2 -963.9638062

y=1,2;

y	Source pressure measurements
1	+ve
2	-ve

If x is greater than 2, an error will be reported
TX> :SOUR:PRES:COMP3?
TX> :SYST:ERR?
RX> :SYST:ERR -114,"Header suffix out of range"

:SOUR:PRES:EFF

Applicability: PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:SOURce[x][:PRESsure]:EFFort?

where x = 1 or 2 is the module number (default - 1)
Short form: :SOUR:EFF?
Function: This query only command returns the effort needed for the controller to achieve the set-point.
Response: Decimal percent number representing controller effort.

Description

This query only command returns the % effort the controller does to achieve the set-point.

Example
TX> :SOUR:PRES:EFF?
RX> :SOUR:PRES:EFF -0.2342882
Percentage of effort the supply or vacuum valve makes to maintain the control point. The return number should be –100 to +100. A positive number indicates the supply valve makes more effort and a minus number indicates the vacuum valve makes more effort. If the controller is off:
TX> :OUTP off
TX> :SOUR:PRES:EFF?
RX> :SOUR:PRES:EFF 0.0
The return number is '0.0', since no valve is making any effort.

:SOUR:PRES:INL

Applicability: PACE5000, PACE6000

Command Syntax

:SOURce[x][:PRESsure]:INLimits <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> in limits value as % full-scale

Short form: :SOUR:INL <number>
Function: The controller has an in-limits set-point indicator. This can generate a service request when the pressure is within limits for a set time period.

Default: 0.01 % full-scale
minimum 0% full-scale
maximum 100% full-scale

Query Syntax

:SOURce[x][:PRESsure]:INLimits?

where x = 1 or 2 is the module number (default - 1)
Short form: :SOUR:INL?
Function: Query in-limits value
Response: Decimal representing in-limits value as % full-scale.

Description

Sets the in-limits value.

Example
TX> :SOUR:PRES:INL?
RX> :SOUR:PRES:INL 0.0200000
The current in-limits set-point is 0.02% full-scale

Can be set to:
TX> :SOUR:PRES:INL 0.01
TX> :SOUR:PRES:INL?
RX> :SOUR:PRES:INL 0.0100000

:SOUR:PRES:INL:TIME

Applicability: PACE5000, PACE6000

Command Syntax

:SOURce[x][:PRESsure]:INLimits:TIME <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <integer> in-limits time in seconds

Short form: :SOUR:INL:TIME <number>
Function: Sets the time that the pressure has to be within limits before generating a service request.

Default: 2 seconds
minimum 2 seconds
maximum 999 seconds

Query Syntax

:SOURce[x][:PRESsure]:INLimits:TIME?

where x = 1 or 2 is the module number (default - 1)
Short form: :SOUR:INL:TIME?
Function: Query in-limits timers.
Response: Number representing in-limits time in seconds.

Description

Sets the in-limits timer value.
Example
TX> :SOUR:PRES:INL:TIME?
RX> :SOUR:PRES:INL:TIME 2

Can be set to 99 by issuing command:
TX> :SOUR:PRES:INL:TIME 99
TX> :SOUR:PRES:INL:TIME?
RX> :SOUR:PRES:INL:TIME 99

Or to 999 by issuing command:
TX> :SOUR:PRES:INL:TIME 999
TX> :SOUR:PRES:INL:TIME?
RX> :SOUR:PRES:INL:TIME 999

:SOUR:PRES:LEV:IMM:AMPL

Applicability: PACE5000, PACE6000

Command Syntax

:SOURce[x][:PRESsure][:LEVel][:IMMediate][:AMPLitude] <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> Pressure in current units

Short form: SOUR <number>
Function: Set the pressure set-point

Default: 0.0

Query Syntax

:SOURce[x][:PRESsure][:LEVel][:IMMediate][:AMPLitude]?

where x = 1 or 2 is the module number (default - 1)
Short form :SOUR?
Function: Programmable set-point value
Response: Decimal number representing pressure set-point in current units.

Description

This command sets the pressure set-point and is the long form of :SOUR:PRES.

Example
TX> :SOUR:PRES?
RX> :SOUR:PRES:LEV:IMM:AMPL 0.4000000
The current controller set-point is 0.4 of the current unit.

It can be set to another value, such as:
TX> :SOUR:PRES 0.5
TX> :SOUR:PRES?
RX> :SOUR:PRES:LEV:IMM:AMPL 0.5000000

:SOUR:PRES:LEV:IMM:AMPL:VENT

Applicability: PACE5000, PACE6000

Command Syntax

:SOURce[x][:PRESsure][:LEVel][:IMMediate][:AMPLitude]:VENT <number>

where x = 1 or 2 is the module number (default - 1)

Parameter: <integer>

0 - abort vent
1 - start vent

Short form: SOUR:VENT <integer>
Function: Vents the user system.

Default: 0

Query Syntax

:SOURce[x][:PRESsure][:LEVel][:IMMediate][:AMPLitude]:VENT?

where x = 1 or 2 is the module number (default - 1)

Short form: :SOUR:VENT?
Function: Query status of vent.

Response:

0 - vent OK
1 - vent in progress
2 - vent completed

Description

This command vents the user system; the command should be queried to get the status of the vent.

Examples

For an instrument switched on without any vent, a query:
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 0 (a return of '0' indicates vent not in progress).

