Programmer's Quick Reference Guide

Publication number 54542-97042
First edition, November 1995

For Safety information, Warranties, and Regulatory information, see the pages at the end of this book.

© Copyright Hewlett-Packard Company 1993-1995
All Rights Reserved

HP 54520C and 54540C Series Oscilloscopes
Introduction

The Quick Reference Guide 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 guide:

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

\[ \text{"is defined as." For example, } \langle A \rangle := \langle B \rangle \text{ indicates that } \langle A \rangle \text{ can be replaced by } \langle B \rangle \text{ in any statement containing } \langle A \rangle. \]

\[ \text{"or." Indicates a choice of one element from a list. For example, } \langle A \rangle | \langle B \rangle \text{ indicates } \langle A \rangle \text{ or } \langle B \rangle \text{ but not both.} \]

\[ \text{An ellipsis (trailing dots) indicate that the preceding element may be repeated one or more times.} \]

[ ] 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:

\[ \begin{align*} 
\text{EX} & := 1 \text{E}18, & \text{M} & := 1 \text{E}-3, \\
\text{PE} & := 1 \text{E}15, & \text{U} & := 1 \text{E}-6, \\
\text{T} & := 1 \text{E}12, & \text{N} & := 1 \text{E}-9, \\
\text{G} & := 1 \text{E}9, & \text{P} & := 1 \text{E}-12, \\
\text{MA} & := 1 \text{E}6, & \text{F} & := 1 \text{E}-15, \\
\text{K} & := 1 \text{E}3, & \text{A} & := 1 \text{E}-18 \\
\end{align*} \]

For more information on specific commands or queries, refer to the HP 54520C and 54540C Series Oscilloscopes Programmer's Reference.
**CLS**

Command

*CLS

**DMC**

Where:

*DMC <ascii_string>,<block_data>
<ascii_string> ::= a quoted ascii string
<block_data> ::= definite block data in IEEE 488.2 # format

**EMC**

Command

*EMC {(OFF 0) | (ON 1)}
*EMC?

Returned Format

{0| 1}<NL>

**ESE**

Command

*ESE <mask_argument>
*ESR?

Query

<mask_argument><NL>

Returned Format

<mask_argument> ::= integer, 0 to 255

Where:

**ESR?**

Query

*ESR?

Returned Format

<status><NL>

<status> ::= integer, 0 to 255

**GMC?**

Query

*GMC? <ascii_string>
[block_data]<NL>
<ascii_string> ::= a quoted string
[block_data] ::= definite block data in # format

Returned Format

Where:
*IDN?

Query
returned format
where:

*IDN?

HEWLETT-PACKARD, 545XXC, YYYYYYYYY, ZZ.ZZ, ZZ.ZZ, ZZ.ZZ,ZZ.ZZ,<NL>

<XXC> ::= model number is 20C (HP 54520C), 22C (HP 54522C),
        40C (HP 54540C), or 42C (HP 54542C)
<YYYYYYYYY> ::= the serial number of the instrument
<ZZ.ZZ> ::= the software revision of the software modules
            (Boot ROM, Flash ROM version of Boot ROM, System, Keyboard
             ROM). 00.00 = not installed
<ZZ.ZZ> ::= the telecommunication mask option
            null = no option, 0.001 = option installed

*LMC

Query
returned format
where:

*LMC?

<ascii_string><NL>
<ascii_string> ::= string list separated by commas

*LRN?

Query
returned format
where:

*LRN?

:SYSTem:SETup <setup><NL>
<setup> ::= #800002048<learn_string>
<learn_string> ::= 2048 bytes in length

*OPC

Command
query
returned format

*OPC
*OPC?
1<NL>

*OPT?

Query
returned format
where:

*OPT?

0<NL> or 000.1<NL>
0 = no option
0.001 = telecom mask option installed
*PMC
Command: *PMC

*RCL
Command: *RCL (0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)

*RST
Command: *RST

*SAV
Command: *SAV (1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)

*SRE
Command: *SRE <mask_argument>
Query: *SRE?
Returned Format: <mask><NL>
Where: <mask_argument> ::= integer, 0 to 255
<mask> ::= sum of all bits that are set - integer, 0 through 255

*STB?
Query: *STB?
Returned Format: <value><NL>
Where: <value> ::= integer, 0 through 255

*TRG?
Command: *TRG
**Programmer's Quick Reference Guide**

<table>
<thead>
<tr>
<th>Command</th>
<th>Query</th>
<th>Returned Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TST?</strong></td>
<td></td>
<td>*TST?</td>
</tr>
<tr>
<td></td>
<td>*TST?</td>
<td>&lt;result&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;result&gt; ::= 0 or non-zero value. 0 indicates the test passed. non-zero indicates the test failed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Query</th>
<th>Returned Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WAI</strong></td>
<td></td>
<td>*WAI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Query</th>
<th>Returned Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACQuire:COMPLETE</strong></td>
<td></td>
<td>:ACQuire:COMPLETE &lt;complete_argument&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[:ACQuire:COMPLETE] &lt;complete_argument&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;complete_argument&gt; ::= integer, 0 to 100 percent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Query</th>
<th>Returned Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACQuire:COUNt</strong></td>
<td></td>
<td>:ACQuire:COUNt &lt;count_argument&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[:ACQuire:COUNt] &lt;count_argument&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;count_argument&gt; ::= integer, 1 to 2048 (depending on the acquisition type)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Query</th>
<th>Returned Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACQuire:POINts</strong></td>
<td></td>
<td>:ACQuire:POINts &lt;points_argument&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[:ACQuire:POINts] &lt;points_argument&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;points_argument&gt; ::= integer, 500 in repetitive mode, 512, 1024, 2048, 4196, 8192, 16384, or 32768 in the real-time mode (sequential mode off), or 4 through 32768 in the real-time mode (sequential mode on)</td>
</tr>
</tbody>
</table>
:ACQuire:TYPE

Command
:ACQuire:TYPE {NORMAL | AVERage | ENvelope | PDETect|
RAWData[,<length>],[<acquisitions>]],[<length>],[<acquisitions>]]
Query
:ACQuire:TYPE?
Returned Format
[:ACQuire:TYPE] {NORMAL | AVERage | ENvelope | PDETect|
RAWData,<length>,<acquisitions><NL>
Where:
<length> ::= integer, 4 to 32768
<acquisitions> ::= dependent on length of acquisitions and
buffer size

:AUToscale

Command
:AUToscale
:AUToscale?
Query
Returned Format
[:AUToscale] {0 | 1}<NL>

:BEEPer

Command
:BEEPer [{OFF | 0} | {ON | 1}]
Query
:BEEPer?
Returned Format
[:BEEPer] {0|1}<NL>

:BLANk

Command
:BLANk <display>
Where:
<display> ::= {CHANNEL<n> | FUNCTION{1|2|3|4} | |
MEMORY{1 | 2 | 3 | 4} | PMEMory{1 | 2})
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
(HP 54540C/54542C)

:BNC

Command
:BNC {PROBe | TRIGger}
Query
:BNC?
Returned Format
[:BNC] {PROBe | TRIGger}<NL>
:CALibrate:DATA:ASCII?

Query
:CALibrate:DATA:ASCII?

Returned Format
::CALibrate:DATA:ASCII? <data>,<data>,...<NL>
<data> ::= calibration data

Where:

:CALibrate:SETup?

Query
:CALibrate:SETup?

Returned Format
::CALibrate:SETup? <null_value_n>,<null_value_n>,<null_value_n><NL>

Where:

<null_value_n> ::= exponential, channel 1 to channel<n> skew,
where n = 2 (HP 54520C/54522C) or 2 through 4
(HP 54540C/54542C) in format

:CALibrate:TNULL1

Command
::CALibrate:TNULL <null_value_n>

Query
:CALibrate:TNULL?

Returned Format
::CALibrate:TNULL1 <null_value_n><NL>

Where:

<null_value_n> ::= exponential, channel 1 to channel<n> skew,
where n = 2 (HP 54520C/54522C) or 2 through 4
(HP 54540C/54542C)

:CHANnel<n>:COUPling

Command
::CHANnel<n>:COUPling {AC | DC | DCFifty}

Query
:CHANnel<n>:COUPling?

Returned Format
::CHANnel<n>:COUPling <AC | DC | DCFifty><NL>

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
(HP 54540C/54542C)

:CHANnel<n>:DISPlay

Command
::CHANnel<n>:DISPlay {OFF | 0} | {ON | 1}

Query
:CHANnel<n>:DISPlay?

Returned Format
::CHANnel<n>:DISPlay <0 | 1><NL>

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
(HP 54540C/54542C)
**:CHANnel<n>:ECL**

Command

:CHANnel<n>:ECL

Where:

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

---

**:CHANnel<n>:HFReject**

Command

:CHANnel<n>:HFReject {{OFF | 0} | {ON | 1}}

Query

:CHANnel<n>:HFReject?

Returned Format

[:CHANnel<n>:HFReject] {0 | 1}<NL>

Where:

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

---

**:CHANnel<n>:LFReject**

Command

:CHANnel<n>:LFReject {{OFF | 0} | {ON | 1}}

Query

:CHANnel<n>:LFReject?

Returned Format

[:CHANnel<n>:LFReject] {0 | 1}<NL>

Where:

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

---

**:CHANnel<n>:OFFSet**

Command

:CHANnel<n>:OFFSet <offset_argument>

Query

:CHANnel<n>:OFFSet?

Returned Format

[:CHANnel<n>:OFFSet] <offset_argument><NL>

Where:

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

+offset_argument> ::= exponential, offset value in volts
:CHANnel<n>:PROBe

Command
:CHANnel<n>:PROBe <probe_argument>

Query
:CHANnel<n>:PROBe?

Returned Format
[:CHANnel<n>:PROBe] <probe_argument><NL>

Where:
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
       (HP 54540C/54542C)
<probe_argument> ::= exponential, 0.9 to 1000.0

:CHANnel<n>:RANGe

Command
:CHANnel<n>:RANGe <range_argument>

Query
:CHANnel<n>:RANGe?

Returned Format
[:CHANnel<n>:RANGe] <range_argument><NL>

Where:
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
       (HP 54540C/54542C)
<range_argument> ::= exponential, full-scale range value

:CHANnel<n>:SETup

Query
:CHANnel<n>:SETup?

Returned Format
:CHAN<n>:Coup {AC|DC|DCP};
    DISP {0 | 1};
    HFR {0 | 1};
    LFR {0 | 1};
    OFFS <offset_argument>;
    PROB <probe_argument>;
    RANG <range_argument><NL>

Where:
<range_argument> ::= exponential, full-scale range value
<offset_argument> ::= exponential, offset value in volts
<probe_argument> ::= exponential, 0.9 to 1000.0
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
       (HP 54540C/54542C)

:CHANnel<n>:TTL

Command
:CHANnel<n>:TTL

Where:
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
       (HP 54540C/54542C)
:DIGitize

Command
 Where: :DIGitize CHANNEL<n>[,CHANNEL<n>[],CHANNEL<n>[],CHANNEL<n>][]]
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:DISK:CDIRectory

Command
 Where: :DISK:CDirectory <directory_name>
<directory_name> ::= 1 to 65 character quoted ASCII string

:DISK:DELete

Command
 Where: :DISK:DElete <file_name>
<file_name> ::= 1 to 8 character quoted ASCII string, if DOS, can have a 0 to 3 character extension

:DISK:DIREctory?

