

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

PRELIMINARY PROGRAMMING INFORMATION

FOR

HP 16500A

Logic Analysis System

c Copyright Hewlett-Packard Company 1987

Printed in U.S.A. December 1987

NOTICE:

This programming reference document contains preliminary data. The programming instructions contained in this document are a subset of the complete command set. Changes between this preliminary command set and the final command set (when released) are documented within.

Table of Contents

1 HP-IB Command Set

1 I) System Command List

3 II) Disc System Command List

4 III) Pattern Generator Command List

6 IV) Oscilloscope Command List

12 V) Detailed Examples

15 Appendix A -- HP 16500A SETUP Command and Configuration Block Data Definition

16 Appendix B -- Definition of SYMBOLS Sections for the HP 16500A Modules

18 Appendix C -- HP 16500A SYSTEM SETUP Command Learn String

24 Appendix D -- HP 16510A STATE/TIMING ANALYZER SETUP Command Learn String

46 Appendix E -- HP 16520A PATTERN GENERATOR SETUP Command Learn String

50 Appendix F -- HP 16530A OSCILLOSCOPE SETUP Command Learn String

HP-IB Command Subset

I) System Command List

System commands allow the user access to the main features of the HP 16500A including data/setup retrieval and downloading, error querying, key access, and run control. At introduction, the majority of the system commands have been included into the HP 16500A system. The following pages describe the commands now available.

SELECT

The SELECT command specifies the current module that HP-IB commands are to be directed.

```
Command Syntax :  SELect <module_number>
                         <module_number> ::= { 0 - System and Intermodule,
                                               1 - Module A,
                                               2 - Module B,
                                               3 - Module C,
                                               4 - Module D,
                                               5 - Module E }
Example        :  OUTPUT 707,"SELECT 3"
```

TYPE

The TYPE command selects the RUN mode. If the TYPE is set to SINGLE, then any START (RUN) command received will cause the module or group of modules to acquire data just one time, then halt. If the TYPE is set to REPETITIVE, then the module or group of modules will restart itself after the run has completed and continue to reacquire and redisplay data until a STOP command is received.

```
Command Syntax :  TYPE <SINGle|REPetitive>
Example        :  OUTPUT 707,"TYPE SINGLE"
```

START

The START command is used to signal a module or group of modules to begin acquiring data. The data is acquired in a manner defined by the TYPE mode. If the TYPE is SINGLE, the START command will cause the module or group of modules to run only once, and then display the acquired data. If the TYPE is set to REPETITIVE, then the START command will cause the module or group of modules to run repetitively.

```
Command Syntax  : STARt
Example         : OUTPUT 707,"START"
```

STOP

The STOP command causes the module or group of modules to stop acquiring data.

```
Command Syntax   : STOP
Example          : OUTPUT 707,"STOP"
```

HEADer

The HEADER command tells the instrument whether or not to output a header for query responses. When HEADER is set to ON, query responses will include the command header.

The HEADER query returns the state of the HEADER command.

```
Command Syntax   : HEADer { {ON|1} | {0|OFF} }
Example          : OUTPUT 707,"HEADER ON"

Query Syntax     : HEADer?
Returned Format  : [:HEADer]{1 | 0}
```

LONGform

The LONGFORM command sets the long form for the HP 16500A's responses to queries. If the LONGFORM command is set to OFF, command headers and alpha arguments are sent from the HP 16500A in abbreviated form. If the LONGFORM command is set ON, the whole word will be output. This command does not affect the input data messages to the HP 16500A. Headers and arguments may be sent to the HP 16500A in either long form or short form regardless of how the LONGFORM command is set.

The LONGFORM query returns the state of the LONGFORM command.

```
Command Syntax   : LONGform { {ON|1} | {0|OFF} }
Example          : OUTPUT 707,"LONGFORM ON"

Query Syntax     : LONGform?
Returned Format  : [:LONGform]{1 | 0}
```

SETUP

The SETUP command is used to configure an entire module in one command. The query returns the entire configuration for the SELECTed module or system. Appendix A through F describes the format of the SETUP command and the BLOCKS OF CONFIGURATION DATA (learn strings) for all available modules.

```
Command Syntax   : SETUP <BLOCK OF CONFIGURATION DATA>
Example          : OUTPUT 707,"SETUP"<BLOCK OF CONFIGURATION DATA>

Query Syntax     : SETUP?
Example          : OUTPUT 707,"SETUP?"
```

II) Disc System Command List

All disc commands in version V01.00 are preceded by DISCn where n is 0 for the rear disc and 1 for the front disc. These disc commands will be modified slightly in subsequent programming releases and other commands will be added. Filenames may be up to 10 characters in length with no embedded blanks and must start with a letter. Only letters, numbers and the underscore may be used in filenames. The file descriptor is used to describe the contents of the file and may contain up to 32 characters.

