

Programmer's
Quick Reference

HP 54600-Series Oscilloscopes

Error Messages

Error Number	Description	Error Number	Description
-100	Command error (unknown command)	-160	Block data error
-101	Invalid character	-161	Invalid block data
-102	Syntax error	-168	Block data not allowed
-103	Invalid separator	-170	Expression error
-104	Data type error	-171	Invalid expression
-105	GET not allowed	-178	Expression data not allowed
-108	Parameter not allowed	-200	Execution error
-109	Missing parameter	-211	Trigger ignored
-112	Program mnemonic too long	-221	Settings conflict
-113	Undefined header	-222	Data out of range
-121	Invalid character in number	-223	Too much data
-123	Numeric overflow	-310	System error
-124	Too many digits	-350	Too many errors
-128	Numeric data not allowed	-400	Query error
-130	Suffix error	-410	Query INTERRUPTED
-131	Invalid suffix	-420	Query UNTERMINATED
-138	Suffix not allowed	-430	Query DEADLOCKED
-140	Character data error	-440	Query UNTERMINATED after indefinite response
-141	Invalid character data		
-144	Character data too long		
-148	Character data not allowed		
-150	String data error		
-151	Invalid string data		
-152	String data not allowed		

Programmer's Quick Reference

Introduction

The following section lists the commands and queries with their corresponding arguments and returned formats. The arguments for each command list the minimum argument required. The part of the command or query listed in uppercase letters refers to the short form of that command or query. The long form is the combination of uppercase and lowercase letters.

Conventions The following conventions are used in this section:

< > Angular brackets enclose words or characters that symbolize a program code parameter or an HP-IB command.

::= "is defined as." For example, <A> ::= indicates that <A> can be replaced by in any statement containing <A>.

| "or." Indicates a choice of one element from a list. For example, <A> | indicates <A> or but not both.

... An ellipsis (trailing dots) indicate that the preceding element may be repeated one or more times.

[] Square brackets indicate that the enclosed items are optional.

{ } When several items are enclosed by braces, one, and only one of these elements may be selected.

Suffix Multipliers The suffix multipliers available for arguments are:

EX ::= 1E18

PE ::= 1E15

T ::= 1E12

G ::= 1E9

MA ::= 1E6

K ::= 1E3

M ::= 1E-3

U ::= 1E-6

N ::= 1E-9

P ::= 1E-12

F ::= 1E-15

A ::= 1E-18

For more information on specific commands or queries, refer to the Programmer's Reference.

***CLS** (Clear Status) command

Command Syntax: *CLS

***ESE** (Event Status Enable) command/query

Command Syntax: *ESE {0 to 255}

Query Syntax: *ESE?

Returned Format: {integer, 0 to 255}<NL>

***ESR** (Event Status Register) query

Query Syntax: *ESR?

Returned Format: {integer, 0 to 255}<NL>

***IDN** (Identification Number) query

Query Syntax: *IDN?

Returned Format: HEWLETT-PACKARD,54600A,0,X.X<NL>

***LRN** (Learn) query

Query Syntax: *LRN?

Returned Format: :SYSTem:SETup #800000121<learn string><NL>

***OPC** (Operation Complete) command/query

Command Syntax: *OPC

Query Syntax: *OPC?

Returned Format: 1<NL>

***OPT** (Option) query

Query Syntax: *OPT?

Returned Format: 0<NL>

***RCL** (Recall) command

Command Syntax: *RCL {1 to 16}

*RST	(Reset)	command
Command Syntax: *RST		
*SAV	(Save)	command
Command Syntax: *SAV {1 to 16}		
*SRE	(Service Request Enable)	command/query
Command Syntax: *SRE {0 to 255}		
Query Syntax: *SRE?		
Returned Format: <mask><NL>		
Where: <mask> ::= sum of all bits set - integer, 0 to 255		
*STB	(Status Byte)	query
Query Syntax: *STB?		
Returned Format: {integer, 0 to 255}<NL>		
*TRG	(Trigger)	command
Command Syntax: *TRG		
*TST	(Test)	query
Query Syntax: *TST?		
Returned Format: {0 or non-zero value}<NL>		
Where: 0 ::= test passed non-zero ::= test failed		
*WAI	(Wait)	command
Command Syntax: *WAI		
:ACQuire:COMPLete		command/query
Command Syntax: :ACQuire:COMPLete {0 to 100}		
Query Syntax: :ACQuire:COMPLete?		
Returned Format: {integer, 0 to 100}<NL>		

:ACquire:COUNT **command/query**

Command Syntax: :ACquire:COUNT {8 | 64 | 256}

Query Syntax: :ACquire:COUNT?

