

## Digital Oscilloscopes

**DOX2040**  
2-channel - 40 MHz - 1GSP/s

**DOX2050**  
2-channel - 50 MHz - 1GSP/s

**DOX2100**  
2-channel - 100 MHz - 1GSP/s

### Remote Programming



Pôle Test et Mesure de CHAUVIN-ARNOUX  
Parc des Glaisins - 6, avenue du Pré de Challes  
F - 74940 ANNECY-LE-VIEUX  
Tél. +33 (0)4.50.64.22.22 - Fax +33 (0)4.50.64.22.00

# Remote Control Manual DOX2000 Series Oscilloscopes

Revision 1.1

**METRIX - Division of CHAUVIN-ARNOUX**

## **Catalogue**

<i>Using Status Registers</i>	3
<i>About these Commands &amp; Queries</i>	3
<i>How are they listed ?</i>	3
<i>How they are described ?</i>	4
<i>When can they be used ?</i>	4
<i>Table of Commands &amp; Queries</i>	5
<i>Commands &amp; Queries</i>	11

## Using Status Registers

A wide range of status registers allows the oscilloscope's internal processing status to be determined quickly at any time. These registers and the instrument's status reporting system are designed to comply with IEEE 488.2 recommendations. Following an overview, starting this page, each of the registers and their roles are described.

Related functions are grouped together in common status registers. Some, such as the Status Byte Register (STB) or the Standard Event Status Register (ESR), are required by the IEEE 488.2 Standard. Other registers are device-specific, and include the Command Error Register (CMR) and Execution Error Register (EXR). Those commands associated with IEEE 488.2 mandatory status registers are preceded by an asterisk <\*>.

## About these Commands & Queries

This section lists and describes the remote control commands and queries recognized by the instrument. All commands and queries can be executed in either local or remote state.

The description for each command or query, with syntax and other information, begins on a new page. The name (header) is given in both long and short form at the top of the page, and the subject is indicated as a command or query or both. Queries perform actions such as obtaining information, and are recognized by the question mark (?) following the header.

## How they are listed ?

The descriptions are listed in alphabetical order according to their long form. Thus the description of ATTENUATION, whose short form is ATTN, is listed before that of AUTO SETUP, whose short form is ASET.

## How they are described?

In the descriptions themselves, a brief explanation of the function performed is given. This is followed by a presentation of the formal syntax, with the header given in Upper- and Lower-Case characters and the short form derived from it in ALL UPPER-CASE characters. Where applicable, the syntax of the query is given with the format of its response.

## When can they be used?

The commands and queries listed here can be used with [DOX2000 Series](#) digital instruments.

## Command Notation

The following notation is used in the commands:

- < > Angular brackets enclose words that are used as placeholders, of which there are two types: the header path and the data parameter of a command.
- : = A colon followed by an equal sign separates a placeholder from the description of the type and range of values that may be used in a command instead of the placeholder.
- { } Braces enclose a list of choices, which one must be performed.
- [ ] Square brackets enclose optional items.
- ... An ellipsis indicates that the items both to its left and right may be repeated a number of times.

As an example, consider the syntax notation for the command to set the vertical input sensitivity: <channel> :VOLT\_DIV <v\_gain>  
<channel> := {C1, C2, C3, C4} <v\_gain> := 2 mV to 10V

The first line shows the formal appearance of the command, with <channel> denoting the placeholder for the header path and <v\_gain> the placeholder for the data parameter specifying the desired vertical gain value. The second line indicates that one of four channels must be chosen for the header path. And the third explains that the actual vertical gain can be set to any value between 2 mV and 10 V.

## Table of Commands & Queries

<i>Short Form</i>	<i>Long Form</i>	<i>Subsystem</i>	<i>What the Command or Query does</i>
ALST?	ALL_STATUS?	STATUS	Reads and clears the content of all status registers.
ARM	ARM_ACQUISITION	ACQUISITION	Changes acquisition state from "stopped" to "single".
ATTN	ATTENUATION	ACQUISITION	Selects the vertical attenuation factor of the probe.
ACAL	AUTO_CALIBRATE	MISCELLANEOUS	Enables or disables automatic calibration.
ASET	AUTO_SETUP	ACQUISITION	Adjusts vertical, time base and trigger parameters.
AUTTS	AUTO_TYPESET	ACQUISITION	Selects the display type of automatic setup.
AVGA	AVERAGE_ACQUIRE	ACQUISITION	Selects the average times of average acquisition.
BWL	BANDWIDTH_LIMIT	ACQUISITION	Enables/disables the bandwidth-limiting low-pass filter.
BUZZ	BUZZER	MISCELLANEOUS	Controls the built-in piezo-electric buzzer.
*CAL?	*CAL?	MISCELLANEOUS	Performs complete internal calibration of the instrument.
CHDR	COMM_HEADER	COMMUNICATION	Controls formatting of query responses.
*CLS	*CLS	STATUS	Clears all status data registers.
CMR?	CMR?	STATUS	Reads and clears the Command error Register (CMR).
CONET	COMM_NET	COMMUNICATION	Specifies network addresses of scope and printers.
CPL	COUPLING	ACQUISITION	Selects the specified input channel's coupling mode.

CRMS	CURSOR_MEASURE	CURSOR	Specifies the type of cursor/parameter measurement.
CRST?	CURSOR_SET?	CURSOR	Allows positioning of any one of eight cursors.
CRVA?	CURSOR_VALUE?	CURSOR	Returns trace values measured by specified cursors.
CRAU	CURSOR_AUTO	CURSOR	Changes the cursor mode to auto mode.
CSVS	CSV_SAVE	SAVE/RECALL	Saves specified waveform data of CSV format to USB device.
COUN	COUNTER	FUNCTION	Enables or disables the cymometer to display on the screen.
CYMT	CYMOMETER	FUNCTION	Returns the current cymometer value displayed on the screen.
DATE	DATE	MISCELLANEOUS	Changes the date/time of the internal real-time clock.
DDR?	DDR?	STATUS	Clears the Device Dependent Register (DDR).
DEF	DEFINE?	FUNCTION	Specifies math expression for function evaluation.
DELF	DELETE_FILE	MASS STORAGE	Deletes files from mass storage.
DIR	DIRECTORY	MASS STORAGE	Creates and deletes file directories.
DTJN	DOT_JOIN	DISPLAY	Controls the interpolation lines between data points.
*ESE	*ESE	STATUS	Sets the Standard Event Status Enable register (ESE).
*ESR?	*ESR?	STATUS	Reads, clears the Event Status Register (ESR).
EXR?	EXR?	STATUS	Reads, clears the Execution error Register (EXR).
FLNM	FILENAME	MASS STORAGE	Changes default filenames.
FRTR	FORCE_TRIGGER	ACQUISITION	Forces the instrument to make one acquisition.

FVDISK	FORMAT_VDISK	MASS STORAGE	Reads the capability of the USB device.
FILT	FILTER	FUNCTION	Enables or disables the filter of specified source.
FILTS	FILT_SET	FUNCTION	Selects the type of filter, and sets the limit value of filter.
FFTW	FFT_WINDOW	FUNCTION	Selects the window of FFT.
FFTZ	FFT_ZOOM	FUNCTION	Selects the zoom in/out times of FFT trace.
FFTS	FFT_SCALE	FUNCTION	Selects the vertical scale of FFT trace.
FFTF	FFT_FULLSCREEN	FUNCTION	Enables or disables to display the FFT trace full screen.
GRDS	GRID_DISPLAY	DISPLAY	Selects the type of grid
GCSV	GET_CSV	WAVEFORMTRANS	Specifies waveform data format to controller.
HMAG	HOR_MAGNIFY	DISPLAY	Horizontally expands the selected expansion trace.
HPOS	HOR_POSITION	DISPLAY	Horizontally positions intensified zone's center.
HCSU	HARDCOPY_SETUP	HARD COPY	Configures the hard-copy driver.
*IDN?	*IDN?	MISCELLANEOUS	For identification purposes.
INTS	INTENSITY	DISPLAY	Sets the grid or trace/text intensity level.
ILVD	INTERLEAVED	ACQUISITION	Enables/disables Random Interleaved Sampling (RIS).
INR?	INR?	STATUS	Reads, clears Internal state change Register (INR).
INVS	INVERT_SET	DISPLAY	Invert the trace or the math waveform of specified source.
LOCK	LOCK	MISCELLANEOUS	Lock keyboard
MENU	MENU	DISPLAY	Enables or disables to display the current menu.
MTVP	MATH_VERT_POS	ACQUISITION	Controls the vertical position of math waveform of specified source.



MTVD	MATH_VERT_DIV	ACQUISITION	Controls the vertical sensitivity of math waveform of specified source.
MEAD	MEASURE_DELY	FUNCTION	Selects the type of delay measure.
OFST	OFFSET	ACQUISITION	Allows output channel vertical offset adjustment.
*OPC	*OPC	STATUS	Sets the OPC bit in the Event Status Register (ESR).
*OPT?	*OPT?	MISCELLANEOUS	Identifies oscilloscope options.
PACL	PARAMETER_CLR	CURSOR	Clears all current parameters in Custom, Pass/Fail.
PACU	PARAMETER_CUSTO M	CURSOR	Controls parameters with customizable qualifiers.
PAVA?	PARAMETER_VALUE ?	CURSOR	Returns current parameter,mask test values.
PDET	PEAK_DETECT	ACQUISITION	Switches the peak detector ON and OFF.
PERS	PERSIST	DISPLAY	Enables or disables the persistence display mode.
PESU	PERSIST_SETUP	DISPLAY	Selects display persistence duration.
PNSU	PANEL_SETUP	SAVE/RECALL	Complements the *SAV/*RST commands.
PFDS	PF_DISPLAY	FUNCTION	Enables or disables to display the test and the message options of pass/fail.
PFST	PF_SET	FUNCTION	Sets the X mask and the Y mask.
PFSL	PF_SAVELOAD	SAVE/RECALL	Saves or recalls the created mask setting.
PFCT	PF_CONTROL	FUNCTION	Selects the "operate", "output" and the "stop on output" which are the options of pass/fail.
PFCM	PF_CREATEM	FUNCTION	Creates the mask of the pass/fail.
PFDD	PF_DATEDIS	FUNCTION	Return the number of the pass/fail monitor which can be displayed on the screen.

*RCL	*RCL	SAVE/RECALL	Recalls one of five non-volatile panel setups.
REC	RECALL	WAVEFORMTRANS	Recalls a file from mass storage to internal memory.
RCPN	RECALL_PANEL	SAVE/RECALL	Recalls a front-panel setup from mass storage.
*RST	*RST	SAVE/RECALL	The *RST command initiates a device reset.
REFS	REF_SET	FUNCTION	Sets the reference waveform and its options.
*SAV	*SAV	SAVE/RECALL	Stores current state in non-volatile internal memory.
SCDP	SCREEN_DUMP	HARD COPY	Causes a screen dump to controller.
SCSV	SCREEN_SAVE	DISPLAY	Controls the automatic screen saver.
*SRE	*SRE	STATUS	Sets the Service Request Enable register (SRE).
*STB?	*STB?	STATUS	Reads the contents of IEEE 488.
STOP	STOP	ACQUISITION	Immediately stops signal acquisition.
STO	STORE	WAVEFORMTRANS	Stores a trace in internal memory or mass storage.
STPN	STORE_PANEL	SAVE/RECALL	Stores front-panel setup to mass storage.
STST	STORE_SETUP	WAVEFORMTRANS	Controls the way in which traces are stored.
SAST	SAMPLE_STATUS	ACQUISITION	Return the acquisition status of the scope
SARA	SAMPLE_RATE	ACQUISITION	Return the sample rate of the scope
SANU	SAMPLE_NUM	ACQUISITION	Return the number of sampled points available from last acquisition and the trigger position
SKEW	SKEW	ACQUISITION	Sets the skew of specified trace.
SET50	SETTO%50	FUNCTION	Sets the trigger level of the trigger source to the centre value of the signal "peak-to-peak" amplitude.
SXSA	SINXX_SAMPLE	ACQUISITION	Sets the type of the interpolation.
TDIV	TIME_DIV	ACQUISITION	Modifies the time base setting.

TMPL	TEMPLATE	WAVEFORM TRANSFER	Produces a complete waveform template copy.
TRA	TRACE	DISPLAY	Enables or disables the display of a trace.
*TRG	*TRG	ACQUISITION	Executes an ARM command.
TRCP	TRIG_COUPLING	ACQUISITION	Sets the coupling mode of the specified trigger source.
TRDL	TRIG_DELAY	ACQUISITION	Sets the time at which the trigger is to occur.
TRLV	TRIG_LEVEL	ACQUISITION	Adjusts the trigger level of the specified trigger source.
TRMD	TRIG_MODE	ACQUISITION	The trigger mode.
TRSE	TRIG_SELECT	ACQUISITION	Selects the condition that will trigger acquisition.
TRSL	TRIG_SLOPE	ACQUISITION	Sets the trigger slope of the specified trigger source.
UNIT	UNIT	ACQUISITION	Sets the unit of specified trace.
VPOS	VERT_POSITION	DISPLAY	Adjusts the vertical position of the FFT trace.
VDIV	VOLT_DIV	ACQUISITION	Sets the vertical sensitivity.
VTCL	VERTICAL	ACQUISITION	Controls the vertical position of the slope trigger line.
WF	WAVEFORM	WAVEFORMTRANS	Gets the waveform from the instrument.
WFSU	WAVEFORM_SETUP	WAVEFORMTRANS	Specifies amount of waveform data to go to controller.
WAIT	WAIT	ACQUISITION	Prevents new analysis until current has been completed.
XYDS	XY_DISPLAY	DISPLAY	Enables or disables to display the XY format

---

# Commands & Queries

<b>STATUS</b>	<b>ALL_STATUS?, ALST?</b> Query
<b>DESCRIPTION</b>	The ALL_STATUS? Query reads and clears the contents of all status registers: STB, ESR, INR, DDR, CMR, EXR and URR except for the MAV bit (bit 6) of the STB register. For an interpretation of the contents of each register, refer to the appropriate status register. The ALL_STATUS? Query is useful in a complete overview of the state of the instrument.
<b>QUERY SYNTAX</b>	ALL_Status?
<b>RESPONSE FORMAT</b>	ALL_Status STB,<value>,ESR,<value>,INR,<value>,DDR,<value>,CMR,<value>,EXR,<value>,URR,<value> <value> : = 0 to 65535
<b>EXAMPLE</b>	The following instruction reads the contents of all status registers:  Command message: ALST?  Response message: ALST STB, 0, ESR, 52, INR, 5, DDR, 0, CMR, 4, EXR, 24, URR, 0
<b>RELATED COMMANDS</b>	*CLS, CMR? , DDR? , *ESR? , EXR? , *STB? , URR?