A vent can be started by sending:
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT 1

Immediate query:
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 1

A return of '1' indicates vent in progress

TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 1
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 1
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 1
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 1

Continuing queries return '1', until the vent finishes and a return of '2':

TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 2
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 2
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 2
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT?
RX> :SOUR:PRES:LEV:IMM:AMPL:VENT 2

To abort the vent process, send:
TX> :SOUR:PRES:LEV:IMM:AMPL:VENT 0

:SOUR:PRES:RANG

Applicability: PACE5000, PACE6000

Command Syntax

:SOURCe[x][:PRESsure]:RANGe <string>

where x = 1 or 2 is the module number (default - 1)
Parameter: <string> range information

Short form: :SOUR:RANG
Function: Selects the control range.

Query Syntax

:SOURCe[x][:PRESsure]:RANGe?

where x = 1 or 2 is the module number (default - 1)
Short form:: SOUR:RANGE?
Function: Asks for currently selected controller range.
Response: String representing selected sense range.

Description

This command selects the range to be used for controlling pressure.
e.g. : SOURe:RANGe “2.0bara”
selects the 2 bar absolute range; the pressure units are always in bar.

Note:
Absolute ranges are pseudo-absolute values, combining barometric and gauge sensor readings.
Example
TX> :SOUR:PRES:RANG?
RX> :SOUR:PRES:RANG "3.50barg"
Queries current range

Can set to another range:
TX> :SOUR:PRES:RANG "4.50bara"
TX> :SOUR:PRES:RANG?
RX> :SOUR:PRES:RANG "4.50bara"

This change can be observed on front panel display.
The parameter is case sensitive.

:SOUR:PRES:RANG:LOW

Applicability: PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Default:

Query Syntax

:SOURce[x]:PRESsure:RANGe:LOW?

where x = 1 or 2 is the module number (default - 1)
Short form: :SOUR:LOW?
Function: Queries the lower full-scale of the module [x].
Response: Decimal number in current selected units.

Description

This query gets the lower full-scale of the module [x].

Example 1
TX> :Sour:Pres:Rang:Low?
RX> :SOUR:PRES:RANG:LOW "-1000mbar g"

Example 2
TX> :Sour1:Pres:Rang:Low?
RX> :SOUR:PRES:RANG:LOW "-1000mbar g"

:SOUR:PRES:SLEW

Applicability: PACE5000, PACE6000

Command Syntax

:SOURCE[x][:PRESsure]:SLEW <number>

where x = 1 or 2 is the module number (default - 1)

Parameter: <decimal> rate in pressure units/second or enumerated string maximum or minimum rate.
Short form: SOUR:SLEW <number> or maximum or minimum rate
Function: Selects the pressure rate used when value rate is selected.
Default: 100

Query Syntax

:SOURCE[x][:PRESsure]:SLEW?

where x = 1 or 2 is the module number (default - 1)

Short form: :SOUR:SLEW?
Function: Query rate value
Response: Decimal number representing rate value in selected units/second.

Description

When the controller rate is selected as value, this command is used to set the controllers rate in selected units/second.

Example

TX> :SOUR:PRES:SLEW?
RX> :SOUR:PRES:SLEW 2.0000000

Change current unit of mbar to bar and query again:
TX> :SOUR:PRES:SLEW?
RX> :SOUR:PRES:SLEW 0.0020000

To set slew rate to other values, such as:
TX> :SOUR:PRES:SLEW 4
TX> :SOUR:PRES:SLEW?
RX> :SOUR:PRES:SLEW 4.0000000
TX> :SOUR:PRES:SLEW max
TX> :SOUR:PRES:SLEW?
RX> :SOUR:PRES:SLEW 99999999.0000000
TX> :SOUR:PRES:SLEW min
TX> :SOUR:PRES:SLEW?
RX> :SOUR:PRES:SLEW 0.0

:SOUR:PRES:SLEW:MODE

Applicability: PACE5000, PACE6000

Command Syntax

:SOURCE[x][:PRESsure]:SLEW:MODE <enumerated>

where x = 1 or 2 is the module number (default - 1)

Parameter: <enumerated>
MAXimum - maximum rate
LINear - user selected linear rate

Short form: SOUR:SLEW:MODE <enumerated>
Function: Select the rate the controller should use to achieve set-point.
Default: MAXimum

Query Syntax

:SOURCE[x][:PRESsure]:SLEW:MODE?

where x = 1 or 2 is the module number (default - 1)

Short form: :SOUR:SLEW:MODE?
Function: Query rate mode.
Response: MAX for maximum rate
LIN for user defined rate

Description

The controller can operate in two rate modes - maximum and value. In maximum rate the controller tries to achieve set-point as quickly as possible. In value mode the controller achieves the set-point at a user selected rate.

Example 1
TX> :SOUR:PRES:SLEW:MODE?
RX> :SOUR:PRES:SLEW:MODE MAX

Example 2
Can be set to user defined rate:
TX> :SOUR:PRES:SLEW:MODE linear
TX> :SOUR:PRES:SLEW:MODE?
RX> :SOUR:PRES:SLEW:MODE LIN

:SOUR:PRES:SLEW:OVER

Applicability: PACE5000, PACE6000

Command Syntax

:SOURCE[x][:PRESsure]:SLEW:OVERshoot[:STATe] <Boolean>

where x = 1 or 2 is the module number (default - 1)

Parameter: <boolean>

0	-	overshoot not allowed
1	-	overshoot allowed

Short form: SOUR:SLEW:OVER <Boolean>
Function: Selects pressure overshoot to `allowed' or `not allowed'.
Default: 1 - overshoot allowed

Query Syntax

:SOURCE[x][:PRESsure]:SLEW:OVERshoot[:STATe]?

where x = 1 or 2 is the module number (default - 1)

Short form: :SOUR:SLEW:OVER?
Function: Query overshoot state
Response: 0 - overshoot not allowed
1 - overshoot allowed

Description

The controller can reach the set-point in one of two modes:

Overshoot `not allowed', the controller changes the pressure to near the set-point. The rate of pressure change slows when approaching the set-point to avoid overshoot.

