OX 8040
OX 8042 / OX 8062
OX 8050 / OX 8100

ANALOGUE / DIGITAL OSCILLOSCOPES

Remote programming manual

906121584 - Ed. 1 - 04/99

CONTENTS

1. INTRODUCTION

The programming instructions described in this manual comply with the IEEE 488.2 standard. They enable the user to control the instrument remotely.

OX 8050-GPIB OX 8100-GPIB	You can choose between GPIB (IEEE 488.1) and RS232C interfaces
OX 8040 OX 8050, OX 8100 OX 8042, OX 8062	Only the RS232C interface is available.

All the commands operate over both types of interface. However, some functions such as Device Clear and Group Executive Trigger are specific to the GPIB standard.

Communication between controller and oscilloscope can be used to:

* Configure the instrument
* Measure physical quantities
* Read data (curve, measurement, configuration)
* Send information (configuration)

2. IEEE 488.2 PROGRAMMING

This section describes the functionalities specific to the GPIB interface and its general concepts.

2.1. Capabilities

These are specified in the IEEE 488.1-1980 standard:

SH1	Source Handshake	Full capability
AH1	Acceptor Handshake	Full capability
T5	Talker	Full capability
L4	Listener	No listen only capability
SR1	Service Request	Full capability
RL1	Remote Local	Full capability
DC1	Device Clear	Full capability
DT1	Device Trigger	Full capability
C0	Controller	No capability

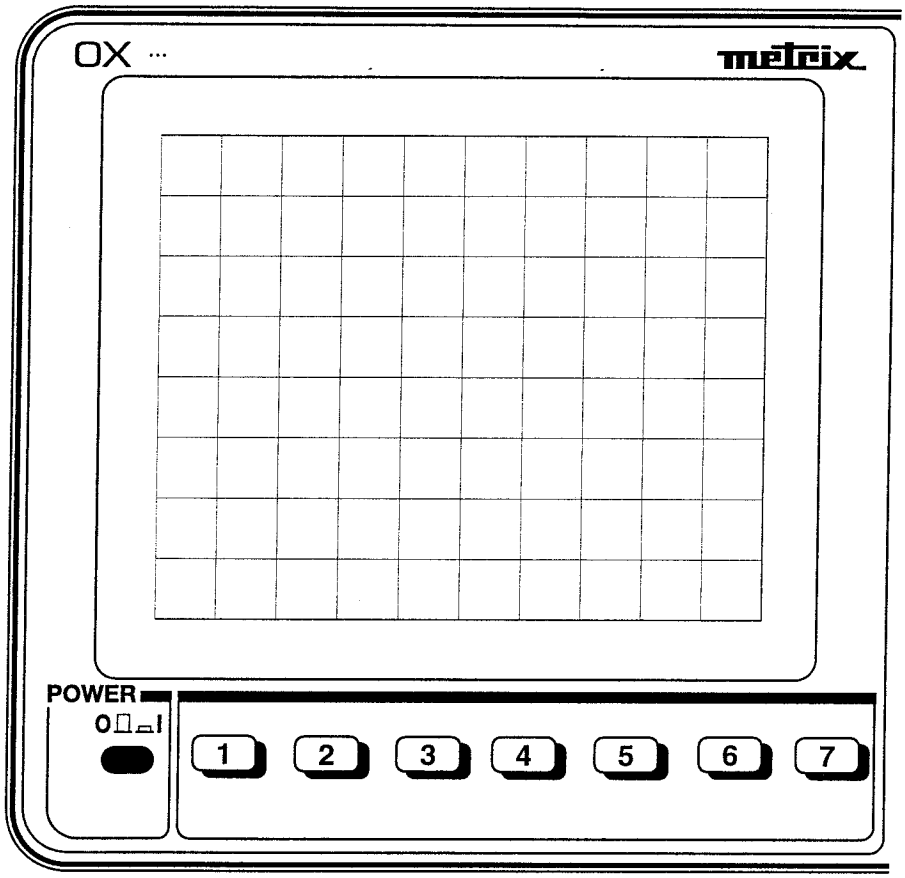
E2 : Three-state ports

2.2. Command and Data Modes

The interface has two operating modes:

* **Command mode**: The ATN line is true (asserted). This mode is used to define the talker and the listeners (via an IEEE code containing their addresses) and to send IEEE commands such as GET (Group Executive Trigger).

* **Data mode**: The ATN line is false (deasserted). This mode is used to transmit via the bus the common IEEE 488.2 commands or instrument-specific commands. All these are described in section 6.



MENU keys to select the communication interfaces

2.3. Addressing

Each instrument connected to the bus resides at an address between 0 and 30.
The controller specifies the talker and the listeners (ATN true) then sets up the dialogue (ATN false).

An instrument can be configured as a talker or a listener or can be "unaddressed" (untalk) by the controller.

An instrument configured as a talker remains a talker until:

* an interface clear message ("IFC" interface clear) is received,
or * another instrument is addressed as a talker ("OTA" Other instrument's Talk Address),
or * the talker is unaddressed ("UNT" untalk).

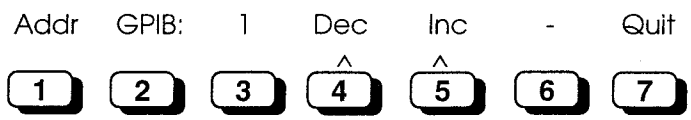
An instrument configured as a listener remains a listener until:

* an interface clear message ("IFC" interface clear) is received,
or * the instrument is addressed as a talker ("MTA" My Talk Address),
or * the listeners are unaddressed ("UNL" unlisten).

To configure the instrument's address, press **UTILITY** and select the IEEE interface in the main menu bar:

OX 8100 :

A new command line sets the address at a number between 0 and 30 :



Increment or decrement the address and then confirm by quitting.

OX 8050 :

To configure the instrument's address, press **UTILITY** and select the IEEE interface in the main menu bar:

2.4. IEEE 488 specific commands

These commands are transmitted over the bus when the ATN line is true (command mode):

Interface Clear (IFC)

This command stops all activity on the bus. It unaddresses all the listeners and the talker, cancels serial searches and returns control to the controller. Only the controller can manage this line.

Device Clear

This command (DCL, *Device Clear*, for a universal command or SDC, *Set Device Clear*, for an addressed command) clears the input and output buffers, cancels current commands and returns the syntax analyzer to the top of the tree structure.

Device Trigger

This command (GET, *Group Executive Trigger*) initiates digital acquisition.

Remote/Local

Instruments can be in:

* **Local Mode**, controlled via the front panel controls and by the controller's commands, or

* **Remote Mode** (REN line asserted), in which case the front panel is locked except for the *ON/OFF* switch and the potentiometers.

Service Request (SRQ)

This line is always managed by the instrument according to its status word. When an instrument wants to attract the controller's attention, it asserts its SRQ line.
The controller thus advised (either by SRQ interrupt or by continuous scanning of the SRQ line) must identify the requesting instrument (there may be several on the same line: hardwired OR) and the cause of the request.

There are two ways of doing this: serial recognition or parallel recognition (see IEEE 488.2).

3. RS232 PROGRAMMING

3.1. Introduction

The IEEE 488.2 standard was designed for the IEEE 488.1 physical interface.

Certain functions are specific to the IEEE standard (Device Clear, Group Executive Trigger, etc) and are not available for remote control via the RS232 interface.

3.2. How it works

The instrument can be controlled remotely using just a three-wire link (XON/XOFF protocol) or a more fully featured link (XON/XOFF or RTS protocol).

Three-wire connection (TD,RD,SG): the XON/XOFF protocol is necessary to control the flow of data.
The receiver controls the incoming data flow while disabling the sender (sends an XOFF (ASCII code, decimal 17) or by reenabling the sender (sending an XON (ASCII code, decimal 19).
After receiving an XOFF, the controller must not send more than 32 characters to the oscilloscope.

Extended connection the protocol can be XON/XOFF or RTS.
A high level on the CTS line enables the oscilloscope to send data and a low level stops it.
A high level on the RTS line enables the controller (or other) to send and a low level stops it.

3.3. Connectors

3.3.1. Three-wire link

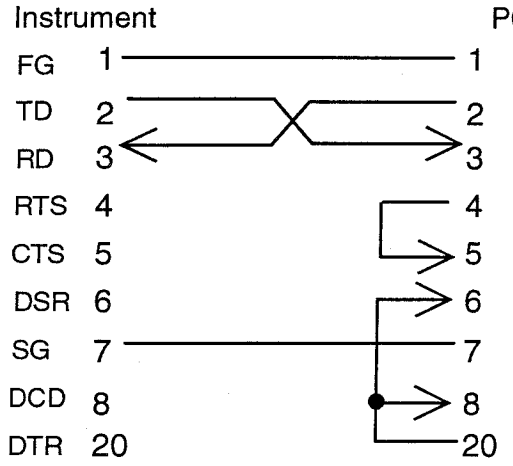
The instrument software controls the incoming and outgoing data (XON/XOFF protocol). In this case, the connection is the bare minimum:

Pin 2	TD (Transmit Data)
Pin 3	RD (Receive Data)
Pin 7	SG (Signal Ground)

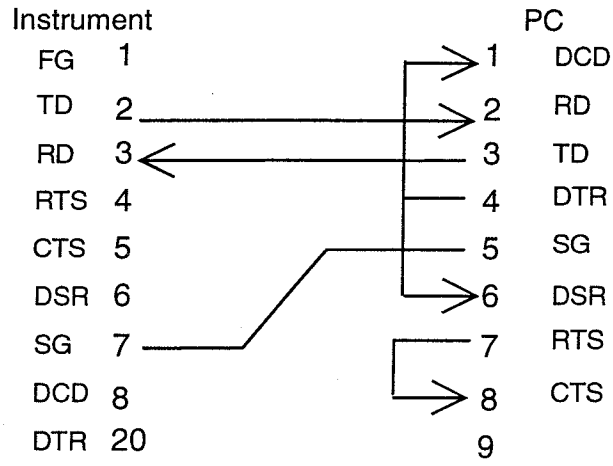
The oscilloscope's TD line must be linked to the controller's RD line.
The oscilloscope's RD line must be linked to the controller's TD line.

Refer to the controller's user manual to set these control lines correctly (on a PC compatible computer, loop RTS (pin 4) to CTS, DTR (pin 20) to DCD (pin 8) and DSR (pin 6) to be sure that it operates correctly).

25-way simplified cable



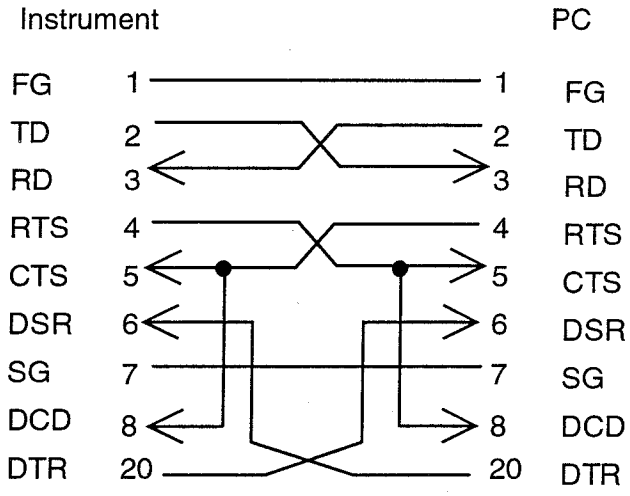
25/9-way simplified cable



3.3.2. Extended link

In this case, either protocol can be used. The connections are as follows:

Pin 2	TD (Transmit Data)	Pin 3	RD (Receive Data)
Pin 7	SG (Signal Ground)	Pin 4	RTS (Request to Send)
Pin 5	CTS (Clear to Send)	Pin 20	DTR (Data Terminal Ready)



The TD line must be linked to the controller's RD line.
The RD line must be linked to the controller's TD line.

The RTS line is an output used to control the flow of incoming data. A high level on this line enables the controller to send and a low level tells the controller to stop transmitting data.

The CTS line is an input which controls the instrument's flow of outgoing data. A high level on this line enables the instrument to send and a low level prevents it.

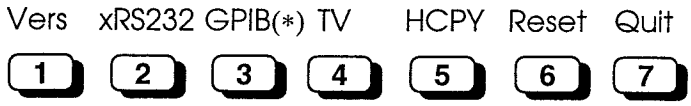
The DTR line is an output that is set high on power up.

The configuration of the RS232C cable depends on your specific application, the main thing being to ensure that the above rules are obeyed. Refer to your controller's user manual to ensure that the required conditions are met.

3.4. Configuration

OX 8100, OX 8042, OX 8062 :

To configure the RS232-C interface, press **UTILITY**; the following menu appears on the screen:



(*) *on OX 8100, if the GPIB option is plugged in.*

Next, select key 44 below the screen to change the menu bar to:

The configuration options are:

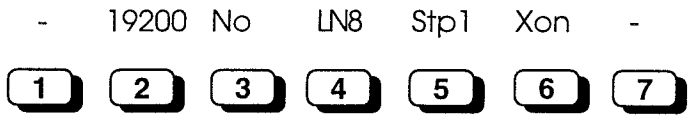
Baud rate	:	75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200
Parity	:	No, odd, even
Length	:	Ln7, Ln8
Stop bit	:	stp1, stp2
Protocol	:	XON, RTS

Quit the menu (key 7) to configure and select the interface RS232C.

OX 8050 :

To configure the RS232-C interface, press **UTILITY**; the following menu appears on the screen:

Press the key 44 to set the parameters of the serial link, the following menu appears on the screen:



The configuration options are:

Baud rate	:	75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200
Parity	:	No, odd, even
Length	:	Ln7, Ln8
Stop bit	:	stp1, stp2
Protocol	:	XON, RTS

OX 8050 :

To configure the RS232-C interface, press **UTILITY**; the following menu appears on the screen:

Press the key 44 to set the parameters of the serial link, the following menu appears on the screen:

The configuration options are:

Baud rate	:	75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200
Parity	:	No, odd, even
Length	:	Ln7, Ln8
Stop bit	:	stp1, stp2
Protocol	:	XON, RTS

4. REMOTE CONTROL PERIPHERAL SELECTION

OX 8100 :

The RS232 interface is selected.

The GPIB interface is selected.

OX 8050 :

Press **UTILITY**; the following menu appears on the screen:

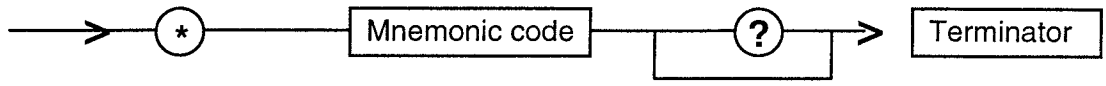
Press the key 4; the following menu appears on the screen:

Press the key 7 to select the peripheral. The menu can be then deleted.