LOAD

The LOAD command instructs the HP 16500A to load all the configuration files specified by <filename> into their appropriate module(s).

```
Command Syntax   : DISCn:LOAD <filename>,"ALL"
Example          : OUTPUT 707,"DISC1:LOAD 'MYFILE__','ALL'"
```

STORE

The STORE disc command instructs the HP 16500A to store a configuration file named <filename> with the appropriate slot specifer and file description <file descriptor> for each module.

```
Command Syntax   : DISCn:STORE "<filename>","<file descriptor>","ALL"
Example          : OUTPUT 707,"DISC0:STORE 'CONFIG','FULL CONFIG','ALL'"
```

III) Pattern Generator Command List

LISTING:DELETE

The LISTing:DELETE commands deletes a pattern generator program listing.

```
Command Syntax   : LISTing:DELete ALL
Example          : OUTPUT 707,"LISTING:DELETE ALL"
```

LISTING:PROGRAM

The LISTing:PROGram command adds a single line to a pattern generator program listing.

```
Command Syntax   : LISTing:PROGram <line_number> , <instruction>
                   [ , opcode_parameter> ]  [ , <pattern> ]

                   <line_number> ::= line number to be programmed.

                   <instruction> ::= pattern generator instruction field
                                     { REPeat|
                                       NOOP|
                                       WIMB|
                                       WAIT|
                                       BREak|
                                       SIGNal}
                   <opcode_parameter> ::= optional repeat count for REPEAT
                                          and WAIT instructions,

                   <pattern> is a string that specifies the fields for the
                           instruction. The format of this field is as follows:

                                     "integer" to specify base 10.
                                     "#Bbbbbbbbb" where b is 0 or 1 for binary
                                     "#Ooooooooo" where o is 0-7 for octal
                                     "#Hhhhhhhhh" where h is 0-F for hex


Example:           OUTPUT 707,"LISTING:PROGRAM 1,NOOP,"3","5"
                   OUTPUT 707,"LISTING:PROGRAM 2,REPEAT,100,"3","5"
```

The <opcode_parameter> for the REPEAT instruction specifies the number of times the line is to be repeated. The <opcode_parameter> for the WAIT instruction specifies which external inputs are to be waited on. The <opcode_parameter> is an integer from 0 to 255 where a zero in a bit position means CONT and 1 means WAIT defined as follows:

WAIT Parameter BITS 2 1 0	Bit Position
0 0 0	0
0 0 1	4
0 1 0	2
0 1 1	6
1 0 0	1
1 0 1	5
1 1 0	3
1 1 1	7

NOTE: The definition of the <opcode_parameter> for the WAIT instruction may change in the next revision.

Future Changes: Macro programs will be able to be specified and Macros will be able to be invoked.

The <pattern> specifier will be expanded to allow for the "auto-fill" feature of the pattern generator. An "X" character in the pattern specifier will indicate that a bit should be auto-filled from the previous instruction.

IV) Oscilloscope Command List

The symbol <nrf> shown in the returned format for certain queries represents the numeric response for those queries.

MEASURE:SOURCE

The MEASURE:SOURCE commands specifies the scope channel from which scope measurements are to be taken.

Command syntax: MEASure:SOURce<channel_number>
Example : OUTPUT 707;"MEASURE:SOURCE CHANNEL1"

MEASURE:RISETIME

The MEASURE:RISETIME query returns the risetime measurement by finding the 10% and 90% points of the first rising edge from the the channel specified by the SOURCE command.

Query Syntax : MEASure:RISetime?
Returned Format : [:MEASure:RISetime] <nrf>

MEASURE:FREQUENCY

The MEASURE:FREQUENCY query returns the frequency measurement from the channel specified by the SOURCE command. This measurement finds the fifty percent points, locates the first and third edges on the screen, and takes the time difference between them. Inverting this gives the frequency.

Query Syntax : MEASure:FREQuency?
Returned Format : [:MEASure:FREQuency] <nrf>

MEASURE:NWIDTH

The MEASURE:NWIDTH query returns the negative pulse width measurement from the channel specified by the SOURCE command. The measurement is made between the 50% points of the first falling and the next rising edge.

Query Syntax : MEASure:NWIDth?
Returned Format : [:MEASure:NWIDth] <nrf>

MEASURE:OVERSHOOT

The MEASURE:OVERSHOOT query returns the overshoot measurement from the channel specified by the SOURCE command. The measurement is made by finding a distortion which follows the first major transition on the screen.

Query Syntax : MEASure:OVERshoot?
Returned Format : [:MEASure:OVERshoot] <nrf>

MEASURE:PERIOD

The MEASURE:PERIOD query returns the period measurement from the channel specified by the SOURCE command. This measurement finds the 50% points, locates the first and third edges on the screen, and takes the time difference between them.

Query Syntax : MEASure:PERiod?
Returned Format : [:MEASure:PERiod] <nrf>

MEASURE:PRESHOOT

The MEASURE:PRESHOOT query returns the preshoot measurement from the channel specified by the SOURCE command. The measurement is made by finding a distortion which precedes the first major transition on the screen.

Query Syntax : MEASure:PREShoot?
Returned Format : [:MEASure:PREShoot] <nrf>

MEASURE:PWIDTH

The MEASURE:PWIDTH query returns the positive pulse width measurement from the channel specified by the SOURCE command. The measurement is made between the 50% points of the first rising and the next falling edge.