Query
 Returned Format
 Where: :DISK:DIREctory?
[D:DISK:DIREctory] <number_of_files><cr><lf><directory>
<number_of_files> ::= integer, number of files (that follow) in the root directory
<directory> ::= (<filename>,<ext>,<date>,<time>,<size>,
<description><cr><lf>...)
<ext> ::= (SETup | WAVEform | TEXT | PIXel)
<date> ::= DDMMYYYY
<time> ::= HH:MM:SS
<size> ::= an integer
<description> ::= LIF format: Model + C1 + HP545XX + Source,
DOS format: Model = DOS file (no description)

:DISK:FORMAT

Command
 Where: :DISK:FORMAT <format_type>
<format_type> ::= (DOS | LIF)
:DISK:LOAD

**Command**
:DISK:LOAD <file_name>,<destination> [,<format>]

**Where:**

- `<file_name>` ::= quoted ascii string. DOS compatible filename.
  - 1 to 8 character ASCII string, if DOS, may have a 0 to 3 character extension.
  - Either .wav or .txt may be used as a suffix after the filename. If no file suffix is specified, the default is .wav.

- `<destination>` ::= (WMEMory {1 | 2 | 3 | 4} | SETup{0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9} | PMEMory{ 1 | 2})

- `<format>` ::= (TEXT | INTERNAL)

:DISK:MDIRectory

**Command**
:DISK:MDirectory <directory_name>

**Where:**

- `<directory_name>` ::= 1 to 65 character quoted ASCII string

:DISK:PWD?

**Query**
:DISK:PWD?

**Returned Format:**
[:DISK:PWD] <present_working_directory>

:DISK:SIMage

**Command**
:DISK:SIMage <file_name> [,<format>],[,<compression>],[,<rendering>]]

**Where:**

- `<file_name>` ::= 1 to 8 character quoted ASCII string, if DOS, can have a 0 to 3 character extension.

- `<format>` ::= {TIFF | PCX | EPS}

- `<compression>` ::= {ON | OFF}

- `<rendering>` ::= {BW | COLOR | GREen}
:DISK:STORE

Command
:DISK:STORE <source>,<file_name>[<format>]

Where:
<source> ::= [CHANnel<n> | WMEMory{1 | 2 | 3 | 4} | SETupN {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9} | PMEMory (1 | 2)
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)
<filename> ::= a descriptive name of the file up to 8 characters long
<format> ::= [INTERNAL | TEXT [{<XYPairs> | <YVALues}> | <VERBOSE}>]]

:DISP:COL

Command
:DISP:COL <column_number>

Query
:DISP:COL?

Returned Format
[:DISP:COL] <column_number><NL>

Where:
<column_number> ::= integer, 0 through 78

:DISP:CON

Command
:DISP:CON {OFF | 0} | {ON | 1}

Query
:DISP:CON?

Returned Format
[:DISP:CON] {0 | 1}<NL>

:DISP:DAT

Command
:DISP:DAT <binary_block>

Query
:DISP:DAT?

Returned Format
[:DISP:DAT] #800016576<16576 bytes of binary data><NL>

Where:
<binary_block> ::= definite block data in # format

:DISP:FOR

Command
:DISP:FOR {1 | 2 | 4}

Query
:DISP:FOR?

Returned Format
[:DISP:FOR] {1 | 2 | 4}<NL>
Programmer’s Quick Reference Guide

:DISPLAY:GRATicule

Command
:DISPLAY:GRATicule {AXES | GRID | FRAME | OFF}
Query
:DISPLAY:GRATicule?
Returned Format
[:DISPLAY:GRATicule] {AXES | GRID | FRAME | OFF}<NL>

:DISPLAY:INVerse

Command
:DISPLAY:INVerse {{OFF | 0} | {ON | 1}}
Query
:DISPLAY:INVerse?
Returned Format
[:DISPLAY:INVerse] {0 | 1}<NL>

:DISPLAY:LINE

Command
:DISPLAY:LINE <ascii_string>
Where:
<ascii_string> ::= any series of ascii characters enclosed in quotes

:DISPLAY:MASK

Command
:DISPLAY:MASK <mask_argument>
Query
:DISPLAY:MASK?
Returned Format
[:DISPLAY:MASK] <mask_argument><NL>
Where:
<mask_argument> ::= integer, 0 through 255

:DISPLAY:PERSistence

Command
:DISPLAY:PERSistence {SINGLE | INFINITE | 0 | {0.5 to 10} | 11}
Query
:DISPLAY:PERSistence?
Returned Format
[:DISPLAY:PERSistence] <value><NL>
Where:
,value> ::= exponential, {0 (minimum) | (0.5 to 10) | 11 (infinite)) in the repetitive mode, {SINGLE | INFINITE) in real-time mode
:DISPlay:ROW

Command
:DISPlay:ROW <row_number>
:DISPlay:ROW?

Query

Returned Format
[:DISPlay:ROW] <row_number><NL>

Where:
<row_number> ::= integer, 0 through 24

:DISPlay:SCReen

Command
:DISPlay:SCReen {OFF | 0} | {ON | 1})

Query
:DISPlay:SCReen?

Returned Format
[:DISPlay:SCReen] (0 | 1)<NL>

:DISPlay:SETup?

Query
:DISP:COL <column_number>;

Returned Format
CONN {0 | 1};
FORM {1 | 2 | 4};
GRAT {AXES | FRAM | GRID | OFF};
INV {0 | 1};
MASK <mask_argument>;
PERS <pers_argument>;
ROW <row_number>;
SCR {0 | 1};
SOUR PMEM {0 | 1 | 2};
MARK {0 | 1}<NL>

Where:
<column_number> ::= integer, 0 through 78

<mask_argument> ::= integer, 0 to 255

<pers_argument> ::= exponential, {0 | .5 to 10 | 11} in the repetitive mode, {SINGle | INFinite} in the real-time mode

<row_number> ::= integer, 0 to 24

:DISPlay:SOURce

Command
:DISPlay:SOURce PMEMory{0 | 1 | 2 }

Query
:DISPlay:SOURce?

Returned Format
[:DISPlay:SOURce] PMEMory{0 | 1 | 2}<NL>
Programmer's Quick Reference Guide

:DISPLAY:STRING

Command
:DISPLAY:STRING <string_argument>

Where:
<string_argument> ::= text string up to 90 characters

:DISPLAY:TEXT

Command
:DISPLAY:TEXT BLANK

:DISPLAY:[MARKer|TMARKer|VMARKer]

Command
:DISPLAY:[MARKer|TMARKer|VMARKer] {{OFF | 0} | (ON | 1)}

Query
:DISPLAY:[MARKer|TMARKer|VMARKer]?

Returned Format
[:DISPLAY:[MARKer|TMARKer|VMARKer] {0 | 1}<NL>

:ERASE

Command
:ERASE {MEMORY0<pmemory_num>}

Where:
<pmemory_num> ::= integer, 1 or 2 #MODULE RULE =

:EXTERNAL:COUPling

Command
:EXTERNAL:COUPling {AC | DC | DCfifty}

Query
:EXTERNAL:COUPling?

Returned Format
[:EXTERNAL:COUPling] {AC | DC | DCfifty}<NL>

:EXTERNAL:HFReject

Command
:EXTERNAL:HFReject {{OFF | 0} | (ON | 1)}

Query
:EXTERNAL:HFReject?

Returned Format
[:EXTERNAL:HFReject] {0 | 1}<NL>
:EXTernal:LFReject

Command
:EXTernal:LFReject {OFF | 0} | {ON | 1}

Query
[:EXTernal:LF Reject?]

Returned Format
[0 | 1]<NL>

:EXTernal:PROBe

Command
:EXTernal:PROBe <probe_argument>

Query
[:EXTernal:PROBe?]

Returned Format
[:EXTernal:PROBe] <probe_argument><NL>

<probe_argument> ::= exponential, 0.9 to 1000.0

:EXTernal:RANGE

Command
:EXTernal:RANGE <range_argument>

Query
[:EXTernal:RANGE?]

Returned Format
[:EXTernal:RANGE] <range_argument><NL>

<range_argument> ::= exponential, full-scale range value

:EXTernal:SETup

Query
:EXTernal:SETup?

:EXT:COUP {AC | DC | DCF};

HFR (0 | 1);

LFR (0 | 1);

PROB <probe_argument>;

RANG <range_argument><NL>

Where:

<range_argument> ::= exponential, full-scale range value

<probe_argument> ::= exponential, 0.9 to 1000.0

:FUNCTION[1 | 2 | 3 | 4]:ADD

Command
:FUNCTION(1 | 2 | 3 | 4):ADD <operand>, <operand>

Where:

<operand> ::= (CHANNEL<n> | WMEMory(1 | 2 | 3 | 4))

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

17
### FUNCTION [1 | 2 | 3 | 4]: DIFF

**Command**: `FUNCTION[1 | 2 | 3 | 4]: DIFF <operand>

**Where**: `(CHANnel<n> | WMEMory{1 | 2 | 3 | 4})`

**<n>**: integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

### FUNCTION [1 | 2 | 3 | 4]: DISPLAY

**Command**: `FUNCTION[1 | 2 | 3 | 4]: DISPLAY {OFF | 0} {ON | 1}`

**Query**: `FUNCTION[1 | 2 | 3 | 4]: DISPLAY?`

**Returned Format**: `[:FUNCTION[1 | 2 | 3 | 4]:DISPLAY] {0 | 1}<NL>

### FUNCTION [1 | 2 | 3 | 4]: FFT

**Command**: `FUNCTION[1 | 2 | 3 | 4]: FFT <operand>

**Where**: `(CHANnel<n> | WMEMory{1 | 2 | 3 | 4})`

**<n>**: integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

### FUNCTION [1 | 2 | 3 | 4]: FREQUENCY

**Command**: `FUNCTION[1 | 2 | 3 | 4]: FREQUENCY <frequency_argument>`

**Query**: `FUNCTION[1 | 2 | 3 | 4]: FREQUENCY?`

**Returned Format**: `[:FUNCTION[1 | 2 | 3 | 4]:FREQUENCY] <frequency_argument><NL>

**Where**: `0 Hz to 1.5X of frequency span`

### FUNCTION [1 | 2 | 3 | 4]: INTEGRATE

**Command**: `FUNCTION[1 | 2 | 3 | 4]: INTEGRATE <operand>`

**Where**: `(CHANnel<n> | WMEMory{1 | 2 | 3 | 4})`

**<n>**: integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)
:FUNCTION[1 2 3 4]:INVert

Command
Where:  
:FUNCTION[1 2 3 4]:INVert <operand>
<operand> ::= {CHANnel<n> | WMEMory[1 2 3 4]}
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:FUNCTION[1 2 3 4]:LEVel

Command
Query
Returned Format
Where:  
:FUNCTION[1 2 3 4]:LEVel <level_argument>
:FUNCTION[1 2 3 4]:LEVel?
[:FUNCTION[1 2 3 4]:LEVel] <level_argument><NL>
<level_argument> ::= exponential, level value 0 to + or - 600 dbm

:FUNCTION[1 2 3 4]:MAGNify

Command
Query
Returned Format
Where:  
:FUNCTION[1 2 3 4]:MAGNify [{OFF | 0} | {ON | 1}]
:FUNCTION[1 2 3 4]:MAGNify?
[:FUNCTION[1 2 3 4]:MAGNify] {0 | 1}<NL>

:FUNCTION[1 2 3 4]:MODE?