Overshoot `allowed', the controller achieves set-point as fast as possible and, when approaching the set-point, may overshoot or undershoot.

Example 1
TX> :SOUR:PRES:SLEW:OVER?
RX> :SOUR:PRES:SLEW:OVER:STAT 0

Example 2
TX> :SOUR:PRES:SLEW:OVER 1
TX> :SOUR:PRES:SLEW:OVER?
RX> :SOUR:PRES:SLEW:OVER:STAT 1

:SOUR:SPE:LEV:IMM:AMPL

Applicability: PACE6000

Command Syntax

:SOURce[x]:SPEed[:LEVel][:IMMediate][:AMPLitude] <number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> Airspeed in current units

Short form: SOUR:SPE <number>
Function: Sets the airspeed set-point

Default: 0.0

Query Syntax

:SOURce[x]:SPEed[:LEVel][:IMMediate][:AMPLitude]?

where x = 1 or 2 is the module number (default - 1)
Short form :SOUR:SPE?
Function: Programmable set-point value
Response: Decimal number representing airspeed set-point in current units.

Description

This command sets the airspeed set-point. This query gets the decimal number representing airspeed set-point in current units.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

:SOUR:SPE:LEV:IMM:AMPL:SLEW

Applicability: PACE6000

Command Syntax

:SOURce[x]:SPEed[:LEVel][:IMMediate][:AMPLitude]:SLEW<number>

where x = 1 or 2 is the module number (default - 1)
Parameter: <decimal> Airspeed rate set-point in current units per second.

Short form: SOUR:SPE:SLEW <number>
Function: Sets the rate set-point for airspeed.

Default: 0.0

Query Syntax

:SOURce[x]:SPEed[:LEVel][:IMMediate][:AMPLitude]:SLEW?

where x = 1 or 2 is the module number (default - 1)
Short form :SOUR:SPE:SLEW?
Function: Queries the rate set-point for airspeed.
Response: Decimal number representing the rate set-point airspeed in currently selected units of speed per second.

Description
This command sets the airspeed rate set-point. The query gets the rate set-point for airspeed.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

STATus

The STATus subsystem supports the OPERation status register as defined in SCPI protocol.

:STAT:OPER:COND

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:STATus:OPERation:CONDition?

Short form: :STAT:OPER:COND?
Function: Query condition register
Response: Contents of condition register

Description
Returns the contents of the 16 bit condition register, see section on status reporting.

:STAT:OPER:ENAB

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:STATus:OPERation:ENABle <integer>

Parameter: <integer> 16 bit value to set enable bits

Short form: STAT:OPER:ENAB <integer>
Function: Controls the status operation enable register.
Default: 0
minimum 0
maximum 32767

Query Syntax

:STATus:OPERation:ENABle?

Short form: :STAT:OPER:ENAB?
Function: Query enable register
Response: 16 bit value of enable register.

Description
Controls the bits that pass through the status reporting system, see status reporting section.

:STAT:OPER:EVEN

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:STATus:OPERation:[EVENt]?

Short form: :STAT:OPER?
Function: Query event register
Response: 16 bit value of event register.

Description
Reads contents of event register, see status reporting section.

:STAT:OPER:PRES:COND

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:STATus:PRESsure:OPERation:CONDition?

Short form: :STAT:OPER:PRES:COND?
Function: Reads the contents of the pressure condition register.
Response: Contents of pressure condition register.

Description

Returns the contents of the 16 bit pressure condition register, see section on status reporting.

:STAT:OPER:PRES:ENAB

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:STATus:OPERation:PRESsure:ENABle <integer>

Parameter: <integer> 16 bit value to set pressure enable bits

Short form: STAT:OPER:PRES:ENAB <integer>
Function: Controls the pressure status operation enable register.
Default: 0
minimum 0
maximum 32767

Query Syntax

:STATus:OPERation:PRESsure:ENABle?

Short form: :STAT:OPER:PRES:ENAB?
Function: Query enable register.
Response: 16 bit value of pressure enable register.

Description
Controls the bits that pass through the status reporting system, see status reporting section.

:STAT:OPER:PRES:EVEN

Applicability: PACE5000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:STATus:OPERation:PRESsure:[:EVENt]?

Short form: :STAT:OPER:PRES?
Function: Reads contents of pressure event register
Response: 16 bit value of pressure event register.

Description

Reads contents of pressure event register, see status reporting section.

Status System Examples

Example *SRE and *STB

```
                //Explanation:
*SRE 255        //To enable every bit of SRE register;
FRED            //Deliberately type a wrong command, such as;
*STB?           //Now queries Status Byte;
*STB 68         //Error Message set EAV and SERIAL POLL set MSS, 2^2+2^6 = 68;
*STB 0          //Previous read STB clears it.
```

Refer to Figure 3-1.

