HP 54600-Series Oscilloscopes
## Error Messages

<table>
<thead>
<tr>
<th>Error Number</th>
<th>Description</th>
<th>Error Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-100</td>
<td>Command error</td>
<td>-160</td>
<td>Block data error</td>
</tr>
<tr>
<td></td>
<td>(unknown command)</td>
<td>-161</td>
<td>Invalid block data</td>
</tr>
<tr>
<td>-101</td>
<td>Invalid character</td>
<td>-168</td>
<td>Block data not allowed</td>
</tr>
<tr>
<td>-102</td>
<td>Syntax error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-103</td>
<td>Invalid separator</td>
<td>-170</td>
<td>Expression error</td>
</tr>
<tr>
<td>-104</td>
<td>Data type error</td>
<td>-171</td>
<td>Invalid expression</td>
</tr>
<tr>
<td>-105</td>
<td>GET not allowed</td>
<td>-178</td>
<td>Expression data not allowed</td>
</tr>
<tr>
<td>-108</td>
<td>Parameter not allowed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-109</td>
<td>Missing parameter</td>
<td>-200</td>
<td>Execution error</td>
</tr>
<tr>
<td>-112</td>
<td>Program mnemonic too long</td>
<td>-211</td>
<td>Trigger ignored</td>
</tr>
<tr>
<td>-113</td>
<td>Undefined header</td>
<td>-221</td>
<td>Settings conflict</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-222</td>
<td>Data out of range</td>
</tr>
<tr>
<td>-121</td>
<td>Invalid character in number</td>
<td>-223</td>
<td>Too much data</td>
</tr>
<tr>
<td>-123</td>
<td>Numeric overflow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-124</td>
<td>Too many digits</td>
<td>-310</td>
<td>System error</td>
</tr>
<tr>
<td>-128</td>
<td>Numeric data not allowed</td>
<td>-350</td>
<td>Too many errors</td>
</tr>
<tr>
<td>-130</td>
<td>Suffix error</td>
<td>-400</td>
<td>Query error</td>
</tr>
<tr>
<td>-131</td>
<td>Invalid suffix</td>
<td>-410</td>
<td>Query INTERRUPTED</td>
</tr>
<tr>
<td>-138</td>
<td>Suffix not allowed</td>
<td>-420</td>
<td>Query UNTERMINATED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-430</td>
<td>Query DEADLOCKED</td>
</tr>
<tr>
<td>-140</td>
<td>Character data error</td>
<td>-440</td>
<td>Query UNTERMINATED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>after indefinite response</td>
</tr>
<tr>
<td>-141</td>
<td>Invalid character data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-144</td>
<td>Character data too long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-148</td>
<td>Character data not allowed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-150</td>
<td>String data error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-151</td>
<td>Invalid string data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-152</td>
<td>String data not allowed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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> ::= <B> indicates that <A> can be replaced by <B> in any statement containing <A>.
- | "or." Indicates a choice of one element from a list. For example, <A> | <B> indicates <A> or <B> 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}

---
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>*RST</td>
<td>(Reset)</td>
<td>command</td>
</tr>
<tr>
<td>Command Syntax: *RST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*SAV</td>
<td>(Save)</td>
<td>command</td>
</tr>
<tr>
<td>Command Syntax: *SAV {1 to 16}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*SRE</td>
<td>(Service Request Enable)</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax: *SRE {0 to 255}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query Syntax: *SRE?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returned Format: (&lt;mask&gt;&lt;NL&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Where: (&lt;mask&gt; ::= \text{sum of all bits set} - \text{integer, 0 to 255})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*STB</td>
<td>(Status Byte)</td>
<td>query</td>
</tr>
<tr>
<td>Query Syntax: *STB?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returned Format: {integer, 0 to 255}&lt;NL&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*TRG</td>
<td>(Trigger)</td>
<td>command</td>
</tr>
<tr>
<td>Command Syntax: *TRG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*TST</td>
<td>(Test)</td>
<td>query</td>
</tr>
<tr>
<td>Query Syntax: *TST?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returned Format: {0 or non-zero value}&lt;NL&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Where: \(0 ::= \text{test passed} \)
\(\text{non-zero ::= test failed}\) |
<p>| *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}&lt;NL&gt; |</p>
<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ACQuire:COUNt</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:ACQuire:COUNt [0</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:ACQuire:COUNt?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{ 8</td>
</tr>
<tr>
<td>:ACQuire:POINts</td>
<td>query</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:ACQuire:POINts?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{integer, 1 to 4000}&lt;NL&gt;</td>
</tr>
<tr>
<td>:ACQuire:Setup</td>
<td>query</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:ACQuire:SETup?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>&lt;string&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td>:ACQuire:TYPE</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:ACQuire:TYPE {NORMal</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:ACQuire:TYPE?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{NORM</td>
</tr>
<tr>
<td>:ASTore</td>
<td>command</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:ASTore</td>
</tr>
<tr>
<td>:AUToscale</td>
<td>command</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:AUToscale</td>
</tr>
<tr>
<td>:BLANK</td>
<td>command</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:BLANK {CHANNEL[1</td>
</tr>
<tr>
<td>:CHANnel{1</td>
<td>2}:BWLimit</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:CHANnel{1</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:CHANnel{1</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{ON</td>
</tr>
</tbody>
</table>