Command
Query
Returned Format
Where:  
:FUNCTION[1 2 3 4]:MODE?
[:FUNCTION[1 2 3 4]:MODE]
<operation>,<operand>[,<operand>]
<operation> ::= {ADD | SUBTract | MULtiply | VERSus | ONLY | INVert | INtegrate | DIFF | FFT}
<operand> ::= {CHANnel<n> | WMEMory[1 2 3 4]}
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:FUNCTION[1 2 3 4]:MULTiply

Command
Where:  
:FUNCTION[1 2 3 4]:MULTiply <operand>,<operand>
<operand> ::= {CHANnel<n> | WMEMory[1 2 3 4]}
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)
:FUNCTION[1 2 3 4]:OFFSET

Command:
:FUNCTION[1 2 3 4]:OFFSET <offset_argument>
Query:
:FUNCTION[1 2 3 4]:OFFSET?
Returned Format:
[:FUNCTION[1 2 3 4]:OFFSET] <offset_argument><NL>
<offset_argument> ::= exponential, offset value of 0 to +200 dBm for FFT

:FUNCTION[1 2 3 4]:ONLY

Command:
:FUNCTION[1 2 3 4]:ONLY <operand>
Query:
:FUNCTION[1 2 3 4]:ONLY?
<operand> ::= {CHANNEL<n> | WMEMory{1 2 3 4}}
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:FUNCTION[1 2 3 4]:PEAK

Command:
:FUNCTION[1 2 3 4]:PEAK <peak1_number>,<peak2_number>
Query:
:FUNCTION[1 2 3 4]:PEAK?
<peak1_number>,<peak2_number><NL>
<peak1_number> ::= integer, 1 through 99
<peak2_number> ::= integer, 1 through 99

:FUNCTION[1 2 3 4]:POINts

Command:
:FUNCTION[1 2 3 4]:POINts <points_argument>
Query:
:FUNCTION[1 2 3 4]:POINts?
Returned Format:
[:FUNCTION[1 2 3 4]:POINts] <points_argument><NL>
<points_argument> ::= integer, 512, 1024, 2048, 4096, 8192, 16384, or 32768

:FUNCTION[1 2 3 4]:RANGE

Command:
:FUNCTION[1 2 3 4]:RANGE <range_argument>
Query:
:FUNCTION[1 2 3 4]:RANGE?
Returned Format:
[:FUNCTION[1 2 3 4]:RANGE] <range_argument><NL>
<range_argument> ::= exponential full scale vertical range in volts, dB, or dBm

20
:FUNCTION{1 | 2 | 3 | 4}:SETup?

Query

:FUNCTION{1 | 2 | 3 | 4}:SETup?

:FUNCTION{1 | 2 | 3 | 4}:DISP {0|1};

{ADD | SUBT | MULT | VERS | ONLY | INV | INT | DIFF | FFT} {CHAN<n> | WMEM {1 | 2 | 3 | 4} [,.{CHAN<n>} | WMEM {1 | 2 | 3 | 4}]};

OFF <offset_argument>;
RANG <range_argument>;
FREQ <frequency_argument>; (FFT only)
LEV <level_argument>; (FFT only)
MAGN {0 | 1}; (FFT only)
PEAK <peak1_number>,peak2_number>; (FFT only)
POIN <points_argument>; (FFT only)
SPAN <span_argument>; (FFT only)
WIND {RECT | HANN | FLAT} (FFT only)<NL>

<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)
<offset_argument> ::= exponential offset value of 0 to ±200 dBm for FFT function
<range_argument> ::= exponential full scale vertical range in volts, dB, or dBm
<frequency_argument> ::= exponential center frequency from 0 Hz to 1.5X of frequency span
<level_argument> ::= exponential level value 0 to ±600 dBm
<peak1_number> ::= integer, 1 through 99
<peak2_number> ::= integer, 1 through 99
<points_argument> ::= integer, 512, 1024, 2048, 4096, 8192, 16384, or 32768
<span_argument> ::= exponential number in hertz

:FUNCTION{1 | 2 | 3 | 4}:SPAN

Command
Query
Returned Format
Where:

:FUNCTION{1 | 2 | 3 | 4}:SPAN <span_argument>

:FUNCTION{1 | 2 | 3 | 4}:SPAN?

{:FUNCTION{1 | 2 | 3 | 4}:SPAN} <span_argument><NL>

<span_argument> ::= exponential number in hertz
Programmer's Quick Reference Guide

:FUNCtion{1 | 2 | 3 | 4}:SUBTract

Command
Where:

:FUNCtion{1 | 2 | 3 | 4}:SUBTract <operand1>, <operand2>
<operand1> ::= {CHAnnel<n> | WMEMory{1 | 2 | 3 | 4}}
<operand2> ::= {CHAnnel<n> | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:FUNCtion{1 | 2 | 3 | 4}:VERSus

Command
Where:

:FUNCtion{1 | 2 | 3 | 4}:VERSus <Y_operand>, <X_operand>
<Y_operand> ::= {CHAnnel<n> | WMEMory{1 | 2 | 3 | 4}}
<X_operand> ::= {CHAnnel<n> | WMEMory{1 | 2 | 3 | 4}}
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:FUNCtion{1 | 2 | 3 | 4}:WINDow

Command
Query
Returned Format

:FUNCtion{1 | 2 | 3 | 4}:WINDow {RECTangular | HANNing | FLATtop}
:FUNCtion:WINDow?
[:FUNCtion{1|2|3|4}:WINDow] {RECTangular | HANNing | FLATtop}<NL>

:HARDcopy:LENGth

Command
Query
Returned Format

:HARDcopy:LENGth {11 | 11.6 | LETTER | A4}
:HARDcopy:LENGth?
[:HARDcopy:LENGth] {11 | 11.6 | LETTER | A4}<NL>
:HARDcopy:MODE

Command
: HARDcopy: MODE {THINKJET | DJ5XBW75DPI | DJ5XBW100DPI |
DJ5XBW150DPI | DJ5XBW300DPI | DJET75DPI | DJET100DPI |
DJET150DPI | DJET300DPI | LASERJET_II | LASERJET_IIP |
PJETXL300 | PJETXL | PAINTJET | COLORPRO | HP7475A |
HP7470A | HP7550A | DJET310C | DJET320C | DJET500C |
DJET540C | DJET560C | DJET660C | DJET850C | DJET855C |
P1200C | QUIETJETALT | EPSON5000} <NL>

Query
: HARDcopy: MODE?

Returned Format
[ :HARDcopy: MODE ] {THINKJET | DJ5XBW75DPI | DJ5XBW100DPI |
DJ5XBW150DPI | DJ5XBW300DPI | DJET75DPI | DJET100DPI |
DJET150DPI | DJET300DPI | LASERJET_II | LASERJET_IIP |
PJETXL300 | PJETXL | PAINTJET | COLORPRO | HP7475A |
HP7470A | HP7550A | DJET310C | DJET320C | DJET500C |
DJET540C | DJET560C | DJET660C | DJET850C | DJET855C |
P1200C | QUIETJETALT | EPSON5000)<NL>

:HARDcopy:PAGE

Command
: HARDcopy: PAGE {MANual | AUTomatic}

Query
: HARDcopy: PAGE?

Returned Format
[ :HARDcopy: PAGE ] {MANual | AUTomatic}<NL>

:HARDcopy:PLOT:AREA

Command
: HARDcopy: PLOT: AREA {ALL | DISPLAY | FACTors | GRATicule | LAbeled}

Query
: HARDcopy: PLOT: AREA?

Returned Format
[ :HARDcopy: PLOT: AREA ] {ALL | DISPLAY | FACTors | GRATicule | LAbeled}<NL>

:HARDcopy:PLOT:INITialize

Command
: HARDcopy: PLOT: INITIALize [{OFF | 0} | {ON | 1}]

Query
: HARDcopy: PLOT: INITIALize?

Returned Format
[ :HARDcopy: PLOT: INITIALize ] {0 | 1}<NL>
:HARDcopy:PLOT:{PEN|COLor}

Command
:HARDcopy:PLOT:{PEN|COLor } <item>,<pen_number>
:HARDcopy:PLOT:{PEN|COLor}? <item>

Query

Returned Format
[:HARDcopy:PLOT:{PEN|COLOR}] <pen_number><NL>

Where:
<item> ::= {CHANNEL<n> | WMEMory{1 | 2 | 3 | 4} |
FUNCTION{1 | 2 | 3 | 4} | PMEMory{1 | 2} | Y{1 | 2}Marker | X{1 | 2}Marker | GRATicule | TRIGger | TIMeBase | MEASure | TITLe}s)

<pen_number> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:LER?

Query

Returned Format
[:LER] {0 | 1}<NL>

:LTER?

Query

Returned Format
[:LTER] {0 | 1}<NL>

:MARKer:DISPLAY

Command
:MARKer:DISPLAY {{OFF | 0} | {ON | 1}}

Query

Returned Format
[:MARKer:DISPLAY]? {0 | 1}<NL>

:MARKer:MODE

Command
:MARKer:MODE {MANual | WAVeform}

Query

Returned Format
[:MARKer:MODE] {MANual | WAVeform}<NL>
:MARKer:SETup?

Query
:MARKer:SETup?
:MARKer:MODE {MAN| WAV}; DISP {0 | 1};
XDEL <xdelta>;
X1P <xposition_argument>;
X2P <xposition_argument>;
XY1Y1 (CHAN<n> | FUNC{1 | 2 | 3 | 4} | WMEM{1 | 2 | 3 | 4});
XY2Y2 (CHAN<n> | FUNC{1 | 2 | 3 | 4} | WMEM{1 | 2 | 3 | 4});
Y1P <yposition_argument>;
Y2P <yposition_argument>;
YDEL <ydelta>;
Y1P <yposition_argument>;
Y2P <yposition_argument><NL> (MAN mode only)

Returned Format
Where:
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4
(HP 54540C/54542C)
<xdelta> ::= exponential, difference in seconds between x1 and
x2 markers
<xposition_argument> ::= exponential, xmarker position in
seconds or hertz
<ydelta> ::= exponential, difference between y1 and y2 markers
<yposition_argument> ::= exponential, ymarker position in volts
or power

:MARKer:X1Position

Command
:MARKer:X1Position <xposition_argument>

Query
:MARKer:X1Position?

Returned Format
[[:MARKer:X1Position] <xposition_argument><NL>
<xposition_argument> ::= exponential, xmarker time in seconds,
or frequency in hertz

:MARKer:X2Position

Command
:MARKer:X2Position <xposition_argument>

Query
:MARKer:X2Position?

Returned Format
[[:MARKer:X2Position] <xposition_argument><NL>
<xposition_argument> ::= exponential, xmarker time in
seconds,
or frequency in hertz

25
:MARKer:X1Y1source

Command
:MARKer:X1Y1source {CHANnel<n> | {FUNCTION(1 | 2 | 3 | 4) | WMEMory(1 | 2 | 3 | 4})

Query
:MARKer:X1Y1source?

Return Format
[:MARKer:X1Y1source] {CHANnel<n> | {FUNCTION(1 | 2 | 3 | 4) | WMEMory(1 | 2 | 3 | 4)})<NL>

Where:
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:MARKer:X2Y2source

Command
:MARKer:X2Y2source {CHANnel<n> | {FUNCTION(1 | 2 | 3 | 4) | WMEMory(1 | 2 | 3 | 4})

Query
:MARKer:X2Y2source?

Return Format
[:MARKer:X2Y2source] {CHANnel<n> | {FUNCTION(1 | 2 | 3 | 4) | WMEMory(1 | 2 | 3 | 4)})<NL>

Where:
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:MARKer:XDELta?

Query
:MARKer:XDELta?

Return Format
[:MARKer:XDELta] <xdelta><NL>

Where:
<xdelta> ::= exponential, difference between x1 and x2 markers

:MARKer:Y1Position

(Command ignored in waveform mode)

Command
:MARKer:Y1Position <yposition_argument>

Query
:MARKer:Y1Position?

Return Format
[:MARKer:Y1Position] <yposition_argument><NL>

Where:
<yposition_argument> ::= exponential, ymarker level in volts, or power in dBm
**:MARKer:Y2Position**  (Command ignored in waveform mode)

Command  
:MARKer:Y2Position <yposition_argument>
:MARKer:Y2Position?