Example of Pressure Event generated SRQ

```
TX> *SRE 128                          //Enable SRE bit 7
TX> :STAT:OPER:ENAB 1024              //Enable Operation Register, bit 10
TX> :STAT:OPER:PRES:ENAB 32767        //Enable Pressure register all 16 bit
TX> :STAT:OPER:PRES:EVEN?             //Query Pressure Event
RX> :STAT:OPER:PRES:EVEN 0            //No any even yet
TX> :SENS:PRES?                       //What's the controller pressure now?
RX> :SENS:PRES 1099.9993896           //It is 1100 mbar
TX> :OUTP 1                           //Switch controller on
TX> :SOUR:PRES 2000                   //To generate an in-limit event
RX> :SRQ 192                          //Receive an SRQ automatically
TX> :STAT:OPER:PRES:EVEN?             //What happened to Pressure event?
RX> :STAT:OPER:PRES:EVEN 4            //In-limit event happened, see Table 3-2
TX> :STAT:OPER:PRES:EVEN?             //Ask again will clear the event register
RX> :STAT:OPER:PRES:EVEN 0            //It is cleared indeed
```

Example of *IDN?

```
TX> *IDN?
RX> *IDN GE Druck,Pace5000 User Interface,58784,01.05.04
```

:STAT:PRES

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:STATus:PRESet

Parameter: <integer>

Short form: :STAT:PRES
Function: Enables the preset values to be used.

Query Syntax

n/a

Short form:
Function:
Response:

Description

Enables the 25 preset values of set-point to be used.

:STAT:QUES:COND

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:STATus:QUEStionable:CONDition?

Short form: :STAT:QUES:COND?
Function: Query questionable data condition register
Response: Contents of questionable data condition register

Description

Returns the contents of the 16 bit questionable data condition register, see section on status reporting.

:STAT:QUES:ENAB

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:STATus:QUEStionable:ENABle <integer>

Parameter: <integer> 16 bit value to set enable bits

Short form: STAT:OPER:ENAB <integer>
Function: Controls the questionable data condition enable register.
Default: 0
minimum 0
maximum 32767

Query Syntax

:STATus:QUEStionable:ENABle?

Short form: :STAT:OPER:ENAB?
Function: Query questionable data condition enable register.
Response: 16 bit value of enable register.

Description
Controls the bits that pass through the questionable data condition reporting system, see status reporting section.

:STAT:QUES:EVEN

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:STATus:QUEStionable:[EVENt]?

Short form: :STAT:QUES?
Function: Query questionable data condition register
Response: 16 bit value of questionable data condition register.

Description
Reads contents of questionable data condition register, see status reporting section.

SYSTem

The SYSTem subsystem consists of general purpose commands.

SYST:COMM:USB

Applicability: PACE1000

Command Syntax

:SYStem:COMMunicate:USB

Parameter: <boolean>

0 - mass storage
1 - communication

Short form: :SYST:COMM:USB
Function: Switches the communication mode between mass storage to communications.

Query Syntax

:SYStem:COMMunicate:USB?

Short form: :SYST:COMM:USB?
Function: Gets the communication mode
Response: 0 - mass storage
1 - communication

:SYST:ERR

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:SYStem:ERRor?

Short form: :SYST:ERR?
Function: Gets next error from the error queue

Response: The follow list of errors are available
-102, “Syntax error”
-104, “Data type error”
-108, “Parameter not allowed”
-109, “Missing parameter”
-110, “Command Header Error”
-111, “Header Separator Error”
-112, “Program mnemonic too long”
-113, “Undefined header”
-114, “Header suffix out of range”
-120, “Numeric data error”
-121, “Invalid character in number”
-123, “Exponent too large”
-124, “Too many digits”
-128, “Numeric data not allowed”
-130, “Suffix error”
-131, “Invalid suffix”
-134, “Suffix too long”
-138, “Suffix not allowed”
-140, “Character data error”
-141, “Invalid character data”
-144, “Character data too long”
-148, “Character data not allowed”
-150, “String data error”
-151, “Invalid string data”

-158, “String data not allowed”
-200, “Execution error”
-201, “Invalid while in local”
-202, “Settings lost due to rtl”
-203 "Command protected"
-220, “Parameter error”
-222, “Data out of range”
-223, “Too much data”
-224, “Illegal parameter value”
-240 "Hardware error"
-241 "Hardware missing"
-310, “System error”
-350, “Queue overflow”
-400, “Query error”
201 , “Query only”
202 , “No query allowed”
203 , “Parameter(s) not expected”
207 , “Enumerated value not in union”
208 , “Illegal number of parameters”
210 , “Run out of memory handle”
211 , “Unit not matched”
212 , “Unit not required”

Description

This command queries the error queue which holds up to five errors. The instrument returns the message “No error” when no more errors are in the queue.

Example

Returns the system error
To test the system, send a wrong command, such as:
TX> :SENS:PRES qwer
Ask for error:
TX> :SYST:ERR?
RX> :SYST:ERR -200,"Execution error;Query or command violation"
This query empties the error stack, another query returns “No error”.
TX> :SYST:ERR?

:SYST:DATE

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:DATE

Parameter: Integer in date format.

Short form: :SYST:DATE
Function: Sets the date.

Query Syntax

:SYSTem:DATE?

Short form: :SYST:DATE?
Function: Queries date
Response: Returns date setting.

Description
Replies with the date settings in year, month, and day order.

Example
The system date can be set by:
TX> :SYST:DATE 9, 5, 2
TX> :SYST:DATE?
RX> :SYST:DATE 9, 5, 2

TX> :SYST:DATE?
RX> :SYST:DATE 9, 4, 1
It returns current date in year, month, and day order.

:SYST:SET

Applicability: PACE5000, PACE6000

Command Syntax

:SYSTem:SET[x]

where x = 1 or 2 is the module number (default - 1)

Parameter: Enumerated (measure or control) and a decimal number of set-point.
Short form: :SYST:SET
Function: This command only effective at switch-on condition and can set the controller on with a set-point.