**ACQUISITION**      **ARM\_ACQUISITION, ARM**  
Command

**DESCRIPTION**      The ARM\_ACQUISITION command enables the signal acquisition process by changing the acquisition state (trigger mode) from "stopped" to "single".

**COMMAND SYNTAX**      ARM acquisition

**EXAMPLE**      The following command enables signal acquisition:

Command message:  
ARM

**RELATED COMMANDS**      STOP, \*TRG, TRIG\_MODE, WAIT

**ACQUISITION**                    **ATTENUATION, ATTN**  
Command/Query

**DESCRIPTION**                    The ATTENUATION command selects the vertical attenuation factor of the probe. Values of 1, 5, 10, 50, 100, 500 and 1000 may be specified.

The ATTENUATION? Query returns the attenuation factor of the specified channel.

**COMMAND SYNTAX** <channel>: ATTeNuation <attenuation>  
<channel> := {C1, C2, C3, C4}  
<attenuation> = {1, 5, 10, 50, 100, 500, 1000}

**QUERY SYNTAX**                    <channel>: ATTeNuation?

**RESPONSE FORMAT** <channel>: ATTeNuation <attenuation>

**EXAMPLE**                            The following command sets to 100 the attenuation factor of Channel 1:

Command message:  
C1:ATTN 100

**ACQUISITION**      **AUTO\_SETUP, ASET**  
Command

**DESCRIPTION**      The AUTO\_SETUP command attempts to identify the waveform type and automatically adjusts controls to produce a usable display of the input signal.

**COMMAND SYNTAX**    AUTO\_SETUP

**EXAMPLE**            The following command instructs the oscilloscope to perform an auto-setup:

Command message:  
ASET

**RELATED COMMANDS**    AUTTS

**ACQUISITION**            **AUTO\_TYPESET, AUTTS**  
Command/Query

**DESCRIPTION**            The AUTO\_TYPESET command selects the specified type of automatically adjusting which is used to display.

**COMMAND SYNTAX**    AUTO\_TYPESET <type>

<type> : = (SP,MP,RS,DRP,RC)  
SP means only one period to be displayed, MP means multiple periods to be displayed, RS means the waveform is triggered on the rise side, DRP means the waveform is triggered on the drop side, and RC means to go back to the state before auto set.

**QUERY SYNTAX**        AUTO\_TYPESET?

**RESPONSE FORMAT**    AUTO\_TYPESET <type>

**EXAMPLE**                The following command sets the type of automatic adjustment to multiple periods:  
Command message:  
AUTTS MP

**RELATED COMMANDS**    ASET



**ACQUISITION**            **AVERAGE\_ACQUIRE, AVGA**  
                                  Command/Query

**DESCRIPTION**            The AVERAGE\_ACQUIRE command selects the average times of average acquisition.  
  
                                  The response to the AVERAGE\_ACQUIRE query indicates the times of average acquisition.

**COMMAND SYNTAX**      AVERAGE\_ACQUIRE <time>  
  
                                  <time> := {4, 16, 32, 64, 128, 256}

**QUERY SYNTAX**            AVERAGE\_ACQUIRE?

**RESPONSE FORMAT**      AVERAGE\_ACQUIRE <time>

**EXAMPLE**                The following command turns the average times of average acquisition to 16:

Command message:  
AVGA 16

---

<b>ACQUISITION</b>	<b>BANDWIDTH_LIMIT, BWL</b> Command/Query
<b>DESCRIPTION</b>	BANDWIDTH_LIMIT enables or disables the bandwidth-limiting low-pass filter. If the bandwidth-limiting filters are on, it will limit the bandwidth to reduce display noise. When you turn Bandwidth Limit ON, the Bandwidth Limit value is set to $\approx 18$ MHz. It also filters the signal to reduce noise and other unwanted high frequency components. The response to the BANDWIDTH_LIMIT? Query indicates whether the bandwidth filters are on or off.
<b>COMMAND SYNTAX</b>	BandWidth_Limit <channel>, <mode> [, <channel>, <mode> [, <channel>, <mode> [, <channel>, <mode>]]] <channel> := {C1, C2, C3, C4} <mode> = {ON, OFF}
<b>QUERY SYNTAX</b>	BandWidth_Limit?
<b>RESPONSE FORMAT</b>	BandWidth_Limit <channel>, <mode> [, <channel>, <mode> [, <channel>, <mode> [, <channel>, <mode>]]]
<b>EXAMPLE</b>	The following turns on the bandwidth filter for all channels, when Global_BWL is on (as it is by default) The following turns the bandwidth filter on for Channel 1 only:  Command message: BWL C1, ON

**MISCELLANEOUS**    **BUZZER, BUZZ**  
Command/Query

**DESCRIPTION**        The BUZZER command enables or disables sound switch.  
The response to the BUZZER? query indicates whether the sound switch is enabled.

**COMMAND SYNTAX**   BUZZer <state>  
                         <state> = {ON, OFF}

**QUERY SYNTAX**      BUZZER?

**RESPONSE FORMAT**   BUZZER <state>

**EXAMPLE**             Sending the following code will let the oscilloscope turn on the sound switch.

Command message:  
BUZZ ON

<b>MISCELLANEOUS</b>	<b>*CAL?</b> Query
<b>DESCRIPTION</b>	The *CAL? query cause the oscilloscope to perform an internal self-calibration and generates a response.
<b>QUERY SYNTAX</b>	*CAL?
<b>RESPONSE FORMAT</b>	*CAL <diagnostics> <diagnostics> : = 0 0 = Calibration successful
<b>EXAMPLE</b>	The following instruction forces a self-calibration:  Command message: *CAL?  Response message: *CAL 0
<b>RELATED COMMANDS</b>	AUTO_CALIBRATE

**COMMUNICATION**    **COMM\_HEADER, CHDR**  
Command/ Query**DESCRIPTION**

The COMM\_HEADER command controls the way the oscilloscope formats responses to queries. There are three response formats: LONG, in which responses start with the long form of the header word; SHORT, where responses start with the short form of the header word; and OFF, for which headers are omitted from the response and units in numbers are suppressed.

Unless you request otherwise, the SHORT response format is used.

This command does not affect the interpretation of messages sent to the oscilloscope. Headers can be sent in their long or short form regardless of the COMM\_HEADER setting.

Querying the vertical sensitivity of Channel 1 may result in one of the following responses:

COMM_HEADER	RESPONSE
LONG	C1:VOLT_DIV 200E-3 V
SHORT	C1:VDIV 200E-3 V
OFF	200E-3

**COMMAND SYNTAX**    Comm\_HeaDeR <mode>  
<mode> := {SHORT, LONG, OFF}

**QUERY SYNTAX**        Comm\_HeaDeR?

**RESPONSE FORMAT**    Comm\_HeaDeR <mode>

**EXAMPLE**

The following code sets the response header format to SHORT:

Command message:  
CHDR SHORT

**STATUS**

**\*CLS  
Command**

**DESCRIPTION**

The \*CLS command clears all the status data registers.

**COMMAND SYNTAX**

\*CLS

**EXAMPLE**

The following command causes all the status data registers to be cleared:

Command message:  
\*CLS

**RELATED COMMANDS**

ALL\_STATUS, CMR, DDR, \*ESR, EXR,  
\*STB, URR

<b>STATUS</b>	<b>CMR?</b> Query
<b>DESCRIPTION</b>	The CMR? Query reads and clears the contents of the Command error Register (CMR)see table next page which specifies the last syntax error type detected by the instrument.
<b>QUERY SYNTAX</b>	CMR?
<b>RESPONSE FORMAT</b>	CMR <value> <value> : = 0 to 14
<b>EXAMPLE</b>	<p>The following instruction reads the contents of the CMR register:</p> <p>Command message: CMR?</p> <p>Response message: CMR 0</p>
<b>RELATED COMMANDS</b>	ALL_STATUS? ,*CLS

**ADDITIONAL INFORMATION**

---

Command Error Status Register Structure (CMR)

Value	Description
1	Unrecognized command/query header
2	Invalid character
3	Invalid separator
4	Missing parameter
5	Unrecognized keyword
6	String error
7	Parameter cannot allowed
8	Command String Too Long
9	Query cannot allowed
10	Missing Query mask
11	Invalid parameter
12	Parameter syntax error
13	Filename too long



**MISCELLANEOUS**

**COMM\_NET, CONET**  
Command/Query

**DESCRIPTION**

The COMM\_NET command changes the IP address of the oscilloscope's internal network interface.

The COMM\_NET? query returns the IP address of the oscilloscope's internal network interface.

**COMMAND SYNTAX**

COMM\_NET <ip\_add0>, <ip\_add1>,  
<ip\_add2>, <ip\_add3>  
< ip\_add >: = 0 to 255

**QUERY SYNTAX**

COMM\_NET?

**RESPONSE FORMAT**

COMM\_NET <ip\_add0>, <ip\_add1>,  
<ip\_add2>, <ip\_add3>

**EXAMPLE**

This instruction will change the IP address to 10.11.0.230:

Command message:  
CONET 10,11,0,230

**ACQUISITION****COUPLING, CPL**  
Command/Query**DESCRIPTION**

The COUPLING command selects the coupling mode of the specified input channel.

The COUPLING? query returns the coupling mode of the specified channel.

**COMMAND SYNTAX**

<channel>: CouPLing <coupling>  
<channel> : = {C1, C2, C3, C4}  
<coupling> : = {A1M, D1M, GND}

The A of the <coupling> is alternating current. The D of the <coupling> is direct current. 1M is the impedance of input.

**QUERY SYNTAX**

<channel>: CouPLing?

**RESPONSE FORMAT**

<channel>: CouPLing <coupling>

**EXAMPLE**

The following command sets the coupling of Channel 2 to D1M:

Command message:  
C2:CPL D1M

<b>CURSOR</b>	<b>CURSOR_MEASURE, CRMS</b> Command/Query
<b>DESCRIPTION</b>	<p>The CURSOR_MEASURE command Specifies the type of cursor or parameter measurement to be displayed.</p> <p>The CURSOR_MEASURE? query indicates which cursors or parameter measurements are currently displayed.</p>
<b>Notation</b>	<p>HREL Selected tract-cursor mode</p> <p>VREL Selected manual-cursor mode and set to voltage type</p> <p>AUTO Selected auto mode</p> <p>OFF Cursors and parameters off</p>
<b>COMMAND SYNTAX</b>	<p>CuRsor_MeaSure &lt;mode&gt; &lt;mode&gt;={ OFF,HREL,VREL,AUTO}</p>
<b>QUERY SYNTAX</b>	<p>CuRsor_MeaSure?</p>
<b>RESPONSE FORMAT</b>	<p>CuRsor_MeaSure &lt;mode&gt;</p>
<b>EXAMPLE</b>	<p>The following command <u>turns</u> cursor Function off:</p>
<u>Command message:</u>	<p>CRMS OFF</p>
<b>RELATED COMMANDS</b>	<p>CURSOR_VALUE, PARAMETER_VALUE</p>

**CURSOR****CURSOR\_SET, CRST**  
Command/Query**DESCRIPTION**

The CURSOR\_SET command allows the user to position any one of the eight independent cursors at a given screen location. The positions of the cursors can be modified or queried even if the required cursor is not currently displayed on the screen. When setting a cursor position, a trace must be specified, relative to which the cursor will be positioned.

The CURSOR\_SET? Query indicates the current position of the cursor(s). The values returned depend on the grid type selected.

Notation	
HREF	The time value of curA under Track cursor mode
HDIF	The time value of curB under Track cursor mode
VREF	The volt-value of curA under manual cursor mode
VDIF	The volt -value of curB under manual cursor mode
TREF	The time value of curA under manual cursor mode
TDIF	The time value of curB under manual cursor mode

**COMMANDSYNTAX**

<trace>:CuRsrOr\_SeT<cursor>,<position>[,<cursor>,<position>,<cursor>,<position>]

< trace > : = {C1, C2, C3, C4} <cursor>:  
=(HREF,HDIF,VREF,VDIF,TREF,TDIF) <position> := 0.1  
to 17.9 DIV (horizontal " track" mode ,  
the range of values is related to screen size)  
<position> := -4 to 4 DIV (vertical)

<position> := -9) to 9 DIV (horizontal " manual" mode ,  
the range of values is related to screen size)

**QUERY SYNTAX**

<trace>: CuRsrOr\_SeT? [<cursor>,...<cursor>]

<cursor>: =( HREF, HDIF, VREF, VDIF, TREF, TDIF)

**RESPONSE FORMAT**

<trace>:CuRsrOr\_SeT <cursor>,<position> [ ,

<cursor>,<position>,<cursor>,<position>]

**EXAMPLE**

The following command positions the VREF and VDIF cursors at +3 DIV and -1 DIV respectively, using C1 as a reference:

Command message:  
C1: CRST VREF, 3DIV, VDIF, -1DIV

**RELATED COMMANDS**

CURSOR\_MEASURE, CURSOR\_VALUE,  
PARAMETER\_VALUE

**CURSOR****CURSOR\_VALUE?, CRVA?**

Query

**DESCRIPTION**

The CURSOR\_VALUE? Query returns the values measured by the specified cursors for a given trace.  
(The PARAMETER\_VALUE? query is used to obtain measured waveform parameter values.)