[y:MARKer:Y2Position] <yposition_argument><NL>
<yposition_argument> ::= exponential, ymarker level in volts, or power in dBm

**Query**

**Returned Format**

**Where:**

**:MARKer:YDELta?**

Query  
:MARKer:YDELta?

[y:MARKer:YDELta] <ydelta><NL>
<ydelta> ::= exponential, difference between y1 and y2 markers

**Query**

**Returned Format**

**Where:**

**:MEASure:ALL?**

Query  
:MEASure:ALL?

[y:MEASure] [:DELay] <result>; [:DUTycycle] <result>; [:FALLtime] <result>; [:FREQuency] <result>; [:NWIDth] <result>; [:OVERshoot] <result>; [:PERiod] <result>; [:PRESShoot] <result>; [:PWIDth] <result>; [:RISeTime] <result>; [:VACRms] <result>; [:VAMPlitude] <result>; [:VAVerage] <result>; [:VBASel] <result>; [:VDCRms] <result>; [:VMAX] <result>; [:VMIN] <result>; [:VPP] <result>; [:VTOP] <result><NL>

<result> ::= exponential, individual measurement results

**Query**  
**Returned Format**

**Where:**

**:MEASure:COMPare**

Command  
:MEASure:COMPare <measurement>,<upper_limit>,<lower_limit>
:MEASure:COMPare?

[y:MEASure:COMPare] <measurement>,<upper_limit>,<lower_limit><NL>

<measurement> ::= [DELay | DUTycycle | FALLtime | FREQuency | NWIDth | OVERshoot | PERiod | PRESShoot | PWIDth | RISeTime | VACRms | VAMPlitude | VAVerage | VBASel | VDCRms | VMAX | VMIN | VPP | VTOP]
<upper_limit> ::= exponential, high limit value
<lower_limit> ::= exponential, low limit value
:MEASure:CURSor?

Query
[[:MEASure:CURSor? {DELTa | START | STOP}]
<time>,<voltage><NL>
<time> ::= exponential, delta time, y1marker time, or y2marker time
<voltage> ::= exponential, delta voltage, x1marker voltage, or x2marker voltage

:MEASure:DEFine

Command
[:MEASure:DEFine <define_argument>]

Query
[:MEASure:DEFine? {DELay | PWIDth | NWIDth}]

Returned Format
[[:MEASure:DEFine] <define_argument><NL>
<define_argument> ::= {DELay, <polarity>, <edge_number>, <level>,
                  <polarity>, <edge_number>, <level> | PWIDth, <level> | NWIDth, <level>}
<polarity> ::= {POSitive | NECative}
<edge_number> ::= integer, 1 to 4000 specifying a displayed edge
<level> ::= {MIDDLE | UPPer | LOWer}

:MEASure:DELay

Command
[:MEASure:DELay]

Query
[:MEASure:DELay?]

Returned Format
[[:MEASure:DELay] <value><NL>
[value] ::= exponential, time value in seconds
:MEASure:DESTination

Command
:MEASure:DESTination <source_argument>,<destination_argument>

Query
:MEASure:DESTination? [CHANnel<n> | FUNCTION(1 | 2 | 3 | 4) | SCReen]?

Returned Format
[:MEASure:DESTination] {{CHANnel<n> | FUNCTION(1 | 2 | 3 | 4) | SCReen}} [WMEMory(1 | 2 | 3 | 4) | PMEMory(1 | 2) | MULTiple | OFF]<NL>

Where:
<source_argument> ::= {{SCReen, {OFF | PMEMory(1 | 2)}} | {{CHANnel<n> | FUNCTION(1 | 2 | 3 | 4)}}}
<destination_argument> ::= (WMEMory(1 | 2 | 3 | 4) | MULTiple | OFF)
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)

:MEASure:DUTycycle

Command
:MEASure:DUTycycle

Query
:MEASure:DUTycycle?

Returned Format
[ :MEASure:DUTycycle ] <value><NL>

Where:
<value> ::= exponential, ratio of positive pulse width to period

:MEASure:ESTArt

Command
:MEASure:ESTArt <slope_and_occurrence>

Query
:MEASure:ESTArt?

Returned Format
[:MEASure:ESTArt] <slope_and_occurrence><NL>

Where:
<slope_and_occurrence> ::= integer, -4000 to 4000 (excluding 0

:MEASure:ESTOp

Command
:MEASure:ESTOp <slope_and_occurrence>

Query
:MEASure:ESTOp?

Returned Format
[:MEASure:ESTOp] <slope_and_occurrence><NL>

Where:
<slope_and_occurrence> ::= integer, -4000 to 4000 (excluding 0
Programmer’s Quick Reference Guide

:MEASure:EANalysis

Command
:MEASure:EANalysis {OFF | 0} | {ON | 1})

Query
:MEASure:EANalysis?

Returned Format
[[:MEASure:EANalysis] {0 | 1}<NL>

:MEASure:FALLtime

Command
:MEASure:FALLtime

Query
:MEASure:FALLtime?

Returned Format
[:MEASure:FALLtime] <value><NL>

Where:
<value> ::= exponential, time value in seconds between lower threshold and upper threshold voltage points

:MEASure:FREQuency

Command
:MEASure:FREQuency

Query
:MEASure:FREQuency?

Returned Format
[:MEASure:FREQuency] <value><NL>

Where:
<value> ::= exponential, frequency in hertz

:MEASure:LIMIttest

Command
:MEASure:LIMIttest {MEASure | OFF}

:MEASure:LOWer

Command
:MEASure:LOWer [<source>,]<lower_threshold>

Query
:MEASure:LOWer? [<source>]

Returned Format
[:MEASure:LOWer] <source>,<lower_threshold><NL>

Where:
<source> ::= (CHANnel<n> | FUNCTION{1 | 2 | 3 | 4})

<lower_threshold> ::= integer, user defined lower threshold in percent or volts (selected by :MEASure:UNITs)

<n> ::= integer 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)
:MEASure:MODE

Command
:MEASure:MODE {STAnDard | USER}
Query
:MEASure:MODE?
Returned Format
[:MEASure:MODE] {STAnDard | USER}<NL>

:MEASure:MWINdow

Command
:MEASure:MWINdow {MARKers | SCReen}
Query
:MEASure:MWINdow?
Returned Format
[:MEASure:MWINdows] {MARKers | SCReen}<NL>

:MEASure:NWIDth

Command
:MEASure:NWIDth
Query
:MEASure:NWIDth?
Returned Format
[:MEASure:NWIDth] <value><NL>
Where:
<value> ::= exponential, negative pulse width in seconds

:MEASure:OVERshoot

Command
:MEASure:OVERshoot
Query
:MEASure:OVERshoot?
Returned Format
[:MEASure:OVERshoot] <value><NL>
Where:
<value> ::= exponential, ratio of overshoot to Vamplitude

:MEASure:PERiod

Command
:MEASure:PERiod
Query
:MEASure:PERiod?
Returned Format
[:MEASure:PERiod] <value><NL>
Where:
<value> ::= exponential, waveform period in seconds

:MEASure:POSTfailure

Command
:MEASure:POSTfailure {CONTinue | STOP}
Query
:MEASure:POSTfailure?
Returned Format
[:MEASure:POSTfailure] {CONTinue | STOP}<NL>
Programmer's Quick Reference Guide

:MEASure:PRESHoot

Command
:MEASure:PRESHoot

Query
:MEASure:PRESHoot?

Returned Format
[[:MEASure:PRESHoot] <value><NL>
<value> ::= exponential, ratio of preshoot to Vamplitude]

:MEASure:PWIDth

Command
:MEASure:PWIDth

Query
:MEASure:PWIDth?

Returned Format
[[:MEASure:PWIDth] <value><NL>
<value> ::= exponential, width of positive pulse in sseconds]

:MEASure:RESults?

Query
:MEASure:RESults?

Returned Format
[:MEASure:RESults] <number_of_meas>[;<measurement>] ...<NL>
<number_of_meas> ::= integer, number of measurements
displayed

on the screen, 0 through 23

<measurement> ::= (DElay <result> | DUTycycle <result> | FALLtime <result> | FREQuency <result> | NWIDTH <result> | OVERShoot <result> | PERiod <result> | PRESHoot <result> | PWIDth <result> | RISetime <result> | TMAX <result> | TMIN <result> | TVOLT <result> | VACRms <result> | VAMPLitude <result> | VAVerge <result> | VBASE <result> | VDCRms <result> | VMAX <result> | VMIN <result> | VPP <result> | VTIme <result> | VTOP <result>)

<result> ::= exponential, individual measurement results

:MEASure:RISetime

Command
:MEASure:RISetime

Query
:MEASure:RISetime?

Returned Format
[[:MEASure:RISetime] <value><NL>
<value> ::= exponential, rise time in seconds]
:MEASure:SCRatch

Command
:MEASure:SCRatch

:MEASure:SOURce

Command
:MEASure:SOURce <source>[,<source>]
:MEASure:SOURce?
Query
[[:MEASure:SOURce] <source>[,<source>]<NL>
<source> ::= {CHANNEL<n> | FUNCTION(1 | 2 | 3 | 4) | 
WHEREy(1 | 2 | 3 | 4))
<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 
(HP 54540C/54542C)

:MEASure:STATistics

Command
:MEASure:STATistics ((OFF | 0) | (ON | 1))
:MEASure:STATistics?
Query
[[:MEASure:STATistics] {0|1}<NL>

MEASure:STATistics:MODE

Command
:MEASure:STATistics:MODE {NORMAL | SDEViation}
:MEASure:STATistics:MODE?
Query
[[:MEASure:STATistics:MODE] {NORMAL | SDEViation}<NL>

:MEASure:TDELta

Query
:MEASure:TDELta?
Returned Format
[[:MEASure:TDELta] <value><NL>
<value> ::= exponential, difference between x2 and x1 markers

:MEASure:TMAX?

Query
:MEASure:TMAX?
Returned Format
[[:MEASure:TMAX] <time><NL>
<time> ::= exponential, time at maximum voltage
:MEASure:TMIN?
Query
:MEASure:TMIN?
[[:MEASure:TMIN] <time><NL>
<time> ::= exponential, time at minimum voltage

:MEASure:TSTArt
Command
:MEASure:TSTArt <tstart_argument>
Query
:MEASure:TSTArt?
Returned Format
[[:MEASure:TSTArt] <tstart_argument><NL>
<tstart_argument> ::= exponential, time at x1marker in seconds

:MEASure:TSTOp
Command
:MEASure:TSTOp <tstop_argument>
Query
:MEASure:TSTOp?
Returned Format
[[:MEASure:TSTOp] <tstop_argument><NL>
<tstop_argument> ::= exponential, time at x2marker in seconds

:MEASure:TVOLt?
Query
:MEASure:TVOLt? <tvolt_argument>,<slope_and_occurrence>
Returned Format
[[:MEASure:TVOLt] <time><NL>
<tvolt_argument> ::= real number representing positive or negative voltage level that the waveform must cross
<slope_and_occurrence> ::= slope is the direction of the waveform when <voltage> is crossed - rising (space character or +) or falling (-)
occurrence is the number of crossings to be reported (if one - the first crossing is reported, if two - the second crossing is reported
<time> ::= exponential, time in seconds of specified voltage crossing

34
:MEASure:UNITs

Command: :MEASure:UNITs {PERCent | VOLTs}
Query: :MEASure:UNITs?
Returned Format: [:MEASure:UNITs] {PERCent | VOLTs}<NL>

:MEASure:UPPer

Command: :MEASure:UPPer [<source>,]<upper_threshold>
Query: :MEASure:UPPer? [<source>]
Returned Format: [:MEASure:UPPer] <source>,<upper_threshold><NL>
Where:
<source> ::= {CHANNEL<n> | FUNCTION{1 | 2 | 3 | 4} | WAVE<nory{1 | 2 | 3 | 4}}
<upper_threshold> ::= integer, user defined upper threshold in percent or volts (selected by :MEASure:UNITs)
<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)