Query Syntax

:SYSTem:SET[x]?

where x = 1 or 2 is the module number (default - 1)

Short form: :SYST:SET?
Function: Queries current system setting.
Response: Returns current settings in either measure or control with current set-point value.

Description

Example
TX> :SYST:SET?
RX> :SYST:SET MEAS, 0.0
At the current system set, when the instrument is switched on, the controller will be in measurement mode and the set-point will be zero.
This command can set the instrument to control mode, immediately after switch-on, with a set-point to 100 mbar, by sending:

TX> :SYST:SET CONT, 100.0
TX> :SYST:SET?
RX> :SYST:SET CONT, 100.0000000

Note:

The controller will not change mode without a switch on/off process. During normal operations use the following command example:

TX> :SOUR 2000
TX> :OUTP 1

This switches on the controller immediately with a set-point of 2000 mbar.

:SYST:TIME

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:TIME

Parameter: Integer in time format.

Short form: :SYST:TIME
Function: Sets the time.

Query Syntax

:SYSTem:TIME?

Short form: :SYST:TIME?
Function: Returns current time in hour, minute and second.
Response: Sends in order: hour, minute and second.

Description

This command sets the time. The command query returns the current time in hour, minute and second order.

Example
TX> :SYST:TIME?
RX> :SYST:TIME 10, 8, 44
It returns current time in hour, minute and second order.

Time can be reset by:
TX> :SYST:TIME 11, 11, 2
TX> :SYST:TIME?
RX> :SYST:TIME 11, 11, 4

:SYST:COMM:SER:CONT:RTS

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:COMMunicate:SERial:CONTrol:RTS

Parameter: Boolean, ON or OFF

Short form: SYST:COMM:SER:RTS[x]
Function: Sets the serial communication hardware handshaking where:
[x] =
0: OFF
1: ON

Query Syntax

:SYSTem:COMMunicate:SERial:CONTrol:RTS?

Short form: SYST:COMM:SER:CONT:RTS?
Function: Queries serial communication handshaking.
Response: Returns 0 or 1.

Description

Selects serial communications hardware handshaking. The command query requests the current serial communication hardware handshaking.

Example: 1	TX> :SYST:COMM:SER:CONT:RTS?	
	RX> :SYST:COMM:SER:CONT:RTS 0	//Hardware handshaking OFF.
Example: 2	TX> :SYST:COMM:SER:CONT:RTS1	//Hardware handshaking ON.
	TX> :SYST:COMM:SER:CONT:RTS?	//Query again
	RX> :SYST:COMM:SER:CONT:RTS 1	//Confirmed

:SYST:COMM:SER:CONT:XONX

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:COMMunicate:SERial:CONTrol:XONXoff

Parameter: Integer

Short form: SYST:COMM:SER:XONX
Function: Sets the serial communication handshaking of:
0: NONE
1: XON/XOFF
2: RTS/CTS

Query Syntax

:SYSTem:COMMunicate:SERial:CONTrol:XONXoff?

Short form: SYST:COMM:SER:CONT:XONX?
Function: Queries serial communication handshaking.
Response: Returns an integer of 0, 1, 2.

Description

Selects serial communications interface handshaking. The command query requests the current serial communication handshaking.

Example:
TX> :SYST:COMM:SER:CONT:XONX?
RX> :SYST:COMM:SER:CONT 0
The current hardware handshaking is NONE

It can be set to RTS/CTS by sending:
TX> :SYST:COMM:SER:CONT 2

The integer of 0, 1, 2 represents:
0: NONE
1: XON/XOFF
2: RTS/CTS

:SYST:COMM:SER:BAUD

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:COMMunicate:SERial:BAUD

Parameter: Integer

Short form: SYST:COMM:SER:BAUD
Function: Sets the baud rate
Note: *The parameter must be a valid Baud-rate number.*

Query Syntax

:SYSTem:COMMunicate:SERial:BAUD?

Short form: :SYST:COMM:SER:BAUD?
Function: Queries current baud setting.
Response: Current baud setting.

Description

This command instructs the instrument to set the baud rate. The command query returns the current baud rate set in the instrument.

Example
TX> :SYST:COMM:SER:BAUD?
RX> :SYST:COMM:SER:BAUD 9600
The current baud-rate is 9600 bit/second.

It can be set to another baud rate (example: 19200) by sending:
TX> :SYST:COMM:SER:BAUD 19200

Notes

1: The parameter must be a valid baud rate number:

2400	9600	38400	115200
4800	19200	576000	

2: Changing to a new baud rate causes a loss of communications until resetting the local PC RS232 to the new baud rate.

:SYST:COMM:SER:TYPE:PAR

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:COMMunicate:SERial[:TYPE]:PARity

Parameter: Enumerate

Short form: SYST:COMM:SER:TYPE:PAR
Function: Sets parity, odd, even or none.

Note:
This command breaks the communication between the PC and the PACE instrument. No further query can be made until after resetting the pc to the same setting. The instrument cannot be brought back to local mode.

Query Syntax

:SYSTem:COMMunicate:SERial:TYPE:PARity?

Short form: :SYST:COMM:SER:TYPE:PAR?
Function: Queries current parity setting.
Response: Returns odd, even or none.

Description

This command instructs the instrument to set serial communication parity. The command query gets the current parity setting.

Example

TX> :SYST:COMM:SER:TYPE:PAR?
RX> :SYST:COMM:SER:TYPE:PAR NONE
The current setting ignores the parity check.