Query Syntax : MEASure:PWIDth?
Returned Format : [:MEASure:PWIDth] <nrf>

MEASURE:FALLTIME

The MEASURE:FALLTIME query returns the falltime measurement by finding the 10% and 90% points of the first falling edge from the the channel specified by the SOURCE command.

Query Syntax : MEASure:FALLtime?
Returned Format : [:MEASure:FALLtime] <nrf>

MEASURE:VAMPLITIDE

The MEASURE:VAMPLITUDE query returns the voltage measurement by finding the relative maximum and minimum points from the channel specified by the SOURCE command.

Query Syntax : MEASure:VAMPlitude?
Returned Format : [:MEASure:VAMPlitude] <nrf>

MEASURE:VBASE

The MEASURE:VBASE query returns the voltage at the base (relative min) from the channel specified by the SOURCE command.

Query Syntax : MEASure:VBASe?
Returned Format : [:MEASure:VBASe] <nrf>

MEASURE:VMAX

The MEASURE:VMAX query returns the absolute maximum voltage from the channel specified by the SOURCE command.

Query Syntax : MEASure:VMAX?

Returned Format : [:MEASure:VMAX] <nrf>

MEASURE:VMIN

The MEASURE:VMIN query returns the absolute minumum voltage from the channel specified by the SOURCE command.

Query Syntax : MEASure:VMIN?
Returned Format : [:MEASure:VMIN] <nrf>

MEASURE:VPP

The MEASURE:VPP query returns the peak-to-peak voltage measurement from the channel specified by the SOURCE command.

Query Syntax : MEASure:VPP?
Returned Format : [:MEASure:VPP] <nrf>

MEASURE:VTOP

The MEASURE:VTOP query returns the voltage at the top (relative max) from the channel specified by the SOURCE command.

Query Syntax : MEASure:VTOP?
Returned Format : [:MEASure:VTOP] <nrf>

MARKER:XTIME

The MARKER:XTIME query returns the X marker to trigger time measurement.

Query Syntax : MARKer:XTIMe?
Returned Format : [:MARKer:TX] <nrf>
Future Change : In the next revision the Returned format will be
[:MARKER:XTIMe] <nrf>

MARKER:OTIME

The MARKER:OTIME query returns the O marker to trigger time measurement.

```
Query Syntax    : MARKer:OTIMe?
Returned Format : [:MARKer:TO] <nrf>
Future Change   : In the next revision the Returned format will be
                  [:MARKER:OTIMe] <nrf>
```

TIMEBASE:RANGE

This command defines the full screen diameter. The Query returns the current time range setting.

```
Command Syntax  : TIMebase:RANGe <nrf>
Query Syntax    : TIMebase:RANGe?
Returned Format : [:TIMebase:RANGe] <nrf>
```

TIMEBASE:DELAY

The DELAY command assigns the time between trigger and the center of the screen if the trigger events count is zero. If the trigger events count is non-zero, the center of the screen is the trigger events plus the delay time. The Query returns the current time delay setting.

```
Command Syntax  : TIMebase:DELay <nrf>
Query Syntax    : TIMebase:DELay?
Returned Format : [:TIMebase:DELay] <nrf>
```

TRIGGER:CONDITION

This command specifies whether the trigger is generated on entry to the specified logic pattern or when exiting it in the PATTERN trigger mode. The query returns the current condition setting.

If the CONDITION selected was ENTER, a trigger will be generated on the first transition that makes the pattern specification for every channel to be true. With the EXIT condition, a trigger will be generated on the first transition that causes the pattern specification to be false, after the pattern was true once.

```
Command Syntax  : TRIGger:CONDition ENTer/EXIt
Query Syntax    : TRIGger:CONDition?
Returned Format : [TRIGger:CONDition] ENTer/EXIt
```

TRIGGER:LEVEL

This command sets the trigger voltage level for the selected SOURCE or PATH. The query returns the trigger level of the current trigger source or path. This command cannot be used in IMMEDIATE trigger mode. With EDGE trigger mode, trigger source is used; with PATTERN mode, trigger path is used for the source of the trigger level.

Command Syntax : TRIGger:LEVel <nrf>
Query Syntax : TRIGger:LEVel?
Returned Format : [TRIGger:LEVel] <nrf>

TRIGGER:LOGIC

This command is used to specify the relation between the signal and the predefined voltage level that must exist before that part of the pattern is considered valid. HIGH indicates a requirement for an input of the selected source or path to be greater than its own trigger level and LOW indicates a requirement for an input of the selected source or path to be less than its own trigger level.

This LOGIC command can be used only in the PATTERN trigger mode and the pattern of the previously selected trigger path will be modified as specified.

The query returns the current condition of the previously selected trigger source or path.