:MEASure:VACRms

Command: :MEASure:VACRms
Query: :MEASure:VACRms?
Returned Format: [:MEASure:VACRms] <value><NL>
Where:
<value> ::= exponential, calculated ac rms voltage

:MEASure:VAMPlitude

Command: :MEASure:VAMPlitude
Query: :MEASure:VAMPlitude?
Returned Format: [:MEASure:VAMPlitude] <value><NL>
Where:
<value> ::= exponential, difference between top and base voltages

:MEASure:VAverage

Command: :MEASure:VAverage
Query: :MEASure:VAverage?
Returned Format: [:MEASure:VAverage] <value><NL>
Where:
<value> ::= exponential, calculated average voltage
Programmer's Quick Reference Guide

<table>
<thead>
<tr>
<th>Command</th>
<th>:MEASure:VBASe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>:MEASure:VBASe</td>
</tr>
<tr>
<td>Returned Format</td>
<td>[:MEASure:VBASe] &lt;value&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td>Where:</td>
<td>&lt;value&gt; ::= exponential, voltage at base of selected waveform</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>:MEASure:VDCRms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>:MEASure:VDCRms</td>
</tr>
<tr>
<td>Returned Format</td>
<td>[:MEASure:VDCRms] &lt;value&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td>Where:</td>
<td>&lt;value&gt; ::= exponential, calculated dc rms voltage</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query</th>
<th>:MEASure:VDELta?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned Format</td>
<td>[:MEASure:VDELta?] &lt;value&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td>Where:</td>
<td>&lt;value&gt; ::= exponential, delta V value in volts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>:MEASure:VFIFty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>:MEASure:VFIFty</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>:MEASure:VMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>:MEASure:VMAX</td>
</tr>
<tr>
<td>Returned Format</td>
<td>[:MEASure:VMAX] &lt;value&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td>Where:</td>
<td>&lt;value&gt; ::= exponential, maximum voltage of selected waveform</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>:MEASure:VMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>:MEASure:VMIN</td>
</tr>
<tr>
<td>Returned Format</td>
<td>[:MEASure:VMIN] &lt;value&gt;&lt;NL&gt;</td>
</tr>
<tr>
<td>Where:</td>
<td>&lt;value&gt; ::= exponential, minimum voltage value of the selected waveform</td>
</tr>
</tbody>
</table>
### :MEASure:VPP

**Command**

`:MEASure:VPP`

**Query**

`:MEASure:VPP?`

**Returned Format**

`[:MEASure:VPP] <value><NL>`

**Where:**

`<value> ::= exponential, voltage peak to peak`

---

### :MEASure:VRELative

**Command**

`:MEASure:VRELative <percent_argument>`

**Query**

`:MEASure:VRELative?`

**Returned Format**

`[:MEASure:VRELative] <percent_argument><NL>`

**Where:**

`<percent_argument> ::= integer, Vmarker2 relative position in percent, from 0 through 100`

---

### :MEASure:VSTArt

**Command**

`:MEASure:VSTArt <vstart_argument>`

**Query**

`:MEASure:VSTArt?`

**Returned Format**

`[:MEASure:VSTArt] <vstart_argument><NL>`

**Where:**

`<vstart_argument> ::= exponential, voltage at y1marker`

---

### :MEASure:VSTOp

**Command**

`:MEASure:VSTOp <vstop_argument>`

**Query**

`:MEASure:VSTOp?`

**Returned Format**

`[:MEASure:VSTOp] <vstop_argument><NL>`

**Where:**

`<vstop_argument> ::= exponential, voltage at y2marker`

---

### :MEASure:VTIME?

**Query**

`:MEASure:VTIME? <vtime_argument>`

**Returned Format**

`[:MEASure:VTIME] <voltage><NL>`

**Where:**

`<vtime_argument> ::= real number representing the displayed time from the trigger in seconds
<voltage> ::= exponential, voltage at specified time`
Programmer's Quick Reference Guide

:MEASURE:VTOP

Command
Query
Returned Format
Where:

:MEASURE:VTOP
:MEASURE:VTOP?
[:MEASURE:VTOP] <value><NL>
,value> ::= exponential, voltage at the top of the waveform

:MEASURE:WCOMpare:ALLOWance

Command
Query
Returned Format
Where:

:MEASURE:WCOMpare:ALLOWance <allow_argument>
:MEASURE:WCOMpare:ALLOWance?
[:MEASURE:WCOMpare:ALLOWance] <allow_argument><NL>
<allow_argument> ::= real number representing number of vertical divisions of allowance, from 0.0 to 8.0

:MEASURE:WCOMpare:COMPare

Command
Query
Returned Format
Where:

:MEASURE:WCOMpare:COMPare {CHANnel<n> | FUNCTION{1 | 2 | 3 | 4}), WMEMemory{1 | 2 | 3 | 4})
:MEASURE:WCOMpare:COMPare?
[:MEASURE:WCOMpare:COMPare] {CHANnel<n> | FUNCTION{1 | 2 | 3 | 4}), WMEMemory{1 | 2 | 3 | 4}<NL>
<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)

:MEASURE:WCOMpare:DESTination

Command
Query
Returned Format
Where:

:MEASURE:WCOMpare:DESTination <source_argument>,<destination_argument>
:MEASURE:WCOMpare:DESTination? {CHANnel<n> | FUNCTION{1 | 2 | 3 | 4}) SCreen}
[:MEASURE:WCOMpare:DESTination] {CHANnel<n> | FUNCTION{1 | 2 | 3 | 4}) WMEMemory{1 | 2 | 3 | 4}
| PMemory{1 | 2) | MULTiple | OFF}<NL>
<source_argument> ::= {SCreen,OFF | PMemory{1 | 2)) | {CHANnel<n> | FUNCTION{1 | 2 | 3 | 4})
<destination_argument> ::= {WMEMemory{1 | 2 | 3 | 4) | MULTiple
| PMemory{1 | 2) | OFF}
<n> ::= integer, 1 or 2 (HP 54520C/54522C) or 1 through 4 (HP 54540C/54542C)
:MEASure:WCOMpare:HALLowance

Command
:MEASure:WCOMpare:HALLowance <allow_argument>
:MEASure:WCOMpare:HALLowance?

Returned Format
[[:MEASure:WCOMpare:HALLowance] <allow_argument><NL>

<allow_argument> ::= real number representing number of
horizontal divisions of allowance, from 0.0 to 8.0

:MEASure:WCOMpare:POSTfailure

Command
:MEASure:WCOMpare:POSTfailure {CONTinue | STOP}
:MEASure:WCOMpare:POSTfailure?

Query

Returned Format
[[:MEASure:WCOMpare:POSTfailure] {CONTinue | STOP}<NL>

:MEASure:WCOMpare:VALLowance

Command
:MEASure:WCOMpare:VALLowance <allow_argument>
:MEASure:WCOMpare:VALLowance?

Query

Returned Format
[[:MEASure:WCOMpare:VALLowance] <allow_argument><NL>

<allow_argument> ::= real number representing number of
vertical divisions of allowance, from 0.0 to 8.0

:MEASure:WCOMpare:WTESt

Command
:MEASure:WCOMpare:WTESt {MEASure | OFF}
:MEASure:WCOMpare:WTESt?

Query

Returned Format
[[:MEASure:WCOMpare:WTESt] {MEASure | OFF}<NL>

:MENU

Command
:MENU {CHANnel<n>|TIMebase | TRIGger | DISK | DISPlay | DELTa
| MATH | SAVE | MEASure | UTILITY | SHOW}

Query

Returned Format
[[:MENU] {CHANnel<n>|TIMebase | TRIGger | DISK | DISPlay | DELTa
| MATH | SAVE | MEASure | UTILITY | SHOW}

<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4
(HP 54540C/54542C)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:MERGe</td>
<td>$\text{:MERGe} \ &lt;\text{memory_num}&gt;$</td>
</tr>
<tr>
<td>Where:</td>
<td>$\text{&lt;memory_num&gt;} ::= \text{integer}, \text{1 or 2}$</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>:MMEMory:DISPlay</td>
<td>$\text{:MMEMory:DISPlay} \ {{\text{OFF} \mid 0} \mid {\text{ON} \mid 1}}$</td>
</tr>
<tr>
<td>Query</td>
<td>$\text{:MMEMory:DISPlay?}$</td>
</tr>
<tr>
<td>Returned Format</td>
<td>$\text{[:MMEMory:DISPlay]} \ {0 \mid 1}\langle\text{NL}\rangle$</td>
</tr>
<tr>
<td>Where:</td>
<td></td>
</tr>
<tr>
<td>:MMEMory:FNUMber</td>
<td>$\text{:MMEMory:FNUMber} \ &lt;\text{failure_number}&gt;$</td>
</tr>
<tr>
<td>Query</td>
<td>$\text{:MMEMory:FNUMber?}$</td>
</tr>
<tr>
<td>Returned Format</td>
<td>$\text{[:MMEMory:FNUMber]} \ &lt;\text{failure_number}&gt;\langle\text{NL}\rangle$</td>
</tr>
<tr>
<td>Where:</td>
<td>$\text{&lt;failure_number&gt;} ::= \text{integer}, \text{0 to 665}$</td>
</tr>
<tr>
<td>:MMEMory:SOURce</td>
<td>$\text{:MMEMory:SOURce} \ (\text{CHANnel&lt;n&gt; \mid FUNCTION}(1 \mid 2 \mid 3 \mid 4)$</td>
</tr>
<tr>
<td>Query</td>
<td>$\text{:MMEMory:SOURce?}$</td>
</tr>
<tr>
<td>Returned Format</td>
<td>$\text{[:MMEMory:SOURce]} \ (\text{CHANnel&lt;n&gt; \mid FUNCTION}(1 \mid 2 \mid 3 \mid 4)\langle\text{NL}\rangle$</td>
</tr>
<tr>
<td>Where:</td>
<td>$\langle\text{n}\rangle ::= \text{integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)}$</td>
</tr>
<tr>
<td>:MMEMory:STORE</td>
<td>$\text{:MMEMory:STORE} \ {\text{MMEMory}(1 \mid 2 \mid 3 \mid 4}$</td>
</tr>
<tr>
<td>:PCFRequency</td>
<td>$\text{:PCFRequency} \ &lt;\text{cal_freq}&gt;$</td>
</tr>
<tr>
<td>Query</td>
<td>$\text{PCFRequency}$</td>
</tr>
<tr>
<td>Returned Format</td>
<td>$\text{[:PCFRequency]} \ &lt;\text{cal_freq}&gt;$</td>
</tr>
</tbody>
</table>
| Where:             | $\text{<cal\_freq>} ::= \text{exponential, probe compensation signal frequency in hertz}$
:PLOT

Query

:PLOT?

:PMEMory{1|2}:CLEar

Command

:PMEMory{1|2}:CLEar

:PMEMory{1|2}:DISPlay

Command

:PMEMory{1|2}:DISPlay \{OFF | 0 \} \{ON | 1\}

Query

:PMEMory{1|2}:DISPlay?

Returned Format

[:PMEMory{1|2}:DISPlay] \{0 | 1\}<NL>

:PMEMory{1|2}:MERGe

Command

:PMEMory{1|2}:MERGe

:PMEMory{1|2}:SETup?

Query

:PMEMory{1|2}:SETup?

Returned Format

PMEMory{1|2}:DISP\{0 | 1\}<NL>

:PLOT?

Query

:PLOT?

:POWerup[RUNning | STOPped]

Command

:POWerup[RUNning | STOPped]

Query

:POWerup?