It can be set to ODD and EVEN by sending:
TX> :SYST:COMM:SER:TYPE:PAR ODD
or
TX> :SYST:COMM:SER:TYPE:PAR EVEN

:SYST:COMM:GPIB:SELF:ADDR

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

Parameter: Integer

Short form: SYST:COMM:GPIB:SELF:ADDR
Function: Sets the instrument's GPIB address.

Query Syntax

:SYSTem:COMMunicate:GPIB:SELF:ADDRess?

Short form: :SYST:COMM:GPIB:SELF:ADDR?
Function: Queries GPIB address
Response: Returns decimal number representing GPIB address.

Description

This command instructs the instrument to set a GPIB address. The query command gets the current address.

Example:

```
:TX> :SYST:COMM:GPIB:SELF:ADDR?
RX> :SYST:COMM:GPIB:SELF:ADDR 1        //Current GPIB address is 1
TX> :SYST:COMM:GPIB:SELF:ADDR 16       //Set address to 16
TX> :SYST:COMM:GPIB:SELF:ADDR?
RX> :SYST:COMM:GPIB:SELF:ADDR 16
```

:SYST:AREA

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:AREA

Parameter: String

Short form: SYST:AREA
Function: Sets a group of parameters for an area of the world.

Query Syntax

:SYSTem:AREA?

Short form: :SYST:AREA?
Function: Queries area setting.
Response: Sends:

EUROPE, or EUR
USA
JAPAN or JAP
ASIA
ROW (rest of the world)

Description

This command instructs the instrument to enable a group of default settings for an area of the world.

:SYST:AREA
TX> :SYST:AREA?
RX> :SYST:AREA EUR
The current area of use is set to Europe.

Can be set to another area, such as Japan:
TX> :SYST:AREA jap
TX> :SYST:AREA?
RX> :SYST:AREA JAP

Set to the rest of the world:
TX> :SYST:AREA row
TX> :SYST:AREA?
RX> :SYST:AREA ROW

:SYST:PASS:CDIS

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:PASSword:CDISable

Parameter: 2317100

Short form: SYST:PASS:CDIS
Function: Disables the calibration with a password.

Query Syntax

n/a

Short form:
Function:
Response:

Description

This command disables calibration with a password.

:SYST:PASS:CEN

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SYSTem:PASSword:CENable:

Parameter: 2317100

Short form: :SYST:PASS:CEN
Function: Enables calibration, default condition - disabled (calibration is not allowed).

Query Syntax

n/a

Short form:
Function:
Response:

Description
This command enables calibration.

:SYST:PASS:CEN:STAT

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:SYSTem:PASSword:CENable:STATe?

Short form: SYST:PASS:CEN:STAT?
Function: Gets the status of the calibration password.
Response: 0 - disabled
1 - enabled

Description
This query asks if the calibration is enabled or disabled.

Example
It can be enabled by sending command and password
TX> :SYST:PASS:CEN 2317100
TX> :SYST:PASS:CEN:STAT?
RX> :SYST:PASS:CEN:STAT 1
When enabled and a 3-point calibration can be carried out.

It can be disabled again by issuing a command and a password, such as:
TX> :SYST:PASS:CDIS 2317100
TX> :SYST:PASS:CEN:STAT?
RX> :SYST:PASS:CEN:STAT 0

:SYST:VERS

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

:SYStem:VERSion?

Short form: :SYST:VERS?
Function: Returns the SCPI version.
Response: version number

Description
Replies with the SCPI version number.

Example
TX> :SYST:VERS?
RX> :SYST:VERS 1995.0

UNIT

The UNIT sub-system configures the instrument's pressure measurement units.

UNIT:ALT

Applicability: PACE1000, PACE6000

Command Syntax

UNIT:ALTitude

Parameter: <name>

Short form UNIT:ALT
Function: Selects altitude units.

Query Syntax

UNIT:ALTitude?

Short form: UNIT:ALT?
Function: Queries what altitude units are selected
Response: SUFFIX DATA

Description

This command selects the altitude units. The query gets the altitude units are selected.

Note: If the instrument is not in aero mode it returns an error: -221. Settings conflict.

UNIT:CONV

Applicability: PACE1000, PACE6000

Command Syntax

UNIT:CONVert

Parameter: <name>

Short form UNIT:CONV
Function: Converts units.

Query Syntax

n/a

Short form:
Function:
Response:

Description

This command converts units.

:UNIT:PRES

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:UNIT[x][:PRESsure] <name>

where x = 1 or 2 is the module number (default - 1)
Parameter: <name>
Valid units: Refer to :INStrument:UNIT for valid unit list.
Note: *Take care when entering the units, the letter 'O' can easily be mistaken for the number '0' or vice versa.*
Short form :UNIT <name>
Function: Selects pressure units

Query Syntax

:UNIT[x][:PRESsure]?

where x = 1 or 2 is the module number (default - 1)
Short form: :UNIT?
Function: Query what pressure units are selected
Response: name as above

Description

This command selects the current pressure units; USER1 to USER4 are the user defined units.

Example
TX> :UNIT:PRES?
RX> :UNIT:PRES MBAR
The current unit is mbar

It can be set to another unit, such as bar by sending:
TX> :UNIT:PRES bar
TX> :UNIT:PRES?
RX> :UNIT:PRES BAR
The reply to this query confirms the change, the front panel display can also confirm the change.

:UNIT:PRES:DEF

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:UNIT[x][:PRESsure]:DEFine[y] <string>,<number>

where: x = 1 or 2 is the module number (default - 1) and y = 1, 2, 3 or 4 is the number of user defined units (default - 1).