Returned Format: { 8 | 64 | 256}<NL>

:ACquire:POINTS **query**

Query Syntax: :ACquire:POINTS?

Returned Format: {integer, 1 to 4000}<NL>

:ACquire:Setup **query**

Query Syntax: :ACquire:SETup?

Returned Format: <string><NL>

:ACquire:TYPE **command/query**

Command Syntax: :ACquire:TYPE {NORMa1 | AVERAge | PEAK}

Query Syntax: :ACquire:TYPE?

Returned Format: {NORM | AVER | PEAK}<NL>

:ASToRe **command**

Command Syntax: :ASToRe

:AUToscale **command**

Command Syntax: :AUToscale

:BLANK **command**

Command Syntax: :BLANK {CHANne1{1 | 2 | 3 | 4}|PMEMory{1 | 2}}

:CHANnel{1 | 2}:BWLimit **command/query**

Command Syntax: :CHANne1{1 | 2}:BWLimit {ON | OFF}

Query Syntax: :CHANne1{1 | 2}:BWLimit?

Returned Format: {ON | OFF}<NL>

:CHANnel{1 | 2 | 3 | 4}:COUPling **command/query**

Command Syntax: :CHANnel{1 | 2}:COUPling {AC | DC | GND} | {3 | 4}:COUPling {DC | GND}
Query Syntax: :CHANnel{1 | 2 | 3 | 4}:COUPling?
Returned Format: {AC | DC | GND}<NL> for Channels 1 and 2
{DC | GND}<NL> for Channels 3 and 4

:CHANnel{1 | 2}:INVert **command/query**

Command Syntax: :CHANnel{1 | 2}:INVert {ON | OFF}
Query Syntax: :CHANnel{1 | 2}:INVert?
Returned Format: {ON | OFF}

:CHANnel:MATH **command/query**

Command Syntax: :CHANnel:MATH {OFF | PLUS | SUBtract}
Query Syntax: :CHANnel:MATH?
Returned Format: {OFF | PLUS | SUB}

:CHANnel{1 | 2 | 3 | 4}:OFFSet **command/query**

Command Syntax: :CHANnel{1 | 2 | 3 | 4}:OFFSet <offset value>
Query Syntax: :CHANnel{1 | 2 | 3 | 4}:OFFSet?
Returned Format: <exponential, offset value><NL>

:CHANnel{1 | 2 | 3 | 4}:PROBe **command/query**

Command Syntax: :CHANnel{1 | 2 | 3 | 4}:PROBe {X1 | X10 | X100}
Query Syntax: :CHANnel{1 | 2 | 3 | 4}:PROBe?
Returned Format: {X1 | X10 | X100}<NL>

:CHANnel{1 | 2 | 3 | 4}:RANGe **command/query**

Command Syntax: :CHANnel{1 | 2}:RANGe<full-scale range> | {3 | 4}:RANGe {HIGH | LOW}
Query Syntax: :CHANnel{1 | 2 | 3 | 4}:RANGe?
Returned Format: <exponential full-scale range><NL> for Channels 1 and 2
{HIGH | LOW}<NL> for Channels 3 and 4

:CHANnel{1 | 2 | 3 | 4}:SETup **query**

Query Syntax: :CHANnel{1 | 2 | 3 | 4}:SETup?
Returned Format: <string><NL>

:CHANnel{1 | 2}:VERNier **command/query**

Command Syntax: :CHANnel{1 | 2}:VERNier {ON | OFF}
Query Syntax: :CHANnel:VERNier?
Returned Format: {ON | OFF}<NL>

:DIGitize **command**

Command Syntax: :DIGitize CHANnel{1 | 2 | 3 | 4}, [,CHANnel{1 | 2 | 3 | 4}]

:DISPlay:COLumn **command/query**

Command Syntax: :DISPlay:COLumn {0 to 63}
Query Syntax: :DISPlay:COLumn?
Returned Format: {integer, 0 to 63}<NL>

:DISPlay:DATA **command/query**

Command Syntax: :DISPlay:DATA #800016257<data>
Query Syntax: :DISPlay:DATA?
Returned Format: #800016257<data><NL>

:DISPlay:GRID **command/query**

Command Syntax: :DISPlay:GRID {ON | OFF}
Query Syntax: :DISPlay:GRID?
Returned Format: {ON | OFF}<NL>

:DISPlay:INVerse **command/query**

Command Syntax: :DISPlay:INVerse {ON | OFF }
Query Syntax: :DISPlay:INVerse?
Returned Format: {ON | OFF}<NL>

:DISPlay:LINE **command**

Command Syntax: :DISPlay:LINE <quoted string>

:DISPlay:PIXel **command/query**

Command Syntax: :DISPlay:PIXel <x>, <y>, <intensity>

Query Syntax: :DISPlay:PIXel? <x>, <y>

Returned Format: <integer, intensity><NL>

:DISPlay:ROW **command/query**

Command Syntax: :DISPlay:ROW {1 to 20}

Query Syntax: :DISPlay:ROW?