Returned Format

[:POWerup] \{STOPped | RUNning\}

41
Programmer's Quick Reference Guide

:PRINT?

Query
:PRINT?

:RUN

Command
:RUN

:SEQUENTIAL:DISPLAY

Command
:SEQUENTIAL:DISPLAY {{OFF | 0} | {ON | 1}}
Query
:SEQUENTIAL:DISPLAY?
Returned Format
[[:SEQUENTIAL:DISPLAY] {0 | 1}<NL>

:SEQUENTIAL:EXCLUDE

Command
:SEQUENTIAL:EXCLUDE <from_argument>,<to_argument>
Query
:SEQUENTIAL:EXCLUDE?
Returned Format
[[:SEQUENTIAL:EXCLUDE] <exclude_list><NL>
Where:
<from_argument> ::= integer, segment number of the lower end of the exclude range
<to_argument> ::= integer, segment number of the upper end of the exclude range
number greater than or equal to the value of <from_argument>
<exclude_list> ::= integer, a list of previously captured segment numbers separated by commas

:SEQUENTIAL:INCLUDE

Command
:SEQUENTIAL:INCLUDE <from_argument>,<to_argument>
Query
:SEQUENTIAL:INCLUDE?
Returned Format
[[:SEQUENTIAL:INCLUDE] <include_list><NL>
Where:
<from_argument> ::= integer, segment number of the lower end of the include range
<to_argument> ::= integer, segment number of the upper end of the include range
<include_list> ::= integer, a list of previously captured segment numbers separated by commas

42
:SEQUential:NPOints

Command
:SEQUential:NPOints <points_argument>

Query
:SEQUential:NPOints?

Returned Format:
[:SEQUential:NPOints] <points_argument><NL>

Where:
<points_argument> ::= integer, 4 to 32768

:SEQUential:NSEGments

Command
:SEQUential:NSEGments <segments_argument>

Query
:SEQUential:NSEGments?

Returned Format:
[:SEQUential:NSEGments] <segments_argument><NL>

Where:
<segments_argument> ::= integer, 1 to 8888 dependent on the
SEQUential:NPOints selection

:SEQUential:SETup?

Query
:SEQ:DTSP {0 | 1};
EXCL <exclude_list>;
INCL <include_list>;
NPO <points_argument>;
NSEG <segment_argument>;
SNUM <segments_argument>; (Sequential Single Shot Mode Only)
SOURCE CHANnel<n><NL> (Sequential Single Shot Mode Only)

Where:
<exclude_list> ::= integer, a list of previously captured
segment numbers separated by commas
<include_list> ::= integer, a list of previously captured
segment numbers separated by commas
<points_argument> ::= integer, 4 to 32768
<segments_argument> ::= integer, 1 to 8888 dependent on the
points selected
<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4
(HP 54540C/54542C)
:SEQuential:SNUMber

Command
:SEQuential:SNUMber <segments_argument>

Query
:SEQuential:SNUMber?

Returned Format
[:SEQuential:SNUMber] <segments_argument><NL>
<segments_argument> ::= integer, 1 to 8888 dependent on the
number of segments acquired and ACQuire:TYPE currently
selected

:SEQuential:SOURce

Command
:SEQuential:SOURce {CHANnel<n>}

Query
:SEQuential:SOURce?

Returned Format
[:SEQuential:SOURce] {CHANnel<n>}
<n> ::= integer 1 or 2 (HP 54526C/54522C), or 1 through 4
(HP 54548C/54542C)

:SEQuential:TTAGs?

Query
:SEQuential:TTAGs? <segments_argument>

Where:
<segments_argument> ::= integer, 1 to 8888 dependent on the
SEQuential:NSEGments selection

:SEQuential:TTDifference?

Query
:SEQuential:TTDifference?

Where:
<segments_argument>, <segments_argument>
<segments_argument> ::= integer, 1 to 8888 dependent on the
SEQuential:NSEGments selection

:SERial

Command
:SERial <serial_number>

Where:
<serial_number> ::= 10 character serial number within quotes
:STATus?

Query

:STATus? <display>

Returned Format

[:STATus] [0 | 1]<NL>

<display> ::= (CHANnel<n> | FUNCTION(1 | 2 | 3 | 4) |

WMEMory(1 | 2 | 3 | 4) | PMEMory(1 | 2)<NL>

<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4

(HP 54540C/54542C)

:STOP

Command

:STOP

:STORe

Command

:STORe <source>,<destination>

Where:

<source> ::= (CHANnel<n> | FUNCTION(1 | 2 | 3 | 4) |

WMEMory(1 | 2 | 3 | 4))

<destination> ::= (WMEMory(1 | 2 | 3 | 4))

<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4

(HP 54540C/54542C)

:SYSTem:DATE

Command

:SYSTem:DATE <year>,<month>,<day>

Query

:SYSTem:DATE?

Returned Format

[:SYSTem:DATE] "DDMMYYYY"<NL>

<year>/<YYYY> ::= integer, 1990 through 2059

<month> ::= integer, 1 through 12

<MMM> ::= three digit alphabetic month

<day>/<DD> ::= integer, 1 through 31

:SYSTem:DSP

Command

:SYSTem:DSP <ascii_string>

Query

:SYSTem:DSP?

Returned Format

[:SYSTem:DSP] <ascii_string><NL>

<ascii_string> ::= string response data containing the last

information written on the advisory line
:SYSTem:ERRor?

Query
:SYSTem:ERRor [{NUMBER | STRING}]
[:SYSTem:ERRor] <error>[:<ascii_string>]<NL>
<error> ::= an integer error code
<ascii_string> ::= an alpha string specifying the error condition

Returned Format

Where:

:SYSTem:HEADer

Command
:SYSTem:HEADer {{OFF | 0} | {ON | 1}}
:SYSTem:HEADer?
[[:SYSTem:HEADer] {0 | 1}<NL>]

Query

Returned Format

:SYSTem:KEY

Command
:SYSTem:KEY <key_code>
:SYSTem:KEY?
[:SYSTem:KEY] <key_code><NL>
<key_code> ::= integer, 0 through 63

Query

Returned Format

Where:

:SYSTem:LONGform

Command
:SYSTem:LONGform {{OFF | 0} | {ON | 1}}
:SYSTem:LONGform?
[[:SYSTem:LONGform] {0 | 1}<NL>]

Query

Returned Format

:SYSTem:PINMacro

Command
:SYSTem:PINMacro <ascii_string>
<ascii_string> ::= name of the defined macro

Where:
::SYStem::SETup

Command

::SYStem::SETup <block_data>

Query

::SYStem::SETup?

Returned Format

[:SYStem::SETup] <block_data><NL>

Where:

[block_data] ::= #800002048<setup_string>

[setup_string] ::= block of binary data bytes

::SYStem::TIME

Command

::SYStem::TIME <hour>,<minute>,<second>

Query

::SYStem::TIME?

Returned Format

[:SYStem::TIME] "HH:MM:SS"<NL>

Where:

<hour>/<HH> ::= integer, 0 through 23

<minute>/<MM> ::= integer, 0 through 59

<second>/<SS> ::= integer, 0 through 59

::SYStem::UTILITY

Command

::SYStem::UTILITY:GMARkers {ON | OFF}

::SYStem::UTILITY:LABELs {ON | OFF}

::SYStem::UTILITY:FACTOr {ON | OFF}

::SYStem::UTILITY:FPANel {ON | OFF}

::SYStem::UTILITY:FPANel:TIMEout {<timeout> | INFINITY}

Query

::SYStem::UTILITY:GMARkers?

::SYStem::UTILITY:LABELs?

::SYStem::UTILITY:FACTOr?

::SYStem::UTILITY:FPANel?

::SYStem::UTILITY:FPANel:TIMEout?

Returned Format

[:SYStem::UTILITY:GMARkers] {ON | OFF}<NL>

<timeout> ::= integer, 0 through 12 (hours)

::TER?

Query

::TER?

Returned Format

[:TER] {0 | 1}<NL>
### :TIMebase:DELay

**Command**

::TIMebase:DELay <delay_value>

**Query**

::TIMebase:DELay?

**Returned Format**

::TIMebase:DELay [ <delay_value> ]<NL>

**Where:**

<delay_value> ::= exponential, time from trigger to display reference in seconds

### :TIMebase:MODE

**Command**

::TIMebase:MODE {AUTO | TRIGgered | SINGLE}

**Query**

::TIMebase:MODE?

**Returned Format**

::TIMebase:MODE [ AUTO | TRIGgered | SINGLE]<NL>

### :TIMebase:RANGE

**Command**

::TIMebase:RANGE <range_value>

**Query**

::TIMebase:RANGE?

**Returned Format**

::TIMebase:RANGE [ <range_value> ]<NL>

**Where:**

<range_value> ::= exponential, 10 ns to 50 s in a 1,2,5 sequence

### :TIMebase:REForce

**Command**

::TIMebase:REForce {LEFT | CENTER | RIGHT}

**Query**

::TIMebase:REForce?

**Returned Format**

::TIMebase:REForce [ LEFT | CENTER | RIGHT]<NL>

### :TIMebase:RLENgth

**Command**

::TIMebase:RLENgth <length_argument>

**Query**

::TIMebase:RLENgth?

**Returned Format**

::TIMebase:RLENgth [ <length_argument> ]<NL>

**Where:**

<length_argument> ::= integer, 500 in the repetitive mode, 512, 1024, 2048, 4195, 8192, 16384, or 32768 in the real-time mode (sequential off), 4 to 32768 in the real-time mode (sequential on), depending on the current :SEQUential:NSEGments value
:TIMebase:SAMPLE

Command
:TIMebase:SAMPLE {REALtime | REPetitive}

Query
:TIMebase:SAMPLE?

Returned Format
[:TIMebase:SAMPLE] {REALtime | REPetitive}<NL>

:TIMebase:SAMPLE:CLOCK

Command
:TIMebase:SAMPLE:CLOCK {AUTO | <clock_value>}

Query
:TIMebase:SAMPLE:CLOCK?

Returned Format
[:TIMebase:SAMPLE:CLOCK] <clock_value><NL>

Where:
<clock_value> ::= exponential, 10S/s to 20S/s

:TIMebase:SETup?

Query
:TIMebase:SETup?

Returned Format
:TIM:DEL <delay_value>;

MODE {AUTO|TRIG|SING};

RANGE <range_value>;

RLEN <length_argument>;

REF {LEFT|CENT|RIGH};

SAMPLE {REAL|REP};

SAMP:CLOC {AUTO | <clock_value}><NL>

Where:
<delay_value> ::= exponential, time from trigger to display reference in seconds

<range_value> ::= exponential, 5ns to 50s

<length_argument> ::= 500 in the repetitive mode, 512, 1024, 2048, 4196, 8192, 16384, or 32768 in the real-time mode (sequential off), or 4 to 32768 in the real-time mode (sequential on), depending on the current SEQkuential:NSEGments value

<clock_value> ::= exponential, 10S/s to 20S/s

:TRIGger:CENTered

Command
:TRIGger:CENTered
.:TRIGger:CONDition

Command
:TRIGger:CONDition {ENTER | EXIT | TRUE | FALSE |
   GT,<gt_argument>| LT,<lt_argument> |
   RANGE,<range_gt>,<range_lt>}

Query
:TRIGger:CONDition?

Returned Format
[:TRIGger:CONDition] <argument><NL>
   <argument> ::= {ENTER | EXIT | GT,<gt_argument> |
      LT,<lt_argument> | RANGE,<range_gt>,<range_lt> | in PATTERN
      or DELay with QUALify:PATTERN selected; {TRUE | FALSE} |in
      STATE or DELay with QUALify:STATE selected; or RANGE,
      <range_gt>,<range_lt> in TV mode
   <gt_argument> ::= exponential, 20 ns to 160 ms
   <lt_argument> ::= exponential, 20 ns to 160 ms
   <range_gt> ::= exponential, 20 ns to 159.999 ms (must be less
      than <range_lt>)
   <range_lt> ::= exponential, 30 ns to 160 ms (must be greater
      than <range_gt>)

.:TRIGger:COUPling

Command
:TRIGger:COUPling {AC | DC | LFReject}

Query
:TRIGger:COUPling?