Programmer's Quick Reference
4

HP 54600A/54601A
Oscilloscopes
:CHANnel{1 | 2 | 3 | 4}:COUPling command/query
Command Syntax:  :CHANnel{1 | 2 | 3 | 4}: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 Syntax: :CHANnel{1 | 2 | 3 | 4}:SETup?
Returned Format: <string><NL>

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

:DIGitize  
Command Syntax: :DIGitize Chan{1 | 2 | 3 | 4}, Chan{1 | 2 | 3 | 4}

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

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

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

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

::DISPlay:PIXel
  Command Syntax: ::DISPlay:PIXel<x>,<y>,<intensity>
  Query Syntax: ::DISPlay:PIXel? <x>,<y>
  Returned Format: <integer,intensity><NL>

::DISPlay:ROW
  Command Syntax: ::DISPlay:ROW{1 to 20}
  Query Syntax: ::DISPlay:ROW?
  Returned Format: {integer,1 to 20}<NL>

::DISPlay:Setup
  Query Syntax: ::DISPlay:SETup?
  Returned Format: <string><NL>

::DISPlay:SOURce
  Command Syntax: ::DISPlay:SOURce PMEMory{1|2}
  Query Syntax: ::DISPlay:SOURce?
  Returned Format: PMEM{1|2}<NL>

::DISPlay:TEXT
  Command Syntax: ::DISPlay:TEXT BLANK

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

::ERASE
  Command Syntax: ::ERASE PMEMory{1|2]
<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Query Syntax</th>
<th>Returned Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>:MEASURE:ALL</td>
<td>:MEASURE:ALL?</td>
<td>&lt;FREQUENCY result&gt;,&lt;PERIOD result&gt;,&lt;PWIDTH result&gt;,&lt;NWIDTH result&gt;,&lt;RISetime result&gt;,&lt;FALLtime result&gt;,&lt;VPP result&gt;,&lt;DUTycycle result&gt;,&lt;VRMS result&gt;,&lt;VMAX result&gt;,&lt;VMIN result&gt;,&lt;VTOP result&gt;,&lt;VBASE result&gt;,&lt;VAVge result&gt;</td>
</tr>
<tr>
<td>:MEASURE:DUTycycle</td>
<td>:MEASURE:DUTycycle</td>
<td>&lt;exponential, dutycycle value&gt;</td>
</tr>
<tr>
<td>:MEASURE:FALLtime</td>
<td>:MEASURE:FALLtime</td>
<td>&lt;exponential, falltime value&gt;</td>
</tr>
<tr>
<td>:MEASURE:FREQuency</td>
<td>:MEASURE:FREQuency</td>
<td>&lt;exponential, frequency value&gt;</td>
</tr>
<tr>
<td>:MEASURE:NWIDth</td>
<td>:MEASURE:NWIDth</td>
<td>&lt;exponential, negative_width value&gt;</td>
</tr>
<tr>
<td>:MEASURE:PERiod</td>
<td>:MEASURE:PERiod</td>
<td>&lt;exponential, period value&gt;</td>
</tr>
<tr>
<td>Command/Query</td>
<td>Syntax</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td><strong>:MEASure:PWIDth</strong></td>
<td>command/query</td>
<td></td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:MEASure:PWIDth</td>
<td></td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:MEASure:PWIDth?</td>
<td></td>
</tr>
<tr>
<td>Returned Format:</td>
<td>&lt;exponential, positive width value&gt;&lt;NL&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>:MEASure:RISetime</strong></td>
<td>command/query</td>
<td></td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:MEASure:RISetime</td>
<td></td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:MEASure:RISetime?</td>
<td></td>
</tr>
<tr>
<td>Returned Format:</td>
<td>&lt;exponential, risetime value&gt;&lt;NL&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>:MEASure:SCRatch</strong></td>
<td>command</td>
<td></td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:MEASure:SCRatch</td>
<td></td>
</tr>
<tr>
<td><strong>:MEASure:SHOW</strong></td>
<td>command/query</td>
<td></td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:MEASure:SHOW {ON</td>
<td>OFF}</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:MEASure:SHOW?</td>
<td></td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{ON</td>
<td>OFF}&lt;NL&gt;</td>
</tr>
<tr>
<td><strong>:MEASure:SOURce</strong></td>
<td>command/query</td>
<td></td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:MEASure:SOURce CHAnnel {1</td>
<td>2</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:MEASure:SOURce?</td>
<td></td>
</tr>
<tr>
<td>Returned Format:</td>
<td>CHAN{1</td>
<td>2</td>
</tr>
<tr>
<td><strong>:MEASure:TDELta</strong></td>
<td>query</td>
<td></td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:MEASure:TDELta?</td>
<td></td>
</tr>
<tr>
<td>Returned Format:</td>
<td>&lt;exponential, delta time markers&gt;&lt;NL&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>:MEASure:TSTArt</strong></td>
<td>command/query</td>
<td></td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:MEASure:TSTArt &lt;start marker time&gt;</td>
<td></td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:MEASure:TSTArt?</td>
<td></td>
</tr>
<tr>
<td>Returned Format:</td>
<td>&lt;exponential, start marker time&gt;&lt;NL&gt;</td>
<td></td>
</tr>
</tbody>
</table>
:MEASure:TSTOp