Command Syntax : TRIGger:LOGIC LOW/HIGH/DONTcare
Query Syntax : TRIGger:LOGIC?
Returned Format : [TRIGger:LOGIC] LOW/HIGH/DONTcare

TRIGGER:MODE

This command allows you to select the trigger mode. The EDGE mode will trigger an oscilloscope on an edge whose slope is determined by the SLOPE command at a voltage determined by the LEVEL command. The PATTERN mode will trigger on entering or exiting a specified pattern of all internal channels and the external trigger. IN the IMMEDIATE mode, the oscilloscope will trigger by itself.

The MODE query will return the current trigger mode.

Command Syntax : TRIGger:MODE EDGE/PATTern/IMMediate
Query Syntax : TRIGger:MODE?
Returned Format : [TRIGger:MODE] EDGE/PATTern/IMMediate

TRIGGER:PATH

This command allows you to select a trigger path which is used for the subsequent LOGIC and LEVEL commands in the PATTERN trigger mode. The query returns the current trigger path.

Command Syntax : TRIGger:PATH CHANnelN/EXTernal
Query Syntax : TRIGger:PATH?
Returned Format : [TRIGger:PATH] CHANnelN/EXTernal

TRIGGER:SLOPE

This command allows you to select the trigger slope for the previously specified trigger SOURCE. It can be used only in the EDGE trigger mode. The query returns the trigger slope of the current trigger SOURCE.

Command Syntax : TRIGger:SLOPe POSitive/NEGative
Query Syntax : TRIGger:SLOPe?
Returned Format : [TRIGger:SLOPe] POSitive/NEGative

TRIGGER:SOURCE

This command is used to specify the trigger source. This command also identifies the source for any subsequent SLOPE and LEVEL commands. The query returns the current trigger source.

NOTE: The SOURCE command can be used only in the EDGE trigger mode.

Command Syntax : TRIGger:SOURce CHANnelN/EXTernal
Query Syntax : TRIGger:SOURce?
Returned Format : [TRIGger:SOURce] CHANnelN/EXTernal

V) Detailed Examples

To start a measurement

Step 1: Specify the module or group run that is to be started.

'SELECT 0' - selects group run
'SELECT 1' - selects module A
'SELECT 2' - selects module B
'SELECT 3' - selects module C
'SELECT 4' - selects module D
'SELECT 5' - selects module E

Step 2: Specify the RUN mode (single or repetitive).

'TYPE SINGLE' - specifies a single run
'TYPE REPETITIVE' - specifies a repetitive run

Step 3: To start the measurement.

'START'

Bugs: The 'SELECT 0' command does not select the group run module properly when using the 'TYPE' and 'START' commands. Instead of selecting group run, select a module that is in the group run. This will yield identical results.

To stop a measurement

Step 1: Specify the module or group run that is to be stopped.

'SELECT 0' - selects group run
'SELECT 1' - selects module A
'SELECT 2' - selects module B
'SELECT 3' - selects module C
'SELECT 4' - selects module D
'SELECT 5' - selects module E

Step 2: To stop the measurement.

'STOP'

Bugs: The 'SELECT 0' command does not select the group run module properly when using the 'STOP' command. Instead of selecting group run, select a module that is in the group run. This will yield identical results.

To add a single line in the pattern generator.

Step 1: Select the pattern generator module by using the 'SELECT' command.

Step 2: Remove all unwanted lines in the stimulus program by deleting them manually, or by using the 'LIST:DELETE ALL' command.

Step 3: To add a line use the 'LIST:PROGRAM' command. The format of this command is as follows:

LIST:PROGRAM <line_number> , <instruction> [, <repeat_count>]
[, <pattern>]

where <line_number> is an integer.

<instruction> is REPEAT|NOOP|WIMB|WAIT|BREAK|SIGNAL.

<repeat_count> is the count for the REPEAT instruction.

<pattern> is a string that specifies the fields for the instruction. The format of this field is as follows:

"integer" to specify base 10.
"#Bbbbbbbbb" where b is 0 or 1 for binary
"#Ooooooooo" where o is 0-7 for octal
"#Hhhhhhhhh" where h is 0-F for hex

Examples: LIST:PROGRAM 1,REPEAT,10,"34","5"
LIST:PROGRAM 2,BREAK,"#HABD","0"

Future changes: Macro programs will be able to be specified and Macros will be able to be invoked.

Query scope measurements

Step 1: Select the scope module by using the 'SELECT' command.

Step 2: Specify the measurement source channel by using the 'SOURCE' command. The 'SOURCE' command has the following syntax:

SOURCE CHANNEL#

Example: SOURCE CHANNEL2

Step 3 Obtain the desired measurement.

```
MEASURE:FALLTIME?
MEASURE:FREQUENCY?
MEASURE:NWIDTH?
MEASURE:OVERSHOOT?
MEASURE:PERIOD?
MEASURE:PRESHOOT?
MEASURE:PWIDTH?
MEASURE:RISETIME?
MEASURE:VAMPLITUDE?
MEASURE:VBASE?
MEASURE:VMAX?
MEASURE:VMIN?
MEASURE:VPP?
MEASURE:VTOP?
```

Query scope marker positions.

Step 1: Select the scope module by using the 'SELECT' command.