Returned Format
[:TRIGger:COUPling] {AC | DC | LFReject}<NL>

.:TRIGger:DE Lay

Command
:TRIGger:DE Lay {TIME,<time_argument> | EVENT,<event_argument>}

Query
:TRIGger:DE Lay?

Returned Format
[:TRIGger:DE Lay] {TIME,<time_argument> | EVENT,<event_argument>}

Where:
   <time_argument> ::= exponential, amount of delay from 30 ns
to
      160 ms
   <event_argument> ::= integer, number of events from 1 to
      16000000
:TRIGger:DELay:SLOPe

**Command**
:TRIGger:DELay:SLOPe {POSitive | NEGative}

**Query**
:TRIGger:DELay:SLOPe?

**Returned Format**
[:TRIGger:DELay:SLOPe] {POSitive | NEGative}<NL>

:TRIGger:DELay:SOURce

**Command**
:TRIGger:DELay:SOURce {CHANnel<n> | EXternal}

**Query**
:TRIGger:DELay:SOURce?

**Returned Format**
[:TRIGger:DELay:SOURce] {CHANnel<n> | EXternal}

**Where:**

<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)

NOTE: EXternal available only on HP 54520C/54522C

:TRIGger:FIELD

**Command**
:TRIGger:FIELD {1 | 2}

**Query**
:TRIGger:FIELD?

**Returned Format**
[:TRIGger:FIELD] {1 | 2}<NL>

:TRIGger:GLITch:CENTered

**Command**
:TRIGger:GLITch:CENTered

:TRIGger:GLITch:HOLDoff

**Command**
:TRIGger:GLITch:HOLDoff <holdoff_time>

**Query**
:TRIGger:GLITch:HOLDoff?

**Returned Format**
[:TRIGger:GLITch:HOLDoff] <holdoff_time><NL>

**Where:**

<holdoff_time> ::= exponential, 40 ns to 320 ns rounded to nearest 20 ns increment
:TRIGger:GLITch:LEVEL

Command
:TRIGger:GLITch:LEVEL <level_argument>
Query
:TRIGger:GLITch:LEVEL?
Returned Format
[:TRIGger:GLITch:LEVEL] <level_argument><NL>
<level_argument> ::= exponential, for internal triggers, ±1.5
x
full-scale voltage from center screen, for external
triggers
(HP 54520C/54522C) ±2 volts with probe attenuation at 1:1

:TRIGger:GLITch:SOURce

Command
:TRIGger:GLITch:SOURce (CHANnel<n> | EXTERNAL),{HIGH | LOW}
Query
:TRIGger:GLITch:SOURce?
Returned Format
[:TRIGger:GLITch:SOURce] (CHANnel<n> | EXTERNAL),{HIGH | LOW}<NL>
<n> ::= integer 1 or 2 (HP 54520C/54522C), or 1 through 4
(HP 54540C/54542C)
NOTE: EXTERNAL available only on HP 54520C/54522C

:TRIGger:GLITch:WIDth

Command
:TRIGger:GLITch:WIDth (GT | LT),<width_argument>
Query
:TRIGger:GLITch:WIDth?
Returned Format
[:TRIGger:GLITch:WIDth] (GT | LT),<width_argument><NL>
<width_argument> ::= exponential, 5 ns to 160 ms

:TRIGger:HOLDoff

Command
:TRIGger:HOLDoff {{TIME,<holdoff_time>} | 
    { EVENT,<event_argument>}}
Query
:TRIGger:HOLDoff?
Returned Format
[:TRIGger:HOLDoff] {TIME,<holdoff_value> | 
    EVENT,<event_argument}><NL>
<holdoff_time> ::= exponential, 40 ns to 320 ms
<event_argument> ::= integer, 1 to 16000000
:TRIGger:LEVel

Command
:TRIGger:LEVel <level_argument>
:TRIGger:LEVel?

Returned Format
[:TRIGger:LEVel] <level_argument><NL>

Where:
[level_argument] := for internal triggers, ±1.5 x full-scale voltage from center screen; for external triggers, ±2 volts with probe attenuation at 1:1

:TRIGger:LINE

Command
:TRIGger:LINE <line_number>
:TRIGger:LINE?

Returned Format
[:TRIGger:LINE] <line_number><NL>

Where:
[line_number] := integer, 1 to 625 (depends on STANDARD and FIELD selection)

:TRIGger:LOGic

Command
:TRIGger:LOGic {HIGH | LOW | DONTcare}
:TRIGger:LOGic?

Returned Format
[:TRIGger:LOGic] {HIGH | LOW | DONTcare}<NL>

:TRIGger:MODE

Command
:TRIGger:MODE {EDGE | PATTERN | STATe | DELay | TV | GLITCh}
:TRIGger:MODE?

Returned Format
[:TRIGger:MODE] {EDGES | PATTERN | STATe | DELay | TV | GLITCh}<NL>

:TRIGger:NREJect

Command
:TRIGger:NREJect {OFF | 0} | {ON | 1})
:TRIGger:NREJect?

Returned Format
[:TRIGger:NREJect] {0 | 1}<NL>
Programmer's Quick Reference Guide

**:TRIGger:OCCurrence**

**Command**

`:TRIGger:OCCurrence <occurrence_argument>

**Query**

`:TRIGger:OCCurrence?

**Returned Format**

`[:TRIGger:OCCurrence] <occurrence_argument><NL>
<occurrence_argument> ::= integer, 1 to 16000000`

**:TRIGger:OCCurrence:SLOPe**

**Command**

`:TRIGger:OCCurrence:SLOPe {POSitive | NEGative} [E]xternal]

**Query**

`:TRIGger:OCCurrence:SLOPe?

**Returned Format**

`[:TRIGger:OCCurrence:SLOPe] {POSitive | NEGative}<NL>

**:TRIGger:OCCurrence:SOURce**

**Command**

`:TRIGger:OCCurrence:SOURce {CHANnel<n> | EXTernal}

**Query**

`:TRIGger:OCCurrence:SOURce?

**Returned Format**

`[:TRIGger:OCCurrence:SOURce] {CHANnel<n> | EXTernal}<NL>
<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)`

**NOTE:** EXTernal available only on HP 54520C/54522C

**:TRIGger:PATH**

**Command**

`:TRIGger:PATH {CHANnel<n> | EXTernal}

**Query**

`:TRIGger:PATH?

**Returned Format**

`[:TRIGger:PATH] {CHANnel<n> | EXTernal}<NL>
<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)`

**NOTE:** EXTernal available only on HP 54520C/54522C

**:TRIGger:POLarity**

**Command**

`:TRIGger:POLarity {POSitive | NEGative} [E]xternal]

**Query**

`:TRIGger:POLarity?

**Returned Format**

`[:TRIGger:POLarity] {POSitive | NEGative}<NL>`
:TRIGger:QUALify

Command Query Returned Format

:TRIGger:QUALify {{EDGE | PATTERN | STATE} | (LOW | HIGH)}
:TRIGger:QUALify?
[:TRIGger:QUALify] {{EDGE | PATTERN | STATE} | (LOW | HIGH)}<NL>

:TRIGger:SETup?

Query Returned Format

:TRIGger:SETup?
:TRIG:MODE EDGE;
COUP {AC | DC | LFR};
HOLD {{TIME,<holdoff_time>} | EVENT, <event_argument>}};
LEV <level_argument>;
NREJ {0 | 1};
SLOP {POS | NEG};
SOUR {CHAN<n> | EXT | LINE | AUX}<NL> (EXT is only valid for the 54620/22)

:TRIG:MODE PATT;
COND {ENT | EXIT | GT,<gt_argument> | LT,<lt_argument> | RANGE,<range_gt>,<range_lt>};
HOLD {{TIME,<holdff_time>} | EVENT, <event_argument>}};
LEV <level_argument>;
LOG {HIGH | LOW | DONT};
NREJ {0 | 1};
PATH {CHAN<n> | EXT}<NL> (EXT is only valid for the 54620/22)

:TRIG:MODE STAT;
COND {TRUE | FALS};
HOLD {{TIME,<holdoff_time>} | {EVEN, <event_argument>}};
LEV <level_argument>;
LOG {HIGH | LOW | DONT};
NREJ {0 | 1};
PATH {CHAN<n> | EXT}; (EXT is only valid for the 54620/22)
SLOP {POS | NSG};
SOUR {CHAN<n> | EXT}<NL> (EXT is only valid for the 54620/22)
:TRIG:MODE DEL;
COND {ENT | EXIT | TRUE | FALS | GT,<gt_argument> | LT,<lt_argument> | RANG,<range_gt>,<range_lt>};
DEL {{TIME,<time_value>} | {EVEN, <event_value>}};
DEL:SLOP {POS | NEG};
DEL:SOUR {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
LEV <level_argument>;
LOG {HIGH | LOW | DONT};
NREJ {0 | 1};
OCC <occurrence_argument>;
OCC:SLOP {POS | NEG};
OCC:SOUR {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
PATH {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
QUAL {EDGE | PATT | STAT};
SLOP {POS | NEG};
SOUR {CHAN<n> | EXT}<NL> (EXT is only valid for the 54520/22)

:TRIG:MODE TV;
COND {RANG,<range_gt>,<range_lt>};
FIEL {1 | 2};
HOLD {{TIME,<holdoff_time>} | {EVEN, <event_argument>}};
LEV <level_argument>;
LINE <line_number>;
NREJ {0 | 1};
OCC <occurrence_argument>;
OCC:SLOP {POS | NEG};
POL {POS | NEG};
QUAL {LOW | HIGH};
SOUR {CHAN<n> | EXT}; (EXT is only valid for the 54520/22)
STAN {525 | 625 | USER}<NL>

:TRIG:MODE GLIT;
GLITCH:HOLD TIME,<holdoff_time>;
GLITCH:LEV <level_argument>;
GLITCH:NREJ {0 | 1};
GLITCH:SOUR {CHAN<n> | EXT}, {LOW | HIGH}; (EXT is only valid for the 54520/22)
GLITCH:WIDTH {GT | LT}, <width_argument><NL>
Where:

\(<n> \) ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)
\(<\text{gt}\_\text{argument}> \) ::= exponential, 20 ns to 160 ms
\(<\text{lt}\_\text{argument}> \) ::= exponential, 20 ns to 160 ms
\(<\text{range}\_\text{gt}> \) ::= exponential, 20 ns to 159.999 ms (must be less than \(<\text{range}\_\text{lt}>\))
\(<\text{range}\_\text{lt}> \) ::= exponential, 30 ns to 160 ms (must be greater than \(<\text{range}\_\text{gt}>\))
\(<\text{time}\_\text{value}> \) ::= exponential, amount of delay from 30ns to 160ms
\(<\text{event}\_\text{value}> \) ::= integer, number of events from 1 to 16000000
\(<\text{holdoff}\_\text{time}> \) ::= exponential, 40 ns to 320 ms rounded to nearest 20 ns increment
\(<\text{level}\_\text{argument}> \) ::= exponential, trigger level in volts
\(<\text{width}\_\text{argument}> \) ::= exponential, 5 ns to 160 ms
\(<\text{event}\_\text{argument}> \) ::= integer, 1 to 16000000
\(<\text{line}\_\text{number}> \) ::= integer, 1 to 625 (depends on STANDard and FIELd selection)
:VIEW