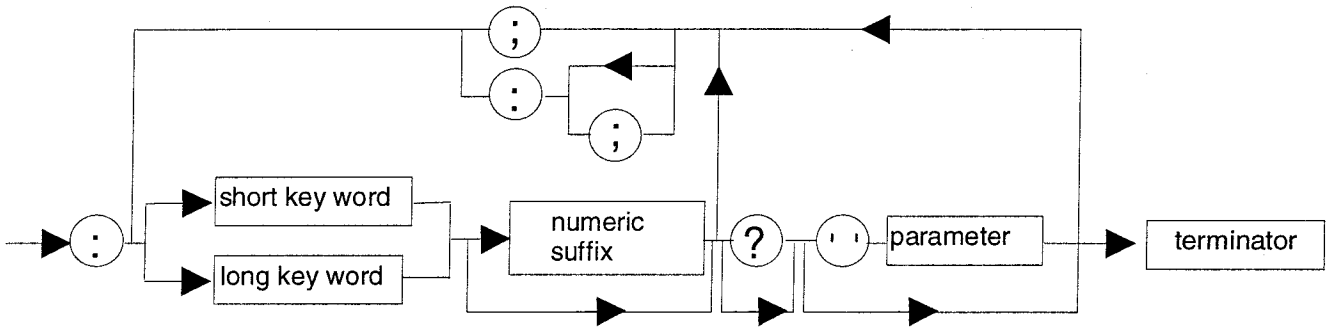
5. COMMAND SYNTAX

The commands are based on a hierarchical tree structure. There are two types of command:

* Common commands



* Instrument-specific commands



Keywords

Square brackets ([ ]) are used either side of a keyword which is optional during programming; this means that the instrument will execute the command whether the optional keyword is there or not. Uppercase and lowercase letters are used to differentiate the short form of the keyword (uppercase) from the long form (complete word).

The instrument accepts uppercase and lowercase without distinction.

Example: ACQ:REF 1 is equivalent to acquire:reference 1

Parameters

Parameters, if there are any, are separated from the keyword by a space (' '). A command can accept defined parameters, literal expressions or a combination of the two.

Defined parameters are between < > (for example: <setup>).

The square brackets [] signify that the parameter is optional.

The braces {} define the list of permitted parameters.

The | can be read as an "or" and is used to separate the different parameters.

For the MEASure and CONFigure commands, the parameter specifying the source is in parentheses (see sections 7.2.7.1. MEASure? and 7.2.7.2. CONFigure).

Separators

The colon separator is used to go down to the next directory or to reposition at the root, when it is preceded by a semicolon.

The semicolon separator separates two commands in the same directory or shows the end of a directory command, which repositions at the root level in adding a colon.

The space separator separates the keyword from the next parameter.

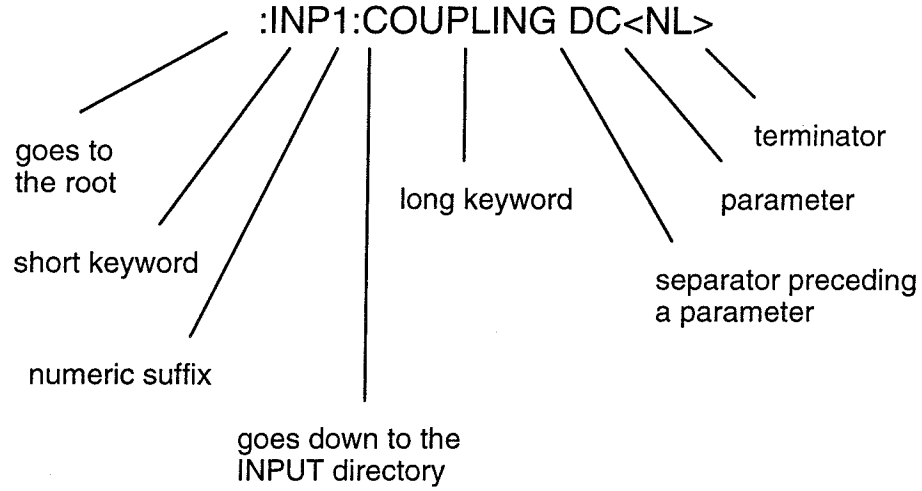
Terminators

In the rest of this manual <NL> will be the general term indicating a terminator.

NL can be: LF, ascii code 10 (ctrl J)
CR, ascii code 13
or both

A command line must not exceed 80 characters and must end with a terminator.

✍ Examples:



:INP1:COUP DC;RANGE 2<NL>

6. PROGRAMMING CONVENTIONS

6.1. Introduction

This section describes the conventions used to program the instrument and describes the command tree and how to move around in it.

6.2. Tree concept

The command tree contains all the instrument-specific commands. The common commands (IEEE 488.2) are listed separately because they do not affect the position of the syntax analyzer in the tree. When the terminator <NL> is sent to the instrument, the analyzer positions itself at the root. When the analyzer is in a directory, the 2 separators ' ; : ' are necessary to go back to the root.

6.3. Complete tree structure

IEEE 488.2 common commands

Commands	Functions	Refer to §
*CLS	Clear status command	7.1.1.
*ESE	Standard event status enable command	7.1.2.
*ESE?	Standard event status enable query	
*ESR?	Standard event status register query	7.1.3.
*IDN?	Identification query	7.1.4.
*LRN?	Instrument configuration query	7.1.5.
*OPC	Operation complete command	7.1.6.
*OPC?	Operation complete query	
*OPT?	Option query	7.1.7.
*RST	Reset command	7.1.8.
*SRE	Service request enable command	7.1.9.
*SRE?	Service request enable query	
*STB?	Read status byte query	7.1.10.
*TRG	Acquisition trigger	7.1.11
*TST?	Self-test query	7.1.12
*WAI	Wait to continue command	7.1.13

OX 8040 / OX 8050 / OX 8100 specific commands

Directory	Command	Function	Refer to §
	ABORt	Aborts current acquisition	7.2.1.1.
	AUToset	Executes Autoset	7.2.1.2.
	INITiate INITiate:IMMediate	Initiates acquisition in single shot mode	7.2.1.3.
	INITiate:CONTinuous INITiate:CONTinuous?	Initiates or stops acquisition in repetitive mode	7.2.1.3.2.
	:MODE :MODE?	Manages analog/digital modes	7.2.1.4.

ACQuire	:GLITch :GLITch?	Manages glitch capture mode	7.2.2.1.
	:MODE :MODE?	Manages acquisition mode	7.2.2.2.
	:PEAK :PEAK?	Manages envelope mode	7.2.2.3.
	:REFerence :REFerence?	Manages the pre-trigger	7.2.2.4.
DISPlay	:ANNotation :ANNotation?	Controls text display	7.2.3.1.
	:MODE :MODE?	Manages the vertical mode	7.2.3.2.
	:SAVE<N> :SAVE?<N>	Saves records	7.2.3.3.
	[:WINDow]:TRACe:X[:SCALe]:LEFT [:WINDow]:TRACe:X[:SCALe]:LEFT?	Manages the window position	7.2.3.4.1.
	[:WINDow]:TRACe:X[:SCALe]:PDIVision [:WINDow]:TRACe:X[:SCALe]:PDIVision?	Manages horizontal magnification	7.2.3.4.2.
	[:WINDow]:TRACe:Y:SEParation [:WINDow]:TRACe:Y:SEParation?	Manages trace separation *(OX 8100)*	7.2.3.4.3.
	[:WINDow]:TRACe:CLEAR	Clears trace display	7.2.3.4.4.
	[:WINDow]:TRACe:CUMUL [:WINDow]:TRACe:CUMUL?	Cumulates ETS ON/OFF	7.2.3.4.5.
	[:WINDow]:CURSor<N>:MODE [:WINDow]:CURSor<N>:MODE?	Cursor display mode	7.2.3.4.6.1.
	[:WINDow]:CURSor<N>:POSition [:WINDow]:CURSor<N>:POSition?	Cursor positioning	7.2.3.4.6.2.
	[:WINDow]:CURSor<N>:VALue?	Reads cursor value	7.2.3.4.6.3.
FORMat	[:DATA] [:DATA]?	Manages data type	7.2.4.1.
	:DINTerchange :DINTerchange?	Selects ADIF format	7.4.2.2.
HARDcopy	:FORMat :FORMat?	Selects hardcopy format	7.2.5.1.
	:GRATicule :GRATicule?	Manages graticule plot	7.2.5.2.
	:STATus :STATus?	Manages instrument configuration plot	7.2.5.3.
INPut<N>	:COUPling :COUPling?	Selects input coupling	7.2.6.1.
	:INVert :INVert?	Manages CH2 invert	7.2.6.2.
	:RANGe :RANGe?	Manages input sensitivity	7.2.6.3.
	:MODE :MODE?	Manages Normal/ Differential modes *(OX 8042 / OX 8062)*	7.2.6.7.
	:VERNier?	Querries gain variable	7.2.6.4.
	:FILTer[:LPASs][:STATe] :FILTer[:LPASs][:STATe]?	Selects bandwidth *(OX 8100)*	7.2.6.5.
	:PROBe :PROBe?	Selects probe coefficient	7.2.6.6.
MEASure	[:SCALar][:VOLTage][:DC]?	Configures/sends DC component measurement	7.2.6.7.
	[:SCALar][:VOLTage]:AC?	Configures/sends AC RMS measurement	7.2.7.1.
	[:SCALar][:VOLTage]:FREQuency?	Configures/sends Frequency measurement	7.2.7.1.
	[:SCALar][:VOLTage]:PERiod?	Configures/sends period measurement	7.2.7.1.
	[:SCALar][:VOLTage]:PHASe?	Configures/sends phase measurement	7.2.7.1.
	[:SCALar][:VOLTage]:AMPLitude?	Configures/sends amplitude measurement	7.2.7.1.
	[:SCALar][:VOLTage]:LOW?	Configures/sends low voltage measurement	7.2.7.1.
	[:SCALar][:VOLTage]:HIGH?	Configures/sends high voltage measurem.	7.2.7.1.
	[:SCALar][:VOLTage]:RISE:TIME? [:SCALar][:VOLTage]:RTIMe?	Configures/sends risetime measurement	7.2.7.1.
	[:SCALar][:VOLTage]:FALL:TIME? [:SCALar][:VOLTage]:FTIMe?	Configures/sends falltime measurement	7.2.7.1.
	[:SCALar][:VOLTage]:NWIDth?	Configures/sends neg. pulse width meas.	7.2.7.1.
	[:SCALar][:VOLTage]:PWIDth?	Configures/sends positive pulse width meas.	7.2.7.1.
	[:SCALar][:VOLTage]:PDUTy?	Configures/sends PWIDth/PERiod ratio measurement	7.2.7.1.

	[:SCALar][:VOLTage]:NDUTy?	Configures/sends NWIDth/PERiod ratio measurement	7.2.7.1.
	[:SCALar][:VOLTage]:MINimum?	Configures/sends minimum voltage measurement	7.2.7.1.
	[:SCALar][:VOLTage]:MAXimum?	Configures/sends minimum voltage measurement	7.2.7.1.
	[:SCALar][:VOLTage]:PTPeak?	Configures/sends p-p voltage measurement	7.2.7.1.
CONFigure?		Sends current type of auto measurement	7.2.7.
CONFigure	[:SCALar][:VOLTage][:DC]	Configures DC component measurement	7.2.7.
	[:SCALar][:VOLTage]:AC	Configures AC RMS measurement	7.2.7.2.
	[:SCALar][:VOLTage]:FREQuency	Configures frequency measurement	7.2.7.2.
	[:SCALar][:VOLTage]:PERiod	Configures period measurement	7.2.7.2.
	[:SCALar][:VOLTage]:PHASe	Configures phase measurement	7.2.7.2.
	[:SCALar][:VOLTage]:AMPLitude	Configures amplitude measurement	7.2.7.2.
	[:SCALar][:VOLTage]:LOW	Configures low voltage measurement	7.2.7.2.
	[:SCALar][:VOLTage]:HIGH	Configures high voltage measurement	7.2.7.2.
	[:SCALar][:VOLTage]:RISE:TIME [:SCALar][:VOLTage]:RTIMe	Configures risetime measurement	7.2.7.2.
	[:SCALar][:VOLTage]:FALL:TIME [:SCALar][:VOLTage]:FTIMe	Configures falltime measurement	7.2.7.2.
	[:SCALar][:VOLTage]:NWIDth	Configures negative pulse width meas.	7.2.7.2.
	[:SCALar][:VOLTage]:PWIDth	Configures positive pulse width meas.	7.2.7.2.
	[:SCALar][:VOLTage]:PDUTy	Configures PWIDth/PERiod ratio meas.	7.2.7.2.
	[:SCALar][:VOLTage]:NDUTy	Configures NWIDth/PERiod ratio meas.	7.2.7.2.
	[:SCALar][:VOLTage]:MINimum	Configures min. voltage measurement	7.2.7.2.
	[:SCALar][:VOLTage]:MAXimum	Configures max. voltage measurement	7.2.7.2.
	[:SCALar][:VOLTage]:PTPeak	Configures p-p voltage measurement	7.2.7.2.
	[:SCALar][:VOLTage]:OFF	Cancels auto measurement	7.2.7.2.
SYSTem?	:ERRor?	Displays the error number	7.2.8.1.
	:KLOCk :KLOCk?	Locks front panel	7.2.8.2.
	:SET :SET?	Manages instrument configuration	7.2.8.3.
	:VERSion?	Reads SCPI version	7.2.8.4.
TIMebase	:DELay :DELay?	Manages horizontal mode	7.2.9.1.
	:EXPH :EXPH?	Manages horizontal magnification (analog)	7.2.9.2.
	:MODE :MODE?	Manages mode (auto or trigger) (analog)	7.2.9.3.
	:RANGe :RANGe?	Configures timebase speed	7.2.9.4.
	:POINts :POINts?	Acquisition size	7.2.9.5.
TRACe	[:DATA]?	Reads records	7.2.10
	:CATalog?	Reads record list	7.2.10.1.
TRIGger	:COUPLing :COUPLing?	Configures coupling	7.2.11.1.
	:MODE :MODE?	Configures trigger mode *(except OX 8040)*	7.2.11.2.
	:SLOPe :SLOPe?	Configures trigger slope	7.2.11.3.
	:SOURce :SOURce?	Configures trigger source	7.2.11.4.
	:B :B?	Configures B timebase trigger *(OX 8100)*	7.2.11.4.
	:VIDeo:FIELd:LPFRame :VIDeo:FIELd:LPFRame?	TV Trigger : standard selection *(OX 8040 / OX 8050 / OX 8100)*	7.2.11.6.1.
	:VIDeo:FIELd:SELect :VIDeo:FIELd:SELect?	TV Trigger : field selection *(OX 8040 / OX 8050 / OX 8100)*	7.2.11.6.2.
	:VIDeo:LINE :VIDeo:LINE?	TV Trigger: line selection *(OX 8040 / OX 8050 / OX 8100)*	7.2.11.6.3
	:VIDeo:SSIGnal :VIDeo:SSIGnal?	TV Trigger : signal polarity	7.2.11.6.4.

6.4. Three types of command

6.4.1. Common commands

These are defined by the IEEE 488.2 standard and are used to execute functions common to all instruments complying with that standard.

The common commands are independent of the tree structure so do not affect the position of the syntax analyzer. They are therefore different from the "root" commands which return the analyzer at the root of the tree.

✍ Example of common command: *RST (Reset)

6.4.2. "Root" commands

The "root" commands control the instrument's basic functions. They reside at the root of the tree.
A command is considered by the syntax analyzer as a "root" command if it is placed at the beginning of a programming message or if it is preceded by a colon (ASCII code 58_{10}).