Parameter:	<string>	ASCII representation of unit name.
	<number>	Conversion factor from Pascals to required units

Short form: :UNIT[x]DEF[y] <string>,<number>

Function: There are four user defined units on the instrument. This command defines the name and conversion factor to use.

minimum	string	""
	number	0.0
maximum	string	8 characters
	number	1.0e10

Default:	string	N/A
	number	1000.0

Query Syntax

:UNIT[x][:PRESsure]:DEFine[y]?

where x = 1 or 2 is the module number (default - 1) and y = 1, 2, 3 or 4 is the number of user defined units (default - 1).

Short form: :UNIT[x]DEF[y]?

Function: Query conversion factor

Response: A string representing name and number corresponding to the conversion factor.

1	USER1
2	USER2
3	USER3
4	USER4

Description

This command defines the name and conversion factor for the special units. This conversion factor is from Pascals to the required units. An index of 1 is for USER1 unit and an index of 2 is for USER2 unit.

1. Example

y = 1, 2, 3 and 4 representing the four user-defined units.

By default:

TX> :UNIT:PRES:DEF?
RX> :UNIT:PRES:DEF "UserUnit1", 1000.0000000
TX> :UNIT:PRES:DEF1?
RX> :UNIT:PRES:DEF "UserUnit1", 1000.0000000
TX> :UNIT:PRES:DEF2?
RX> :UNIT:PRES:DEF2 "UserUnit2", 1000.0000000
TX> :UNIT:PRES:DEF3?
RX> :UNIT:PRES:DEF3 "UserUnit3", 1000.0000000
TX> :UNIT:PRES:DEF4?
RX> :UNIT:PRES:DEF4 "UserUnit4", 1000.0000000

The first parameter is the unit name and the second the equivalent pressure in Pascals.

2. Example - Defining 'MyUnit'

Setting USER4 as 'MyUnit':
TX> :UNIT:PRES:DEF4 "MyUnit", 2000.0
TX> :UNIT:PRES:DEF4?
RX> :UNIT:PRES:DEF4 "MyUnit", 2000.0000000

Select 'MyUnit' for use:
TX> :UNIT:PRES USER4
TX> :UNIT:PRES?
RX> :UNIT:PRES USER4
The front panel display shows "MyUnit" and uses it for measurement, set-point, full-scales and source pressures.

3. Example

The following query:
TX> :SENS:PRES?
RX> :SENS:PRES 5.0000187
Returns a number in 'MyUnit'

To confirm:
TX> :UNIT:PRES?
RX> :UNIT:PRES USER4

To find conversion factor for 'MyUnit'
TX> :UNIT:PRES:DEF4?
RX> :UNIT:PRES:DEF4 "MyUnit", 2000.0000000
Calculate equivalent in Pascals: 5.0000187 multiplied by 2000.0000000.

UNIT:SPE

Applicability: PACE1000, PACE6000

Command Syntax

UNIT:SPEed

Parameter: <name>

Short form UNIT:SPE
Function: Selects airspeed units.

Query Syntax

UNIT:SPEed?

Short form: UNIT:SPE?
Function: Queries the selected airspeed units.
Response: SUFFIX DATA

Description

This command selects the airspeed units. The query gets the selected airspeed units.

4.2 Common Commands

The commands identified with * are SCPI common commands.

*CLS

Command Syntax

*CLS

Parameter: none

Short form *CLS
Function: This command clears the all queues.

Query Syntax

n/a

Parameter:
Short form:
Function:

Description

Clears all event and condition register, see status reporting section.

*ESE

Command Syntax

*ESE <integer>

Parameter: integer 8 bit value of enable mask

Short form: *ESE <integer>
Function: Sets the Standard Event Status enable register.
minimum 0
maximum 255
Default: 0

Query Syntax

*ESE?

Short form: *ESE?
Function: Query Standard Event Status Enable register.
Response: 8 bit integer of contents of Standard Event Status Enable register.

Description

See Standard Event Group, section 3-2 and Figure 3-1.

*ESR

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

*ESR?

Short form: *ESR?
Function: Queries the Standard Event Status Register
Response: 8 bit integer of contents of Standard Event Status register.

Description

See Standard Event Group, section 3-2 and Figure 3-1.

*IDN?

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

*IDN?

Short form:	*IDN?
Function:	Queries the identification of the instrument.
Response:	A comma separated list containing manufacture, model, serial number and software version.

Description

Return identification

e.g., *IDN Druck,PACEnnnn,1234,01.00.00

where: n = PACE model number

*OPC

Command Syntax

*OPC

Parameter: None

Short form: *OPC
Function: When all operations have been completed, event 800 is written to the error/event queue (if enabled by STAT:QUE:ENAB). At the same time, this command allows the OPC bit in the standard event register to be set. This can be used to trigger a serial poll, see *ESR for details. If there are no pending operations these actions are performed immediately. This command should be sent each time the OPC bit is allowed to be set.

Default: -

Note:
*It is intended that the *OPC command should be the last command in a program message.*

Query Syntax

*OPC?

Short form: *OPC?
Function: When all operations have been completed a '1' is written to the output queue this can be used to trigger a serial poll, see *SRE for details.
Response: If there are no pending operations a '1' is written to the queue immediately.

Note:
*It is intended that the *OPC command should be the last command in a program message.*

Description

This command sets the OPC bit in the standard event register. The query returns '1' when all operations have been completed.