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

:MEASure:TVOLt

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

:MEASure:VAVerage

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

:MEASure:VBASe

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

:MEASure:VDELta

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

:MEASure:VMAX

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

:MEASure:VMIN

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

Programmer's Quick Reference

10

HP 54600A/54601A
Oscilloscopes
:MEASure:VPP

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

:MEASure:VRMS (DC RMS)

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

:MEASure:VSTArt

Command Syntax: :MEASure:VSTArt <marker1 voltage>
Query Syntax: :MEASure:VSTArt?
Returned Format: <exponential, vmarker1 voltage><NL>

:MEASure:VSTOp

Command Syntax: :MEASure:VSTOp <marker2 voltage>
Query Syntax: :MEASure:VSTOp?
Returned Format: <exponential, vmarker2 voltage><NL>

:MEASure:VTIMe

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

:MEASure:VTOP

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

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 Syntax: :MERGe PMEMory {1 | 2}

:PRINt

Query Syntax: :PRINt? [HRes]

:RUN

Command Syntax: :RUN

:STATUs

Query Syntax: :STATUs? [CHANNEL {1 | 2 | 3 | 4} | PMEMory {1 | 2}]
Returned Format: {ON | OFF}<NL>

:STOP

Command Syntax: :STOP

Programmer's Quick Reference

HP 54600A/54601A Oscilloscopes
:SYSTem:DSP

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

:SYSTem:ERRor

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

Where:

<error number>::=
+0, No error
-100, Command error (unknown command)
-101, Invalid character
-102, Syntax error
-103, Invalid separator
-104, Data type error
-105, GET not allowed
-108, Parameter not allowed
-109, Missing parameter
-112, Program mnemonic too long
-113, Undefined header
-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
-199, Execution error
-200, Execution error
-211, Trigger ignored
-221, Settings conflict
-222, Data out of range
-223, Too much data
-224, Too much data
-310, System error
-350, Too many errors
-400, Query error
-410, Query INTERRUPTED
-420, Query UNTERMINATED
-430, Query DEADLOCKED
-440, Query UNTERMINATED
after indefinite response
:SYSTem:KEY

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

Where:

<integer>::= -1 for NO KEY
0 for AUTOSCALE
1 for CH1
2 for CH2
3 for CH3
4 for CH4
5 for +/-
6 for TRIG_SRC
7 for TRIG_MODE
8 for TRIG_SLOPE
9 for MAIN/DELAYED
10 for TIME
11 for VOLTAGE
12 for CURSORS
13 for SAVE_TRACE
14 for SAVE_SETUP
15 for DISPLAY
16 for PRINT/UTILITY
17 NA
18 NA
19 for RUN
20 for AUTOSTORE
21 for STOP
22 for ERASE
23 for SOFTKEY_1
24 for SOFTKEY_2
25 for SOFTKEY_3
26 for SOFTKEY_4
27 for SOFTKEY_5
28 for SOFTKEY_6
29 for CH1_VOLT CW
30 for CH1_VOLT CCW
31 for CH1_POS CW
32 for CH1_POS CCW
33 for CH2_VOLT CW
34 for CH2_VOLT CCW
35 for CH2_POS CW
36 for CH2_POS CCW
37 for CH3_POS CW
38 for CH3_POS CCW
39 for CH4_POS CW
40 for CH4_POS CCW
41 for S/DIV CW
42 for S/DIV CCW
43 for DELAY CW
44 for DELAY CCW
45 for TRIG_LEVEL CW
46 for TRIG_LEVEL CCW
47 for TRIG_HOLD CW
48 for TRIG_HOLD CCW
49 for CURSOR_Knob CW
50 for CURSOR_Knob CCW