✍ Example of a "root" command: ABORT

6.4.3. Directory commands

Some commands are combined in a directory under the root so that all the programming commands are structured. When the instrument is powered up, the analyzer is automatically positioned at the root.

✍ Example of a directory command: :INPut1:COUPLing AC
(the colon preceding INPut1 is optional).

6.5. Changing directories

The analyzer returns to the root in one of the following cases:

- on power up
- on receipt of a programming message terminator (<NL> or EOI true with the last character (IEEE 488))
- on receipt of a colon (ASCII code 58_{10}) at the head of the complete path (after a semicolon separator).

To access the directory commands, you must specify the directory name and then go down to this directory by adding the colon.
The command can then be accessed by adding its name to the string.

✍ Example: ":INPut1:COUPLing AC<NL>"

:INPut1	moves the syntax analyzer to the INPut1 directory
:COUPLing AC	accesses the directory command
<NL>	returns the analyzer to the root

When you specify a directory, you can access the commands in that directory one after the other without having to redefine the complete path.
A semicolon separator (ASCII code 59_{10}) between each command is sufficient.

✍ Example: ":INPut2:COUPLing AC;RANGE 2;INVert ON<NL>"

:INPut2	moves the syntax analyzer to the INPut2 subdirectory
COUPLing AC;RANGE 2;INVert ON	accesses the commands in this directory
<NL>	returns the syntax analyzer to the root

To access commands in different directories using the same programming message, you must specify the complete path then return to the root and insert the colon (ASCII code 58_{10}) after a semicolon separator (ASCII code 59_{10}).

✍ Example: ":INPut2:INVert ON;:TIMebase:RANGe 100e-3<NL>"

:INPut2	positions the syntax analyzer at the INPut2 subdirectory
:INVert ON;	accesses the command in the INPut2 directory: returns the syntax analyzer to the root
:TIMebase	positions the syntax analyzer at the TIMebase subdirectory
:RANGe 100e-3	accesses the command in the TIMebase directory
<NL>	returns the syntax analyzer to the root

☞ ***Note:*** ***in all these examples, the colon (:) at the beginning of the message is optional (the analyzer automatically goes to the root).***

6.6. Directories

The command set includes the common commands, the "root" commands and 10 sets of commands in directories.
Each command has a short form (uppercase letters). The optional part is indicated in lowercase. The short form speeds up communication (fewer characters to transmit); the long form makes the application program easier to read.

DIRECTORIES

ACQuire	configures the digital acquisition environment
DISPlay	configures the display environment
FORMat	defines the format of data sent by the instrument (curves)
HARDcopy	configures the hard copy environment
INPut<N>	configures the analogue inputs
MEASure	configures and/or reads the measurement results
SYSTem	controls the instrument's basic functions
TIMEbase	configures the timebase
TRACe	controls the reading of traces
TRIGger	configures the trigger

7. COMMAND DETAILS

7.1. IEEE 488.2 common commands

The common commands are defined in the IEEE 488.2 standard and operate on all IEEE 488.2 compliant instruments. They control the basic functionalities such as identification, reset, read configuration, read and reset status and event registers.
If a command containing one or more subdirectories is received and a common command is stacked, the instrument remains in that subdirectory and executes the commands normally.

7.1.1. *CLS (Clear Status)

(Command)
The common command ***CLS** resets the event and status registers.

Command syntax: *CLS

Example: OUTPUT 705 ; "*CLS"

Note: see section 8.3. Status and event management, for a more detailed explanation.

7.1.2. *ESE (Event Status Enable)

(Command and Query)
The common command ***ESE** sets the event mask status. A 1 enables the event register bit to generate an event.
The query ***ESE?** returns the current content of the event mask register.

Command syntax: *ESE <mask>
where <mask>::=0 to 255

Example: OUTPUT 707 : "*ESE 32"

This example enables the CME event (Command error), bit 5 of the event register.

Query syntax: *ESE?

Response format: <mask> <NL>
where <mask>:: = 0 to 255 (integer in NR1 format)

Example: OUTPUT 707 ; "*ESE?"
ENTER 707 ; MEVENT
PRINT MEVENT

Description of the Event Mask register:

Bit	Weight	Event Enabled
7	128	PON (Power ON): not used (0)
6	64	URQ (User Request): not used (0)
5	32	CME (command error)
4	16	EXE (execution error)
3	8	DDE (instrument error)
2	4	QYE - Query Error
1	2	RQC - not used (0)
0	1	OPC - operation complete

☞ ***Note: see section 8.3 Status and event management, for a more detailed explanation.***

7.1.3. *ESR? (Event Status Register)

(Query)
The query ***ESR?** returns the content of the event register.
The event register is reset when it has been read.

Query syntax: *ESR?

Response format: <state> <NL>
where <state>:: = 0 to 255 (integer in NR1 format)

✍ Example:

```
OUTPUT 707 ;"*ESR?"
ENTER 707 ; Event
PRINT Event
```

Description of the event register

Bit	Weight	Name	Condition
7	128	PON	not used (0)
6	64	URQ	not used (0)
5	32	CME	0 : no command error 1 : a command error has been detected
4	16	EXE	0 : no execution error 1 : an execution error has been detected
3	8	DDE	0 : no instrument-specific error 1 : an instrument-specific error has been detected
2	4	QYE	0 : no query error 1 : a query error has been detected
1	2	RQL	not used (0)
0	1	OPC	0 : current command not complete 1 : current command complete

7.1.4. *IDN? (Identification Number)

(Query)
In response to the query ***IDN?**, the oscilloscope returns the instrument type and the software version.

Query syntax: *IDN?

Response format:
"manufacturer, *instrument*,FV<firmware version>< instrument code ><NL>

<manufacturer>	Manufacturer
<instrument>	Instrument reference
<firmware version>	Software version
< instrument code >	Instrument code, independent of the manufacturer and instrument code

Example:

```
OUTPUT 707 ; "*IDN?"
ENTER 707 ;ID$
PRINT ID$
```

7.1.5. *LRN? (learn)

(Query)
In response to the query ***LRN?**, the oscilloscope returns the instrument's complete configuration preceded by a header "#800000062" This command is equivalent to the :SYSTEM:SET? command.

Query syntax: *LRN?

Response format: <setup>
where <setup> : : = #800000062 <setup data string>

Example:

```
OUTPUT 707 ; "*LRN?"
ENTER 707 ;config$
PRINT config$
```

Note*: *<setup data string> contains 62 bytes in binary code.

7.1.6. *OPC (Operation Complete)

(Command/Query)
The command ***OPC** Enable sets the OPC bit at 1 in the event register when the current operation is complete.
In response to the query ***OPC?**, the oscilloscope returns the ASCII character "1" when the current operation is complete.
This command lets you synchronize the oscilloscope and the controller's application program (see section 8.3.2.).

Command syntax: *OPC

Example:

```
OUTPUT 707;"*OPC"
```

Query syntax: *OPC?

Response format: 1<NL>

Example:

```
OUTPUT 707;"*OPC?"
ENTER 707;END$
PRINT END$
```

7.1.7. *OPT? (Option)

(Query)
In response to the query ***OPT?**, the instrument returns the list of options.

Query syntax: *OPT?

Response format: NO<NL> no option has been fitted.
GPIB<NL> GPIB option has been fiffed.

Example:
```
OUTPUT 707;"*OPT?"
ENTER 707;OPTION$
PRINT OPTION$
```

7.1.8. *RST (Reset)

(Command)
The command ***RST** reconfigures the instrument to the default configuration below:

Command syntax: *RST

Example:
```
OUTPUT 707;"*RST"
```

Operating mode	Analogue
System configuration	
. Front panel	active
Channel configuration	
. CH1/CH2 coupling	AC
. CH2 inversion	NO *(OX 8040 / OX 8050 / OX 8100)*
. CH1/CH2 sensitivity	0.1 V/div (or > 0.1 V/div) *(OX 8040 / 8050 / 8100)*
	1 v/div. *(OX 8042 / OX 8062)*
. Vertical mode	CHOP
. Bandwidth limit	OFF *(OX 8100)*
. Probe coefficient	x1 *(OX 8042 / OX 8062)*
Timebase configuration	
. range	50 µs/div
. auto mode	YES
. horizontal mode	time base A
. x 10	no
. trace separation	0 *(OX 8100)*
Trigger configuration	
. coupling	DC
. level	as front panel
. mode	NORMAL
. slope	+
. source	CH1
. trigger B	OFF *(OX 8100)*
Acquisition configuration	
. acquisition	not initiated
. mode	REFResh
. pre-trigger	0 K
. envelope mode	NO
. glitch capture mode	NO
. record size	1 K
. Dot cumul in ETS mode	YES

Display configuration (digital)

. horizontal expansion	x1
. window position	0000
. save channels	NO

Hardcopy configuration

. format	IBM Proprinter
. interface	Centronics
. graticule	NO
. partitioning	NO
. configuration	NO

Measurement configuration

. cursor selected	C1
. cursor displayed	NONE
. measurement selected	NONE
. H cursor positions	C1: -4 div C2 = +4 div
. V cursor positions	C1: -5 div C2 = +5 div C3 = 0 div
. source	CH1

Digital data configuration
- . ASCII format
- . header OFF

☞ ***Note: This default configuration is the factory setting except for the interface-related part which, for a *RST command is not changed.***

7.1.9. *SRE (Service Request Enable)

(Command/Query)
The command ***SRE** enables the service request mask register. A bit at 1 enables the corresponding bit in the status register to request a service (status register bit 6 at 1). A bit at 0 disables it.
In response to the query ***SRE?**, the oscilloscope returns the service request mask register value.

Command syntax: *SRE <mask>
where <mask> ::= 0 to 255

✍ Example: OUTPUT 707;":*SRE 32"
(enables the ESB bit to request a service).

Query syntax: *SRE?

Response format: <mask><NL> where <mask> : : = 0 to 255 (NR1 notation)

✍ Example:
```
OUTPUT 707;":*SRE?"
ENTER 707 ; MASK$
PRINT MASK$
```

SERVICE REQUEST MASK REGISTER

Bit	Weight	Enables
7	128	not used
6	64	not used
5	32	ESB - Event Status Bit
4	16	MAV = Message Available
3	8	not used
2	4	not used
1	2	not used
0	1	not used

7.1.10. *STB? (Status Byte)

(Query)
In response to the query ***STB?**, the oscilloscope returns the content of its Status Byte Register.

When bit 6 is returned, it indicates the MSS (Master Summary Status) value (1 if the instrument has a reason for requesting a service).
Unlike RQS, it is not reset after the status register is read (RQS can be accessed by serial recognition only and forced to zero after it).
For further details, see section 8.3, Status and Event Management.

Query syntax: *STB?

Example: OUTPUT 707;"*STB"

Response format: <value><NL>
where <value>: := 0 to 255 (integer in NR1 format)

Example:
```
OUTPUT 707;"*STB"
ENTER 707 ; STATUS$
PRINT STATUS$
```

STATUS REGISTER

Bit	Weight	Name	Value
7	128	-	not used (0)
6	64	RQS/MSS	0=no request 1=service request
5	32	ESB	0=no event recognized 1=an authorized event has occurred
4	16	MAV	0=no message available 1=output message available
3	8	-	not used (0)
2	4	-	not used (0)
1	2	-	not used (0)
0	1	-	not used (0)

7.1.11. *TRG (Trigger)

(Command)
This command initiates acquisition according to the current mode (SINGLE, REFRESH or ROLL).

Command syntax: *TRG

Example: OUTPUT 707;"*TRG"

7.1.12. *TST? (Test)

(Query)
In response to the query ***TST?**, the oscilloscope responds automatically with a 0.

Query syntax: *TST?

Example: OUTPUT 707;"*TST"

Response format: 0 <NL>

Example:
```
OUTPUT 707;"*TST?"
ENTER 707;Result$
PRINT Result$
```

7.1.13. *WAI (Wait)

(Command)
The command ***WAI** stops the oscilloscope executing other commands until the current command is complete. This can be used to synchronize the oscilloscope with the application program running on the controller (see the section on Oscilloscope/Controller Synchronization).

Command syntax: *WAI

Example:
```
OUTPUT 707;":INITiate;*WAI;:MEAS:AC?"
ENTER 707;MEAS$
PRINT MEA$
```

In this particular instance, since the oscilloscope has received *WAI, it will wait until the end of acquisition before executing the measurements and transmitting the results.

7.2. Instrument-specific commands

7.2.1. "Root" commands

The "root" commands control the oscilloscope's basic functions. They are recognized as such if they are placed at the head of the message or if they are preceded by a colon (ASCII code 58).

Commands:

ABORt
AUToset
INITiate
[:IMMediate]
:CONTinuous
MODE

7.2.1.1.ABORt

(Command)
The command **:ABORt** stops the current acquisition.

Command syntax: ABORt

Example: OUTPUT 707;"ABORt"

7.2.1.2.AUToset

(Command)
The command **:AUToset** assesses the input signals and sets the timebase parameters (range according to the slowest signal) and vertical sensitivity. If no signal is present, the oscilloscope returns to its initial state (for further details, see the instrument's user manual).

Command syntax: AUToset

Example: OUTPUT 707;"AUToset"

7.2.1.3.INITiate

The INITiate command initiates or stops acquisition.

7.2.1.3.1.[:IMMediate]

(Command)
The command **INITiate[:IMMediate]** initiates the oscilloscope in single shot mode (SINGLE or ROLL according to what has been selected in ACQuire:MODE).

Command syntax: INITiate[:IMMediate]

Example: OUTPUT 707;"INITiate"

7.2.1.3.2.:CONTinuous

(Command/Query)
The command **INITiate:CONTinuous** starts (ON) or stops (OFF) an acquisition in repetitive mode (REFRESH mode) irrespective of the ACQuire:MODE command.

In response to the query **INITiate:CONTinuous?**, the oscilloscope returns the continuous mode state.

Command syntax: INITiate:CONTinuous {OFF|ON}

Example: OUTPUT 707 ; "INIT:CONT OFF"

Query syntax: INITiate:CONTinuous?

Response format: {ON|OFF}

Example:
```
OUTPUT 707 ; "INIT:CONT?"
ENTER 707 ; CONT$
PRINT CONT$
```

7.2.1.4.MODE

(Command/Query)
The command **:MODE** sets analogue or digital mode.
In response to the query **:MODE?**, the oscilloscope returns the current mode.

Command syntax: MODE {ANIUNM]

Example: OUTPUT 707 ; "NUM MODE"

Query syntax: MODE?

Response format: {ANINUM}<NL>

Example:
```
OUTPUT 707 ; "MODE?"
ENTER 707 ;MODE$
PRINT MODE$
```

After a *RST, the oscilloscope is in analogue mode.

7.2.2. ACQuire commands

The commands in this directory control the acquisition configuration:

GLITch
MODE
PEAK
REFerence

7.2.2.1.GLITch

(Command/Query)
The command **:ACQuire:GLITch:** configures glitch capture mode.

In response to the query **ACQuire:GLITch?**, the oscilloscope returns the glitch capture mode configuration.

After a *RST, glitch capture mode is OFF.

Command syntax: ACQuire:GLITch {ON|OFF}

Example: OUTPUT 707 ; ":ACQuire:GLITch ON"

Query syntax: ACQuire:GLITch?