Returned Format: {integer, 1 to 20}<NL>

:DISPlay:Setup **query**

Query Syntax: :DISPlay:SETup?

Returned Format: <string><NL>

:DISPlay:SOURce **command/query**

Command Syntax: :DISPlay:SOURce PMEMory{1 | 2}

Query Syntax: :DISPlay:SOURce?

Returned Format: PMEM{1 | 2}<NL>

:DISPlay:TEXT **command**

Command Syntax: :DISPlay:TEXT BLANK

:DITher **command/query**

Command Syntax: :DITher {ON | OFF}

Query Syntax: :DITher?

Returned Format: {ON | OFF}<NL>

:ERASe **command**

Command Syntax: :ERASe [PMEMory{1 | 2}]

:MEASure:ALL **query**

Query Syntax: :MEASure:ALL?
Returned Format: <FREQuency result>,<PERiod result>, <PWIDth result>,<NWIDTH result>,<RISetime result>,<FALLtime result>,<VPP result>, <DUTycycle result>,<VRMS result>,<VMAX result>,<VMIN result>,<VTOP result>,<VBASe result>,<VAverage result><NL>

:MEASure:DUTycycle **command/query**

Command Syntax: :MEASure:DUTycycle
Query Syntax: :MEASure:DUTycycle?
Returned Format: <exponential, dutycycle value><NL>

:MEASure:FALLtime **command/query**

Command Syntax: :MEASure:FALLtime
Query Syntax: :MEASure:FALLtime?
Returned Format: <exponential, falltime value><NL>

:MEASure:FREQuency **command/query**

Command Syntax: :MEASure:FREQuency
Query Syntax: :MEASure:FREQuency?
Returned Format: <exponential, frequency value><NL>

:MEASure:NWIDTH **command/query**

Command Syntax: :MEASure:NWIDTH
Query Syntax: :MEASure:NWIDTH?
Returned Format: <exponential, negative_width value><NL>

:MEASure:PERiod **command/query**

Command Syntax: :MEASure:PERiod
Query Syntax: :MEASure:PERiod?
Returned Format: <exponential, period value><NL>

:MEASure:PWIDth **command/query**

Command Syntax: :MEASure:PWIDth
Query Syntax: :MEASure:PWIDth?
Returned Format: <exponential, positive_width value><NL>

:MEASure:RISetime **command/query**

Command Syntax: :MEASure:RISet ime
Query Syntax: :MEASure:RISet ime?
Returned Format: <exponential, risetime value><NL>

:MEASure:SCRatch **(Clear Results)** **command**

Command Syntax: :MEASure:SCRatch

:MEASure:SHOW **command/query**

Command Syntax: :MEASure:SHOW {ON | OFF}
Query Syntax: :MEASure:SHOW?
Returned Format: {ON |OFF}<NL>

:MEASure:SOURce **command/query**

Command Syntax: :MEASure:SOURce CHAnne1 {1 | 2 | 3 | 4}
Query Syntax: :MEASure:SOURce?
Returned Format: CHAN{1 | 2 | 3 | 4}<NL>

:MEASure:TDELta **query**

Query Syntax: :MEASure:TDELta?
Returned Format: <exponential, delta time markers><NL>

:MEASure:TSTArt **command/query**

Command Syntax: :MEASure:TSTArt <start marker time>
Query Syntax: :MEASure:TSTArt?
Returned Format: <exponential, start marker time><NL>

:MEASure:TSTOp **command/query**

Command Syntax: :MEASure:TSTOp <stop marker time>
Query Syntax: :MEASure:TSTOp?
Returned Format: <exponential, stop marker time><NL>

:MEASure:TVOLT **query**

Query Syntax: :MEASure:TVOLT? <voltage>,<slope><occurrence>
Returned Format: <exponential, time of voltage crossing><NL>