Step 2: Query the marker positions using the 'XVOLT'and 'OVOLT' commands.

```
Examples: MARKER:XTIME?
          MARKER:OTIME?
```

Appendix A -- HP 16500A SETUP Command and Configuration Block Data Definition

The HP 16500A system and modules may be quickly configured using the SETUP command. The SETUP command transmits to or from the HP 16500A a block of configuration data with values for the parameters to operate the system or module.

-- Format of the SETUP command for HP 16500A system and modules

Syntax : SETUP<BLOCK OF CONFIGURATION DATA>

-- Format of the SETUP query for the HP 16500A system and modules

Query syntax : SETUP?
Returned format : SETUP<BLOCK OF CONFIGURATION DATA>

-- Definition of BLOCKS OF CONFIGURATION DATA

The block of configuration data is made up of a block length specifier and a variable number of sections.

<BLOCK LENGTH SPECIFIER><SECTION 1>..<SECTION N>

The block length specifier is defined as follows:

#N<LENGTH> where N is the number of digits needed to represent <LENGTH> and <LENGTH> is the total length of all the sections. For example, if the total length of the block (all the sections) is 1732 bytes, the block length specifier would be "#41732" since that length is represented with four digits.

Sections consist of a section header followed by the section data as follows:

<Section header><Section data> where section header is defined as:

10 bytes for the section name
1 byte reserved (always 0)
1 byte for the module ID code (0 for system module)
4 bytes for the length of the section data in bytes

The section data format varies for each section and may be of any length.

NOTE: The total length of a section is 16 (for the section header) plus the length of the section data. Thus when calculating the length of a block of configuration data care should be taken not to forget to add the length of the section headers.

Appendix B -- Definition of SYMBOL Sections for the HP 16500A Modules

The symbol sections have a special structure defined for compatibility between modules. Otherwise, they are the same as any other section for a module's setup and may be the only section sent to a module.

A section is defined in Appendix A. The name of the section is always "SYMBOLS " except for the HP 16510A module (and HP 165X instrument) since these products treat symbols independently for two internal instruments (machines). The names for these products is "SYMBOLS A " or "SYMBOLS B ". The contents of the symbols section is defined as follows:

```
1 byte specifier followed by:
     if the byte is a 1,
          6 bytes of label name for next list of symbols.
          1 byte reserved (name terminator, always 0)
     if the byte is a 2,
          16 bytes of pattern symbol name
          1 byte reserved (name terminator, always 0)
          4 bytes to represent the pattern
          4 bytes to represent a don't care mask for the pattern
     if the byte is a 3,
          16 bytes of range symbol name
          1 byte reserved (name terminator, always 0)
          4 bytes to represent start value for range
          4 bytes to represent stop value for range
     if the byte is a 0,
          end of the symbol section, no more bytes.
```

When the section is read in, all previous symbols are discarded. The process then continues by reading in the first label specifier (1) and searching the label list for the label that matches the read name. If the label is found, the symbols that follow are read and assigned to that label. Otherwise, the symbols are skipped until the next label specifier (1) or end of symbol section (0) is encountered.

Example:

We want to give label STATUS two symbols -- OPCODE_FETCH and MEMORY_ACCESS. OPCODE_FETCH occurs whenever the 8th bit of STATUS (MSB) is set and MEMORY_ACCESS occurs whenever the value of status is 1, 2, or 3. Assuming that STATUS is 8 bits wide, we can define the two symbols as follows:

```
OPCODE_FETCH   pattern symbol  1XXXXXXXB
MEMORY_ACCESS  range symbol    00000001B   00000011B
```

The section would look as follows:

section header (16 bytes):

```
10 bytes section name ("SYMBOLS   " for most modules)
1 byte reserved (name terminator, always 0)
1 byte id code of module
4 bytes section length (61 for this example)
```

section data (61 bytes):

```
1 byte specifies a label follows (1)
6 bytes label name ("STATUS")
1 byte reserved (name terminator, always 0)
1 byte specifies pattern symbol (2)
16 bytes symbol name ("OPCODE_FETCH    ")
1 byte reserved (name terminator, always 0)
4 bytes pattern value (128=10000000B)
4 bytes don't care mask value (127=01111111B)
1 byte specifies range symbol (3)
16 bytes symbol name ("MEMORY_ACCESS   ")
1 byte reserved (name terminator, always 0)
4 bytes range start value (1)
4 bytes range stop value (3)
1 byte specifies end of symbol section (0)
```

Appendix C -- HP 16500A SYSTEM SETUP Command Learn String

The SETUP command for the SYSTEM module is used to configure system parameters such as the screen colors, RS-232C configuration, printer selection, etc.

As with all HP 16500A SETUPs, the system configuration consists of several sections. The format of the SETUP command and sections is described in Appendix A. The system module has an ID code of 00.

The system SETUP consists of up to 6 sections (643 bytes). Default values are indicated with '*' whenever applicable and numbers are listed in the left hand column to specify the byte in the learn string that the following value starts in.

The six sections are:

1. "CARD_CAGE " (10 bytes) - Current card cage configuration.