Response format: {ON|OFF} <NL>

Example:
```
OUTPUT 707 ; "ACQuire:GLIT?"
ENTER 707 ;GLITCH$
PRINT GLITCH$
```

7.2.2.2.:MODE

(Command/Query)
The command **:ACQuire:MODE** configures acquisition mode.

In response to the query **ACQuire:MODE?**, the oscilloscope returns the current acquisition mode.

After a *RST, REFRESH mode is configured.

Command syntax: ACQuire:MODE {REFResh|ROLLISINGle}

Example: OUTPUT 707 ; ":ACQUIRE:ROLL MODE"

Query syntax: ACQuire:MODE?

Response format: {REFRESH|ROLLISINGLE}<NL>

Example:
```
OUTPUT 707 ; "ACQ:MODE?"
ENTER 707 ;MODE$
PRINT MODE$
```

7.2.2.3.:PEAK

(Command/Query)
The command **ACQuire:PEAK** configures envelope mode.

Envelope mode is on only if the oscilloscope is in REFRESH mode.

In response to the query **ACQuire:PEAK?**, the oscilloscope returns the envelope mode configuration.

After a *RST, envelope mode is OFF.

Command syntax: ACQuire:PEAK {ON|OFF}

Example: OUTPUT 707 ; ":ACQUIRE:PEAK ON"

Query syntax: ACQuire:PEAK?

Response format: {ON|OFF}<NL>

Example: OUTPUT 707 ; "ACQuire:PEAK?"
ENTER 707 ;ENV$
PRINT ENV$

7.2.2.4.:REFerence

(Command/Query)
The command **ACQuire:REFerence** configures the pre-trigger value in dot.

In response to the query **ACQuire:REFerence?**, the oscilloscope returns the pre-trigger value.

After a *RST, the pre-trigger is set at 0 dot.

Command syntax: ACQuire:REFerence<value><NL>
with <value> ::= {OFF| 0| 250| 500| 750| 1000}
for a record size configuration of 1 K

(OX 8042 / OX 8062 / OX 8100) ::={OFF| 0| 1000| 2000| ...| 8000}
for a record size configuration of 8 K

(OX 8042 / OX 8062 / OX 8100) ::={OFF| 0| 1000| 2000| ...| 16000}
for a record size configuration of 16 K

(OX 8040 / OX 8050) ::={OFF| 0| 2000| 4000| 6000| 8000}
for a record size configuration of 8 K

(OX 8040 / OX 8050) ::={OFF| 0| 4000| 8000| 12000| 16000}
for a record size configuration of 16 K

OFF corresponds to the uninterrupted ROLL mode.

Example : OUTPUT 707 ; ":ACQUIRE:REF 250"

Query syntax: ACQuire:REFerence?

Example:
```
OUTPUT 707 ; "ACQuire:REF?"
ENTER 707 ;REF$
PRINT REF$
```

7.2.3. DISPlay commands

The DISPLAY directory contains the display configuration commands:

ANNotation
MODE
SAVe
WINDow

7.2.3.1.ANNotation

(Command/Query)
The command **DISPlay:ANNotation** switches the oscilloscope's screen text display on or off.

In response to the query **DISPlay:ANNotation?**, the oscilloscope returns the current text display configuration.

After a *RST, the text is displayed on screen (DISPlay:ANNotation ON).

Command syntax: DISPlay:ANNotation {ON|OFF}

Example: OUTPUT 707 ; "DISP:ANN ON"

Query syntax: DISPlay:ANNotation?

Response format: {ON|OFF}<NL>

Example:
```
OUTPUT 707 ; "DISP:ANN?"
ENTER 707 ; TEXTE$
PRINT TEXTE$
```

7.2.3.2.:MODE

(Command/Query)
The command **DISPlay:MODE:** configures the oscilloscope's vertical mode.

In response to the query **DISPlay:MODE?**, the oscilloscope returns its vertical mode configuration.

After a *RST, vertical mode is CHOP.

Command syntax: DISPlay:MODE <vertical>

(OX 8100) with <vertical>: : = {CH1|CH2|ALT|CHOP|ADD|XY}
(OX 8050 / OX 8042 / OX 8062) {CH1|CH2|ALT|CHOP|ADD|MULT|XY}
(OX 8040) {CH1|CH2|DUAL|ADD|MULT|XY}

Example: OUTPUT 707 ; "DISPlay:MODE XY"

Query syntax: DISPlay:MODE?

Response format: <vertical><NL>

where <vertical> is defined above

Example:

```
OUTPUT 707 ; "DISP:MODE?"
ENTER 707 ; MODE$
PRINT MODE$
```

7.2.3.3.:SAVE<N>

(Command/Query)
The command **DISPlay:SAVE<N>** configures the specified channel save in digital mode.

In response to the query **DISPlay:SAVE<N>?**, the oscilloscope returns the channel save configuration in digital mode.

After a *RST, no channel is saved (OFF).

Command syntax: DISPlay:SAVE<N> {ON|OFF}

where <N>: : = 1, 2

Example: OUTPUT 707 ; "DISPlay:SAVE1 ON"

Query syntax: DISPlay:SAVE<N>?

where <N>: : = 1, 2

Response format: {ON|OFF}<NL>

Example:

```
OUTPUT 707 ; "DISP:SAVE2?"
ENTER 707 ; SAVE$
PRINT SAVE$
```

7.2.3.4.[:WINDow]

The DISPlay:WINDow command set manages the display of error messages and traces (selects feeds, windows and horizontal expansion).

The structure of this command set is:

```
DISPlay
    [: WINDow]
        :TRACe
            :X
              [:SCALe]
                  :LEFT
                  :PDIVision
            :Y
              :SEParation        (OX 8100)
            :CLEAR
            :CUMUL
        :CURSor
            :MODE
            :POSition
            :VALue
```

7.2.3.4.1.:TRACe:X[:SCALe]:LEFT

(Command/Query)
The command **DISPlay[:WINDow]:TRACe:X[:SCALe]:LEFT** configures the window origin in digital mode.

In response to the query **DISPlay[:WINDow]:TRACe:X[:SCALe]:LEFT?**, the oscilloscope returns the window position in digital mode.

After a *RST, the window is at its origin point, 0.

Command syntax: DISPlay[:WINDow]:TRACe:X[:SCALe]:LEFT<numeric-value>

with <numeric-value>::= 0 to 800 for a record size configured at 1 K

::= 0 to 7800 for a recrod size configured at 8 K

::= 0 to 15800 for a record size configured at 16 K

☞ ***Note :*** ***The real value of the window position will be the even value ≤ requested value.***

✍ Example: OUTPUT 707;"DISP:TRAC:X:LEFT 1000"

Query syntax: DISPlay[:WINDow]:TRACe:X[:SCALe]:LEFT?

Response format: <numeric-value><NL>

where <numeric-value> is defined above.

7.2.3.4.2.:TRACe:X[:SCALe]:PDIVision

(Command/Query)
The command **DISPlay[:WINDow]:TRACe:X[:SCALe]:PDIVision** configures horizontal magnification used in digital mode.

In response to the query **DISPlay[:WINDow]:TRACe:X[:SCALe]:PDIVision?**, the oscilloscope returns the horizontal magnification value.

After a *RST, horizontal magnification is configured at 4 (*).

Command syntax:
DISPlay[:WINDow]:TRACe:X[:SCALe]:PDIVision <numeric-value>

avec <numeric-value> ::= 4 to 6 for a record size configured at 1 K
::= 1 to 6 for a record size configured at 8 K
::= 0 to 6 for a record size configured at 16 K

✍ Example: OUTPUT 707;"DISP:TRACe:X:PDIV 4"

Query syntax: DISPlay[:WINDow]:TRACe:X[:SCALe]:PDIVision?

Response format: <numeric-value><NL>

where <numeric-value> is defined above.

(*) Details relating to the index

Record size			Index	Magnification factor	Visualized record size
16 K			0	x 0.05 | 0.04	20000 | 25000
	8 K		1	x 0.1	10000
			2	x 0.25 | x 0.2	4000 | 5000
			3	x 0.5 | 0.4	2000 | 2500
		1 K	4	x 1	1000
			5	x 2.5 | x 2	400 | 500
			6	x 5 | x 4	200 | 250

7.2.3.4.3.:TRACe:Y:SEParation (OX 8100)

(Command)
The command **DISPlay[:WINDow]:TRACe:Y:SEParation** in alternate horizontal mode configures the trace separation.

In response to the query **DISPlay[:WINDow]:TRACe:Y:SEParation?**, the oscilloscope sends the trace separation value.

After a *RST, the trace separation is configured at value 0 (minimum separation).

Command syntax :

DISPlay[:WINDow]:TRACe:Y:SEParation <numeric-value>
with <numeric-value> ::= 0 to 15

Example : OUTPUT 707;"DISP:TRACe:Y:SEP 7"

Query syntax : DISPlay[:WINDow]:TRACe:Y:SEParation?

Response format : <numeric-value><NL>

were <numeric-value> is defined above.

7.2.3.4.4.:TRACe:CLEAR

(Command)
The command erase the displayed numeric curves.

Command syntax : DISPlay[:WINDow]:TRACe:CLEAR

7.2.3.4.5.:TRACe:CUMUL {ON|OFF}

(Command/Query)
After a *RST, the cumul is on.

Command syntax : DISPlay[:WINDow]:TRACe:CUMUL{ON|OFF}

Parameters : ON authorises the cumul of points.
OFF at each new acquisition,
the previous points are erased
the new points are displayed.

Query syntax : DISPlay[:WINDow]:TRACe:CUMUL?

Response format : {ON|OFF}LF>

7.2.3.4.6.:CURSor

Configuration of cursor measurements. Four modes are available :

- Voltage measurement between two cursors (DV)
- Time measurement between two cursors (DT)
- Frequency measurement (F)
- Phase measurement (PHI)

7.2.3.4.6.1.:CURSor<N>:MODE

(Command/Query)

After a *RST, no cursor is displayed.

Command syntax :
DISPlay[:WINDow]:CURSor{1|2|3}:MODE {OFF|DV|DT|F|PHI}

Parameters :

OFF	no cusor displayed
DV	ΔV measurement; 2 displayed vertical cursors
DT	ΔT measurement; 2 displayed vertical cursors
F	1/dT measurement; 2 displayed vertical cursors
PHI	phase measurement ; 3 displayed vertical cursors

☞ ***Note :*** ***The {1|2|3} numeric sufix has no effect.***

✍ Example : Erase cursor display
DISP:CURS1:MODE OFF<LF>

Query syntax : DISPlay[:WINDow]:CURSor{1|2|3}:MODE?

☞ ***Note :*** ***The {1|2|3} numeric sufix has no effect.***

Response syntax : {OFF|dV|dT|F|Phi|AUTO}<LF>
The AUTO value shows that the instrument has bèen configured in automatic measurement mode.
The CONF?<LF> command indicates the used type of automatic measurement.

7.2.3.4.6.2.:CURSor<N>:POSition

(Command/Query)
Individual cursor positionning.

Command syntax : DISPlay[WINDow]:CURSor{1|2|3}:POSition {<position>}

Parameters : <position> is a whole at NR1 format
Horizontal cursors : min. value (at the bottom of the screen) : 25
max. value (at the top of the screen) : 225
Vertical cursors : min. value (on the left hand side of the screen) : 0
max. value (on the right hand side of the screen) : 250

Example : Vertical cursors : left and right end stop

```
DISP:CURS1: MODE DT<LF>
DISP:CURS1:POS 0<LF>
DISP:CURS2:POS 250<LF>
```

Query syntax : DISPlay[WINDow]:CURSor{1|2|3}POSition?

Response format : <position><LF>

7.2.3.4.6.3.:CURSor<N>:VALue?

(Query)
Informs about the value indicated by the cursors.

Query syntax : DISPlay[:window]:cursor:VALue?

Response format : <measurement><LF>

Example : DISP:CURSor:VALue?<LF>

7.2.4. FORMat commands

The commands in the FORMat directory configure the format of the numeric data transferred using the **TRACe:[DATA]** commands (traced signals).

The structure of the FORMAT command set is:

```
FORMat
    [:DATA]
    :DINTerchange
```

7.2.4.1.[:DATA]

(Command/Query)
The command **FORMat[:DATA]** selects the type of numeric data.

In response to the query **FORMat[:DATA]?**, the oscilloscope returns the selected type.

After a *RST, ASCii is configured.

The available types are: ASCii, INTeger, HEXadecimal, BINary

ASCii

The numeric data is transferred in ASCii characters according to <NR1> numbering. Each number is separated by a comma. The values transferred are between 0 and 255 and correspond to the value of the samples acquired on the 8-bit analogue/digital converters.

Example: 0, 34, 128, 253, 255 : <NR1> notation

INTeger

INTeger reflects transfers of curves in blocks. The numeric data is transferred in 8-bit unsigned integers.

Example:

FORMat:DATA INTeger

11111111 +255
10000000 +128
01111111 +127
00000000 0

HEXadecimal

HEXadecimal is for transfers of <Numeric> curves. The numeric data is transferred in unsigned integers in base 16, preceded by "#H" as specified in IEEE 488.2 and separated by commas.

Example: #H00,#HFF,#H80 <==> 0,255,128

BINary

BINary is for transfers of <Numeric> curves. The numeric data is transferred in unsigned integers in base 2, preceded by "#B" as specified in IEEE 488.2 and separated by commas.

Example:
#B00000000,#B11111111,#B10000000 <==> 0,255,128

Command syntax:

FORMat[:DATA] {ASCii|INTeger|HEXadecimal|BINary}

Example: OUTPUT 707;"FORM ASC"

Query syntax: FORMat[:DATA]?

Response format: {ASCII|INT|HEX|BIN}<NL>

Example: OUTPUT 707;"FORM?"
ENTER 707;Format$
PRINT 707;

7.2.4.2.:DINTerchange

(Command/Query)
The command **FORMat:DINTerchange** specifies whether the numeric data transferred by the TRACe[:DATA] command is preceded by a SCPI format header known as ADIF: see below.

In response to the query **FORMat:DINTerchange?**, the oscilloscope returns the configured state.

After a *RST, there is no ADIF header.

Command syntax: FORMat:DINTerchange {ON|OFF}

Example: OUTPUT 707;"FORM:DINT ON"

Query syntax: FORMat:DINTerchange?

Response format: {ON|OFF}<NL>

Example:
```
OUTPUT 707;"FORM:DINT?"
ENTER 707;DINT$
PRINT 707;DINT$
```

Structure of ADIF data (SCPI)

If the "FORMat:DINTerchange ON" command has been sent to the instrument, the data is in the ADIF format shown below:

Example: Integer format with header:

(ADIF=**S1** (STD(Version 1992.0) DIM=X(TYPE IMPL SCALE **5E-3** SIZE 1000) DIM=Y(TYPE EXPL SCALE **4E-3** OFFSET 128 SIZE 255) DATA (CURVE(VAL**# An<n bytes in binary code>**))))<NL>

#An : The "n" bytes number is coded on A digits.