Notation	
HREL	the cursor value under track cursor mode
VREL	the delta volt-value under manual cursor mode

**QUERY SYNTAX**

```
<trace>: CuRsr_ Value? [<mode>, ...<mode>]
<trace> := { C1, C2, C3, C4}
<mode> := { HREL, VREL }
```

**RESPONSE FORMAT**

```
<trace> : CuRsr_ Value HREL,
<B->T - A->T>, <B->V - A->V>, <A->T>,
<B->T>,
<B->V - A->V>/(<B->T - A->T)>
<trace> : CuRsr_ Value VREL, <delta_vert>
```

**EXAMPLE**

The following query reads the delta Volt value under manual cursor mode (VREL) on Channel 2:

Command message:  
C2:CRVA? VREL

Response message:  
C2:CuRsr\_ Value VREL 1.00V

**RELATED COMMANDS**

CURSOR\_SET, PARAMETER\_VALUE

**CURSOR**

**CURSOR\_AUTO, CRAU**  
Command

**DESCRIPTION**

The CURSOR\_AUTO command changes the cursor mode to auto mode

**COMMAND SYNTAX**

CRAU

**EXAMPLE**

The following code changes the cursor mode to auto mode

Command message:  
CRAU

**SAVE/RECALL**

**CSV\_SAVE, CSVS**

Command/Query

**DESCRIPTION**

The CSV\_SAVE command selects the specified option of storing CSV format waveform.

The CSV\_SAVE? query returns the option of storing waveform data of CSV format.

**COMMAND SYNTAX**

CSV\_SAVE DD,<DD>,SAVE,<state>

The option DD is the data depth which is saved as. The option

SAVE is that if the waveform data is stored with parameter.

<DD> :=(MAX,DIS) the meaning of MAX is saved as the maximum data depth. The meaning of DIS is saved as the data depth which is displayed on the screen

<save> := {OFF, ON}

**QUERY SYNTAX**

CSV\_SAVE?

**RESPONSE FORMAT**

CSV\_SAVE DD, <DD>, SAVE, <state>

**EXAMPLE**

The following command sets the save data depth as the maximum and "parameter" save to off

Command message:

CSV\_SAVE DD,MAX,SAVE,OFF



<b>FUNCTION</b>	<b>COUNTER, COUN</b> Command/Query
<b>DESCRIPTION</b>	The COUNTER command enables or disables the cymometer display on the screen of instrument. The response to the COUNTER? query indicates whether the cymometer is displayed on the screen of instrument.
<b>COMMAND SYNTAX</b>	COUNTER <state> < state > := {ON, OFF}
<b>QUERY SYNTAX</b>	COUNTER?
<b>RESPONSE FORMAT</b>	COUNTER < state >
<b>EXAMPLE</b>	The following command enables the cymometer display  Command message: COUN ON

<b>FUNCTION</b>	<b>CYMOMETER, CYMT</b> Query
<b>DESCRIPTION</b>	<p>The response to the CYMOMETER? query is the value of cymometer displayed on the screen of the instrument.</p> <p>When the signal frequency is less than 10Hz, it returns 10Hz.</p>
<b>QUERY SYNTAX</b>	CYMOMETER?
<b>RESPONSE FORMAT</b>	CYMOMETER <option>
<b>EXAMPLE</b>	<p>The following instruction returns the value of cymometer displayed on the screen of the instrument.</p> <p>Response message: CYMT 10Hz</p>



<b>STATUS</b>	<b>DDR?</b> Query
<b>DESCRIPTION</b>	The DDR? Query reads and clears the contents of the Device Dependent or device specific error Register (DDR). In the case of a hardware failure, the DDR register specifies the origin of the failure.
<b>QUERY SYNTAX</b>	DDR?
<b>RESPONSE FORMAT</b>	DDR <value> <value> : = 0 to 65535
<b>EXAMPLE</b>	<p>The following instruction reads the contents of the DDR register:</p> <p>Command message: DDR?</p> <p>Response message: DDR 0</p>
<b>RELATED COMMANDS</b>	ALL_STATUS?, *CLS

**FUNCTION**                      **DEFINE, DEF**  
Command/Query

**DESCRIPTION**                      The DEFINE command specifies the mathematical expression to be evaluated by a function.

**COMMAND SYNTAX**                  DEFine EQN,'<equation>'  
<equation> the mathematical expression

Function Equations	
<source1> + <source2>	Addition
<source1> - <source2>	Subtraction
<source1>*<source2>	Multiplication
<source1>/<source2>	Ratio
FFT(source x)	FFT

**QUERY SYNTAX**                      DEFine?

**RESPONSE FORMAT**                  DEFine EQN,'<equation>'

**EXAMPLE**                                  Command message:  
DEFine EQN,'C1\*C2'

**MASS STORAGE**

**DELETE\_FILE, DELF**  
Command

**DESCRIPTION**

The DELETE\_FILE command deletes files from the currently selected directory on mass storage.

**COMMAND SYNTAX**

DELeTe\_File DISK, <device>, FILE,'<filename>'  
<device>={UDSK}  
<filename> = a file of specified directory and the specified file should up to eight characters.

**EXAMPLE**

The following command deletes a front-panel setup from the directory named SETUP in a USB memory device:

Command message:  
DELF DISK, UDSK, FILE, '/ SETUP /001.SET'

**RELATED COMMANDS**

DIRECTORY

**MASS STORAGE****DIRECTORY, DIR**  
Command/Query**DESCRIPTION**

The DIRECTORY command is used to manage the creation and deletion of file directories on mass storage devices. It also allows selection of the current working directory and listing of files in the directory. The query response consists of a double-quoted string containing a DOS-like listing of the directory.

**COMMAND SYNTAX**

Directory DISK, <device>, ACTION, <action>, '<directory>'

**QUERY SYNTAX**

Directory? DISK, <device> [, '<directory>']  
<device> = {UDSK}  
<action> = {CREATE, DELETE}  
<directory> = A legal DOS path or filename. (This can include the '/' character to define the root directory.)

**RESPONSE FORMAT**

DIRectory DISK, <device> "<directory>"

**EXAMPLE**

The following asks for a listing of the directory of a USB memory device:

Command message:  
DIR? DISK, UDSK

Response message:  
DIRectory DISK, UDSK,"A":  
DOX2100  
DOX2100AA  
BB.SET 2.00 KB  
DOX00001.SET 2.00 KB  
DOX00002.SET 2.00 KB

3 File(s), 2 DIR(s)

**RELATED COMMANDS**

DELF

<b>DISPLAY</b>	<b>DOT_JOIN, DTJN</b> Command/Query
<b>DESCRIPTION</b>	The DOT_JOIN command controls the interpolation lines between data points.
<b>COMMAND SYNTAX</b>	DoT_JoiN <state> <state> := {ON, OFF}
<b>QUERY SYNTAX</b>	DoT_JoiN?
<b>RESPONSE FORMAT</b>	DoT_JoiN <state>
<b>EXAMPLE</b>	The following instruction turns off the interpolation lines:  Command message: DTJN OFF



<b>STATUS</b>	<b>*ESE</b> Command/Query
<b>DESCRIPTION</b>	The *ESE command sets the Standard Event Status Enable register (ESE). This command allows one or more events in the ESR register to be reflected in the ESB summary message bit (bit 5) of the STB register.
<b>COMMAND SYNTAX</b>	*ESE <value> <value> : = 0 to 255
<b>QUERY SYNTAX</b>	*ESE?
<b>RESPONSE FORMAT</b>	*ESE <value>
<b>EXAMPLE</b>	<p>The following instruction allows the ESB bit to be set if a user request (URQ bit 6, i.e. decimal 64) and/or a device dependent error (DDE bit 3, i.e. decimal 8) occurs. Summing these values yields the ESE register mask <math>64+8=72</math>.</p> <p>Command message: *ESE 72</p>
<b>RELATED COMMANDS</b>	*ESR

<b>STATUS</b>	<b>*ESR?</b> Query
<b>DESCRIPTION</b>	The *ESR? query reads and clears the contents of the Event Status Register (ESR). The response represents the sum of the binary values of the register bits 0 to 7.
<b>QUERY SYNTAX</b>	*ESR?
<b>RESPONSE FORMAT</b>	*ESR <value> <value> : = 0 to 255
<b>EXAMPLE</b>	<p>The following instruction reads and clears the contents of the ESR register:</p> <p>Command message: *ESR?</p> <p>Response message: *ESR 0</p>
<b>RELATED COMMANDS</b>	ALL_STATUS, *CLS, *ESE

## ADDITIONAL INFORMATION

Standard Event Status Register (ESR)					
Bit	Bit Value	Bit Name	Description		Note
15~8			0	reserved by IEEE 488.2	
7	128	PON	1	Power OFF-to-ON transition as occurred	(1)
6	64	URQ	1	User Request has been issued	(2)
5	32	CME	1	Command parser Error has been detected	(3)
4	16	EXE	1	Execution Error detected	(4)
3	8	DDE	1	Device specific Error occurred	(5)
2	4	QYE	1	Query Error occurred	(6)
1	2	RQC	1	Instrument never requests bus control	(7)
0	1	OPC	1	Instrument never requests bus control	(8)

**Notes**

- (1) The Power On (PON) bit is always turned on (1) when the unit is powered up.
- (2) The User Request (URQ) bit is set true (1) when a soft key is pressed. An associated register URR identifies which key was selected. For further details refer to the URR? query.
- (3) The CoMmand parser Error bit (CME) is set true (1) whenever a command syntax error is detected. The CME bit has an associated CoMmand parser Register (CMR) which specifies the error code. Refer to the query CMR? for further details.
- (4) The EXecution Error bit (EXE) is set true (1) when a command cannot be executed due to some device condition (e.g. oscilloscope in local state) or a semantic error. The EXE bit has an associated Execution Error Register (EXR) which specifies the error code. Refer to query EXR? for further details.
- (5) The Device specific Error (DDE) is set true (1) whenever a hardware failure has occurred at power-up, or execution time, such as a channel overload condition, a trigger or a Timebase query.
- (6) The Query Error bit (QYE) is set true (1) whenever (a) an attempt is made to read data from the Output Queue when no output is either present or pending, (b) data in the Output Queue has been lost, (c) both output and input buffers are full (deadlock state), (d) an attempt is made by the controller to read before having sent an <END>, (e) a command is received before the response to the previous query was read (output buffer flushed).
- (7) The ReQuest Control bit (RQC) is always false (0), as the oscilloscope has no GPIB controlling capability.
- (8) The OPeration Complete bit (OPC) is set true (1) whenever \*OPC has been received, since commands and queries are strictly executed in sequential order. The oscilloscope starts processing a command only when the previous command has been entirely executed.

<b>STATUS</b>	<b>*EXR?</b> Query
<b>DESCRIPTION</b>	The EXR? query reads and clears the contents of the Execution error Register (EXR). The EXR register specifies the type of the last error detected during execution.
<b>QUERY SYNTAX</b>	EXR?
<b>RESPONSE FORMAT</b>	EXR <value> <value> : = to
<b>EXAMPLE</b>	<p>The following instruction reads the contents of the EXR register:</p> <p>Command message: EXR?</p> <p>Response message (if no fault): EXR 0</p>
<b>RELATED COMMANDS</b>	ALL_STATUS, *CLS

## ADDITIONAL INFORMATION

Execution Error Status Register Structure (EXR)	
Value	Description
21	Permission error. The command cannot be executed in local mode.
22	Environment error. The instrument is not configured to correctly process a command. For instance, the oscilloscope cannot be set to RIS at a slow Timebase.
23	Option error. The command applies to an option which has not been installed.
25	Parameter error. Too many parameters specified.
26	Non-implemented command.
32	Waveform descriptor error. An invalid waveform descriptor has been detected.
36	Panel setup error. An invalid panel setup data block has been detected.
50	No mass storage present when user attempted to access it.
53	Mass storage was "write" protected when user attempted to create, or to delete a file.
58	Mass storage file not found.
59	Requested directory not found.
61	Mass storage filename not DOS compatible, or illegal filename.
62	Cannot write on mass storage because filename already exists.

**MASS STORAGE      FILENAME, FLNM**  
Command/Query

**DESCRIPTION**      The FILENAME command is used to change the default filename given to any traces, setups and hard copies when they are being stored to a mass storage device.

**COMMAND SYNTAX**      FileNaMe TYPE, <type>, FILE, '<filename>'  
<type>:={ C1,C2,C3, C4, SETUP,TA, TB, TC, TD, HCOPIY}  
<filename> : = an alphanumeric string of up to 8 characters forming a legal DOS filename.

Note: the file's extension can be specified automatically by the oscilloscope.

**QUERY SYNTAX**      FileNaMe? TYPE, <type>  
<type> :={ ALL, C1, C2, C3, C4, SETUP, TA, TB, TC, TD, HCOPIY}

**RESPONSE FORMAT**      FileNaMe TYPE, <type>, FILE, "<filename>"  
[,TYPE, <type>, FILE, "<filename>"...]

**EXAMPLE**      The following command designates Channel 1 waveform files to be "TESTWF.DAV":

Command message:  
FLNM TYPE, C1, FILE, 'TESTWF'

**RELATED COMMANDS**      DIRECTORY, DELETE\_FILE

**ACQUISITION**

**FORCE\_TRIGGER, FRTR**  
Command

**DESCRIPTION**

Causes the instrument to make one acquisition.