   NOTE: The next 10 bytes should not be changed.

17 5 bytes - ID codes for installed cards. (FFh=no card)
22 5 bytes - slot that owns this card (0=none, 1..5=A..E)

2. "RS-232 " (10 bytes) - RS-232C communication parameters.

43 2 bytes - data bits (0=7-bits, 1=8-bits*)
45 2 bytes - parity (0=none*, 1=odd, 2=even)
47 2 bytes - stop bits (0=1-bit*, 1=1.5-bits, 2=2-bits)
49 2 bytes - baud rate (0=110, 1=300, 2=600, 3=1200, 4=2400, 5=4800, 6=9600*, 7=19200)
51 2 bytes - protocol (0=none, 2=XON/XOFF*)

3. "HP-IB " (10 bytes) - HP-IB and printer parameters.

   NOTE: These next 4 bytes should not be changed.

69 2 bytes controlling interface (0=HP-IB*, 1=RS-232C)
71 2 bytes - HP-IB address (0..30, 7*)
73 2 bytes - printer type (0=ThinkJet, 1=QuietJet, 2=LaserJet 3=PaintJet, 4=Alternate)
75 2 bytes - print width columns for listings (0=80*, 1=132)
77 2 bytes - page length (0=11"*, 1=12")

4. "HIL " (20 bytes) - Touch screen/HIL parameters.

95 2 bytes - sound on/off (0=OFF, 1=ON*)
97 2 bytes - touch on/off (0=OFF, 1=ON*)

16 bytes - touch calibration values

99 2 bytes - lower left x (10*)
101 2 bytes - lower left y (33*)
103 2 bytes - upper right x (46*)
105 2 bytes - upper right y (16*)
107 2 bytes - x scaling factor (36*)
109 2 bytes - y scaling factor (19*)
111 2 bytes - x offset (-17*)
113 2 bytes - y offset (-5*)

5. "COLORS " (24 bytes) - Screen display color parameters.

3 bytes for each color 0 through 7 (8 colors)

131 1 byte - hue value (0..100)
132 1 byte - saturation value (0..100)
133 1 byte - luminosity value (0..100)

*The default color values are as follows:

0, 0, 0 (Color 0 - Black*)
13, 43, 76 (Color 1 - Tan*)
0, 0,100 (Color 2 - White*)
60,100, 60 (Color 3 - Blue*)
60, 45, 90 (Color 4 - Light Blue*)
33,100, 75 (Color 5 - Green*)
0,100,100 (Color 6 - Red*)
15,100,100 (Color 7 - Yellow*)

6. "INTERMODUL" (473 bytes) - Intermodule configuration.

9 bytes - intermodule setup for correlatible modules

171 4 bytes reserved (0*)
175 1 byte - module A (0 = not correlatible*, 1 = corr)
176 1 byte - module B (0 = not correlatible*, 1 = corr)
177 1 byte - module C (0 = not correlatible*, 1 = corr)
178 1 byte - module D (0 = not correlatible*, 1 = corr)
179 1 byte - module E (0 = not correlatible*, 1 = corr)

180 1 byte reserved (0*)

9 bytes - intermodule setup for modules with trigger.

4 bytes reserved
1 byte - module A (0 = no trigger*, 1 = triggered)
1 byte - module B (0 = no trigger*, 1 = triggered)
1 byte - module C (0 = no trigger*, 1 = triggered)
1 byte - module D (0 = no trigger*, 1 = triggered)
1 byte - module E (0 = no trigger*, 1 = triggered)

190 1 byte reserved (0*)

10 bytes - intermodule configuration tree

191 4 bytes reserved
195 1 byte - module A (0*,3..8)
196 1 byte - module B (0*,3..8)
197 1 byte - module C (0*,3..8)
198 1 byte - module D (0*,3..8)
199 1 byte - module E (0*,3..8)
200 1 byte - Port Out (0*,4..8)

0: Independent module or Off for Port Out*
3: Module armed from group run key
4: Module armed from the module A
5: Module armed from the module B
6: Module armed from the module C
7: Module armed from the module D
8: Module armed from the module E

201 2 bytes - Port In specification (0 = Port In not active*)
(1 = Port In active)

36 bytes intermodule trigger positions

203 16 bytes reserved (IEEE float 0*)
219 4 bytes - module A trigger position (IEEE float 0*)
223 4 bytes - module B trigger position (IEEE float 0*)
227 4 bytes - module C trigger position (IEEE float 0*)
231 4 bytes - module D trigger position (IEEE float 0*)
235 4 bytes - module E trigger position (IEEE float 0*)

NOTE: If an HP 16510A card is configured as a dual state and timing analyzer, the trigger position represents the timing analyzer trigger.

If an HP 16510A card is configured as a dual state analyzer, the trigger position represents the state analyzer that triggered the IMB bus.