Example : #41000 ; #516000

General syntax

(ADIF=**<trace-name>** (STD (Version 1992.0) DIM=X(TYPE IMPL SCALE **<Numeric>** SIZE **<Numeric>**) DIM=Y (TYPE EXPL SCALE **<Numeric>** OFFSET **<Numeric>** SIZE 255) DATA(CURVE (VAL**<Block|Numeric>**))))<NL>

Syntax detail:

<trace-name> Trace name (see list in section 7.2.10.1 : CATalog?)

DIMension=X (TYPE IMPL SCALE **<Numeric>** SIZE **<Numeric>**)

1st **<Numeric>** time between two samples on the X axis (in seconds)

2nd **<Numeric>** number of samples in the record

DIMension=Y (TYPE EXPL SCALE **<Numeric>** OFFSET **<Numeric>** SIZE 255)

1st **<Numeric>** potential difference for one unit in y axis (in volts)
Note : The value takes the probe coefficient into account.

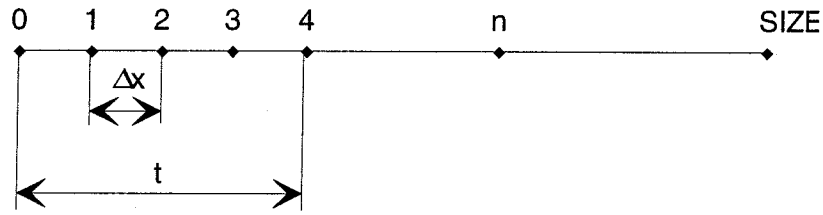
2nd **<Numeric>** value of Y corresponding to zero volt

DATA(CURVE(VAL<**Block|Numeric**>))))

<Block|Numeric>	contains curve-related data in "Block" or "Numeric" form
"Block" form	INTeger type format with an ASCII header and binary coded data (see section 7.2.4. FORMat commands)
"Numeric" format	ASCII, HEXadecimal or BINary format (see section 7.2.4. FORMat commands)

Interpreting scales

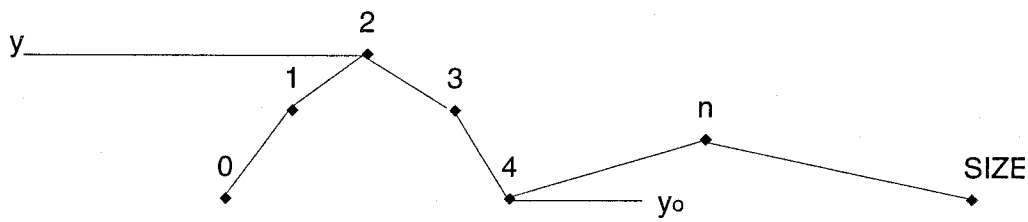
* Point in time:



The point in time of the "nth" sample in the record expressed in seconds is:

$t = n * \Delta x$ where Δx is given by SCALE **<Numeric>**

* Absolute potential of a point in volts:



$$V = (y - y_O)^* \Delta y$$

where: y : value of the point

yo : value of the Y corresponding to zero volts given by OFFSET **<Numeric>**

Δy : potential difference for a unit in y, given by SCALE **<Numeric>**

7.2.5. HARDcopy commands

The HARDcopy: directory contains the commands for configuring the screen hard copy:

:FORMat
:GRATicule
:STATus

7.2.5.1.:FORMat

(Command/Query)
The command **HARDcopy:FORMat** can be used to select a data format for executing a hard copy. The format defines HPGL type (HPGL), IBM Proprinter (IBM_Pro), ESC P2 (ESC_P2) or HP-PCL (HP_PCL).

Caution: ***The IBM Proprinter, ESC P2, HP-PCL or HPGL printer format cannot be used correctly on an XON/XOFF serial link.***

In response to the query **HARDcopy:FORMat?**, the oscilloscope returns the current format.

After a *RST, the IBM_PRO format is configured.

Command syntax: HARDcopy:FORMat {HPGL|IBM_PRO|ESC_P2|HP_PCL}

Example:
```
OUTPUT 707 ; "HARDcopy:FORMat"
```

Query syntax: HARDcopy:FORMat?

Response format: {HPGL|IBM_PRO|ESC_P2|HP_PCL}<NL>

Example:
```
OUTPUT 707 ; "HARDcopy:FORM?"
ENTER 707 ; FORMAT$
PRINT FORMAT$
```

7.2.5.2.:GRATicule

(Command/Query)
The command **HARDcopy:GRATicule** selects or deselects reproduction of the graticule on the hard copy.
In response to the query **HARDcopy:GRATicule?**, the oscilloscope returns the configuration of the graticule option.

After a *RST, the graticule is not reproduced on the hard copy.

Command syntax: HARDcopy:GRATicule {ON|OFF}

Example:
```
OUTPUT 707 ; "HARDcopy:GRAT ON"
```

Query syntax: HARDcopy:GRATicule?

Response format: {ON|OFF}<NL>

Example:
```
OUTPUT 707 ; "HARDcopy:GRAT?"
ENTER 707 ; GRATICULE$
PRINT GRATICULE$
```

7.2.5.3.:STATus

(Command/Query)
The command **HARDcopy:STATus** selects or deselects printout of the instrument's internal configuration.

In response to the query **HARDcopy:STATus?**, the oscilloscope returns the status of the option.

After a *RST, the instrument's configuration is not printed.

Command syntax: HARDcopy:STATus {ON|OFF}

Example: OUTPUT 707 ; "HARDcopy:STAT ON"

Query syntax: HARDcopy:STATus?

Response format: {ON|OFF}<NL>

Example:
```
OUTPUT 707 ; ":HARD:STAT?"
ENTER 707 ; STAT$
PRINT STAT$
```

7.2.6. INPut<N> commands

The INPut<N> directory contains the commands for configuring input channels CH1 and CH2:

COUPling
INVert *(OX 8040 / OX 8050 / OX 8100)*
RANGe
VERNier
FILTer
PROBe *(OX 8042 / OX 8062)*

7.2.6.1.:COUPling

(Command/Query)
The command **INPut<N>:COUPling:** configures the input coupling of the specified channel.

In response to the query **INPut<N>:COUPling?**, the oscilloscope returns the input coupling configuration of the specified channel.

After a *RST, AC coupling is configured on both channels.

Command syntax: INPut<N>:COUPling {AC|DC|GND}
where <N> : : = 1 or 2 (channel 1 or channel 2)

Example: OUTPUT 707 ; "INPut1:COUP AC"

Query syntax: INPut<N>:COUPling?
where <N> : : = 1 or 2 (channel 1 or channel 2)

Response format: {AC|DC|GND}<NL>

Example:
```
OUTPUT 707 ; "INPUT1:COUP?"
ENTER 707 ; COUP$
PRINT COUP$
```

7.2.6.2.:INVert *(OX 8040 / OX 8050 / OX 8100)*

(Command/Query)
The command **INPut<N>:INVert:** configures channel 2 inversion (<N> can only be 2).

In response to the query **INPut<N>:INVert?**, the oscilloscope returns the input coupling configuration of the specified channel.

For channel 1, the response is always OFF.

After a *RST, CH2 is not inverted (OFF).

Command syntax: INPut<N>:INVert {ON|OFF}
where <N> : : = 2 (channel 2 only)

Example: OUTPUT 707 ; ":INPUT2:INV ON"

Query syntax: INPut<N>:INVert?
where <N> : : = 1 or 2 (channel 1 or channel 2)

Response format: {ON|OFF}<NL>

Example: OUTPUT 707 ; ":INP1:INVert?"
ENTER 707 ;INVERT$
PRINT INVERT$

7.2.6.3.:RANGe

(Command/Query)
The command **INPut<N>:RANGe:** configures the input sensitivity of the specified channel (in volt/division and sequences 1-2-5) without considering the selected probe coefficient.

In response to the query **INPut<N>:RANGe?**, the oscilloscope returns the sensitivity of the specified channel (in volt/division).

After a *RST, the instrument is at 0.1 V/div. *(OX 8040 / OX 8050 / OX 8100)*
1 V/div. *(OX 8042 / OX 8062)*

Command syntax: INPut<N>:RANGe <sensitivity>
where <N> : : = 1 or 2 (channel 1 or channel 2) and <sensitivity> : : =
(OX 8100) {2E-3|5E-3|10E-3|20E-3|50E-3|100E-3|200E-3|500E-3|1|2|5}
(OX 8050) {1E-3|2E-3|5E-3|10E-3|20E-3|50E-3|100E-3|200E-3|500E-3|1|2|5|10|20}
(OX 8042 / OX 8062) {10E-3|20E-3|50E-3|100E-3|200E-3|500E-3|1|2|5|10|20|50|100|200}
(OX 8040) {5E-3|10E-3|20E-3|50E-3|100E-3|200E-3|500E-3|1|2|5|10|20}

Example: OUTPUT 707 ; "INPUT1:RANGe 1E-1"

Query syntax: INPut<N>:RANGe?
where <N> : : = 1 or 2 (channel 1 or channel 2)

Response format: <sensitivity> <NL>
where <sensitivity> is defined above

Example: OUTPUT 707 ; "INP1:RANGe?"
ENTER 707 ; RANGE$
PRINT RANGE$

7.2.6.4.:VERNier

(Query)
In response to the query **INPut<N>:VERNier?**, the oscilloscope returns the variable gain (calibrated (OFF) or not (ON)) for the specified channel.

Query syntax: INPut<N>:VERNier?
where <N> : : = 1 or 2 (channel 1 or channel 2)

Response format: {ON|OFF}<NL>

Example:
```
OUTPUT 707 ; "INP1:VERNier?"
ENTER 707 ; VERNIER$
PRINT VERNIER$
```

7.2.6.5.:FILTer[:LPASs][:STATe] *(OX 8100)*

(Command/Query)
The command **INPut<N>:FILTer[:LPASs][:STATe]** configures the oscilloscope bandwidth for the 2 inputs channel what ever is the mark N (reduction of the band to 20 MHz :ON / No filter :OFF).

Note : The bandwidth limit is disabled in digital mode.

In response to the query **INPut<N>:FILTer[:LPASs][:STATe]?**, the oscilloscope returns the input filter configuration.

After a *RST, the filter is OFF.

Command syntax : INPut<N>:FILTer[:LPASs][:STATe]{ON|OFF}
where <N> : : = 1 or 2 (channel 1 or 2)

Example :
```
OUTPUT 707 ; "INPUT1:FILTer ON"
```

Query syntax : INPut<N>:FILTer[:LPASs][:STATe]?
where <N> : : = 1 or 2 (channel 1 or 2)

Response syntax : {ON|OFF} <NL>

Example :
```
OUTPUT 707 ; "INP1:FILT?"
ENTER 707 ; filter$
PRINT filter$
```

7.2.6.6.:PROBe

(Command/Query)
The probe coefficient affects the sensitivity, the cursor measurements and the automatic measurements.
Three coefficients are available : 1, 10 and 100 *(OX 8040 / OX 8050 / OX 8100)*

After a *RST, a x1 probe coefficient is configured on both channels.

Command syntax : INPut{1|2}:PROBE {X1|X10|X100} *(OX 8040 / OX 8050 / OX 8100)*

Example : INP1:PROB X10
x10 probe on channel 1 *(OX 8042 / OX 8062)*

Query syntax : INPut{1|2}:PROBe? *(OX 8040 / OX 8050 / OX 8100)*

Response format : {X1|X10|X100}<LF> *(OX 8042 / OX 8062)*

7.2.6.7.:MODE (OX 8042 / OX 8062)

(Command/Query)
INPut<N>:MODE: configures the oscilloscope using mode(<N> can be 1 or 2).

In response to the query **INPut<N>:MODE?,]?**, the oscilloscope returns the input filter configuration.

After a *RST, the differential mode is active on both channels.

Command syntax : INPut<N>:MODE{NORIDIFF}
with <N>: : = 1 ou 2 (voie 1 ou 2)

Example : OUTPUT 707 ; ":INPut1:MODE DIFF"

Query syntax : INPut<N>:MODE ?
with <N>: : = 1 or 2 (channel 1 or 2)

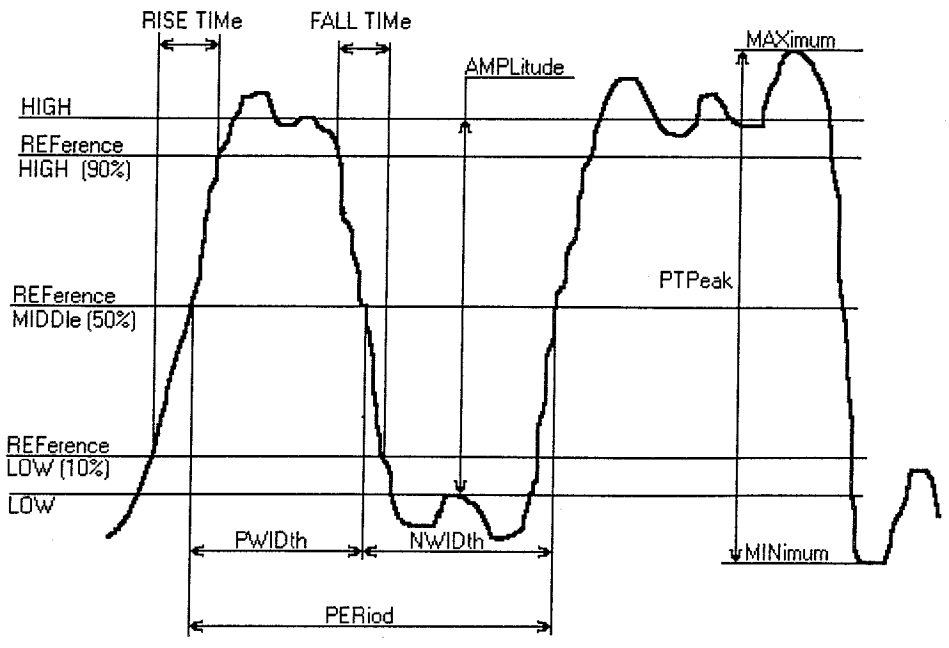
Response format : {NORIDIFFI}<NL>

Example : OUTPUT 707 ; ":INPut1:MODE?"
ENTER 707 ; MODE$
PRINT MODE$

7.2.7. MEASure/CONFigure commands

The MEASure directory is used to configure the type of measurement and return the results in a single command. The CONFigure directory merely configures the type of measurement: the results are displayed on the oscilloscope screen but are not sent to the controller.

MEASure<function>? is used in the query form only.
CONFigure<function> is used in both command and query forms. For a query, the oscilloscope returns the current measurement type configured by MEASure or CONFigure.



7.2.7.1.MEASure?

(Query)
In response to the query **MEASure[:SCALar][:VOLTage]:<type>? (@<source>)**, the oscilloscope configures the measurement type and source specified in the command and returns the result.

After a *RST, no measurement is configured.