**COMMAND SYNTAX**

Force\_TriGger

**EXAMPLE**

Either of the following pairs of instructions make one acquisition:

Command message1:  
TRMD SINGLE;ARM;FRTR

Command message2:  
TRMD STOP;ARM;FRTR



<b>MASS STORAGE</b>	<b>FORMAT_VDISK, FVDISK</b> Query
<b>DESCRIPTION</b>	The FORMAT_VDISK? query reads the capability of the USB memory device.
<b>QUERY SYNTAX</b>	Format_VDISK?
<b>RESPONSE FORMAT</b>	Format_VDISK <capability> <capability>:= the capability of the USB memory device.
<b>EXAMPLE</b>	<p>The following query reads the capability of the USB device.</p> <p>Command message: Format_VDISK?</p> <p>Response message: Format_VDISK 963 MB</p>

<b>FUNCTION</b>	<b>FILTER T</b>
<b>DESCRIPTION</b>	The FILTER command enables or disables filter of the specified trace.  The response to the FILTER? query indicates whether the filter of specified trace is enabled
<b>COMMAND SYNTAX</b>	<channel>:FILTER <state> <channel> := {C1,C2,C3,C4} <state> := {ON,OFF}
<b>QUERY SYNTAX</b>	<channel>:FILTER?
<b>RESPONSE FORMAT</b>	<channel>:FILTER
<b>EXAMPLE</b>	The following command enables the filter of  channel 1:  Command message: C1:FILT ON
<b>RELATED COMMANDS</b>	FILTS

<b>FUNCTION</b>	<b>FILT_SET, FILTS</b>
<b>DESCRIPTION</b>	<p>The FILT_SET command selects the specified type of filter, and sets the limit value of filter.</p> <p>The response to the FILT_SET? query indicates current parameter of the filter</p>
<b>COMMAND SYNTAX</b>	<pre>&lt;channel&gt;: FILT_SET TYPE,&lt;type&gt;, &lt;limit&gt;,&lt;limit_value&gt; &lt;channel&gt; : = {C1,C2,C3,C4} &lt;type&gt; : = {LP,HP,BP,BR} LP is lowpass, HP is highpass, BP is bandpass, BR is bandreject &lt;limit&gt; : = {UPPLIMIT,LOWLIMIT} if seted the &lt;limit&gt; and the &lt;type&gt; must be related</pre>
<b>QUERY SYNTAX</b>	<pre>&lt;channel&gt;: FILT_SET?</pre>
<b>RESPONSE FORMAT</b>	<pre>&lt;channel&gt;:FILTER TYPE,&lt;type&gt;,&lt;limit&gt;,&lt;limit_value &gt;</pre> <p>The following command changes the type of filter to bandpass, and sets the upplimit to 200 kHz and the lowlimit to 100 kHz:</p>
<b>EXAMPLE</b>	<pre>Command message: C1:FILTS TYPE,BP,UPPLIMIT,200kHz,LOWLIMIT,100kHz</pre>
<b>RELATED COMMANDS</b>	FILT

---

<b>FUNCTION</b>	<b>FFT_WINDOW, FFT W</b>
<b>DESCRIPTION</b>	The FFT_WINDOW command selects the window of FFT (Fast Fourier Transform algorithm). The response to the FFT_WINDOW? query indicates current window of FFT
<b>COMMAND SYNTAX</b>	FFT_WINDOW <window> < window > := {RECT,BLAC,HANN,HAMM} RECT is short for rectangle. BLAC is short for Blackman. HANN is short for hanning. HAMM is short for hamming.
<b>QUERY SYNTAX</b>	FFT_WINDOW ?
<b>RESPONSE FORMAT</b>	FFT_WINDOW,<window>
<b>EXAMPLE</b>	The following command sets the FFT window to hamming.  Command message: FFTW HAMM

<b>FUNCTION</b>	<b>FFT_ZOOM, FFT Z</b>
<b>DESCRIPTION</b>	<p>The FFT_ZOOM command selects the specified zoom of FFT.</p> <p>The response to the FFT_ZOOM? query indicates current zoom in/out times of FFT</p>
<b>COMMAND SYNTAX</b>	<p>FFT_ZOOM &lt;zoom&gt; &lt; zoom &gt; : = {1,2,5,10}</p>
<b>QUERY SYNTAX</b>	FFT_ZOOM ?
<b>RESPONSE FORMAT</b>	FFT_ZOOM,<zoom>
<b>EXAMPLE</b>	<p>The following command sets the zoom factor of FFT to 1X:</p> <p>Command message: FFTZ 1</p>

<b>FUNCTION</b>	<b>FFT_SCALE, FFT S</b>
<b>DESCRIPTION</b>	<p>The FFT_SCALE command selects the specified scale of FFT(Fast Fourier Transform algorithm).</p> <p>The response to the FFT_SCALE? query indicates current vertical scale of FFT waveform.</p>
<b>COMMAND SYNTAX</b>	<p>FFT_SCALE &lt;scale&gt; &lt; scale &gt; := {VRMS,DBVRMS}</p>
<b>QUERY SYNTAX RESPONSE FORMAT</b>	<p>FFT_SCALE FFT_SCALE,&lt; scale &gt;</p>
<b>EXAMPLE</b>	<p>The following command turns the vertical scale of FFT to dBVrms:</p> <p>Command message:</p> <p>FFTS DBVRMS</p>

<b>FUNCTION</b>	<b>FFT_FULLSCREEN, FFT F</b>
<b>DESCRIPTION</b>	<p>The FFT_FULLSCREEN command enables or disables to display the FFT waveform full screen.</p> <p>The response to the FFT_FULLSCREEN? query indicates whether the FFT waveform is full screen displayed.</p>
<b>COMMAND SYNTAX</b>	<p>FFT_FULLSCREEN &lt;state&gt; &lt; state &gt; := {ON,OFF}</p>
<b>QUERY SYNTAX</b>	FFT_FULLSCREEN ?
<b>RESPONSE FORMAT</b>	FFT_FULLSCREEN < state >
<b>EXAMPLE</b>	<p>The following command enables to display the FFT waveform full screen:</p> <p>Command message: FFTF ON</p>

<b>DISPLAY</b>	<b>GRID_DISPLAY, GRDS</b>
<b>DESCRIPTION</b>	<p>The GRID_DISPLAY command selects the type of the grid which is used to display.</p> <p>The response to the GRID_DISPLAY? query indicates current type of the grid</p>
<b>COMMAND SYNTAX</b>	<p>GRID_DISPLAY &lt;type&gt; &lt; type &gt; : = {FULL,HALF,OFF}</p>
<b>QUERY SYNTAX</b>	GRID_DISPLAY ?
<b>RESPONSE FORMAT</b>	GRID_DISPLAY < type >
<b>EXAMPLE</b>	<p>The following command changes the type of grid to full grid:</p> <p>Command message: GRID_DISPLAY FULL</p>



**WAVEFORMTRANS GET\_CSV, GCS  
V**

**DESCRIPTION** The response to the GET\_CSV? query indicates current waveform of CSV format.

The GET\_CSV? query have two options to set. They are the same as the options of CSVS.

**QUERY SYNTAX** GET\_CSV DD,<DD>,<SAVE>,<state>

The option DD is the data depth of the CSV format waveform. The option SAVE is that if the waveform data have parameters.

<DD> = {MAX,DIS} the meaning of MAX is that the CSV waveform's depth is maximum. The meaning of DIS is that CSV waveform's depth is the data which is displayed on the screen.

<save> = {OFF,ON}

**RESPONSE FORMAT** The waveform data of CSV format

**EXAMPLE** The following command transfers the waveform data of CSV format to the controller. It has the maximum depth of waveform data with parameters information.

Command message:  
GET\_CSV DD,MAX,SAVE,ON

---

<b>DISPLAY</b>	<b>HOR_MAGNIFY, HMAG</b> Command/Query
<b>DESCRIPTION</b>	<p>The HOR_MAGNIFY command horizontally expands the selected expansion trace by a specified factor.</p> <p>Magnification factors not within the range of permissible values will be rounded off to the closest legal value.</p> <p>If the specified factor is too large for any of the expanded traces (depending on their current source), it is reduced to an acceptable value and only then applied to the traces. The VAB bit (bit 2) in the STB register is set when a factor outside the legal range is specified.</p> <p>The HOR_MAGNIFY query returns the current magnification factor for the specified expansion function.</p>
<b>COMMAND SYNTAX</b>	<pre>&lt;exp_trace&gt;: Hor_MAGnify &lt;factor&gt; &lt;exp_trace&gt;: = {TA, TB, TC, TD} &lt;factor&gt; : = 1 to 50,000,000 The range of &lt;factor&gt; is related to the current timebase and the range of the timebase</pre>
<b>QUERY SYNTAX</b>	<pre>&lt;exp_trace&gt; : Hor_MAGnify ?</pre>
<b>RESPONSE FORMAT</b>	<pre>&lt;exp_trace&gt;: Hor_MAGnify &lt;factor&gt;</pre>
<b>EXAMPLE</b>	<p>The following instruction horizontally magnifies Trace A (TA) by a factor of 5:</p> <p>Command message: TA: HMAG 5.00</p>
<b>RELATED COMMANDS</b>	HPOS

**DISPLAY****HOR\_POSITION, HPOS**  
Command/Query**DESCRIPTION**

The HOR\_POSITION command horizontally positions the geometric center of the intensified zone on the source trace. Allowed positions range from division -9 to 9. If this would cause the horizontal position of any expanded trace to go outside the left or right screen boundaries, the difference of positions is adapted and then applied to the traces.

The VAB bit (bit 2) in the STB register is set if a value outside the legal range is specified.

The HOR\_POSITION query returns the position of the geometric center of the intensified zone on the source trace.

**COMMAND SYNTAX**

<exp\_trace>: Hor\_Position <hor\_position>  
<exp\_trace>: = (TA, TB, TC, TD)  
<hor\_position>: = -9 to 9 DIV (The range of the value is the size of the screen). The range of the <hor\_position> is related to the magnification factors of command HMAG. When the range after magnifying is beyond what the screen could display, it will be adjusted to the proper value.

**QUERY SYNTAX**

<exp\_trace>: Hor\_Position?

**RESPONSE FORMAT**

<exp\_trace>: Hor\_Position <hor\_position>

**EXAMPLE**

The following instruction positions the center of the intensified zone on the trace currently viewed by Trace A (TA) at division 3:

Command message:  
TA: HPOS 3

**RELATED COMMANDS** HMAG

<b>HARD COPY</b>	<b>HARDCOPY_SETUP, HCSU</b> Command/Query
<b>DESCRIPTION</b>	The HARDCOPY_SETUP command configures the instrument's hard-copy driver.
<b>COMMAND SYNTAX</b>	HCSU PSIZE, <page_size>, ISIZE, <image_size>, FORMAT, <format>, BCKG, <bckg>, PRTKEY, <printkey> <page_size> : = { DEFAULT } <printkey> : = { SAVE, PRINT } <format> : = { PORTRAIT, LANDSCAPE } <bckg> : = { BLACK, WHITE } <image_size> : = { DEFAULT, A4, LETTER }.
<b>QUERY SYNTAX</b>	HCSU?
<b>RESPONSE FORMAT</b>	HCSU PSIZE, <page_size>, ISIZE, <image_size>, FORMAT, <format>, BCKG, <bckg>, PRTKEY, <printkey>
<b>EXAMPLE</b>	The following example selects PORTRAIT format, sets the size of the image to "6*8CM":  Command message: HCSU ISIZE, 6*8CM, FORMAT, PORTRAIT
<b>RELATED COMMANDS</b>	SCDP

**MISCELLANEOUS****\*IDN?**  
Query**DESCRIPTION**

The \*IDN? query is used for identification purposes. The response consists of four different fields providing information on the manufacturer, the scope model, the serial number and the firmware revision level.

**QUERY SYNTAX****\*IDN?****RESPONSE FORMAT**

**\*IDN METRIX\_<model>, <serial\_number>, <firmware\_level>**  
**<model> :** = A n eleven characters model identifier  
**<serial\_number> :** = A 14-digit decimal code  
**<firmware\_level> :** = similar to k.xx.yy.zz

**EXAMPLE**

This example issues an identification request to the scope:

Command message:  
**\*IDN?**

Response message:  
**\*IDN**  
**METRIX,,DOX2100,NEU00002130019, 5.01.02.27R1**

**DISPLAY**

**INTENSITY, INTS**  
Command/Query

**DESCRIPTION**

The INTENSITY command sets the intensity level of the grid or the trace.

The intensity level is expressed as a percentage (PCT). A level of 100 PCT corresponds to the maximum intensity whilst a level of 0 PCT sets the intensity to its minimum value. The minimum intensity value of the trace is 30PCT

The response to the INTENSITY? Query indicates the grid and trace intensity levels.

**COMMAND SYNTAX**

INTensity GRID, <value>, TRACE, <value>  
<value> : = 0(or 30) to 100 [PCT]

Note 1: Parameters are grouped in pairs. The first of the pair names the variable to be modified, whilst the second gives the new value to be assigned. Pairs may be given in any order and be restricted to those variables to be changed.

Note 2: The suffix PCT is optional.

**QUERY SYNTAX**

INTensity?

**RESPONSE FORMAT**

INTensity TRACE, <value>, GRID, <value>

**EXAMPLE**

The following instruction enables remote control of the intensity, and changes the grid intensity level to 75%:

Command message:  
INTS GRID, 75

<b>ACQUISITION</b>	<b>INTERLEAVED, ILVD</b> Command/Query
<b>DESCRIPTION</b>	<p>The INTERLEAVED command enables or disables random interleaved sampling (RIS) for timebase settings where both single shot and RIS mode are available.</p> <p>The response to the INTERLEAVED? Query indicates whether the oscilloscope is in RIS mode.</p>
<b>COMMAND SYNTAX</b>	InterLeaVeD <mode> <mode> := {ON, OFF}
<b>QUERY SYNTAX</b>	InterLeaVeD?
<b>RESPONSE FORMAT</b>	InterLeaVeD <mode>
<b>EXAMPLE</b>	<p>The following instructs the oscilloscope to use RIS mode:</p> <p>Command message: ILVD ON</p>
<b>RELATED COMMANDS</b>	TIME_DIV, TRIG_MODE

**STATUS****INR?**

Query

**DESCRIPTION**

The INR? query reads and clears the contents of the Internal state change Register (INR). The INR register (table below) records the completion of various internal operations and state transitions.

Note : This command only supports 0 bit and 13 bit.

Internal State Register Structure (INR)		
Bit	Bit Value	Description
15...14	0	Reserved for future use
13	8192	1 Trigger is ready
12	4096	1 Pass/Fail test detected desired outcome
11	2048	1 Waveform processing has terminated in Trace D
10	1024	1 Waveform processing has terminated in Trace C
9	512	1 Waveform processing has terminated in Trace B
8	256	1 Waveform processing has terminated in Trace A
7	128	1 A memory card, floppy or hard disk exchange has been detected
6	64	1 Memory card, floppy or hard disk has become full in "AutoStore Fill" mode
5	32	0 Reserved
	16	1 A segment of a sequence waveform has been acquired
3	8	1 A time-out has occurred in a data block transfer
2	4	1 A return to the local state is detected
1	2	1 A screen dump has terminated
0	1	1 A new signal has been acquired

**QUERY SYNTAX**

INR?

**RESPONSE FORMAT**

INR <value>  
 <value> : = 0 to 65535

**EXAMPLE**

If we send INR? query after having triggered the INR register:

Command message1:  
 INR?



Response message1:  
INR 8913

If we send INR? query while the instrument didn't trigger, the INR register:

Command message2:  
INR?