36 bytes intermodule start acquisition positions

239 16 bytes reserved (IEEE float 0*)
255 4 bytes - mod A start acquisition position (IEEE float 0*)
259 4 bytes - mod B start acquisition position (IEEE float 0*)
263 4 bytes - mod C start acquisition position (IEEE float 0*)
267 4 bytes - mod D start acquisition position (IEEE float 0*)
271 4 bytes - mod E start acquisition position (IEEE float 0*)

36 bytes intermodule end acquisition positions

275 16 bytes reserved
291 4 bytes - module A end acquisition position (IEEE float 0*)
295 4 bytes - module B end acquisition position (IEEE float 0*)
299 4 bytes - module C end acquisition position (IEEE float 0*)
303 4 bytes - module D end acquisition position (IEEE float 0*)
307 4 bytes - module E end acquisition position (IEEE float 0*)

311 36 bytes reserved

36 bytes hardware adjustment values.

347 16 bytes unused
363 4 bytes Module A hardware adjustment (*)
367 4 bytes Module B hardware adjustment (*)
371 4 bytes Module C hardware adjustment (*)
375 4 bytes Module D hardware adjustment (*)
379 4 bytes Module E hardware adjustment (*)

(*) - HP 16510A -50E-9 == B356BF9FH IEEE float
(*) - HP 16520A 0 == 00000000H IEEE float
(*) - HP 16515A -23E-9 == B2C59189H IEEE float
(*) - HP 16530A -57E-9 == B374D02AH IEEE float
(*) - other cards use 0

NOTE: In next software release these values will not need to be specified.

7 bytes - Run mode

383 3 bytes reserved
386 1 byte - Intermodule (0 = Single*, 1 = Repetitive)
387 1 byte - Module A (0 = Single*, 1 = Repetitive)
388 1 byte - Module B (0 = Single*, 1 = Repetitive)
389 1 byte - Module C (0 = Single*, 1 = Repetitive)

NOTE: With current learn string, run mode cannot be set for modules D and E. This will be available with the next release in the next two bytes (390 & 391).

12 bytes - current intermodule settings

NOTE: This section will not need to be specified in the next software release.

390 2 bytes - Module A current setting
392 2 bytes - Module B current setting
394 2 bytes - Module C current setting
396 2 bytes - Module D current setting
398 2 bytes - Module E current setting

Current setting choices for modules:

1 = Independent*
2 = Group
3 = Module A
4 = Module B
5 = Module C
6 = Module D
7 = Module E

400 2 bytes - Port out current setting

Current setting for Port out:

0 = OFF*
3 = Module A
4 = Module B
5 = Module C
6 = Module D
7 = Module E

96 bytes - intermodule blowup choices

NOTE: This section will not need to be specified in the next software release.

402 16 bytes - Module A blowup choices

6 bytes reserved (-8*, 0*, 0*)
2 bytes - Module A field blowup (0 = on, -1 = off*)
2 bytes - Module B field blowup (0 = on, -1 = off*)
2 bytes - Module C field blowup (0 = on, -1 = off*)
2 bytes - Module D field blowup (0 = on, -1 = off*)
2 bytes - Module E field blowup (0 = on, -1 = off*)

418 16 bytes - Module B blowup choices

6 bytes reserved (-8*, 0*, 0*)
2 bytes - Module A field blowup (0 = on, -1 = off*)
2 bytes - Module B field blowup (0 = on, -1 = off*)
2 bytes - Module C field blowup (0 = on, -1 = off*)
2 bytes - Module D field blowup (0 = on, -1 = off*)
2 bytes - Module E field blowup (0 = on, -1 = off*)

434 16 bytes - Module C blowup choices

6 bytes reserved (-8*, 0*, 0*)
2 bytes - Module A field blowup (0 = on, -1 = off*)
2 bytes - Module B field blowup (0 = on, -1 = off*)
2 bytes - Module C field blowup (0 = on, -1 = off*)
2 bytes - Module D field blowup (0 = on, -1 = off*)
2 bytes - Module E field blowup (0 = on, -1 = off*)

450 16 bytes - Module D blowup choices

6 bytes reserved (-8*, 0*, 0*)
2 bytes - Module A field blowup (0 = on, -1 = off*)
2 bytes - Module B field blowup (0 = on, -1 = off*)
2 bytes - Module C field blowup (0 = on, -1 = off*)
2 bytes - Module D field blowup (0 = on, -1 = off*)
2 bytes - Module E field blowup (0 = on, -1 = off*)

466 16 bytes - Module E blowup choices

6 bytes reserved (-8*, 0*, 0*)
2 bytes - Module A field blowup (0 = on, -1 = off*)
2 bytes - Module B field blowup (0 = on, -1 = off*)
2 bytes - Module C field blowup (0 = on, -1 = off*)
2 bytes - Module D field blowup (0 = on, -1 = off*)
2 bytes - Module E field blowup (0 = on, -1 = off*)

482 16 bytes - Output port blowup choices

6 bytes reserved (8*, -1*, -1*)
2 bytes - Module A field blowup (0 = on, -1 = off*)
2 bytes - Module B field blowup (0 = on, -1 = off*)
2 bytes - Module C field blowup (0 = on, -1 = off*)
2 bytes - Module D field blowup (0 = on, -1 = off*)
2 bytes - Module E field blowup (0 = on, -1 = off*)

NOTE: If a module is not in group run then its blow up choices should include all modules that are in group run.