:MEASure:VAverage **command/query**

Command Syntax: :MEASure:VAverage
Query Syntax: :MEASure:VAverage?
Returned Format: <exponential, average voltage><NL>

:MEASure:VBASe **command/query**

Command Syntax: :MEASure:VBASe
Query Syntax: :MEASure:VBASe?
Returned Format: <exponential, base voltage><NL>

:MEASure:VDELta **query**

Query Syntax: :MEASure:VDELta?
Returned Format: <exponential, delta voltage markers><NL>

:MEASure:VMAX **command/query**

Command Syntax: :MEASure:VMAX
Query Syntax: :MEASure:VMAX?
Returned Format: <exponential, maximum voltage><NL>

:MEASure:VMIN **command/query**

Command Syntax: :MEASure:VMIN
Query Syntax: :MEASure:VMIN?
Returned Format: <exponential, minimum voltage><NL>

:MEASure:VPP **command/query**

Command Syntax: :MEASure:VPP
Query Syntax: :MEASure:VPP?
Returned Format: <exponential, peak-to-peak voltage><NL>

:MEASure:VRMS (DC RMS) **command/query**

Command Syntax: :MEASure:VRMS
Query Syntax: :MEASure:VRMS?
Returned Format: <exponential, dc_rms voltage><NL>

:MEASure:VSTArt **command/query**

Command Syntax: :MEASure:VSTArt <marker1 voltage>
Query Syntax: :MEASure:VSTArt?
Returned Format: <exponential, marker1 voltage><NL>

:MEASure:VSTOp **command/query**

Command Syntax: :MEASure:VSTOp <marker2 voltage>
Query Syntax: :MEASure:VSTOp?
Returned Format: <exponential, marker2 voltage><NL>

:MEASure:VTIME **query**

Query Syntax: :MEASure:VTIME? <time from trigger>
Returned Format: <exponential, voltage at specified time><NL>

:MEASure:VTOp **command/query**

Command Syntax: :MEASure:VTOp
Query Syntax: :MEASure:VTOp?
Returned Format: <exponential, top_voltage><NL>

:MENU**command/query**

Command Syntax: :MENU {0 to 16}
Query Syntax: :MENU?
Returned Format: {integer, 1 to 16}<NL>
Where: <integer>::= 0 = Clear Menu
1 = Channel 1
2 = Channel 2
3 = Channel 3
4 = Channel 4
5 = Math
6 = Trigger Source
7 = Trigger Mode
8 = Trigger Slope
9 = Main/Delayed (Horizontal)
10 = Time Measurements
11 = Voltage Measurements
12 = Cursors
13 = Trace
14 = Setup
15 = Display
16 = Utility

:MERGe**command**

Command Syntax: :MERGe PMEMory{1 | 2}

:PRINt**query**

Query Syntax: :PRINt? [HIRes]

:RUN**command**

Command Syntax: :RUN

:STATus**query**

Query Syntax: :STATus? {CHANnel{1 | 2 | 3 | 4 } | PMEMory{1 | 2}}
Returned Format: {ON | OFF}<NL>

:STOP**command**

Command Syntax: :STOP

:SYSTem:DSP**command**

Command Syntax: :SYSTem:DSP <quoted ASCII string>

:SYSTem:ERRor**query**Query Syntax: :SYSTem:ERRor?
Returned Format: <integer, error number><NL>**Where:**

<error number>::=

- | | |
|---------------------------------------|-----------------------------------|
| +0, No error | |
| -100, Command error (unknown command) | -160, Block data error |
| -101, Invalid character | -161, Invalid block data |
| -102, Syntax error | -168, Block data not allowed |
| -103, Invalid separator | |
| -104, Data type error | -170, Expression error |
| -105, GET not allowed | -171, Invalid expression |
| -108, Parameter not allowed | -178, Expression data not allowed |
| -109, Missing parameter | |
| | |
| -112, Program mnemonic too long | -200, Execution error |
| -113, Undefined header | |
| | |
| -121, Invalid character in number | -211, Trigger ignored |
| -123, Numeric overflow | |
| -124, Too many digits | -221, Settings conflict |
| -128, Numeric data not allowed | -222, Data out of range |
| | -223, Too much data |
| | |
| -130, Suffix error | |
| -131, Invalid suffix | |
| -138, Suffix not allowed | -310, System error |
| | |
| -140, Character data error | -350, Too many errors |
| -141, Invalid character data | |
| -144, Character data too long | -400, Query error |
| -148, Character data not allowed | -410, Query INTERRUPTED |
| | -420, Query UNTERMINATED |
| -150, String data error | -430, Query DEADLOCKED |
| -151, Invalid string data | -440, Query UNTERMINATED |
| | after indefinite response |
| -158, String data not allowed | |