Response message2:  
INR 8912

If we send INR? query after having sent an INR? query and the mode of the instrument is STOP, the INR register:

Command message3:  
INR?

Response message3:  
INR 0

If we send INR? query while there is no and then make the instrument triggered, the INR register:

Command message4:  
INR?

Response message4:  
INR 1

**RELATED COMMANDS** ALL\_STATUS?, \*CLS

<b>DISPLAY</b>	<b>INVERTSET, INVS</b> Command/Query
<b>DESCRIPTION</b>	The INVERTSET command inverts the specified traces or the waveform of math.  The response to the INVERTSET? query indicates whether the specified waveform is invert.
<b>COMMAND SYNTAX</b>	<trace>:INVERTSET < state > < trace > : = {C1,C2,C3,C4,MATH}  < state >:= {ON,OFF}
<b>QUERY SYNTAX</b>	<trace>:INVERTSET?
<b>RESPONSE FORMAT</b>	<trace>:INVERTSET < state >
<b>EXAMPLE</b>	The following instruction inverts the trace of channel 1:  Command message: C1:INVS ON

**MISCELLANEOUS**

**LOCK, LOCK**  
Command/Query

**DESCRIPTION**

The LOCK command enables or disables the panel keyboard of the instrument.

When any command or query is executed in either local or remote state, the functions of the panel keys except "FORCE" are not available. When the panel keyboard of the instrument is locked, pressing "FORCE" key enables the panel keyboard function.

The LOCK? query returns the status of the panel keyboard of the instrument.

**COMMAND SYNTAX**

LOCK < status >  
<status>:= {ON,OFF}

**QUERY SYNTAX**

LOCK?

**RESPONSE FORMAT**

LOCK < status >

**EXAMPLE**

The following instruction enables the functions of the panel keys:

Command message:  
LOCK ON

<b>DISPLAY</b>	<b>MENU, MENU</b> Command/Query
<b>DESCRIPTION</b>	The MENU command enables or disables to display the menu. The response to the MENU? query indicates whether the menu is displayed.
<b>COMMAND SYNTAX</b>	MENU < status > <status>:= {ON,OFF}
<b>QUERY SYNTAX</b>	MENU?
<b>RESPONSE FORMAT</b>	MENU < status >
<b>EXAMPLE</b>	The following instruction enables the display of the menu:  Command message: MENU ON

**ACQUISITION**

**MATH\_VERT\_POS, MTVP**  
Command/Query

**DESCRIPTION**

The MATH\_VERT\_POS command controls the vertical position of the math waveform with specified source.

The FFT waveform isn't included. But we have another command which is called VPOS to control its vertical position.

The response to the MATH\_VERT\_POS? query indicates the value of the vertical position of the math waveform.

**COMMAND SYNTAX**

MATH\_VERT\_POS <position>  
<position>:= the position is related to the screen center. For example, if we set the position of MTVP to 25. The math waveform will be displayed 1 grid up to the vertical center of the screen. Namely one grid is 25.

**QUERY SYNTAX**

MATH\_VERT\_POS?

**RESPONSE FORMAT**

MATH\_VERT\_POS < position >

**EXAMPLE**

The following instruction changes the vertical position of the math waveform to 1 grid up to the screen vertical centre:

Command message:  
MTVP 25

<b>ACQUISITION</b>	<b>MATH_VERT_DIV, MTVD</b> Command/Query
<b>DESCRIPTION</b>	<p>The MATH_VERT_DIV command controls the vertical sensitivity of the math waveform of specified source. We can only set the value of existing</p> <p>The FFT waveform isn't included.</p> <p>The response to the MATH_VERT_DIV? query indicates the specified scale of math waveform of specified source.</p>
<b>COMMAND SYNTAX</b>	MATH_VERT_DIV < scale > < scale >:= 1PV/div ~ 100V/div.
<b>QUERY SYNTAX</b>	MATH_VERT_DIV?
<b>RESPONSE FORMAT</b>	MATH_VERT_DIV < scale >
<b>EXAMPLE</b>	<p>The following instruction changes the vertical sensitivity of the math waveform of specified source to 1V/div.</p> <p>Command message: MTVD 1V</p>

<b>FUNCTION</b>	<b>MEASURE_DELY, MEAD</b> Command/Query
<b>DESCRIPTION</b>	<p>The MEASURE_DELY command selects the type of delay measure.</p> <p>The response to the MEASURE_DELY? query indicates the type of delay measure.</p>
<b>COMMAND SYNTAX</b>	<p>MEASURE_DELY SOURCE,&lt;mode&gt;,TYPE,&lt;type&gt; &lt;mode&gt;:= {C1-C2, C1-C3, C1-C4, C2-C3, C2-C4, C3-C4} &lt;type&gt;:= {PHA,FRR,FRF,FFR,FFF,LRR,LRF,LFR,LFF},</p> <p>The PHA is phase, the others are the same as the specified type of the instrument's delay measure</p>
<b>QUERY SYNTAX</b>	MEASURE_DELY?
<b>RESPONSE FORMAT</b>	MEASURE_DELY SOURCE,<mode>,TYPE,<type>
<b>EXAMPLE</b>	<p>The following instruction sets the type of delay measure to phase between C1 and C2.</p> <p>Command message: MEAD SOURCE,C1-C2,TYPE,PHA</p>

<b>ACQUISITION</b>	<b>OFFSET, OFST</b> Command/Query
<b>DESCRIPTION</b>	<p>The OFFSET command allows adjustment of the vertical offset of the specified input channel. The maximum ranges depend on the fixed sensitivity setting.</p> <p>If an out-of-range value is entered, the oscilloscope is set to the closest possible value and the VAB bit (bit 2) in the STB register is set.</p> <p>The OFFSET? query returns the offset value of the specified channel.</p>
<b>COMMAND SYNTAX</b>	<p>&lt;channel&gt;: OffSeT &lt;offset&gt; &lt;channel&gt; : = {C1, C2, C3,C4} &lt;offset&gt; : = See the <a href="#">DOX2000</a> specifications.</p>
<b>QUERY SYNTAX</b>	<channel>: OffSeT?
<b>RESPONSE FORMAT</b>	<channel>: OffSeT <offset>
<b>EXAMPLE</b>	<p>The following command sets the offset of Channel 2 to -3 V:</p> <p>Command message: C2: OFST -3V</p>



<b>STATUS</b>	<b>*OPC</b> Command/Query
<b>DESCRIPTION</b>	<p>The *OPC (Operation Complete) command sets to true the OPC bit (bit 0) in the standard Event Status Register (ESR). This command has no other effect on the operation of the oscilloscope because the instrument starts parsing a command or query only after it has completely processed the previous command or query.</p> <p>The *OPC? query always responds with the ASCII character "1" because the oscilloscope only responds to the query when the previous command has been entirely executed.</p>
<b>COMMAND SYNTAX</b>	*OPC
<b>QUERY SYNTAX</b>	*OPC?
<b>RESPONSE FORMAT</b>	*OPC 1

**MISCELLANEOUS** \*OPT  
Query

**DESCRIPTION** The \*OPT? query identifies oscilloscope options: installed software or hardware that is additional to the standard instrument configuration. The response consists of a series of response fields listing all the installed options.

**QUERY SYNTAX** \*OPT?

**RESPONSE FORMAT** \*OPT <option>

NOTE: If no option is present, the character 0 will be returned.

EXAMPLE The following instruction queries the installed options:

\*OPT?  
Return: \*OPT RS232,NET,USBTMC

***CURSOR***

**PARAMETER\_CLR, PACL**  
Command

**DESCRIPTION**

The PARAMETER\_CLR command clears the P/F test counter and starts it again at 0.

**COMMAND SYNTAX**

PArAmeter\_CLr

**RELATED COMMANDS**

PARAMETER\_VALUE PFDD

**CURSORS****PARAMETER\_CUSTOM, PACU**

Command/Query

**DESCRIPTION**

The PARAMETER\_CUSTOM command controls the parameters that have customizable qualifiers.

Note: The measured value of a parameter setup with PACU may be read using PAVA?

**COMMAND SYNTAX**

PParameter\_CUstom <line>,  
<parameter>,<qualifier><line> : = 1 to 5  
<parameter> : =(PKPK, MAX, MIN, AMPL, TOP, BASE,  
CMEAN, MEAN, RMS, CRMS, OVSN, FPRES,  
OVSP,  
RPRE, PER, FREQ, PWID, NWID, RISE, FALL, WID,  
DUTY, NDUTY )  
<qualifier> : = Measurement qualifier specific to  
each(source option)

**QUERY SYNTAX**

PParameter\_CUstom? <line>

**RESPONSE FORMAT**

PParameter\_Custom <line>,<parameter>,<qualifier>

**EXAMPLE**

Command Example  
PACU 2, PKPK, C1

Query/Response Examples  
PACU? 2 returns:  
PACU 2, PKPK, C1  
PAVA? CUST2 returns:  
C2: PAVA CUST2, 160.00mV

**RELATED**

COMMANDS PARAMETER\_CLR,  
PARAMETER\_VALUE

**CURSORS****PARAMETER\_VALUE?, PAVA?**

Query

**DESCRIPTION**

The PARAMETER\_VALUE query returns the measurement values.

Parameters Available on All Models			
ALL	all parameters	NDUTY	negative duty cycle
AMPL	amplitude	NWID	negative width
BASE	base	OVSN	negative overshoot
CMEAN	mean for cyclic waveform	OVSP	positive overshoot
CRMS	root mean square for cyclic part of waveform	PKPK	peak-to-peak
DUTY	duty cycle	PER	period
FALL	falltime	RPRE	(Vmin-Vbase)/ Vamp before the waveform rising transition
FREQ	frequency	PWID	positive width
FPRE	(Vmin-Vbase)/ Vamp before the waveform falling transition	RMS	root mean square
MAX	maximum	RISE	risetime
MIN	minimum	TOP	top
MEAN	mean	WID	width

**QUERY SYNTAX**

```
<trace>: PArAmeter_VAlue? [<parameter>, ... , <parameter>]
```

```
<trace>: = { C1, C2, C3, C4}
```

```
<parameter> := See table of parameter names on previous table.
```

**RESPONSE FORMAT**

```
<trace>: PArAmeter_VAlue <parameter>,<value> [ ... , <parameter>,<value>]
```

**EXAMPLE**

The following query reads the risetime of Channel 2

Command message:

```
C2: PAVA? RISE
```

Response message : C2: PAVA RISE, 3.6E-9S

**RELATED COMMANDS** CURSOR\_MEASURE, CURSOR\_SET,  
PARAMETER\_CUSTOM

<b>ACQUISITION</b>	<b>PEAK_DETECT, PDET</b> Command/Query
<b>DESCRIPTION</b>	The PEAK_DETECT command switches ON or OFF the peak detector built into the acquisition system. The PEAK_DETECT? query returns the current status of the peak detector.
<b>COMMAND SYNTAX</b>	Peak_DETEct <state> <state> : = {ON, OFF}
<b>QUERY SYNTAX</b>	Peak_DETEct?
<b>RESPONSE FORMAT</b>	PDET <state>
<b>EXAMPLE</b>	The following instruction turns on the peak detector:  Command message: PDET ON

<b>DISPLAY</b>	<b>PERSIST, PERS</b> Command/Query
<b>DESCRIPTION</b>	The PERSIST command enables or disables the persistence display mode.
<b>COMMAND SYNTAX</b>	PERSist <mode> <mode> := {ON, OFF}
<b>QUERY SYNTAX</b>	PERSist?
<b>RESPONSE FORMAT</b>	PERSist <mode>
<b>EXAMPLE</b>	<p>The following code turns the persistence display ON:</p> <p>Command message: PERS ON</p>
<b>RELATED COMMANDS</b>	PERSIST_SETUP



<b>DISPLAY</b>	<b>PERSIST_SETUP, PESU</b> Command/Query
<b>DESCRIPTION</b>	The PERSIST_SETUP command selects the persistence duration of the display, in seconds, in persistence mode. The PERSIST_SETUP? query indicates the current status of the persistence.
<b>COMMAND SYNTAX</b>	PErsist_SetUp <time> <time> =(1,2,5,Infinite)
<b>QUERY SYNTAX</b>	PErsist_SetUp?
<b>RESPONSE FORMAT</b>	PErsist_SetUp <time>
<b>EXAMPLE</b>	<p>The following instruction sets the variable persistence at 5 Seconds:</p> <p>Command message: PESU 5</p>
<b>RELATED COMMANDS</b>	PERSIST

**SAVE/RECALL SETUP**

**PANEL\_SETUP, PNSU**

Command/Query

**DESCRIPTION**

The PANEL\_SETUP command complements the \*SAV or \*RST commands. PANEL\_SETUP allows you to archive panel setups in encoded form on external storage media. Only setup data read by the PNSU? query can be recalled into the oscilloscope.

**COMMAND SYNTAX**

PaNel\_SetUp <setup>  
<setup> : = A setup previously read by  
PNSU?

**QUERY SYNTAX**

PaNel\_SetUp?

**RESPONSE FORMAT**

PaNel\_SetUp <setup>

**EXAMPLE**

The following instruction saves the oscilloscope's current panel setup in the file PANEL.SET:  
Command message:  
PNSU?

**RELATED COMMANDS**

\*RCL, \*SAV

<b>FUNCTION</b>	<b>PF_DISPLAY, PFDS</b> Command/Query
<b>DESCRIPTION</b>	<p>The PF_DISPLAY command enables or disables to turn on the test and display the message in the pass/fail option.</p> <p>The response to the PF_DISPLAY? query indicates whether the test is enabled and the message of pass/fail is displayed</p>
<b>COMMAND SYNTAX</b>	PF_DISPLAY TEST,<state>,DISPLAY,<state> <state> := {ON, OFF}
<b>QUERY SYNTAX</b>	PF_DISPLAY TEST?
<b>RESPONSE FORMAT</b>	PF_DISPLAY TEST <state>,DISPLAY,<state>
<b>EXAMPLE</b>	<p>The following instruction enables to turn on the test and display the message of pass/fail:</p> <p>Command message: PFDS TEST,ON,DISPLAY,ON</p>

<b>FUNCTION</b>	<b>PF_SET, PFST</b> Command/Query
<b>DESCRIPTION</b>	<p>The PF_SET command sets the X mask and the Y mask of the mask setting in the pass/fail option.</p> <p>The response to the PF_SET? query indicates the value of the X mask and the Y mask.</p>
<b>COMMAND SYNTAX</b>	PF_SET XMASK, <div>,YMASK, <div> <div> : = 0.04div~4.0div
<b>QUERY SYNTAX</b>	PF_SET?
<b>RESPONSE FORMAT</b>	PF_SET XMASK, <div>,YMASK, <div>
<b>EXAMPLE</b>	<p>The following instruction sets the X mask to 0.4div and the Y mask to 0.5div of the mask setting in the pass/fail option:</p> <p>Command message: PFST XMASK,0.4,YMASK,0.5</p>
<b>RELATED COMMANDS</b>	PFSL PFST

**SAVE/RECALL**

**PF\_SAVELOAD, PFSL**  
Command

**DESCRIPTION**

The PF\_SAVELOAD command saves or recalls the created mask setting.