*SRE

Command Syntax

*SRE <integer>

Parameter:	integer	8 bit value of enable mask
Short form:	*SRE <integer>	
Function:	Sets the Service Request Enable register.	
	minimum	0
	maximum	255
Default:	0	

Query Syntax

*SRE?

Short form:	*SRE?
Function:	Query Service Request Enable register.
Response:	8 bit integer of contents of Service Request Enable register.

Description

See Status Byte Group, section 3-4 and Figure 3-1.

*STB?

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

*STB?

Short form: *STB?
Function: Queries the Status Register
Response: 8 bit integer of contents of Status register.

Description

See status reporting section.

*TST?

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

*TST?

Short form: *TST?
Function: Queries the self-test status.
Response: Returns ‘0’ indicates fail, ‘1’ indicates pass.

Description

On a self-test failure an error message can be stored in the error/event queue. The unit’s hardware status cannot be changed during that test.

*WAI

Command Syntax

*WAI

Parameter: None

Short form: *WAI
Function: No further commands are carried out until the completion of all pending operations. This command is ignored if all operations have been completed.

Query Syntax

N/A

Short form:
Function:
Response:

Description

This command stops further commands from being carried out until the completion of all pending operations.

4.3 Instrument Control Commands

The following commands identified with : are SCPI instrument control commands.

:GTL

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:GTL

Parameter: none

Short form :GTL
Function: Go to local

Query Syntax

n/a

Parameter:
Short form:
Function:

Description

Takes the instrument out of local lockout mode; the key-pad on the instrument becomes active.

:LLO

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:LOC

Parameter: none

Short form :LLO
Function: Local lock-out

Query Syntax

n/a

Parameter:
Short form:
Function:

Description

Locks the instrument out of local mode. The instrument will no longer be in local mode and cannot be operated from the front panel.

:LOC

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:LOC

Parameter: none

Short form :LOC
Function: Local mode

Query Syntax

n/a

Parameter:
Short form:
Function:

Description

Puts the instrument into local mode. The instrument will no longer be in remote mode and can be operated from the front panel.

:REM

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:REM

Parameter: none

Short form :REM
Function: Selects remote mode

Query Syntax

n/a

Parameter:
Short form:
Function:

Description

Sets the instrument in remote mode. The instrument will no longer be in local mode and cannot be operated from the front panel.

:SRQ

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SRQ

Parameter: none

Short form :SRQ
Function: Sends a service request to read messages.

Query Syntax

n/a

Parameter:
Short form:
Function:

Description

A service request causes all messages to be read from the register.

:SRQ:ENAB

Applicability: PACE1000, PACE5000, PACE6000

Command Syntax

:SRQ:ENABle

Parameter: none

Short form :SRQ:ENAB
Function: Sets a service request to read messages.

Query Syntax

:SRQ:ENABle?

Parameter: none
Short form: :SRQ:ENAB?
Function: Gets the messages in the register.

Description

A service request causes all messages to be read from the register.

SYN

Applicability: PACE1000, PACE6000

Command Syntax

n/a

Parameter:

Short form:
Function:

Query Syntax

SYN?

Short form: SYN?
Function: Queries synchronisation.
Response: <boolean>

0	-	not synchronised
1	-	synchronised

Description

Query for synchronisation purposes.

5 ERRORS

Negative error numbers are used for standard SCPI errors. Positive error numbers are device specific errors. Following the error number, a message describes the error. An error, when detected, is held in the error queue.
When SYST:ERR? is sent any error in the error queue sets the error bit in the event status register.

Error numbers
Each error detected causes an error number with an error message to be returned as follows:

Error code	Error message	Error code	Error message
-102	-102, "Syntax error"	-128	-128, "Numeric data not allowed"
-104	-104, "Data type error"	-130	-130, "Suffix error"
-108	-108, "Parameter not allowed"	-131	-131, "Invalid suffix"
-109	-109, "Missing parameter"	-134	-134, "Suffix too long"
-110	-110, "Command Header Error"	-138	-138, "Suffix not allowed"
-111	-111, "Header Separator Error"	-140	-140, "Character data error"
-112	-112, "Program mnemonic too long"	-141	-141, "Invalid character data"
-113	-113, "Undefined header"	-144	-144, "Character data too long"
-114	-114, "Header suffix out of range"	-148	-148, "Character data not allowed"
-120	-120, "Numeric data error"	-150	-150, "String data error"
-121	-121, "Invalid character in number"	-151	-151, "Invalid string data"
-123	-123, "Exponent too large"	-158	-158, "String data not allowed"
-124	-124, "Too many digits"		

Table 5-1 Errors -100 to -199

Error code	Error message	Error code	Error message
-200	-200, “Execution error”	-222	-222, “Data out of range”
-201	-201, “Invalid while in local”	-223	-223, “Too much data”
-202	-202, “Settings lost due to rtl”	-224	-224, “Illegal parameter value”
-203	-203, “Command protected”	-240	-240, “Hardware error”
-220	-220, “Parameter error”	-241	-241, “Hardware missing”
-221	-221, “Settings conflict”		

Table 5-2 Errors -200 to -299

Error code	Error message
-310	-310, “System error”
-350	-350, “Queue overflow”
-400	-400, “Query error”

Table 5-3 Errors -300 to -400

Error code	Error message
201	201, “Query only”
202	202, “No query allowed”
203	203, “Paramerter(s) not expected”
207	207, "Emumerated value not in union"
208	208, “Illegal number of parameters”
210	210, “Run out of memory handle”
211	211, “Unit not matched”
212	212, “Unit not required”

Table 5-4 Errors +201 to +212