:SYSTEM:KEY**command/query**

Command Syntax: :SYSTEM:KEY {-1 to 50}
Query Syntax: :SYSTEM:KEY?
Returned Format: {integer, -1 to 50}<NL>

Where:

<integer>::=	-1 for NO KEY	21 for STOP	43 for DELAY_CW
	0 for AUTOSCALE	22 for ERASE	44 for DELAY_CCW
	1 for CH1	23 for SOFTKEY_1	45 for TRG_LEVEL_CW
	2 for CH2	24 for SOFTKEY_2	46 for TRG_LEVEL_CCW
	3 for CH3	25 for SOFTKEY_3	47 for TRG_HOLD_CW
	4 for CH4	26 for SOFTKEY_4	48 for TRG_HOLD_CCW
	5 for +/-	27 for SOFTKEY_5	49 for CURSOR_KNOB_CW
	6 for TRG_SRC	28 for SOFTKEY_6	50 for CURSOR_KNOB_CCW
	7 for TRG_MODE	29 for CH1_VOLT_CW	
	8 for TRG_SLOPE	30 for CH1_VOLT_CCW	
	9 for MAIN/DELAYED	31 for CH1_POS_CW	
	10 for TIME	32 for CH1_POS_CCW	
	11 for VOLTAGE	33 for CH2_VOLT_CW	
	12 for CURSORS	34 for CH2_VOLT_CCW	
	13 for SAVE_TRACE	35 for CH2_POS_CW	
	14 for SAVE_SETUP	36 for CH2_POS_CCW	
	15 for DISPLAY	37 for CH3_POS_CW	
	16 for PRINT/UTILITY	38 for CH3_POS_CCW	
	17 NA	39 for CH4_POS_CW	
	18 NA	40 for CH4_POS_CCW	
	19 for RUN	41 for S/DIV_CW	
	20 for AUTOSTORE	42 for S/DIV_CCW	

CW denotes clockwise rotation of the knob.

CCW denotes counter-clockwise rotation of the knob.

:SYSTem:LOCK **command/query**

Command Syntax: :SYSTem:LOCK {ON |OFF}
Query Syntax: :SYSTem:LOCK?
Returned Format: {ON |OFF}

:SYSTem:SETup **command/query**

Command Syntax: :SYSTem:SETup #800000121<setup data string>
Query Syntax: :SYSTem:SETup?
Returned Format: #800000121<setup data string><NL>

:TER **(Trigger Event Register)** **query**

Query Syntax: :TER?
Returned Format: {1 | 0}<NL>

:TIMEbase:DELay **command/query**

Command Syntax: :TIMEbase:DELay <delay time>
Query Syntax: :TIMEbase:DELay?
Returned Format: <exponential, delay time><NL>

:TIMEbase:MODE **command/query**

Command Syntax: :TIMEbase:MODE {NORMal | DELayed | XY | ROLL}
Query Syntax: :TIMEbase:MODE?
Returned Format: {NORM| DEL | XY | ROLL}<NL>

:TIMEbase:RANGE **command/query**

Command Syntax: :TIMEbase:RANGE {20 ns to 50 s}
Query Syntax: :TIMEbase:RANGE?
Returned Format: {exponential,20 ns to 50 s}<NL>

:TIMEbase:REFerence **command/query**

Command Syntax: :TIMEbase:REFerence {LEFT | CENTer} ({CENTer | RIGHT} in roll mode)
Query Syntax: :TIMEbase:REFerence?
Returned Format: {LEFT | CENT}<NL>

:TIMebase:SETup **query**

Query Syntax: :TIMebase:SETup?
Returned Format: <string><NL>

:TIMebase:VERNier **command/query**

Command Syntax: :TIMebase:VERNier {ON | OFF}
Query Syntax: :TIMebase:VERNier?
Returned Format: {ON | OFF}<NL>

:TRIGger:COUPling **command/query**

Command Syntax: :TRIGger:COUPling {AC |DC}
Query Syntax: :TRIGger:COUPling?
Returned Format: {AC |DC}<NL>

:TRIGger:HOLDoff **command/query**

Command Syntax: :TRIGger:HOLDoff <time>
Query Syntax: :TRIGger:HOLDoff?
Returned Format: <time><NL>
Where: <time> ::= exponential, 40 ns to 320 ms