**COMMAND SYNTAX**

PF\_SAVELOAD LOCATION,  
<location>,ACTION, <action>

The <location> means to save the created mask setting to the internal memories or the external memories.  
<location> := {IN,EX}  
IN means to save the mask setting to the internal memories while EX means the external memories.  
<action> := {SAVE,LOAD}

SAVE means to save the mask setting while LOAD means recall the stored mask setting.

**EXAMPLE**

The following instruction saves the mask setting to the internal memories:

Command message:  
PFSL LOCATION,IN,ACTION,SAVE

**RELATED COMMANDS**

PFCM

---

<b>FUNCTION</b>	<b>PF_CONTROL, PFCT</b> Command/Query
<b>DESCRIPTION</b>	<p>The PF_CONTROL command controls the pass/fail controlling options: "operate", "output" and the "stop on output".</p> <p>See instrument's Operator Manual for these options</p> <p>The response to the PF_CONTROL? query indicates the controlling options of the pass/fail.</p>
<b>COMMAND SYNTAX</b>	<pre>PF_CONTROL TRACE,&lt;trace&gt;,CONTROL,&lt;control&gt;,OUTPUT,&lt;output&gt;, OUTPUTSTOP,&lt;state&gt; &lt;trace&gt; := {C1,C2,C3,C4} &lt;control&gt; := {START,STOP} &lt;output&gt; := {FAIL,PASS} &lt;state&gt; := {ON,OFF}</pre>
<b>QUERY SYNTAX</b>	PF_CONTROL?
<b>RESPONSE FORMAT</b>	<pre>PF_CONTROL TRACE,&lt;trace&gt;,CONTROL,&lt;control&gt;, OUTPUT,&lt;output&gt;,OUTPUTSTOP,&lt;state&gt;</pre>
<b>EXAMPLE</b>	<p>The following instruction sets source to channel 1, "operate" to "start", "output" to "pass" and "stop on output" to "off":</p> <p>Command message: PFCT TRACE,C1,CONTROL,START, OUTPUT,PASS,OUTPUTSTOP,OFF</p>

<b>FUNCTION</b>	<b>PF_CREATEM, PFCM</b> Command
<b>DESCRIPTION</b>	The PF_CREATEM command creates the mask of the pass/fail.
<b>COMMAND SYNTAX</b>	PF_CREATEM
<b>EXAMPLE</b>	The following instruction creates the mask of the pass/fail.:  Command message: PFCM
<b>RELATED COMMANDS</b>	PFSL PFST

<b>FUNCTION</b>	<b>PF_DATADIS, PFDD</b> Query
<b>DESCRIPTION</b>	The PF_DATADIS? query returns the displayed_number of fail, pass and total .
<b>QUERY SYNTAX</b>	PF_DATADIS?
<b>RESPONSE FORMAT</b>	PF_DATADIS FAIL,<num>,PASS,<num>,total,<num>
<b>EXAMPLE</b>	The following instruction returns the number of displayed fail/pass/total:  Command message: PFDD FAIL,0,PASS,0,TOTAL,0
<b>RELATED COMMANDS</b>	PACL



**SAVE/RECALL****SETUP \*RCL**  
Command**DESCRIPTION**

The \*RCL command sets the instrument, using one of the ten non-volatile panel setups, by recalling the complete front-panel setup of the instrument.  
Panel setup 0 corresponds to the default panel setup.  
The \*RCL command produces the opposite effect of the \*SAV command.

If the desired panel setup is not acceptable, the Execution error status Register (EXR) is set and the EXE bit of the standard Event Status Register (ESR) is set.

**COMMAND SYNTAX**

\*RCL <panel\_setup>  
<panel\_setup>:= 0 to 20

**EXAMPLE**

The following recalls the instrument setup previously stored in panel setup 3:

Command message:  
\*RCL 3

**RELATED COMMANDS**

PANEL\_SETUP, \*SAV, EXR

**WAVEFORM TRANSFER    RECALL, REC**  
Command

**DESCRIPTION**                    The RECALL command recalls a waveform file from the current directory on mass storage into any or all of the internal memories M1 to M10.

**COMMAND SYNTAX**                <memory>: RECALL DISK, <device>, FILE, '<filename>'  
<memory> := {M1-M10}  
<device> := {UDSK}  
<filename> = A waveform file under a legal DOS path  
A filename-string of up to eight characters, with the extension  
".DAV" (This can include the '/' character to define the root directory).

**EXAMPLE**                            The following recalls a waveform file called "C1WF.DAV" from the memory card into Memory M1:

Command message:  
M1: REC DISK, UDSK FILE, 'C1WF.DAV'

**RELATED COMMANDS**                STORE, INR?

**SAVE/RECALL SETUP**

**RECALL\_PANEL, RCPN**  
Command

**DESCRIPTION**

The RECALL\_PANEL command recalls a front-panel setup from the current directory on mass storage.

**COMMAND SYNTAX**

ReCall\_PaNel DISK, <device>, FILE,  
'<filename>'  
<device> : = {UDSK}  
<filename> = A waveform file under a  
legal DOS path . A  
filename-string of up to eight characters, with the  
extension".SET"  
(This can include the '/' character to define the  
root directory).

**EXAMPLE**

The following recalls the front-panel setup from  
file SEAN.SET in a USB memory device:

Command message:  
RCPN DISK, UDSK, FILE,'SEAN.SET'

**RELATED COMMANDS**

PANEL\_SETUP, \*SAV, STORE\_PANEL,  
\*RCL

**SAVE/RECALL SETUP**    **\*RST**  
Command

**DESCRIPTION**            The \*RST command initiates a device reset.  
The \*RST sets recalls the default setup.

**COMMAND SYNTAX**      \*RST

**EXAMPLE**                This example resets the oscilloscope:

Command message:  
\*RST

**RELATED COMMANDS**    \*CAL, \*RCL

<b>FUNCTION</b>	<b>REF_SET, REFS</b> Command/Query
<b>DESCRIPTION</b>	<p>The REF_SET command sets the reference waveform and its options.</p> <p>The response to the REF_SET? query indicates whether the specified reference waveform is turned on.</p>
<b>COMMAND SYNTAX</b>	<p>REF_SET TRACE,&lt;trace&gt;REF,&lt;ref&gt;,state,&lt;state&gt;,SAVE,DO            &lt;trace&gt; := {C1,C2,C3,C4,C1OFF,C2OFF,C3OFF,C4OFF}            If the trace is closed, the specified trace will be CxOFF,(x is 1,2,3,4), the closed trace couldn't be saved or set            &lt;ref&gt; := {RA,RB,RC,RD}            The Rx(x is A,B,C,D) is that which one can be stored or displayed            &lt;state&gt; := {ON,OFF}            The state enables or disables to display the specified reference waveform.            If the command syntax have the option that SAVE,DO, means that the specified trace will be saved to the specified reference waveform.</p>
<b>QUERY SYNTAX</b>	REF_SET? REF,<ref>
<b>RESPONSE FORMAT</b>	REF_SET REF,<ref>,STATE,<state>
<b>EXAMPLE</b>	<p>The following instruction saves the channel 1 waveform to the REFA, and turns on REFA:</p> <p>Command message:            REFS TRACE,C1,REF,RA,STATE,ON,SAVE,DO</p>

**SAVE/RECALL SETUP**

**\*SAV**  
Command

**DESCRIPTION**

The \*SAV command stores the current instrument setup in internal memory. The \*SAV command stores the complete front-panel setup of the instrument at the time the command is issued.

**COMMAND SYNTAX**

\*SAV <panel\_setup>  
<panel\_setup>: = 1 to 20

**EXAMPLE**

The following saves the current instrument setup in Panel Setup 3:

Command message:  
\*SAV 3

**RELATED COMMANDS**

PANEL\_SETUP, \*RCL

**HARD COPY**

**SCREEN\_DUMP, SCDP**  
Command

**DESCRIPTION**

The SCREEN\_DUMP command is used to obtain the screen information of image format.

**COMMAND SYNTAX**

SCreen\_DumP

**EXAMPLE**

The following command transfers the screen information of image format to the controller

Command message:  
SCDP

<b>DISPLAY</b>	<b>SCREEN_SAVE, SCSV</b> Command/Query
<b>DESCRIPTION</b>	<p>The SCREEN_SAVE command controls the automatic Screen Saver, which automatically shuts down the internal color monitor after a preset time.</p> <p>The response to the SCREEN_SAVE? query indicates whether the automatic screen saver feature is on or off. Note: When the screen saver is in effect, the oscilloscope is still fully functional.</p>
<b>COMMAND SYNTAX</b>	SCreen_SaVe <enabled> <enabled> : = {YES, NO}
<b>QUERY SYNTAX</b>	SCreen_SaVe?
<b>RESPONSE FORMAT</b>	SCreen_SaVe <enabled>
<b>EXAMPLE</b>	<p>The following enables the automatic screen saver:</p> <p>Command message: SCSV YES</p>



**STATUS****\*SRE**  
Command/Query**DESCRIPTION**

The \*SRE command sets the Service Request Enable register (SRE). This command allows the user to specify which summary message bit(s) in the STB register will generate a service request.

A summary message bit is enabled by writing a '1' into the corresponding bit location. Conversely, writing a '0' into a given bit location prevents the associated event from generating a service request (SRQ). Clearing the SRE register disables SRQ interrupts.

The \*SRE? query returns a value that, when converted to a binary number, represents the bit settings of the SRE register.

Note: that bit 6 (MSS) cannot be set and its returned value is always zero.

**COMMAND SYNTAX**

\*SRE <value>  
<value> : = 0 to 255

**QUERY SYNTAX**

\*SRE?

**RESPONSE FORMAT**

\*SRE <value>

**EXAMPLE**

The following instruction allows an SRQ to be generated as soon as the MAV summary bit (bit 4, i.e. decimal 16) or the INB summary bit (bit 0, i.e. decimal 1) in the STB register, or both, are set. Summing these two values yields the SRE mask  $16+1 = 17$ .

Command message:  
\*SRE 17

**STATUS**

**\*STB?**  
Query

**DESCRIPTION**

The \*STB? query reads the contents of the 488.1 defined status register (STB), and the Master Summary Status (MSS). The response represents the values of bits 0 to 5 and 7 of the Status Byte register and the MSS summary message.

The response to a \*STB? Query is identical to the response of a serial poll except that the MSS summary message appears in bit 6 in place of the RQS message.

**QUERY SYNTAX**

\*STB?

**RESPONSE FORMAT**

\*STB <value>  
<value> : = 0 to 255

**EXAMPLE**

The following reads the status byte register:

Command message:  
\*STB?

Response message:  
\*STB 0

**RELATED COMMANDS** ALL\_STATUS, \*CLS, \*SRE

## ADDITIONAL INFORMATION

Status Byte Register (STB)				
Bit	Bit Value	Bit Name	Description	
7	128	DIO7	0 reserved for future use	
6	64	MSS/RQS MSS=1 RQS=1	at least 1 bit in STB masked by SRE is 1 service is requested	(1) (2)
5	32	ESB	1 an ESR enabled event has occurred	(3)
4	16	MAV	1 output queue is not empty	(4)
3	8	DIO3	0 reserved	
2	4	VAB	1 a command data value has been adapted	(5)
1	2	DIO1	0 reserved	
0	1	INB	1 an enabled INternal state change has occurred	(6)

Notes :

(1) The Master Summary Status (MSS) indicates that the instrument requests service, whilst the Service Request status — when set — specifies that the oscilloscope issued a service request. Bit position 6 depends on the polling method:

Bit 6 = MSS if an \*STB? Query is received  
= RQS if serial polling is conducted

(2) Example: If SRE=10 and STB=10 then MSS=1. If SRE=010 and STB=100 then MSS=0.

(3) The Event Status Bit (ESB) indicates whether or not one or more of the enabled IEEE 488.2 events have occurred since the last reading or clearing of the Standard Event Status Register (ESR). ESB is set if an enabled event becomes true (1).

(4) The Message Available bit (MAV) indicates whether or not the Output queue is empty. The MAV summary bit is set true (1) whenever a data byte resides in the Output queue.

(5) The Value Adapted Bit (VAB) is set true (1) whenever a data value in a command has been adapted to the nearest legal value. For instance, the VAB bit would be set if the timebase is redefined as 2  $\mu$ s/div since the adapted value is 2.5  $\mu$ s/div.

(6) The INternal state Bit (INB) is set true (1) whenever certain enabled internal states are entered. For further information, refer to the INR query.

**ACQUISITION**

**STOP**  
Command

**DESCRIPTION**

The STOP command immediately stops the acquisition of a signal. If the trigger mode is AUTO or NORM.

**COMMAND SYNTAX**

STOP

**EXAMPLE**

The following stops the acquisition process:

Command message:  
STOP

**RELATED COMMANDS**

ARM\_ACQUISITION, TRIG\_MODE, WAIT



**SAVE/RECALL SETUP**    **STORE\_PANEL, STPN**  
Command

**DESCRIPTION**            The STORE\_PANEL command stores the complete front-panel setup of the instrument, at the time the command is issued, into a file on the specified-DOS path directory in a USB memory device.

**COMMAND SYNTAX**        STore\_PaNel DISK, <device>, FILE,  
'<filename>'  
<device> := {UDSK}  
< directory > := A legal DOS path or filename.  
A filename-string of up to 8 characters, with the extension ".SET" (This can include the '/' character to define the root directory).