Query syntax:

MEASure[:SCALar][:VOLTage]:<type>? (@<source>)

where <type>::={ACIDCIFREQuencyIPERiodIPHASeIAMPLitudeILOWIHIGHI
RISE:TIMEIRTIMeIFALL:TIMEIFTIMeINWIDthIPWIDthIPDUTyI
NDUTyIMINimumIMAXimumIPTPeak}

AC	root mean square voltage	Vrms
DC	average voltage	Vavg
FREQuency	frequency	F
PERiod	period	T
PHASe	phase	φ
AMPLitude	amplitude	Vamp
LOW	low voltage	Vlow
HIGH	high voltage	Vh
RISE:TIME or RTIMe	rise time	tr
FALL:TIME or FTIMe	fall time	tf
NWIDth	negative pulse width	W-
PWIDth	positive pulse width	W+
PDUTy	duty cycle (PWIDth/PERiod)	DC+
NDUTy	duty cycle (NWIDth/PERiod)	DC-
MINimum	minimum voltage	Vmin
MAXimum	maximum voltage	Vmax
PTPeak	peak-to-peak voltage	Vpp

and <source>::= 1 or 2

Response format: <value> <NL>
where <value>::=result in <NR2> or <NR3> format

If the measurement is decalibrated (display in div.) or impossible, the instrument answers : "not applicable"<NL>.

Example:
```
OUTPUT 707 ; "MEAS? (@2)"
ENTER 707 ; VOLT2$
PRINT VOLT2$;
```

Notes: ***. The abbreviated form MEAS? configures the [:DC] measurement and returns the result.***

. If (@<source>) is not specified according to the selected vertical mode :

CH1 :	***CH1 source selection***
CH2 :	***CH2 source selection***
ALT :	***Source selection in the CURS menu***
CHOP :	***Source selection in the CURS menu***
DUAL :	***Source selection in the CURS menu*** *(OX 8040)*

XY : ***Source selection in the CURS menu***

7.2.7.2.CONFigure

(Command/Query)
The command **CONFigure[:SCALar][:VOLTage]:<type> (@<source>)** configures the specified type of measurement.

In response to the query **CONF?**, the oscilloscope returns the current measurement type.

After a *RST, no measurement is configured.

Command syntax: CONFigure[:SCALar][:VOLTage]:<type> (@<source>)

where <type>::={OFF|AC|DC|FREQuency|PERiod|PHASe|AMPLitude|LOW|HIGH|
RISE:TIME|RTIMe|FALL:TIME|FTIMe|NWIDth|PWIDth|PDUTy|
NDUTy|MINimum|MAXimum|PTPeak}

OFF	automatic measurements are disabled	
AC	root mean square voltage	Vrms
DC	average voltage	Vavg
FREQuency	frequency	F
PERiod	period	T
PHASe	phase	φ
AMPLitude	amplitude	Vamp
LOW	low voltage	Vlow
HIGH	high voltage	Vh
RISE:TIME or RTIMe	rise time	tr
FALL:TIME or FTIMe	fall time	tf
NWIDth	negative pulse width	W-
PWIDth	positive pulse width	W+
PDUTy	duty cycle (PWIDth/PERiod)	DC+
NDUTy	duty cycle (NWIDth/PERiod)	DC-
MINimum	minimum voltage	Vmin
MAXimum	maximum voltage	Vmax
PTPeak	peak-to-peak voltage	Vpp

and <source>::= 1 or 2

Example: OUTPUT 707 ; "CONF (@2)"

Notes: ***. The abbreviated form CONF configures the [:DC] measurement***
. If (@<source>) is not specified
according to the selected vertical mode :

CH1 :	***CH1 source selection***
CH2 :	***CH2 source selection***
ALT :	***Source selection in the CURS menu***
CHOP :	***Source selection in the CURS menu***
DUAL :	***Source selection in the CURS menu*** *(OX 8040)*
XY :	***Source selection in the CURS menu***

Command syntax: CONF?

Response format: <type> <NL>

where <type> is defined above.

Example:

```
OUTPUT 707 ; "CONF?"
ENTER 707 ; TYPE$
PRINT TYPE$
```

7.2.8. SYSTem commands

The SYSTem directory is structured as follows:

:ERRor?
:KLOCk
:SET
:VERSion?

7.2.8.1.:ERRor?

(Query)
In response to the query **SYSTem:ERRor?**, the oscilloscope returns the error number at the top of the queue. The queue has ten numbers and operates on the first in, first out principle.

With each SYSTem:ERRor? query received, the oscilloscope returns the next error number in order of arrival until the queue is empty. Any additional SYSTem:ERRor? query will provoke a negative response: "0" (ASCII code 48). If the queue is full, the cell at the top of the queue takes the value -350 (Queue full).

The queue is emptied:

- on power up
- on receipt of a *CLS
- on reading of the last error

Query syntax: SYSTem:ERRor?

Response format: <error><NL>
where <error>::=negative integer or null (see values in the table)

Example:

```
OUTPUT 707;"SYSTem:ERRor?"
ENTER 707;error$
PRINT error$
```

List of errors:

* Command errors: (-199 to -100)
These indicate that a syntax error has been detected by the syntax analyzer and set bit 5 of the event register to 1, CME (command Error).

-100	:	Command error
-101	:	Invalid Character
-102	:	Syntax error
-103	:	Invalid separator
-104	:	Data type error

-105	:	GET not allowed
-108	:	Parameter not allowed
-109	:	Missing parameter
-111	:	Header separator error
-112	:	Program mnemonic too long
-113	:	Undefined header
-114	:	Header suffix out of range
-121	:	Invalid character in number
-123	:	Numeric overflow
-124	:	Too many digits
-128	:	Numeric data not allowed
-130	:	Suffix error
-131	:	Invalid suffix
-138	:	Suffix not allowed
-140	:	Character data error
-141	:	Invalid character data
-144	:	Character data too long
-148	:	Character data not allowed
-150	:	String data error
-151	:	Invalid string data
-158	:	String data not allowed
-160	:	Block data error
-161	:	Invalid block data
-168	:	Block data not allowed
-170	:	Expression error
-171	:	Invalid expression
-178	:	Expression data not allowed

* Execution errors: (-299 to -200)
  These indicate that an error has been detected on execution of a command and set bit 4 of the event register to 1, EXE (Execution Error).

-200	:	Execution error
-211	:	Trigger ignored
-221	:	Settings conflict
-222	:	Data out of range
-223	:	Too much data

* Device-dependent errors: (-399 to -300)
  These indicate that a fault has been detected during execution of a task and set bit 3 of the event register to 1, DDE (Device Dependent Error).

-300	:	Device specific error
-310	:	System error
-350	:	Too many errors

* Query errors: (-499 to -400)
  These indicate that a fault has occurred in the data interchange protocol and set bit 2 of the event register to 1, QYE (QuerY Error).

-400	:	Query error
-410	:	Query INTERRUPTED
-420	:	Query UNTERMINATED
-430	:	Query DEADLOCKED

-440 : Query UNTERMINATED after indefinite response

7.2.8.2.:KLOCk

(Command/Query)
The command **SYSTem:KLOCk** locks (ON) or unlocks (OFF) the instrument's front panel.

In response to the query **SYSTem:KLOCk?**, the oscilloscope returns the state of its front panel.

After a *RST, the front panel is unlocked (:SYSTem:KLOCk OFF).

Command syntax: SYSTem:KLOCk {ON|OFF}

Example: OUTPUT 707;":SYSTem:KLOCk ON"

Query syntax: SYSTem:KLOCk?

Response format: {ON|OFF}<NL>

Example: OUTPUT 707;"SYSTem:KLOCk?"
ENTER 707;KLOCk$
PRINT KLOCk$

7.2.8.3.:SET

(Command/Query)
The command **SYSTem:SET** configures the oscilloscope according to the data stored in the configuration sent by the controller.

The instrument's configuration is stored in 62 binary-coded bytes excluding the header "#800000062" *(OX 8040 / OX 8050 / OX 8100).*

The instrument's configuration is stored in 64 binary-coded bytes excluding the header "#800000064" *(OX 8042 / OX 8062).*

This data does not affect the configuration of the communication interfaces.

In response to the query **SYSTem:SET?**, the oscilloscope returns its current configuration preceded by the header "#800000062" *(OX 8040 / OX 8050 / OX 8100)*
"#800000064" *(OX 8042 / OX 8062)*
This command is equivalent to the common command ***LRN?**.

The instrument configuration is sent and received in the form of blocks of binary data. The data transmission format is as stipulated in the IEEE 488.2 standard.

Command syntax: SYSTem:SET <setup>
where <setup>::=#800000062<setup data string> *(OX 8040 / 8050 / 8100)*
#800000064<setup data string> *(OX 8042 / OX 8062)*

Example: OUTPUT 707;":SYSTem:SET <setup>"

Query syntax: SYSTem:SET?

Response format: <setup>

Example: OUTPUT 707;"SYSTem:SET?"
ENTER 707;SETUP$
PRINT SETUP$

7.2.8.4.:VERSion?

(Query)
In response to the query **SYSTem:VERSion?**, the instrument returns a numeric format (NR2) corresponding to the SCPI version it supports. The response includes the year and the version number.

Query syntax: SYSTem:VERSion?

Response format: <YYYY.V>
where Y represents the year and V the version

Example:
```
OUTPUT 707;"SYSTem:VERS?"
ENTER 707;VERS$
PRINT VERS$
```

7.2.9. TIMebase commands

The <TIMEbase> directory contains commands relating to the timebase :

DELay
EXPH
MODE
RANGe
POINts

7.2.9.1.:DELay

(Command/Query)
The command **TIMebase:DELay** configures the horizontal mode.

In response to the query **TIMebase:DELay?**, the oscilloscope returns the horizontal mode.

After a *RST, the horizontal mode is time base A mode.

Command syntax: TIMEbase:DELay"TIMebase:DELay DEL"

Example:
```
OUTPUT 707 ; "TIMebase:DELay DEL"
```

Query syntax: TIMEbase:DELay?

Response format: {OFFIALTIDELay}<NL> *(OX 8100)*
{OFFISEARchIDELay}<NL> *(OX 8040 / OX 8050 OX 8042 / OX 8062)*

Example:
```
OUTPUT 707 ; "TIM:DEL?"
ENTER 707;DELAY$
PRINT DELAY$
```

7.2.9.2.:EXPH

(Command/Query)
The command **TIMebase:EXPH** configures x10 horizontal expansion. This applies in analogue mode only.
In response to the query **TIMebase:EXPH?**, the oscilloscope returns the x10 state.

After a *RST, x10 expansion is OFF.

Command syntax: TIMEbase:EXPH {ON|OFF}

Example: OUTPUT 707 ; "TIMebase:EXPH ON"

Query syntax: TIMEbase:EXPH?

Response format: {ON|OFF}<NL>

Example:
```
OUTPUT 707 ; "TIMebase:EXPH?"
ENTER 707 ; EXPH$
PRINT EXPH$
```

7.2.9.3.:MODE

(Command/Query)
The command **TIMebase:MODE** sets the timebase to AUTO or trigger mode.

In response to the query **TIMebase:MODE?**, the oscilloscope returns the current timebase mode.

After a *RST, AUTO mode is configured.

Command syntax: TIMEbase:MODE {AUTO|NORM}

Example: OUTPUT 707 ; "TIMebase:MODE AUTO"

Query syntax: TIMEbase:MODE?

Response format: {AUTO|NORM}<NL>

Example:
```
OUTPUT 707 ; "TIMe:MODE?"
ENTER 707 ; TIMODE$
PRINT TIMODE$
```

7.2.9.4.:RANGe

(Command/Query)

OX 8040 / OX 8050 / OX 8042 / OX 8062 :
The command **TIMebase:RANGe** configure the timebase A range.

OX 8100 :
The command **TIMebase:RANGe** configure the timebase A range if the horizontal mode is normal (OFF) and the timebase B if the horizontal mode is alternate or delayed.
The timebase B must always be inferior or equal to the timebase A.
In response to the query **TIMebase:RANGe?**, the oscilloscope returns the timebase range A or B according to the horizontal mode configuration.

After a *RST, the instrument is configured at 50 µs/div.

Command syntax: TIMEbase:RANGe <range>
(OX 8050 / OX 8100 / OX 8042 / OX 8062)
where <range>::= {5E-9|10E-9|20E-9|50E-9|0.1E-6|0.2E-6|0.5E-6|1E-6|2E-6|5E-6|10E-6|20E-6|50E-6|01.E-3|0.2E-3|0.5E-3|1E-3|2E-3|5E-3|10E-3|20E-3|50E-3|100E-3|200E-3|500E-3|1|2|5|10|20|50|100|200}

(OX 8040) where <range>::={10E-9|20E-9|50E-9|0.1E-6|0.2E-6|0.5E-6|1E-6|2E-6|5E-6|5E-9|10E-6|20E-6|50E-6|0.1E-3|0.2E-3|0.5E-3|1E-3|2E-3|5E-3|10E-3|5E-9|20E-3|20E-9|50E-3|100E-3|200E-3|500E-3|1|2|5|10|20|50|100|200}

Example: OUTPUT 707 ; "TIMebase:RANGe 1E-3"

Query syntax: TIMEbase:RANGe?

Response format: <range><NL>

Example: OUTPUT 707 ; ":TIMe:RANG?"
ENTER 707 ; RANGE$
PRINT RANGE$

Note :

Analog mode ***<range>: : =*** ***50E-9 to 100E-3 (OX 8100)***
50E-9 to 200E-3 (OX 8040 / OX 8050)
0.2E-6 to 200E-3 (OX 8062)
0.5E-6 to 200E-3 (OX 8042)

Digital mode
SINGle or REFResh mode
<range>: : = ***5E-9 to 200 (OX 8050 / OX 8100 / OX 8042 / OX 8062)***
10E-3 to 200 (OX 8040)
Mode Roll ***<range>: : =*** ***100E-3 to 200 (all versions)***

7.2.9.5.:POINts

(Command/Query)
TIMebase:POINts command configures size and acquisition in points.

After a *RST, the size of acquisition is configured at 1000 points.

Command syntax : TIMEbase:POINts {1000|8000|16000}

Example : TIMEbase:POINts 1000

Query syntax : TIMEbase:POINts?

Response format : {1000|8000|16000}<LF>

7.2.10. TRACe commands

The commands in this directory are used to transfer data relating to signals that have been saved or displayed.
They are structured as follows:

TRACe
:CATalog?
[:DATA]

7.2.10.1.:CATalog?

(Query)
In response to the query **TRACe:CATalog?**, the oscilloscope returns the list of available signals; each name is separated by a comma.

Query syntax: TRACe:CATalog?

Response format: <name 1>, <name 2>, ..., <name n><NL>

Example:
```
OUTPUT 707;"TRACe:CATalog?"
ENTER 707;CAT$
PRINT CAT$
```

List of signals:

CH1 :	CH1 saved	(1, 8, 16 k) (*)
CH2 :	CH2 saved	(1, 8, 16 k) (*)
S1 :	signal 1 displayed	(1 k)
S2 :	signal 2 displayed	(1 k)
CH1L :	CH1 acquired in envelope mode (low peak)	(1, 8, 16 k) (*)
CH1H :	CH1 acquired in envelope mode (high peak)	(1, 8, 16 k) (*)
CH2L :	CH2 acquired in envelope mode (low peak)	(1, 8, 16 k) (*)
CH2H :	CH2 acquired in envelope mode (high peak)	(1, 8, 16 k) (*)

(*) acc. to size of acquisition

Note: ***S1 is the signal that has its parameters displayed at the top left of the screen. S2 is the signal that has its parameters displayed at the top right (does not exist unless vertical mode is set to ALT, CHOP, XY).***

7.2.10.2.[:DATA]?