:TRIGger:LEVel **command/query**

Command Syntax: :TRIGger:LEVel <level>
Query Syntax: :TRIGger:LEVel?
Returned Format: <exponential, trigger level in volts><NL>

:TRIGger:MODE **command/query**

Command Syntax: :TRIGger:MODE {AUTLevel | AUTO | NORMa1 | SINGle | TV}
Query Syntax: :TRIGger:MODE?
Returned Format: {AUTL | AUTO | NORM | SING | TV}<NL>

:TRIGger:NREJect **command/query**

Command Syntax: :TRIGger:NREJect:{ON | OFF}
Query Syntax: :TRIGger:NREJect?
Returned Format: {ON | OFF}<NL>

:TRIGger:POLarity **command/query**

Command Syntax: :TRIGger:POLarity {POSitive | NEGative}

Query Syntax: :TRIGger:POLarity?

Returned Format: {POS | NEG}<NL>

:TRIGger:REJect **command/query**

Command Syntax: :TRIGger:REJect {Off | LF | HF}

Query Syntax: :TRIGger:REJect?

Returned Format: {Off | LF | HF}<NL>

:TRIGger:Setup **query**

Query Syntax: :TRIGger:SETup?

Returned Format: <string><NL>

:TRIGger:SLOPe **command/query**

Command Syntax: :TRIGger:SLOPe {POSitive | NEGative}

Query Syntax: :TRIGger:SLOPe?

Returned Format: {POS | NEG}<NL>

:TRIGger:SOURce **command/query**

Command Syntax: :TRIGger:SOURce {CHANnel{1 | 2 | 3 | 4} | EXTerNa1 | LINE}

Query Syntax: :TRIGger:SOURce?

Returned Format: {CHAN{1 | 2 | 3 | 4} | EXT | LINE}<NL>

:TRIGger: TVHFreject **command/query**

Command Syntax: :TRIGger:TVHFreject {ON | OFF}

Query Syntax: :TRIGger:TVHFreject?

Returned Format: {ON | OFF}<NL>

:TRIGger: TVMode **command/query**

Command Syntax: :TRIGger:TVMode {FIELD1 | FIELD2 | LINE}

Query Syntax: :TRIGger:TVMode?

Returned Format: {FILED1 | FIELD2 | LINE}<NL>

:VIEW**command**

Command Syntax: :VIEW {CHANnel{1 | 2| 3| 4}| PMEMory{1 | 2} }

:WAVEform:BYTeorder**command/query**

Command Syntax: :WAVEform:BYTeorder {LSBFfirst | MSBFfirst}

Query Syntax: :WAVEform:BYTeorder?

Returned Format: {LSBF | MSBF}<NL>

:WAVEform:DATA**command/query**

Command Syntax: :WAVEform:DATA <binary block data in # format>

Query Syntax: :WAVEform:DATA?

Returned Format: <binary block data in IEEE 488.2 format><NL>

:WAVEform:FORMat**command/query**

Command Syntax: :WAVEform:FORMat {ASCii | WORD | BYTE}

Query Syntax: :WAVEform:FORMat?

Returned Format: {ASC | WORD | BYTE}<NL>

:WAVEform:POINts**query**

Command Syntax: :WAVEform:POINts {100 | 200 | 250 | 400 | 500 | 800 | 1000 | 2000 | 4000}

Query Syntax: :WAVEform:POINts?

Returned Format: {100 | 200 | 250 | 400 | 500 | 800 | 1000 | 2000 | 4000}<NL>

:WAVeform:PREamble**query**

Query Syntax: :WAVeform:PREamble?
Returned Format: <preamble block><NL>
Where: <preamble block> ::= <format NR1>, <type NR1>, <points NR1>, <count NR1>, <xincrement NR3>, <xorigin NR3>, <xreference NR1>, <yincrement NR3>, <yorigin NR3>, <yreference NR1>

<format> ::= 0 for ASCII format
1 for BYTE format
2 for WORD format

<type> ::= 0 for AVERAGE type
1 for NORMAL type
2 for PEAK DETECT type

:WAVeform:SOURce**command/query**

Command Syntax: :WAVeform:SOURce CHANnel{1 | 2 | 3 | 4}
Query Syntax: :WAVeform:SOURce?
Returned Format: CHAN{1 | 2 | 3 | 4}<NL>

:WAVeform:TYPE**query**

Query Syntax: :WAVeform:TYPE?
Returned Format: {NORM | PEAK | AVER}<NL>

:WAVeform:XINCrement**query**

Query Syntax: :WAVeform:XINCrement?
Returned Format: <exponential, x-increment value><NL>