**EXAMPLE**                    The following code saves the current instrument setup to root directory of the USB memory device in a file called "SEAN.SET":

Command message:  
STore\_PaNel DISK,UDSK,FILE,'SEAN.SET'

The following code saves the current instrument setup to specified-directory of the USB memory device in a file called "SEAN.SET":

Command message:  
STore\_PaNel DISK,UDSK,FILE,'/AAA/SEAN'

**RELATED COMMANDS**    \*SAV, RECALL\_PANEL, \*RCL

<b>WAVEFORM TRANSFER</b>	<b>STORE_SETUP, STST</b> Command/Query
<b>DESCRIPTION</b>	The STORE_SETUP command controls the way in which traces will be stored. A single trace or all displayed traces may be enabled for storage.
<b>COMMAND SYNTAX</b>	STore_SeTup [<trace>, <dest>] <trace> : = {C1,C2,C3,C4,ALL_DISPLAYED }  <dest>: =(M1~M10,UDSK)
<b>QUERY SYNTAX</b>	STore_SeTup?
<b>RESPONSE FORMAT</b>	STore_SeTup <trace>, <dest>
<b>EXAMPLE</b>	The following command selects <sup>2</sup> Channel 1 to be stored.  Command message:  STST C1, UDSK
<b>RELATED COMMANDS</b>	STORE, INR

**ACQUISITION**

**SAMPLE\_STATUS, SAST**  
Query

**DESCRIPTION**

The SAST? query returns the acquisition status of the scope.

**QUERY SYNTAX**

SAST?

**RESPONSE FORMAT**

SAST < status >

**EXAMPLE**

The following command reads the acquisition status of the scope.

Command message:  
SAST?

Response message:  
SAST trig'd



**ACQUISITION**

**SAMPLE\_RATE, SARA**  
Query

**DESCRIPTION**

The SARA? query returns the sample rate of the scope.

**QUERY SYNTAX**

SARA?

**RESPONSE FORMAT**

SARA <value>

**EXAMPLE**

The following command reads the sample rate of the scope.

Command message:  
SARA?

Response message:  
SARA 500.0kSa

**ACQUISITION**                    **SAMPLE\_NUM, SANU**  
Query

**DESCRIPTION**                    The SANU? query returns the number of sampled points available from last acquisition and the trigger position.

**QUERY SYNTAX**                    SANU? <channel>

**RESPONSE FORMAT**                SANU <value>

**EXAMPLE**                            The following command reads the number of sampled points available from last acquisition from the Channel 2.

Command message:  
SANU? C2

Response message:  
SANU 6000

**ACQUISITION**

**SKEW, SKEW**  
Command

**DESCRIPTION**

The SKEW command sets the skew value of the specified trace.  
The response to the SKEW? query indicates the skew value of the specified trace.

**COMMAND SYNTAX**

<trace>:SKEW <skew>  
<trace> : = {C1,C2,C3,C4 }  
<skew>: = it is a value about time.

**QUERY SYNTAX**

<trace>:SKEW?

**RESPONSE FORMAT**

<trace>:SKEW <skew>

**EXAMPLE**

The following command sets channel 1 skew value to 3ns

Command message:  
C1:SKEW 3NS

<b>FUNCTION</b>	<b>SET50, SET50</b> Command
<b>DESCRIPTION</b>	The SET50 command sets the trigger level of the specified trigger source to the <u>mid-point</u> of the signal <u>peak-to-peak</u> amplitude.
<b>COMMAND SYNTAX</b>	SET50
<b>EXAMPLE</b>	<p>The following command sets the trigger level of the specified trigger source to the mid-point of the signal peak-to-peak amplitude.</p> <p>Command message: SET50</p>

¶

**ACQUISITION**            **SINXX\_SAMPLE, SXSA**  
Command/Query

**DESCRIPTION**            The SINXX\_SAMPLE command sets the way of interpolation.  
The response to the SINXX\_SAMPLE? query indicates the way of interpolation.

**COMMAND SYNTAX**      SINXX\_SAMPLE, <state>  
<state> : = {ON,OFF}  
ON means sine interpolation, and OFF means linear interpolation

**QUERY SYNTAX**         SINXX\_SAMPLE?

**RESPONSE FORMAT**     SINXX\_SAMPLE <state>

**EXAMPLE**                The following instruction sets the way of the interpolation to sine interpolation:

Command message:  
SXSA ON

**ACQUISITION**                    **TIME\_DIV, TDIV**  
Command/Query

**DESCRIPTION**

The TIME\_DIV command modifies the timebase setting. The new timebase setting may be specified with suffixes: NS for nanoseconds, US for microseconds, MS for milliseconds, S for seconds, or KS for kiloseconds. An out-of-range value causes the VAB bit (bit 2) in the STB register to be set. The TIME\_DIV? query returns the current timebase setting.

**COMMAND SYNTAX**            Time\_DIV <value>  
<value>:={1NS(not every type has this value),  
2.5NS,5NS,10NS,25NS,50NS,100NS,250NS,500NS,  
1US,2.5US,5US,10US,25US,50US,100US,250US,500US,  
1MS,2.5MS,5MS,10MS,25MS,50MS,100MS,250MS,500  
MS,1S,2.5S,5S,10S,25S,50S}

**QUERY SYNTAX**                Time\_DIV?

**RESPONSE FORMAT**          Time\_DIV <value>

**EXAMPLE**                        The following sets the time base to 500  $\mu$ s /div:

Command message:  
TDIV 500US

**RELATED COMMANDS** TRIG\_DELAY, TRIG\_MODE

**WAVEFORM TRANSFER TEMPLATE, TEMPL**  
Query

**DESCRIPTION** The TEMPLATE? query produces a copy of the template that describes the various logical entities making up a complete waveform. In particular, the template describes in full detail the variables contained in the descriptor part of a waveform.

**QUERY SYNTAX** TeMPLate?

**RESPONSE FORMAT** TeMPLate "<template>"  
<template> : = A variable length string detailing the structure of a waveform.

**RELATED COMMANDS** WF

**DISPLAY**                      **TRACE, TRA**  
Command/Query

**DESCRIPTION**                      The TRACE command enables or disables the display of a trace. An environment error is set if an attempt is made to display more than four waveforms.

The TRACE? query indicates whether the specified trace is displayed or not.

**COMMAND SYNTAX**                      <trace>: TRAcE <mode>  
<trace> : = {C1, C2, C3, C4, TA, TB, TC, TD}  
<mode> : = {ON, OFF}

**QUERY SYNTAX**                      <trace>: TRAcE?

**RESPONSE FORMAT**                      <trace>: TRAcE <mode>

**EXAMPLE**                                      The following command displays Channel 1 (C1):

Command message:  
C1: TRA ON



**ACQUISITION**

**\*TRG**  
Command

**DESCRIPTION**

The \*TRG command executes an ARM command.

**COMMAND SYNTAX**

\*TRG

**EXAMPLE**

The following command enables signal acquisition:

Command message:  
\*TRG

**RELATED COMMANDS** ARM\_ACQUISITION, STOP, WAIT

<b>ACQUISITION</b>	<b>TRIG_COUPLING, TRCP</b> Command/Query
<b>DESCRIPTION</b>	The TRIG_COUPLING command sets the coupling mode of the specified trigger source. The TRIG_COUPLING? query returns the trigger coupling of the selected source.
<b>COMMAND SYNTAX</b>	<trig_source>: TRig_CouPling <trig_coupling> <trig_source>: = {C1, C2, C3, C4, EX, EX5, LINE} <trig_coupling>: = {AC,DC,HFREJ,LFREJ}
<b>QUERY SYNTAX</b>	<trig_source>: TRig_CouPling?
<b>RESPONSE FORMAT</b>	<trig_source>: TRig_CouPling <trig_coupling>
<b>EXAMPLE</b>	The following command sets the coupling mode of the trigger source Channel 2 to AC:  Command message: C2: TRCP AC
<b>RELATED COMMANDS</b>	TRIG_COUPLING, TRIG_DELAY, TRIG_LEVEL, TRIG_MODE, TRIG_SELECT, TRIG_SLOPE

**ACQUISITION**

**TRIG\_DELAY, TRDL**  
Command/Query

**DESCRIPTION**

The TRIG\_DELAY command sets the time at which the trigger is to occur with respect to the first acquired data point.

This mode is called pre-trigger acquisition, as data are acquired before the trigger occurs. Negative trigger delays must be given in seconds. This mode is called post-trigger acquisition, as the data are acquired after the trigger has occurred.

If a value outside the range, the trigger time will be set to the nearest limit and the VAB bit (bit 2) will be set in the STB register. The response to the TRIG\_DELAY? Query indicates the trigger time with respect to the first acquired data point.

**COMMAND SYNTAX**

TRig\_DeLay <value>  
<value> = the range of value is related to the timebase.

Note: The suffix S is optional and assumed.

**QUERY SYNTAX**

TRig\_DeLay?

**RESPONSE FORMAT**

TRig\_DeLay <value>

**EXAMPLE**

The following command sets the trigger delay to -2ms (post-trigger):  
Command message: TRDL  
-2MS

**RELATED COMMANDS**

TIME\_DIV, TRIG\_COUPLING, TRIG\_LEVEL,  
TRIG\_MODE, TRIG\_SELECT, TRIG\_SLOPE

---

<b>ACQUISITION</b>	<b>TRIG_LEVEL, TRLV</b> Command/Query
<b>DESCRIPTION</b>	The TRIG_LEVEL command adjusts the trigger level of the specified trigger source. An out-of-range value will be adjusted to the closest legal value and will cause the VAB bit (bit 2) in the STB register to be set. The TRIG_LEVEL? query returns the current trigger level.
<b>COMMAND SYNTAX</b>	<trig_source>: TRig_LeVel <trig_level> <trig_source>: = {C1, C2, C3, C4, EX, EX5} <trig_level>: = -6DIV* volt/div to 6DIV * volt/div  Note: The suffix V is optional and assumed.
<b>QUERY SYNTAX</b>	<trig_source>: TRig_LeVel?
<b>RESPONSE FORMAT</b>	<trig_source>: TRig_LeVel <trig_level>
<b>EXAMPLE</b>	The following code adjusts the trigger level of Channel 3 to 52.00mv:  Command message: C3:TRig_LeVel 52.00mv
<b>RELATED COMMANDS</b>	TRIG_COUPLING, TRIG_DELAY, TRIG_MODE, TRIG_SELECT, TRIG_SLOPE

**ACQUISITION**

**TRIG\_MODE, TRMD**  
Command/Query

**DESCRIPTION**

The TRIG\_MODE command specifies the trigger mode.

The TRIG\_MODE? query returns the current trigger mode.

NOTE: STOP is a part of the option of this command, but is not a trigger mode of the instrument

**COMMAND SYNTAX**

TRig\_MoDe <mode>  
<mode>: = {AUTO, NORM, SINGLE,STOP}

**QUERY SYNTAX**

TRig\_MoDe

**RESPONSE FORMAT**

TRig\_MoDe <mode>

**EXAMPLE**

The following selects the normal mode:

Command message:  
TRMD NORM

**RELATED COMMANDS**

ARM\_ACQUISITION, STOP,  
TRIG\_SELECT,TRIG\_COUPLING,  
TRIG\_LEVEL, TRIG\_SLOP

**ACQUISITION****TRIG\_SELECT, TRSE**

Command/Query

**DESCRIPTION**

The TRIG\_SELECT command is used to set the

trigger type and the type's option HT which is an option of the TRIG\_SELECT command is related to the TRSL command. The TRSL command could set the <trig\_slope>. The HT's polarity will also be changed.

The TRIG\_SELECT? query returns the current trigger type.

**COMMAND SYNTAX**

TRig\_SelEct <trig\_type>,SR,<source>,HT,  
<hold\_type>,HV,<hold\_value>

TRig\_SelEct<trig\_type>,SR,<source>,CHAR,  
<characteristicse>,POL,<polarity>,SYNC,<sync\_type>,  
LINE,<line>

**OPTION**

TRig\_SelEct INTV,SR,<source>,VERT,<vertical>

<trig\_type>: = {EDGE, GLIT,INTV,TV,}  
GLIT means pulse trigger, INTV means slope  
trigger and TV means video trigger.

**Options:** SR HT HV POL CHAR SYNC  
LINE VERT

HT,<hold\_type>:is used to set pulse type.  
<hold\_type> : = {TI, PS, PL,PE, IS, IL,IE}  
TI means holdoff, PS means that the pulse width  
is smaller than the set value. PL means that the  
pulse width is larger than the set value. PE means  
that the pulse width is equal with the set value. If  
you want to set the Px(x is S,L,E), the  
<trig\_type> must be set to GLIT.  
IS means that the interval is smaller than the set  
value. IL means that the interval is larger than the  
set value is interval larger. IE means that the  
interval is equal with the set value. If you want to  
set the lx(x is S,L,E), the <trig\_type> must be set  
to INTV.  
HV,<hold\_value>:is used to set trigger time

<hold\_value> := See instrument Operator's Manual for valid values

**SR**, <source > := used to set the trigger's channel. If you want to set the other option. You must set it.

<source>: = {C1, C2, C3,C4,EX, EX5}

**CHAR**, <characteristics>:= used to set the standard .if you want to set it, the <trig\_type> must be set to TV.

<characteristics>:={NTSC, PALSEC}

**SYNC**,<sync\_type>:=is used to set sync. If you Want to set it. You must set <trig\_type> to TV

<sync\_type> : = {AL,LN,OF,EF}

AL means all lines; LN means line num; OF means odd field; EF means even field

**LINE**,<line>:=is used to set the line num. if you want to set it. The SYNC must be set to LINENUM

**POL**,<polarity>:= is used to set polarity. If you want to set it. You must set <trig\_type> to TV <polarity>:= {PO,NE}

PO means positive. NE means negative.