(Query)
In response to the query **TRACe[:DATA]?**, the instrument returns the values of the specified trace, according to the format specified in the FORMat command (see section 7.2.4.).
These values can be structured according to the ADIF SCPI model (see FORMat:DINTerchange).

Query syntax: TRACe[:DATA]?<trace-name>
where trace-name ::={CH1|CH2|S1|S2|CH1L|CH1H|CH2L|CH2H}

Example: OUTPUT 707;"TRACe? S1"

Formats: ASCii, INTeger, HEXadecimal, BINary

ASCii all numbers (0 to 255) are in ASCII decimal code and separated by commas.

Example: 0, 34, 128, 253, 255,

HEXadecimal all numbers (0 to 255) are in ASCII hexadecimal code, preceded by #H and separated by commas.

Example: # H00,# H22,# H80,# HFD,# HFF,

BINary all numbers (0 to 255) are in ASCII binary code, preceded by #B and separated by commas.

Example: #B00000000,#B00100010,#B10000000,#B11111101, #B11111111,

INTeger all numbers are sent in binary code. An ASCII header defining the number of binary values transmitted precedes the data block.

Example: #41000 <binary block> (read S1 or S2)
#51600 (read CH1 or CH2 with a size of acquisition of 16 k)

where <binary block> contains the list of 1024, 4096 or 8192 binary values.

7.2.11. TRIGger commands

The TRIGger directory contains the following commands:

COUPling
MODE
SLOPe
SOURce
VIDeo:FIELd:LPFRame
FIELd:SELect
LINE
SSIGnal

7.2.11.1.:COUPling

(Command/Query)
The command **TRIGger:COUPling** configures the trigger input coupling.

In response to the query **TRIGger:COUPling?**, the oscilloscope returns the trigger input coupling.

After a *RST, DC coupling is configured.

Command syntax: TRIGger:COUPling
{DC|AC|LF|HF|TV|HITVV} *(OX 8100 / OX 8042 / OX 8062)*
{DC|AC|LF|HF|TV} *(OX 8040 / OX 8050)*

Example: OUTPUT 707 ; "TRIG:COUP HF"

Query syntax: TRIGger:COUPling?

Response format: {DC|AC|LF|HF|TV|HITVV}<NL> *(OX 8100 / OX 8042 / OX 8062)*
{DC|AC|LF|HF|TV}<NL> *(OX 8040 / OX 8050)*

Example: OUTPUT 707 ; "TRIG:COUP?

```
                                        ENTER 707 ; TRIGCOUP$
                                        PRINT TRIGCOUP$"
```

7.2.11.2.:MODE *(OX 8050 / OX 8100)*

(Command/Query)
The command **TRIGger:MODE** configures the trigger in NORMAL or PEAK TO PEAK mode.

In response to the query **TRIGger:MODE?**, the oscilloscope returns the current trigger mode.

After a *RST, NORMAL mode is configured.

Command syntax: TRIGger:MODE {NORM|PTP}

Example:
```
OUTPUT 707 ; "TRIG:MODE PTP"
```

Query syntax: TRIGger:MODE?

Response format: {NORM|PTP}<NL>

Example:
```
OUTPUT 707 ; "TRIG:MODE?"
ENTER 707 ; MODE$
PRINT MODE$"
```

7.2.11.3.:SLOPe

(Command/Query)
The command **TRIGger:SLOPe** configures the trigger edge (positive-going or negative-going).

In response to the query **TRIGger:SLOPe?**, the oscilloscope returns the current trigger slope.

After a *RST, the positive edge slope is configured.

Command syntax: TRIGger:SLOPe {+|-}

Example:
```
OUTPUT 707 ; "TRIG:SLOPE +"
```

Query syntax: TRIGger:SLOPe?

Response format: {+|-}<NL>

Example:
```
OUTPUT 707 ; "TRIG:SLOP?"
ENTER 707 ; front$
```

7.2.11.4.:SOURce

(Command/Query)
The command **TRIGger:SOURce** configures the trigger source.

In response to the query **TRIGger:SOURce?**, the oscilloscope returns the current trigger source.

After a *RST, CH1 is the source.

Command syntax: TRIGger:SOURce {CH1|CH2|ALT|LINE|EXT}

Example: OUTPUT 707 ; "TRIG:SOURCE CH2"

Query syntax: TRIGger:SOURce?

Response format: {CH1|CH2|ALT|LINE|EXT}<NL>

Example: OUTPUT 707 ; "TRIG:SOUR?"
ENTER 707 ; source$
PRINT source$

7.2.11.5.:B (OX 8100)

(Command/Query)
The command **TRIGger:B** configure the trigger mode of the timebase B.

RUN after Delay : TRIG:B OFF
TRIG after Delay : TRIG:B ON

In response to the query **TRIGger:B?**, the oscilloscope return the trigger mode of the timebase B.

After a *RST, TRIGger:B is OFF.

Command syntax : TRIGger:B {ON/OFF}

Example : OUTPUT 707 ; "TRIger:B ON"

Query syntax : TRIGger:B?

Response syntax : {ON|OFF}<NL>

Example : OUTPUT 707 ; "TRIGger:B?"
ENTER 707 ; trigb$
PRINT trigb$

7.2.11.6.:VIDeo *(OX 8040 / OX 8050 / OX 8100)*

7.2.11.6.1.:FIELd:LPFRame

(Command/Query)
TV standard selection.

After a *RST, the standard 625 lines is configured.

Command syntax : TRIGger:VIDeo:FIELd:LPFRame {625|525}

Parameters : 625 : standard 625 lines
525 : standard 525 lines

Example : TRIG:VID:FIEL:LPFR 625

Querry syntax : TRIGger:VIDeo:FIELd:LPFRame?

Response format : {625|525}<LF>

7.2.11.6.2.:FIELd:SELect

(Command/Query)
This command selects the field (odd or even) in which the video trigger line is selected.

After a *RST, odd field selection (ODD).

Command syntax : TRIGger:VIDeo:FIELd:SELect {ODD|EVEN}

Parameters : ODD : odd field
EVEN :even field

Example : TRIG:VID:FIEL:SEL ODD

Query syntax : TRIGger:VIDeo:FIELd:SELect?

Response format : {ODD|EVEN}<LF>

7.2.11.6.3.:LINE

(Command/Query)
Trigger line selection in active field.

After a *RST, line no. 1 selection.

Command syntax : TRIGger:VIDeo:LINE {<ligne>}

Parameters : <line> is a integer. It cannot be higher than the number of lines of the selected standard.

Standard 525 lines :
odd raster : line <1...263>
even raster : line <1...262>

Standard 625 lines :

odd raster :	line <1...313>
display :	<1...313>
even raster :	line <1...312>
display :	<314...625>

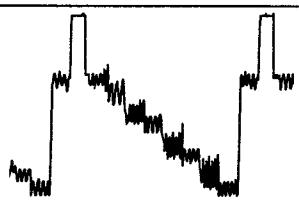
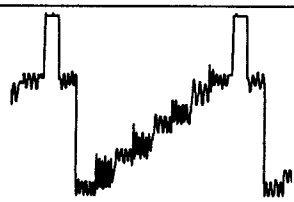
Example : TRIG:VID:LINE 1

Query syntax : TRIGger:VIDeo:LINE ?

Response format : <line><LF>

7.2.11.6.4.:SSIGnal

(Command/Query)

Video signal polarity	
Positive polarity video signal	**Negative polarity video signal**
	

After a *RST, positive video selection.

Command syntax : TRIGger:VIDeo:SSIGnal {POSitive|NEGative}

Parameters :

POSitive :	positive video
NEGative :	negative video

Example : TRIG:VID:SSIGnal POS

Query syntax : TRIG:VID:SSIGnal?

Response format : {POS|NEG}<LF>

8. USING THE INTERFACES

8.1. Introduction

This section describes the functionalities of the interfaces according to the IEEE 488.2 standard.

IEEE 488.2 defines the message transfer protocol during communication between a controller and an instrument and certain common commands that are found on all IEEE 488.2 compliant instruments.

IEEE 488.2 is designed for the IEEE 488.1 hardware interface. When the RS232C interface is used, IEEE488.2 is observed within the limitations of any hardware differences.

8.2. Protocols

8.2.1. Functional elements

Input buffer

The instrument's input buffer is a memory area in which the commands and queries are stored before being unstacked, analyzed and executed. This buffer has a capacity of 128 bytes. The filling of this buffer is handled internally in the instrument.

Output buffer

The instrument's output buffer is a memory area in which the output data is stored until the destination is read. It also has a 128-byte capacity.
The stored data is response messages when remote programming or codes for controlling a digital plotter or printer when executing a hard copy.

Syntax analyzer

The syntax analyzer is the functional block which interprets the commands temporarily stored in the input buffer and decides what action to take.
The analysis of a command starts when a terminator is received or when the input buffer is full.

8.2.2. General information on protocols

The instrument and controller communicate using programming and response messages. These may contain more than one programming command or response.

The programming messages are sent by the controller to the instrument.
The response messages are sent by the instrument to the controller in response to a query message.
A query message is defined as a programming message containing one or more questions. The instrument replies only if the query is valid and it has something to say.

The controller must not try to read a response unless it has previously sent a query message (otherwise it will jam in read mode). It must read the message before sending another programming message (see the following section on "Detailed description" for the instances in which these rules do not apply).

8.2.3. Detailed description

When the instrument is powered up or a "device clear" command is received (IEEE 488), the input and output buffers are reset and the syntax analyzer returns to the top of the tree.

The instrument and controller communicate by sending complete programming and response messages. This means that a controller must always send a complete programming message before reading a response.

If a query message is sent, the next message over the bus is a response message. The controller must always read the response before sending another programming message.

The instrument can handle multiple-query messages. The individual queries are then separated by commas as are the related responses.

The commands are executed in their order of arrival (first in, first executed).

EXCEPTIONS

Command error

Occurs when there is a syntax error or an unknown command header.

Execution error

Occurs if a parameter is out of limits or if the instrument's current configuration does not allow execution of the command.

Query error

Occurs if the response read protocol is not followed.

Incomplete read

If the controller does not read the entire response and tries to send another command message, the instrument generates a query error. The unread part of the response (or all of it) is cleared from the output buffer. The programming message stored in the input buffer is not affected.

Incomplete write

If the controller tries to read a response message before terminating a programming message, the instrument generates a query error. The syntax analyzer returns to the top of the tree and the response is cleared from the output buffer without being sent over the bus.

Talker with nothing to say

If the instrument is addressed as the talker before receiving a query (the controller tries to read a response without having asked a question), it indicates a query error and sends no characters over the bus.

If the instrument receives an invalid query, it generates a command error and sends no characters over the bus.

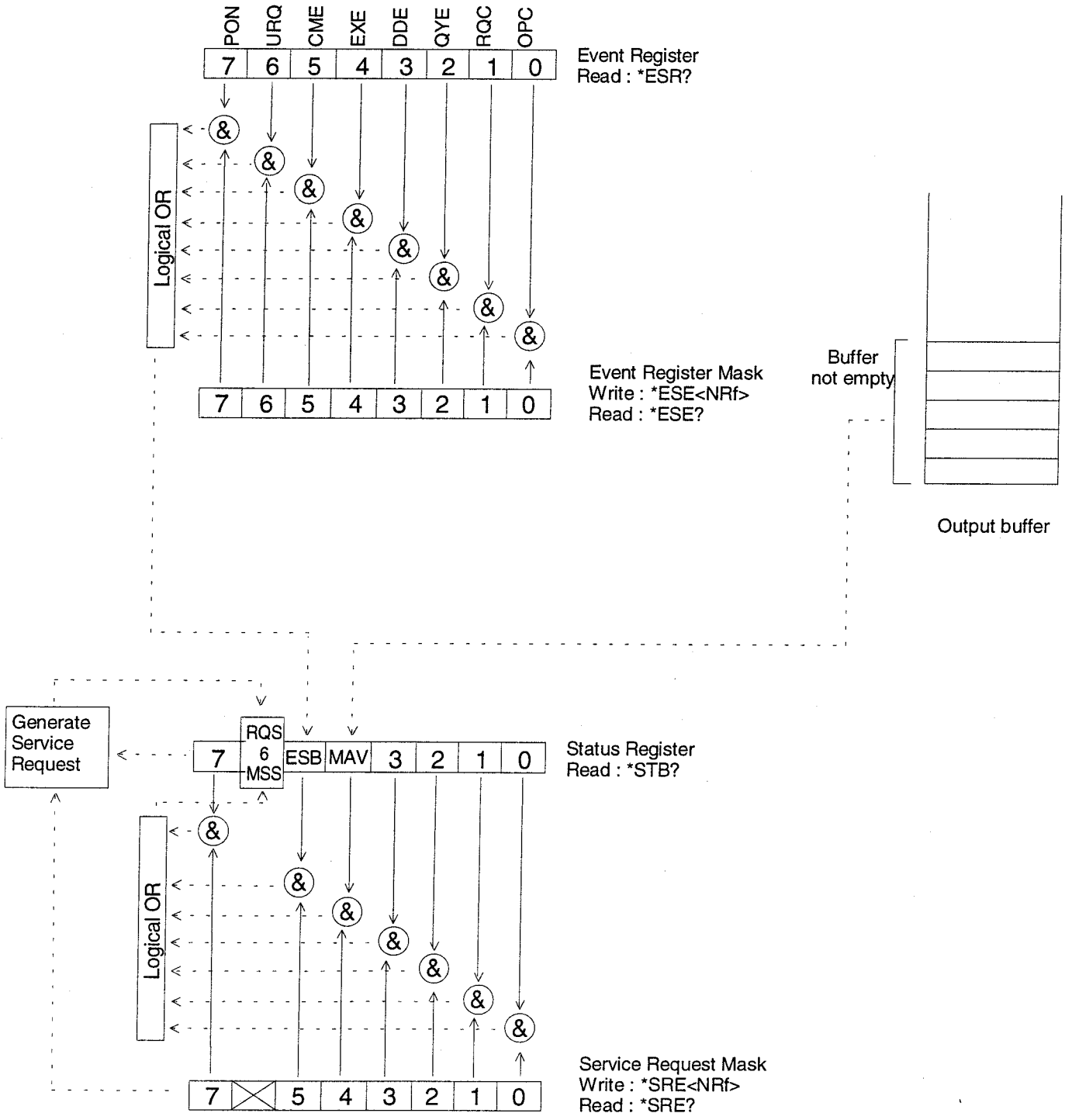
The only solution for the controller is to allow the maximum waiting time to time out and abort the current read before continuing its program.

Talker without listener(s) on the bus

In this case, the talker waits for a listener or for the controller to resume control (untalk).

8.3. Status and Event Management

8.3.1. IEEE 488.2 Structures and Concepts



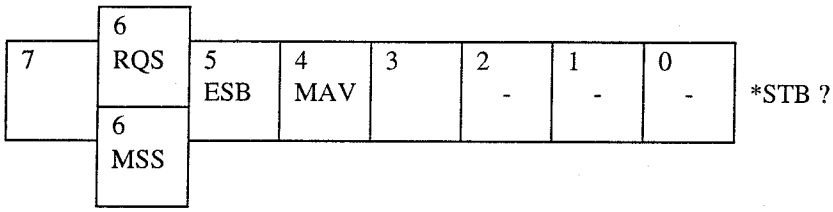
Status register

This is a read only register which is accessible in two different ways:

* by common command *STB?
  In this case, bit 6 (MSS) is returned and remains in its original state before the read (see section *STB (Status Byte))
* by serial recognition (IEEE 488.1)
  In this case, bit 6 (RQS) is returned and goes to zero after the read

The common command *CLS resets the complete register.