Command
:VIEW <display>
Where:
<display> ::= CHANnel<n> | FUNCtion{1 | 2 | 3 | 4} |
          PMEMory{1 | 2} | WMEMory{1 | 2 | 3 | 4})
<n> ::= integer, 1 or 2 (HP 54520C/54522C), or 1 through 4
       (HP 54540C/54542C)
<n> ::= integer 1 or 2 (HP 54520C/54522C)

:WAVEform:DATA

Command
:WAVEform:DATA <block_data>
Query
:WAVEform:DATA?
Returned Format
[:WAVEform:DATA] <block_data><NL>
Where:
[block_data] ::= definite block data in IEEE 488.2 # format

:WAVEform:FORMAT

Command
:WAVEform:FORMAT {ASCII | WORD | BYTE | COMPRESSED}
Query
:WAVEform:FORMAT?
Returned Format
[:WAVEform:FORMAT] {ASCII | WORD | BYTE | COMPRESSED}<NL>

:WAVEform:POINTS

Query
:WAVEform:POINTS?
Returned Format
[:WAVEform:POINTS] {512 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768}<NL>
:WAVeform:PREamble

Command
:WAVeform:PREamble <preamble_data>
:WAVeform:PREamble?
[:WAVeform:PREamble] <preamble block><NL>
<preamble_data> ::= <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
    4 for COMPressed format
<type> ::= 0 for INVALID type
    1 for NORMAL type or REALTIME
    2 for AVERAGE type
    3 for ENVELOPE type
    4 for RAWDATA type
    5 for PDFTect type

:WAVeform:SOURce

Command
:WAVeform:SOURce [CHANnel<n> | WMEMory{1 | 2 | 3 | 4}]
:WAVeform:SOURce?
[:WAVeform:SOURce] [CHANnel<n> | WMEMory{1 | 2 | 3 | 4}]<NL>
<n> ::= integer 1 or 2 (HP 54520C/54522C), or 1 through 4
    (HP 54540C/54542C)

:WAVeform:TYPE?

Query
:WAVeform:TYPE?
[:WAVeform:TYPE] (INVALID | AVERAGE | ENVELOPE | NORMAL |
    PDFTect | RAWData)<NL>

:WAVeform:XINCrement?

Query
:WAVeform:XINCrement?
[:WAVeform:XINCrement] <value><NL>
<value> ::= exponential, x-increment in the current preamble
### :WAVEform:XORigin?

<table>
<thead>
<tr>
<th>Query</th>
<th>Returned Format</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WAVEform:XORigin?</td>
<td>[:WAVEform:XORigin] &lt;value&gt;[,&lt;value&gt;]...&lt;NL&gt;</td>
<td>&lt;value&gt; ::= exponential, x-origin value in the current preamble</td>
</tr>
</tbody>
</table>

### :WAVEform:XREFerence?

<table>
<thead>
<tr>
<th>Query</th>
<th>Returned Format</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WAVEform:XREFerence?</td>
<td>[:WAVEform:XREFerence] &lt;value&gt;&lt;NL&gt;</td>
<td>&lt;value&gt; ::= integer, x-reference value in the current preamble</td>
</tr>
</tbody>
</table>

### :WAVEform:YINCrement?

<table>
<thead>
<tr>
<th>Query Syntax</th>
<th>Returned Format</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WAVEform:YINCrement?</td>
<td>[:WAVEform:YINCrement] &lt;value&gt;&lt;NL&gt;</td>
<td>&lt;value&gt; ::= exponential, y-increment value in the current preamble</td>
</tr>
</tbody>
</table>

### :WAVEform:YORigin?

<table>
<thead>
<tr>
<th>Query</th>
<th>Returned Format</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WAVEform:YORigin?</td>
<td>[:WAVEform:YORigin] &lt; value&gt;&lt;NL&gt;</td>
<td>&lt;value&gt; ::= exponential, y-origin in the current preamble</td>
</tr>
</tbody>
</table>

### :WAVEform:YREFerence?

<table>
<thead>
<tr>
<th>Query</th>
<th>Returned Format</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WAVEform:YREFerence?</td>
<td>[:WAVEform:YREFerence] &lt;value&gt;&lt;NL&gt;</td>
<td>&lt;value&gt; ::= integer, y-reference value in the current preamble</td>
</tr>
</tbody>
</table>

### :WMEMory{1 | 2 | 3 | 4}:DISPlay

<table>
<thead>
<tr>
<th>Command</th>
<th>Query</th>
<th>Returned Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WMEMory{1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
:WMEMemory[1 2 3 4]:GET

Command
:WMEMemory[1 2 3 4]:GET {CHANNEL<n> | WMEMemory[1 2 3 4] FUNCTION[1 2 3 4]}

Where:
<n> ::= integer 1 or 2 (HP 54520C/54522C), or 1 through 4 (HP 54540C/54542C)

:WMEMemory[1 2 3 4]:PROTect

Command
:WMEMemory[1 2 3 4]:PROTect {{OFF | 0} | {ON | 1}}

Query
:WMEMemory[1 2 3 4]:PROTect?

Returned Format
[:WMEMemory[1 2 3 4]:PROTect] (0 | 1)<NL>

:WMEMemory[1 2 3 4]:SETup?

Query
:WMEMemory[1 2 3 4]:SETup?

Returned Format
:WMEMemory[1 2 3 4]:DISP (0 | 1);
PROT (0 | 1);
XOFF <offset_argument>;
XRANG <range_argument>;
YOFF <offset_argument>;
YRANG <range_argument><NL>

Where:
+offset_argument> ::= exponential, 5 ns to 50 sec
+range_argument> ::= exponential, time from trigger to display reference in seconds
+offset> ::= exponential, offset value in volts
+range> ::= exponential, full-scale range value

:WMEMemory[1 2 3 4]:XOFFSET

Command
:WMEMemory[1 2 3 4]:XOFFSET <offset_argument>

Query
:WMEMemory[1 2 3 4]:XOFFSET?

Returned Format
[:WMEMemory[1 2 3 4]:XOFFSET] <offset_argument><NL>

Where:
+offset_argument> ::= exponential, time from trigger to the on screen delay reference point. The maximum value depends on the :WMEMemory:XRange setting
**Programmer's Quick Reference Guide**

---

**:WMEMory{1 | 2 | 3 | 4}:XRANge**

**Command**

:WMEMory{1 | 2 | 3 | 4}:XRANge <range_argument>

**Query**

:WMEMory{1 | 2 | 3 | 4}:XRANge?

**Returned Format**

[:WMEMory{1 | 2 | 3 | 4}:XRANge] <range_argument><NL>

**Where:**

<range_argument> ::= exponential, 5 ns to 50 s in a 1,2,5 sequence

---

**:WMEMory{1 | 2 | 3 | 4}:YOFFset?**

**Query**

:WMEMory{1 | 2 | 3 | 4}:YOFFset?

**Returned Format**

[:WMEMory{1 | 2 | 3 | 4}:YOFFset] <yoffset><NL>

**Where:**

<yoffset> ::= exponential, offset value in volts

---

**:WMEMory{1 | 2 | 3 | 4}:YRANge?**

**Query**

:WMEMory{1 | 2 | 3 | 4}:YRANge?

**Returned Format**

[:WMEMory{1 | 2 | 3 | 4}:YRANge] <yrange><NL>

**Where:**

<yrange> ::= exponential, full-scale range value

---

62
© Copyright Hewlett-Packard Company 1993-1996
All Rights Reserved.

Reproduction, adaptation, or translation without prior
written permission is prohibited, except as allowed
under the copyright laws.

Document Warranty
The information contained in
this document is subject to
change without notice.

Hewlett-Packard makes
no warranty of any kind
with regard to this
material, including, but
not limited to, the implied
warranties of
merchantability or fitness
for a particular purpose.

Hewlett-Packard shall not be
liable for errors contained
herein or for damages in
connection with the
furnishing, performance, or
use of this material.

Safety
This apparatus has been
designed and tested in
accordance with IEC
Publication 348, Safety
Requirements for Measuring
Apparatus, and has been
supplied in a safe condition.
This is a Safety Class I
instrument (provided with
terminal for protective
earthing). Before applying
power, verify that the correct
safety precautions are taken
(see the following warnings).
In addition, note the external
markings on the instrument
that are described under
"Safety Symbols."

Warning
- Before turning on the
  instrument, you must connect
  the protective earth terminal
  of the instrument to the
  protective conductor of the
  (mains) power cord. The
  mains plug shall only be
  inserted in a socket outlet
  provided with a protective
  earth contact. You must not
  negate the protective action
  by using an extension cord
  (power cable) without a
  protective conductor
  (grounding). Grounding one
  conductor of a two-conductor
  outlet is not sufficient
  protection.
- Only fuse the
  required rated current,
  voltage, and specified type
  (normal blow, time delay,
  etc.) should be used. Do not
  use repaired fuses or
  short-circuited fuseholders.
  To do so could cause a shock
  of fire hazard.
- Service instructions are for
  trained service personnel. To
  avoid dangerous electric
  shock, do not perform any
  service unless qualified to do
  so. Do not attempt internal
  service or adjustment unless
  another person, capable of
  rendering first aid and
  resuscitation, is present.
- If you energize this
  instrument by an auto
  transformer (for voltage
  reduction), make sure the
  common terminal is
  connected to the earth
  terminal of the power source.
- Whenever it is likely that
  the ground protection is
  impaired, you must make the
  instrument inoperative
  and secure it against any
  unintended operation.
- Do not operate the
  instrument in the presence of
  flammable gasses or fumes.
  Operation of any electrical
  instrument in such an
  environment constitutes a
  definite safety hazard.
- Do not install substitute
  parts or perform any
  unauthorized modification to
  the instrument.
- Capacitors inside the
  instrument may retain a
  charge even if the instrument
  is disconnected from its
  source of supply.
- Use caution when exposing
  or handling the flat panel
  display. Handling or
  replacing the display shall be
done only by qualified
  maintenance personnel.

Safety Symbols

⚠️
Instruction manual symbol: the product is marked with
this symbol when it is
necessary for you to refer to
the instruction manual in
order to protect against
damage to the product.

Electric shock hazard symbol:

ヶ
Hazardous voltage symbol.

ヶ
Earth terminal symbol: Used
to indicate a circuit common
connected to grounded
chassis.

WARNING
The Warning sign denotes a
hazard. It calls attention to a
procedure, practice, or the
like, which, if not correctly
performed or adhered to,
could result in personal
injury. Do not proceed
beyond a Warning sign until
the indicated conditions are
fully understood and met.

CAUTION
The Caution sign denotes a
hazard. It calls attention to an
operating procedure,
practice, or the like, which, if
not correctly performed or
adhered to, could result in
damage to or destruction of
part or all of the product. Do
not proceed beyond a
Caution symbol until the
indicated conditions are fully
understood or met.
Product Warranty
This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of three years from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective. For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard. For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country. Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument software, or firmware will be uninterrupted or error free.

Limitation of Warranty
The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. Hewlett-Packard specifically disclaims the implied warranties of merchantability or fitness for a particular purpose.

Exclusive Remedies
The remedies provided herein are the buyer's sole and exclusive remedies. Hewlett-Packard shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Assistance
Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products. For any assistance, contact your nearest Hewlett-Packard Sales Office.

Certification
Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

About this edition
This is the first edition of the HP 54520C and 54540C Series Oscilloscope Programmer's Quick Reference Guide.

Publication number 54542-97042
Printed in USA.
Edition dates are as follows:
First edition, November 1995
New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by you. The dates on the title page change only when a new edition is published.
A software or firmware code may be printed before the date. This code indicates the version level of the software or firmware of this product at the time the manual or update was issued. Many product updates do not require manual changes; and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

The following list of pages gives the date of the current edition and of any changed pages to that edition.

All pages original edition