**VERT**,<vertical>:= is used to set vertical. If you want to set it. You must set <trig\_type> INTV

<vertical>:= {UP,DOWN,BOTH}

#### QUERY SYNTAX

TRig\_SelEct

#### RESPONSE FORMAT

TRig\_SelEct <mode>,the other options

#### EXAMPLE

The following sets the trigger type to video, the trigger source to C1,the standard to NTSC, the polarity to positive, the sync to line num and the line num to 5:  
TRSE TV,SR,C1,CHAR,NTSC,POL, PO,SYNC,LN,LINE,5

#### RELATED COMMANDS

TRSL VTCL

<b>ACQUISITION</b>	<b>TRIG_SLOPE, TRSL</b> Command/Query
<b>DESCRIPTION</b>	<p>The TRIG_SLOPE command sets the trigger slope of the specified trigger source.</p> <p>The TRIG_SLOPE? query returns the trigger slope of the selected source.</p>
<b>COMMAND SYNTAX</b>	<p>&lt;trig_source&gt;: TRIG_Slope &lt;trig_slope&gt; &lt;trig_source&gt;: = {C1, C2, C3, C4, EX,EX5 LINE}  &lt;trig_slope&gt;: = {NEG, POS, WINDOW}</p>
<b>QUERY SYNTAX</b>	<trig_source> : TRig_Slope?
<b>RESPONSE FORMAT</b>	<trig_source>: TRig_Slope <trig_slope>
<b>EXAMPLE</b>	<p>The following sets the trigger slope of Channel 2 to negative:</p> <p>Command message: C2: TRSL NEG</p>
<b>RELATED COMMANDS</b>	TRIG_COUPLING, TRIG_DELAY, TRIG_LEVEL, TRIG_MODE, TRIG_SELECT, TRIG_SLOPE



**ACQUISITION**            **UNIT, UNIT**  
Command/Query

**DESCRIPTION**            The UNIT command sets the unit of the specified trace.

                                 The UNIT query returns the unit of the specified trace.

**COMMAND SYNTAX**       <channel>: UNIT <type>  
                                 <channel> = {C1, C2, C3, C4}  
                                 <type> = {V,A}

**QUERY SYNTAX**           <channel> : UNIT

**RESPONSE FORMAT**      <channel>: UNIT <type>

**EXAMPLE**                 The following command sets the unit of the channel 1 to V:

Command message:  
C1: UNIT V

**DISPLAY**

**VERT\_POSITION, VPOS**  
Command/Query

**DESCRIPTION**

The VERT\_POSITION command adjusts the vertical position of the specified FFT trace on the screen. It does not affect the original offset value obtained at acquisition time.

The VERT\_POSITION? query returns the current vertical position of the specified FFT trace.

**COMMAND SYNTAX**

<trace>: Vert\_POSITION <display\_offset>  
<trace>: = {TA, TB, TC, TD}  
<display\_offset> = -40 DIV to 40 DIV

Note: The suffix DIV is optional.

**QUERY SYNTAX**

<trace>: Vert\_Position?

**RESPONSE FORMAT**

<trace>: Vert\_POSITION <display\_offset>

**EXAMPLE**

The following shifts FFT Trace A (TA) upwards by +3 divisions relative to the position at the time of acquisition:

Command message:  
TA: VPOS 3DIV

**ACQUISITION**            **VOLT\_DIV, VDIV**  
Command/Query

**DESCRIPTION**            The VOLT\_DIV command sets the vertical sensitivity in Volts/div. The VAB bit (bit 2) in the STB register is set if an out-of-range value is entered.

The VOLT\_DIV query returns the vertical sensitivity of the specified channel.

**COMMAND SYNTAX**      <channel>: Volt\_DIV <v\_gain>  
<channel> := {C1, C2, C3, C4}  
<v\_gain> = 2mV to 10V  
Note: The suffix V is optional.

**QUERY SYNTAX**         <channel> : Volt\_DIV ?

**RESPONSE FORMAT**    <channel>: Volt\_DIV <v\_gain>

**EXAMPLE**                The following command sets the vertical sensitivity of channel 1 to 50 mV/div.

Command message:  
C1: VDIV 50MV

**ACQUISITION**

**VERTICAL, VTCL**  
Command/Query

**DESCRIPTION**

The VERTICAL command controls the vertical position of the slope trigger line. It is related to the TRSE command. The VERT option of the TRSE command changes the controlling type of the slopes trigger line.

When the slope trigger lines are both controlled, the vertical position of the slope trigger line is the up one's position.

The VERTICAL query returns the vertical position of the slope trigger line.

**COMMAND SYNTAX**

<channel>: VERTICAL <pos>  
<channel> = {C1, C2, C3, C4}  
<pos> := the position is related to the screen vertical center. For example, if we set the vertical position of the slope trigger line to 25, it will be displayed 1 grid up to the screen vertical center. Namely one grid is 25.

**QUERY SYNTAX**

<channel> : VERTICAL ?

**RESPONSE FORMAT**

<channel>: VERTICAL <pos>

**EXAMPLE**

The following command sets the vertical position of the slope trigger line to 25 that what is the distance from the up of centre about 1 grid:

Command message:  
C1: VTCL 25

**RELATED COMMANDS** TRSE

**WAVEFORM  
TRANSFER****WAVEFORM, WF****Query****DESCRIPTION**

A WAVEFORM? Query transfers a waveform from the oscilloscope to the controller.

A waveform consists of several distinct entities:

1. the descriptor (DESC)
2. the user text (TEXT)
3. the time (TIME) descriptor
4. the data (DAT1) block, and, optionally
5. a second block of data (DAT2)

The WAVEFORM? Query instructs the oscilloscope to transmit a waveform to the controller. The entities may be queried independently. If the "ALL" parameter is specified, all four or five entities are transmitted in one block in the order enumerated above.

Note: The format of the waveform data depends on the current settings specified by the last WAVEFORM\_SETUP command.

**QUERY SYNTAX**

<trace>: WaveForm?

<trace> : = { C1,C2,C3,C4}

**RESPONSE FORMAT**

<trace>: WaveForm <waveform\_data\_block>

**EXAMPLE**

The following command reads waveform data block of Channel 2:

Command message:

**RELATED COMMANDS** WAVEFORM\_SETUP

**WAVEFORM  
TRANSFER****DESCRIPTION****WAVEFORM\_SETUP, WFSU**

Command/Query

The WAVEFORM\_SETUP command specifies the amount of data in a waveform to be transmitted to the controller. The command controls the settings of the parameters listed below.

Note: This command currently only support NP

Notation			
FP	first point	NP	number of points
SP	sparsing		

Sparsing (SP): The sparsing parameter defines the interval between data points. For example:

SP = 0 sends all data points

SP = 1 sends all data points

SP = 4 sends every 4th data point

Number of points (NP): The number of points parameter indicates how many points should be transmitted. For example:

NP = 0 sends all data

points NP = 1 sends 1

data point

NP = 50 sends a maximum of 50 data

points NP = 1001 sends a maximum of

1001 data points

First point (FP): The first point parameter specifies the address of the first data point to be sent. For waveforms acquired in sequence mode, this refers to the relative address in the given segment. For example:

FP = 0 corresponds to the first data point

FP = 1 corresponds to the second data point

FP = 5000 corresponds to data point 5001

The WAVEFORM\_SETUP? query returns the transfer parameters currently in use.

**COMMAND SYNTAX**

WaveForm\_SetUp SP, <sparsing>, NP, <number>, FP, <point>

**QUERY SYNTAX**

WaveForm\_SetUp ?

Note 1: After power-on, SP is set to 4, NP is set to 1000, and FP is set to 0.

Note 2: Parameters are grouped in pairs. The first of the pair names the variable to be modified, whilst the second gives the new value to be assigned. Pairs may be given in any order and may be restricted to those variables to be changed.

**RESPONSE FORMAT**

WaveForm\_SetUp SP, <sparsing>, NP, <number>, FP, <point>

**EXAMPLE**

The following command specifies that every 3rd data point (SP=3) starting at address 200 should be transferred:

Command message:  
WFSU SP, 3, FP, 200

**RELATED COMMANDS** WAVEFORM

**ACQUISITION**

**WAIT, WAIT**

Command

**DESCRIPTION**

The WAIT command prevents the instrument from analyzing new commands until the oscilloscope has completed the current acquisition.

The instrument will be waiting for trigger or the limit time over (if we set it) or the device time out when we sent this command.

**COMMAND SYNTAX**

WAIT <time>

Note: This command have two ways to use. One sets the limited time, another one doesn't set the limited time.

**EXAMPLE**

If we move the trigger level of the source to the position where the trace isn't triggered. Then we send an ARM command to set the trigger mode to single. Finally we send the WAIT command. The instrument will be waiting for triggering until the time over (if we set it) or time out.

If we move the trigger level of the source, and the instrument is triggered. Then we send an ARM command to set the trigger mode to single. Finally we send the WAIT command. The WAIT command will be finished if we send a FRTR for triggering.

Command message:  
WAIT



<b>DISPLAY</b>	<b>XY_DISPLAY, XYDS</b> Command/Query
<b>DESCRIPTION</b>	The XY_DISPLAY command enables or disables to display the XY format The response to the XY_DISPLAY? query indicates whether the XY format display is enabled.
<b>COMMAND SYNTAX</b>	XY_DISPLAY <state> <state> = {ON, OFF}
<b>QUERY SYNTAX</b>	XY_DISPLAY ?
<b>RESPONSE FORMAT</b>	XY_DISPLAY <state>
<b>EXAMPLE</b>	The following command enables to display the XY format:  Command message: XYDS

# Index

## A

ALL\_STATUS?, ALST?, Query,  
ARM\_ACQUISITION, ARM, Command,  
ATTENUATION, ATTN, Command/Query,  
AUTO\_CALIBRATE, ACAL, Command/Query,  
AUTO\_SETUP, ASET, Command,  
AUTO\_TYPESET, AUTTS, Command/Query,  
AVERAGE\_ACQUIRE, AVGA, Command/Query,

## B

BANDWIDTH\_LIMIT, BWL, Command/Query,  
BUZZER, BUZZ, Command,

## C

CAL?, Query,  
CLS, Command,  
CMR?, Query,  
COMM\_NET, CONET, Command/Query,  
COUPLING, CPL, Command/Query,  
CURSOR\_SET, CRST, Command/Query,  
CURSOR\_VALUE?, CRVA?, Query,  
CURSOR\_AUTO, CRAU, Command,  
CSV\_SAVE, CSVS, Command/Query,  
COUNTER, COUN, Command/Query,  
CYMOMETER, CYMT, Query,

## D

DATE, Command/Query,  
DDR?, Query,  
DEFINE, DEF, Command/Query,  
DELETE\_FILE, DELF, Command,  
DIRECTORY, DIR, Command/Query,  
DOT\_JOIN, DTJN, Command/Query,

## E

ESE, Command/Query,  
ESR?, Query,  
EXR?, Query,

**F**

FILENAME, FLNM, Command/Query,  
FORMAT\_VDISK, FVDISK, Query,  
FILTER, FILT, Command/Query,  
FLT\_SET, FILTS, Command/Query,  
FFT\_WINDOW, FFTW, Command/Query,  
FFT\_ZOOM, FFTZ, Command/Query,  
FFT\_SCALE, FFTS, Command/Query,  
FFT\_FULLSCREEN, FFTF, Command/Query,

**G**

GRID\_DISPLAY, GRDS, Command/Query,  
GCSV, GET\_CSV, Query,

**H**

HARDCOPY\_SETUP, HCSU,  
HOR\_MAGNIFY, HMAG, Command/Query,  
HOR\_POSITION, HPOS, Command/Query,

**I**

IDN?, Query,  
INTENSITY, INTS, Command/Query,  
INTERLEAVED, ILVD, Command/Query,  
INR, INR, Query,  
INVERT\_SET, INVS, Command/Query,

**L**

LOCK, Command/Query,

**M**

MENU, MENU, Command/Query,  
MATH\_VERT\_POS, MTVP, Command/Query,  
MATH\_VERT\_DIV, MTVD, Command/Query,  
MEASURE\_DELY, MEAD, Command/Query,

**O**

OFFSET, OFST, Command/Query,  
OPC, Command/Query,

**P**

PARAMETER\_CLR, PACL, Command,

PARAMETER\_CUSTOM, PACU, Command/Query,  
PARAMETER\_VALUE?, PAVA?, Query,  
PEAK\_DETECT, PDET, Command/Query,  
PERSIST, PERS, Command/Query,  
PERSIST\_SETUP, PESU, Command/Query,  
PANEL\_SETUP, PNSU, Command/Query,  
PF\_DISPLAY, PFDS, Command/Query,  
PF\_SET, PFST, Command/Query,  
PF\_SAVELOAD, PFSL, Command,  
PF\_CONTROL, PFCT, Command/Query,  
PF\_CREATEM, PFCM, Command,  
PF\_DATEDIS, PFDD, Query,

**R**

RCL, Command,  
RECALL, REC, Command,  
RECALL\_PANEL, RCPN, Command,  
RST, Command,  
REF\_SET, REFS, Command/Query,

**S**

SAV, Command,  
SCREEN\_DUMP, SCDP, Command/Query,  
SRE, Command/Query,  
STB? Query,  
STOP, Command,  
STORE, STO, Command,  
STORE\_PANEL, STPN, Command,  
STORE\_SETUP, STST, Command/Query,  
SAMPLE\_STATUS, SAST/ Query,  
SAMPLE\_RATE, SARA/ Query,  
SAMPLE\_NUM, SANU/ Query,  
SKEW, SKEW, Command,  
SETTO%50, SET50, Command,  
SINXX\_SAMPLE, SXSA, Command/Query,

**T**

TIME\_DIV, TDIV, Command/Query,  
TRACE, TRA, Command/Query,  
TRG, Command,  
TRIG\_COUPLING, TRCP, Command/Query,  
TRIG\_DELAY, TRDL, Command/Query,  
TRIG\_LEVEL, TRLV, Command/Query,  
TRIG\_MODE, TRMD, Command/Query,

## **DOX2000 Series**

---

TRIG\_SELECT, TRSE, Command/Query,  
TRIG\_SLOPE, TRSL, Command/Query,

### **U**

UNIT, UNIT, Command/Query,

### **V**

VOLT\_DIV, VDIV, Command/Query,  
VERTICAL, VTCL, Command/Query,

### **W**

WAIT, Command,  
WAVEFORM, WF, Command/Query,  
WAVEFORM\_SETUP, WFSU, Command/Query,

### **X**

XY\_DISPLAY, XYDS, Command/Query,