:WAVeform:XORigin**query**

Query Syntax: :WAVeform:XORigin?
Returned Format: <exponential, x-origin value><NL>

:WAVeform:XREFerence**query**

Query Syntax: :WAVeform:XREFerence?
Returned Format: <integer, x-reference value><NL>

:WAVeform:YINCrement**query**

Query Syntax: :WAVeform:YINCrement?
Returned Format: <exponential, y-increment value><NL>

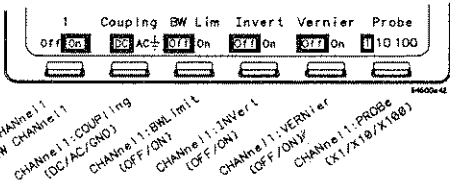
:WAVeform:YORigin**query**

Query Syntax: :WAVeform:YORigin?
Returned Format: <exponential, y-origin value><NL>

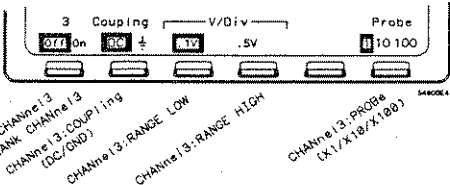
:WAVeform:YREFerence**query**

Query Syntax: :WAVeform:YREFerence?
Returned Format: <integer, y-reference value><NL>