Detailed description



RQS Request Service (bit 6)
Indicates whether the instrument is requesting a service. This information is accessible by serial recognition only (IEEE 488). It is reset after a read and cannot go to 1 unless the event register is reset (by a read or a *CLS).

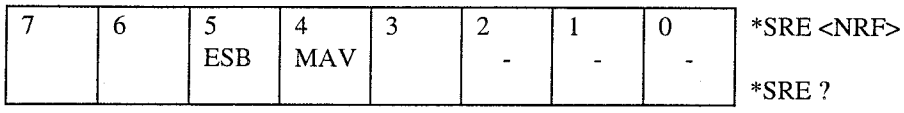
MSS Master Summary Status (bit 6)
Indicates whether the instrument has a reason for requesting a service. This information is accessible only by a status register (command *STB?) and remains unchanged after the read.

ESB Event Status Bit (bit 5)
Indicates whether at least one of the conditions in the event register is satisfied and not masked.

MAV Message Available (bit 4)
Indicates whether at least one response is in the output buffer.

Service Request Mask Register

This register can be read and written using *SRE? and *SRE<NRF> respectively (see the section on *SRE (Standard Request Enable)).



Event Register

This is a read only register accessible via the *ESR command (see the section on *ESR (Event Status Register)). A read resets it.

Detailed description

7	6	5	4	3	2	1	0	
0	0	CME	EXE	DDE	QYE	RQC	OPC	*ESR ?

PON Power On (bit 7)
Always zero.

URQ User request (bit 6)
Always zero.

CME Command Error (bit 5)
Indicates whether the syntax analyzer has found an error.

EXE Execution Error (bit 4)
Indicates whether a parameter is out of limits or whether a command cannot be executed because of the instrument's current configuration.

DDE Device Dependent Error (bit 3)
Indicates whether the instrument has not been able to complete an operation for instrument-specific reasons.

QYE Query Error (bit 2)
Indicates whether the query protocol has been violated.

RQC Request Control (bit 1)
Always zero.

OPC Operation Complete (bit 0)
Indicates whether the instrument has completed all the current operations.

Associated mask register

This register can be read and written using the commands *ESE? and *ESE<NRF> respectively (see the section on *ESE (Event Status Enable)).

7	6	5	4	3	2	1	0	
-	-	CME	EXE	DDE	QYE	TRG	OPC	*ESE <NRF> *ESE ?

Summary table

REGISTER	READ	WRITE
Status register	*STB?	NO
Status Register Mask	*SRE?	*SRE<NRF>
Event register	*ESR?	NO
Event Register Mask	*ESE?	*ESE<NRF>

8.3.2. Oscilloscope/Controller Synchronization

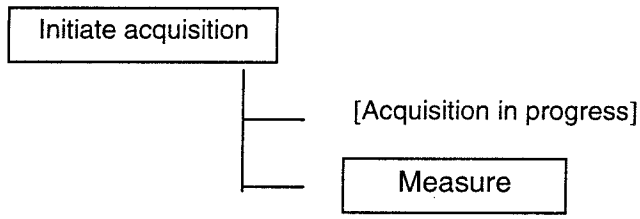
There are two possibilities:

* The controller and the instrument work at different speeds. There may be considerable time differences between transmission of the command by the controller and its execution by the instrument. Synchronization ensures that the controller and instrument work synchronously.
* Some commands are linked to the end of acquisition. These commands must therefore not be executed until acquisition is complete.

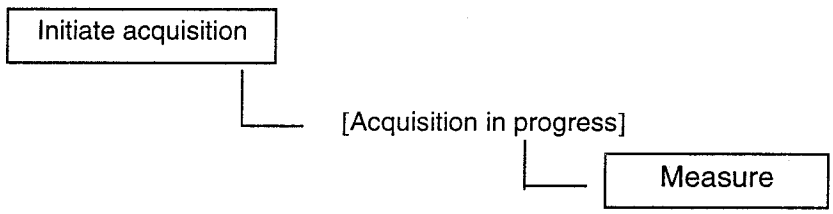
Example:

* Run a single shot acquisition
* Measure the amplitude of the acquired signal

Unless special precautions are taken, the amplitude measurement can be false if the current acquisition is incomplete.



By synchronizing on the end of acquisition, a true measurement is obtained:



Synchronization methods

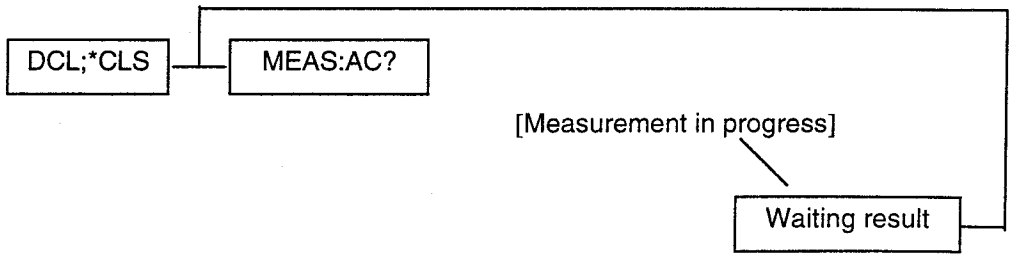
* questions/responses
* OPC? query
* *WAIT command

Synchronization by Questions/Responses

The simplest way to synchronize the controller with the instrument is for the controller to ask regular questions. The controller waits for the response before continuing with its program.

Example:

```
OUTPUT 707;"DCL;*CLS"          ;Reset
FOR i=0 TO 10                  ;
      OUTPUT 707;"MEAS:AC?"    ;measurement command
      ENTER 707;measure$       ;read results
NEXT i
```

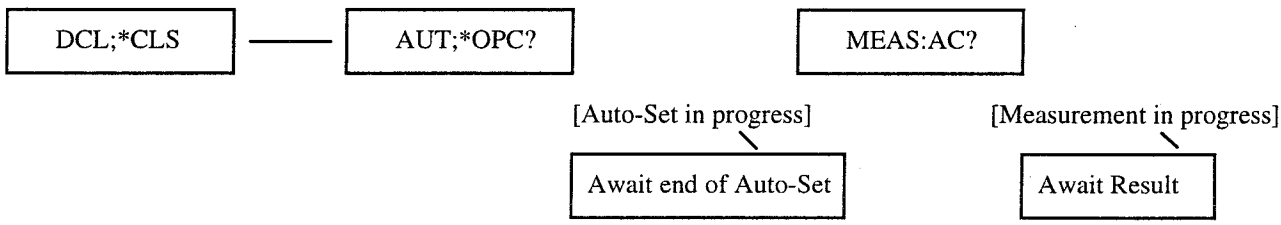


***OPC? query**

The *OPC? query synchronizes the controller and instrument by reading the response or status word (MAV bit, message available). When this bit is at 1, indicating that the response is ready, a service request can be initiated if the status and event registers are correctly initialized.

Example: Read output buffer

```
OUTPUT 707;"DCL;*CLS"        ;Reset
OUTPUT 707;"AUT;*OPC?"       ;Run Auto-Set and request indication
                              of completion
ENTER 707;opc                ;Await completion
OUTPUT 707;"MEAS:AC?"        ;Measurement command
ENTER 707;measure$           ;read results
```



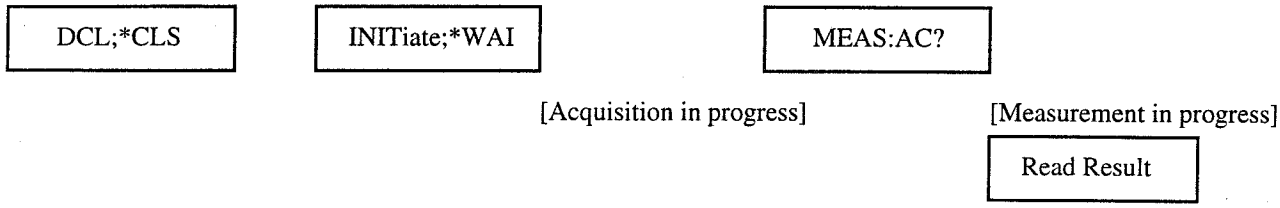
Note: ***Read time must not exceed the controller time-out.***

***WAI command**

All commands stored in the oscilloscope's input buffer are executed sequentially. Only the duration of acquisition is uncontrolled (depends on the input signal and the instrument's configuration). In some cases, the next command cannot be executed until an acquisition has finished:

Example: Measurement with synchronization on end of acquisition:

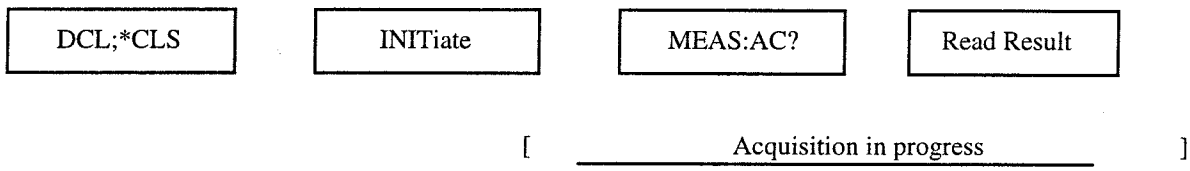
```
OUTPUT 707;"DCL;*CLS"        ;Reset
OUTPUT 707;"INITiate;*WAI"   ;Initiate acquisition &
                              ;Await end
OUTPUT 707;"MEAS:AC?"        ;Measurement command
ENTER 707;measure$           ;read results
```



☞ ***Note:*** ***The *WAI command makes the instrument wait until the end of acquisition before measuring the signal.***

✍ Example: Measurement without synchronization:

```
OUTPUT 707;"DCL;*CLS"
OUTPUT 707;"INITiate"
OUTPUT 707;"MEAS:AC?"
ENTER 707;measure
```



☞ ***Note:*** ***The measurement relates to the signal stored before the current acquisition.***

8.3.3. Error messages

* Command errors: (-199 to -100)

These indicate that a syntax error has been detected by the syntax analyzer and set bit 5 of the event register to 1, CME, Command Error.

-100	:	Command error
-101	:	Invalid Character
-102	:	Syntax error
-103	:	Invalid separator
-104	:	Data type error
-105	:	GET not allowed
-108	:	Parameter not allowed
-109	:	Missing parameter
-111	:	Header separator error
-112	:	Program mnemonic too long
-113	:	Undefined header
-114	:	Header suffix out of range
-121	:	Invalid character in number
-123	:	Numeric overflow
-124	:	Too many digits
-128	:	Numeric data not allowed
-130	:	Suffix error
-131	:	Invalid suffix
-138	:	Suffix not allowed
-140	:	Character data error
-141	:	Invalid character data
-144	:	Character data too long
-148	:	Character data not allowed
-150	:	String data error
-151	:	Invalid string data
-158	:	String data not allowed
-160	:	Block data error
-161	:	Invalid block data
-168	:	Block data not allowed
-170	:	Expression error
-171	:	Invalid expression
-178	:	Expression data not allowed

* Execution errors: (-299 to -200)

These indicate that an error has been detected on execution of a command and set bit 4 of the event register to 1, EXE (Execution Error).

-200	:	Execution error
-211	:	Trigger ignored
-221	:	Settings conflict
-222	:	Data out of range
-223	:	Too much data

* Device-dependent errors: (-399 to -300)

These indicate that a fault has been detected during execution of a task and set bit 3 of the event register to 1, DDE (Device Dependent Error).

-300	:	Device specific error
-310	:	System error
-350	:	Too many errors

* Query errors: (-499 to -400)

These indicate a fault in the interchange protocol and set bit 2 of the event register to 1, QYE (QuerY Error).

-400	:	Query error
-410	:	Query INTERRUPTED
-420	:	Query UNTERMINATED
-430	:	Query DEADLOCKED
-440	:	Query UNTERMINATED after indefinite response

APPENDIX 1
ASCII table

B7 B6 B5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1

B4	B3	B2	B1	Control		Capital number		Capital letter		Small letter	
0	0	0	0	0 **NUL** 0	10 **DLE** 16	20 **SP** 32	30 **0** 48	40 **@** 64	50 **P** 80	60 ` 96	70 **p** 112
0	0	0	1	1 **SOH** 1	11 **DC1** 17	21 **!** 33	31 **1** 49	41 **A** 65	51 **Q** 81	61 **a** 97	71 **q** 113
0	0	1	0	2 **STX** 2	12 **DC2** 18	22 **"** 34	32 **2** 50	42 **B** 66	52 **R** 82	62 **b** 98	72 **r** 114
0	0	1	1	3 **ETX** 3	13 **DC3** 19	23 **#** 35	33 **3** 51	43 **C** 67	53 **S** 83	63 **c** 99	73 **s** 115
0	1	0	0	4 **EOT** 4	14 **DC4** 20	24 **$** 36	34 **4** 52	44 **D** 68	54 **T** 84	64 **d** 100	74 **t** 116
0	1	0	1	5 **ENQ** 5	15 **NAK** 21	25 **%** 37	35 **5** 53	45 **E** 69	55 **U** 85	65 **e** 101	75 **u** 117
0	1	1	0	6 **ACK** 6	16 **SYN** 22	26 **&** 38	36 **6** 54	46 **F** 70	56 **V** 86	66 **f** 102	76 **v** 118
0	1	1	1	7 **BEL** 7	17 **ETB** 23	27 **'** 39	37 **7** 55	47 **G** 71	57 **W** 87	67 **g** 103	77 **w** 119
1	0	0	0	8 **BS** 8	18 **CAN** 24	28 **(** 40	38 **8** 56	48 **H** 72	58 **X** 88	68 **h** 104	78 **x** 120
1	0	0	1	9 **HT** 9	19 **EM** 25	29 **)** 41	39 **9** 57	49 **I** 73	59 **Y** 89	69 **i** 105	79 **y** 121
1	0	1	0	A **LF** 10	1A **SUB** 26	2A ***** 42	3A **:** 58	4A **J** 74	5A **Z** 90	6A **j** 106	7A **z** 122
1	0	1	1	B **VT** 11	1B **ESC** 27	2B **+** 43	3B **;** 59	4B **K** 75	5B **[** 91	6B **k** 107	7B **{** 123
1	1	0	0	C **FF** 12	1C **FS** 28	2C **,** 44	3C **<** 60	4C **L** 76	5C **** 92	6C **l** 108	7C **|** 124
1	1	0	1	D **CR** 13	1D **CS** 29	2D **-** 45	3D **=** 61	4D **M** 77	5D **]** 93	6D **m** 109	7D **}** 125
1	1	1	0	E **S0** 14	1E **RS** 30	2E **.** 46	3E **>** 62	4E **N** 78	5E **^** 94	6E **n** 110	7E **~** 126
1	1	1	1	F **S1** 15	1F **US** 31	2F **/** 47	3F **?** 63	4F **O** 79	5F **_** 95	6F **o** 111	7F **DEL** 127

Hexadecimal i
XY
Decimal j