CW denotes clockwise rotation of the knob.
CCW denotes counter-clockwise rotation of the knob.
<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SYSTem:LOCK</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:SYSTem:LOCK {ON</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:SYSTem:LOCK?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{ON</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SYSTem:SETup</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:SYSTem:SETup #600000121&lt;setup data string&gt;</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:SYSTem:SETup?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>#600000121&lt;setup data string&gt;&lt;NL&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TER</td>
<td>(Trigger Event Register)</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:TER?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TIMebase:DELas</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:TIMebase:DELas &lt;delay time&gt;</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:TIMebase:DELas?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>&lt;exponential, delay time&gt;&lt;NL&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TIMebase:MODE</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:TIMebase:MODE {NORMal</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:TIMebase:MODE?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{NORM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TIMebase:RANGE</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:TIMebase:RANGE {20 ns to 50 s}</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:TIMebase:RANGE?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{exponential, 20 ns to 50 s}&lt;NL&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command/Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TIMebase:REReference</td>
<td>command/query</td>
</tr>
<tr>
<td>Command Syntax:</td>
<td>:TIMebase:REReference {LEFT</td>
</tr>
<tr>
<td>Query Syntax:</td>
<td>:TIMebase:REReference?</td>
</tr>
<tr>
<td>Returned Format:</td>
<td>{LEFT</td>
</tr>
</tbody>
</table>
:TIMebase:SETup

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

:TIMebase:VERNier

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

:TRIGger:COUPling

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

:TRIGger:HOLDoff

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

:TRIGger:LEVEL

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

:TRIGger:MODE

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

:TRIGger:NREJect

Command Syntax: :TRIGger:NREJect:{ON | OFF}
Query Syntax: :TRIGger:NREJect?
Returned Format: {ON | OFF}<NL>
:TRIGger:POLarity
Command Syntax: :TRIGger:POLarity {POSitive | NEGative}
Query Syntax: :TRIGger:POLarity?
Returned Format: {POS | NEG}<NL>

:TRIGger:REJect
Command Syntax: :TRIGger:REJect {Off | LF | HF}
Query Syntax: :TRIGger:REJect?
Returned Format: {Off | LF | HF}<NL>

:TRIGger:Setup
Query Syntax: :TRIGger:SETup?
Returned Format: <string><NL>

:TRIGger:SLOPe
Command Syntax: :TRIGger:SLOPe {POSitive | NEGative}
Query Syntax: :TRIGger:SLOPe?
Returned Format: {POS | NEG}<NL>

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

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

:TRIGger:TVMode
Command Syntax: :TRIGger:TVMode {FIELD1 | FIELD2 | LINE}
Query Syntax: :TRIGger:TVMode?
Returned Format: {FIELD1 | FIELD2 | LINE}<NL>
:VIEW

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

:WAVEform:BYTeorder

command/query

Command Syntax: :WAVEform:BYTeorder {LSBFirst | MSBFirst}
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 {ASC11 | 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 Syntax: :WAVeform:PREamble?
Returned Format: <preamble block><NL>
Where:
<preamble block> ::= <format NR1>,<type NR1>, <points NR1>,<count NR1>,<increment NR3>,<reference NR1>,<increment NR1>,<increment NR1>,<type 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 Syntax: :WAVeform:SOURce CHANNEL1 | 2 | 3 | 4
Query Syntax: :WAVeform:SOURce?
Returned Format: CHANNEL1 | 2 | 3 | 4<NL>

:WAVeform:TYPE

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

:WAVeform:XINCrement

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

:WAVeform:XORigin

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

:WAVeform:XREFerence

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

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

:WAVeform:YORigin

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

:WAVeform:YREFerence

Query Syntax: :WAVeform:YREFerence?
Returned Format: <integer, y-reference value><NL>
The HP54600A has two identical channel subsystems. The HP54600A 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.