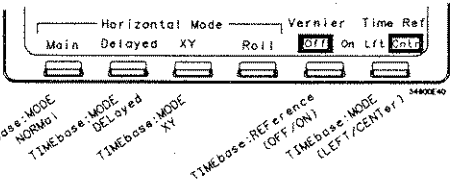
1
OR
2



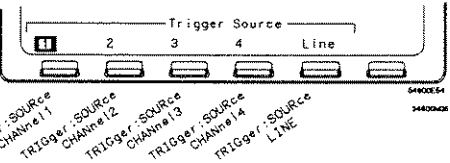
3
OR
4



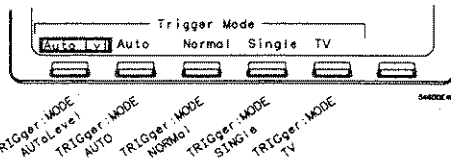
Main
Delayed



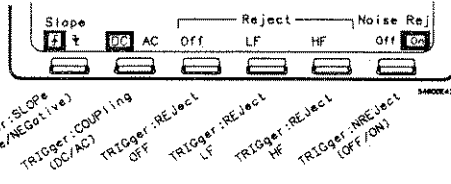
Source



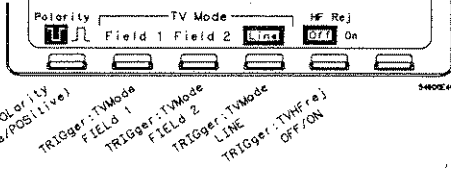
Mode



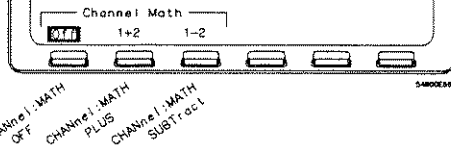
Slope
Coupling



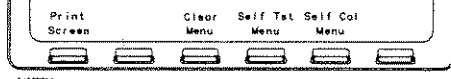
TV MODE
Slope
Coupling

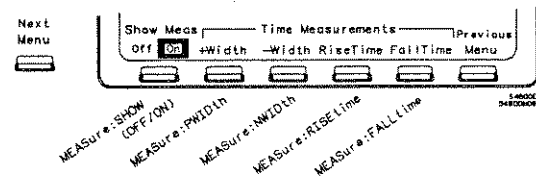
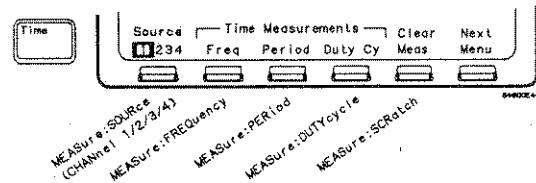
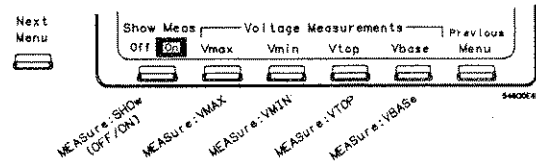
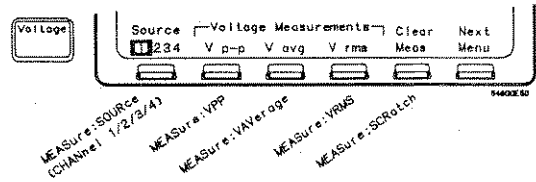
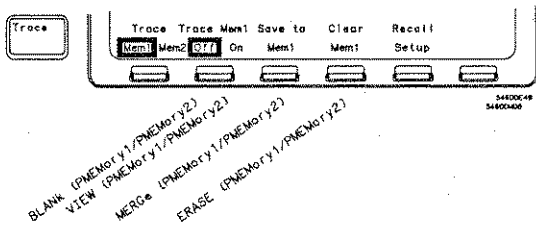
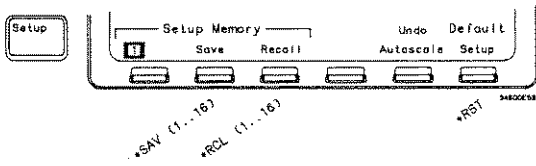
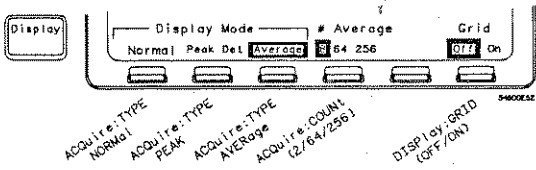
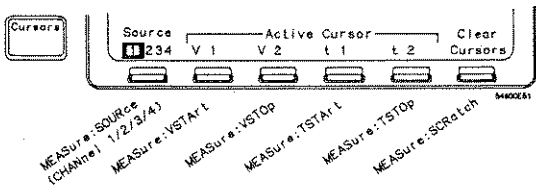


+

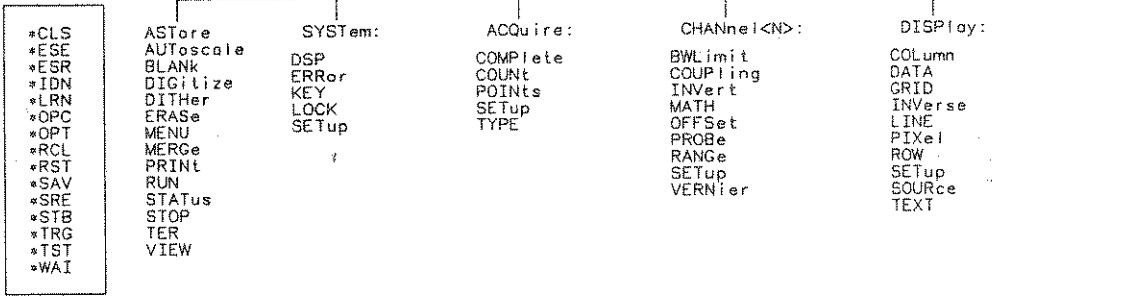


Print
Utility





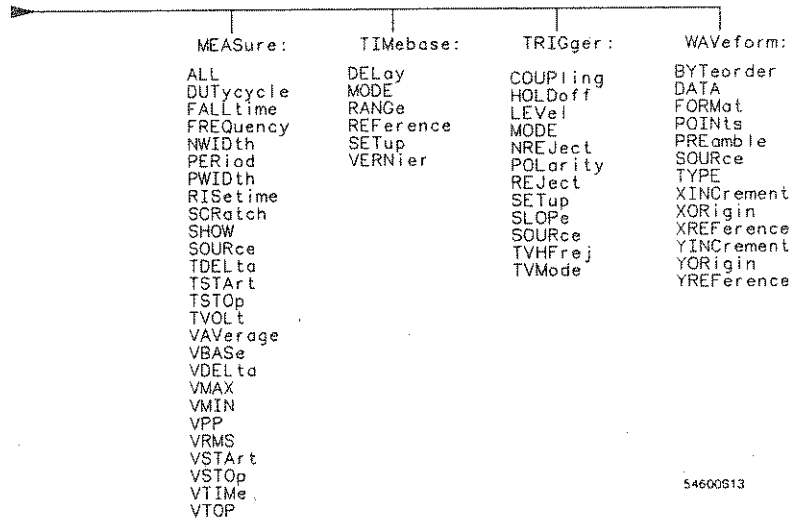
:(root)



Common
Commands
(IEEE 488.2)

The HP54600A has two identical channel subsystems. The HP54601A channels 1 and 2 are identical and fully attenuated. Channels 3 and 4 are identical and can be set for .1V or .5V/div with dc or ground coupling.

54600S12



54600S13