If a module is in group run then its blow up choices should include all modules that are not descendants in the intermodule configuration configuration.

36 bytes intermodule skew adjustment

498 16 bytes reserved (IEEE float 0*)
514 4 bytes - Module A adjust (IEEE float 0*)
518 4 bytes - Module B adjust (IEEE float 0*)
522 4 bytes - Module C adjust (IEEE float 0*)
526 4 bytes - Module D adjust (IEEE float 0*)
530 4 bytes - Module E adjust (IEEE float 0*)

534 108 bytes reserved (0*)

642 2 bytes number of modules in intermodule tree (0*)

The length of the SETUP may be calculated by adding up the length of each individual section and adding the number of sections times 16. For this setup, the length is 10+10+10+20+24+473+(16*6) = 643.

Appendix D -- HP 16510A STATE/TIMING ANALYZER SETUP Command Learn String

The SETUP command for the STATE/TIMING ANALYZER module is used to configure system parameters such as the pod and bit assignments, input thresholds, strobe values, clock rates, etc.

As with all HP 16500A SETUPs, the state/timing analyzer configuration consists of a block of configuration data. The block of configuration data contains one or more sections and is defined in Appendix A. The ID code of the state/timing analyzer module is 31.

The block of configuration data for the state/timing analyzer module varies depending on the number of sections in it and whether it is generated on an HP 16500A or on an HP 165X. All sections are optional and may be omitted when sending the SETUP command. This appendix will only describe the sections used by the HP 16510A module. Version V01.00 will send all sections (except the symbol sections when no symbols have been specified) when the query is requested. Later versions will not send the DATA section for the query.

An HP 16510A module configuration consists of 7 sections. Default values are indicated with '*' whenever applicable and numbers are listed in the left hand column to specify the byte in the learn string that the following value starts in.

The seven sections are:

1. "CONFIG " (2720 bytes)

17 50 bytes - POD information

14 bytes - Machine definition Analyzer 1

17 10 bytes - Machine name Analyzer1 (e.g. "ANALYZER 1"

27 2 bytes - Reserved (always 00)

29 1 byte - Machine mode Analyzer1
(0=OFF, 1=TIMING, 2=STATE)

30 1 byte - List of pods Analyzer1
(The pod list is an 8 bit quantity [1 byte], where a "1" in a given bit means that this pod is assigned to this analyzer and a "0" means this pod does not belong to this analyzer.)

Bit8	Bitx	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
unused	unused	Pod1	Pod2	Pod3	Pod4	Pod5	unused

31 14 bytes - Machine definition Analyzer 2

45 1 byte - unassigned pods
(see explanation for list of pods assigned)

46 1 byte - Range term owner
Which machine owns the range term in state trace
(1=Analyzer1, 2=Analyzer2)

47 20 bytes - Reserved

940 bytes - FORMAT information

20 bytes - Pod specification

4 bytes - Pod 5 Specification
67 1 byte - Demux mode
(0=NORMAL CLOCKING, 1=DEMULTIPLEX,
2=MIXED CLOCKS)
68 1 byte - Threshold
(0=TTL, 1=ECL, 2=User defined)
69 1 byte - User defined threshold
(If User defined is selected, the value of
the threshold can range from -99 to 99
representing voltage levels from -9.9V
to 9.9V.)

70 1 byte - Reserved

71 4 bytes - Pod 4 Specification

75 4 bytes - Pod 3 Specification

79 4 bytes - Pod 2 Specification

83 4 bytes - Pod 1 Specification

460 bytes - Format specification Analyzer1

87 1 byte - First label

NOTE: When the labels are displayed in the trace menu, the order is determined by the first label variable and the next label variable described below. The leftmost label in the list is the first label (value 0 - 19).

88 1 byte - Reserved

22 bytes - Label 1 Information

89 6 bytes - Label name (e.g. "STATUS")
95 2 bytes - Reserved (always 00)
97 1 byte - Display base
(0=BINARY, 1=OCTAL, 2=DECIMAL, 3=HEXADECIMAL, 4=ASCII, 5=SYMBOL)
98 1 byte - Label polarity
(0=NEGATIVE , 1=POSITIVE)
99 8 bytes - Reserved
107 1 byte - Number of channels assigned to label
108 1 byte - Label visibility
(0=label OFF, 1=label ON)
109 1 byte - Next label

NOTE: When the labels are displayed in the trace menu, the order in which they are displayed is determined by the first label variable described above, along with the next label variables. Next label (value 0 - 19) is the label number of the next label in the trace menu label list. When we have traversed the entire label list, the LAST label will have a value of -1 here.

110 1 byte - Symbol base/width for this label

NOTE: This variable is used to hold BOTH the symbol viewing width and the symbol viewing base. The 1 byte value is broken into 2 nibbles. The lower nibble holds the viewing width. The values in this lower nibble range from 2 (width of 3) to 15 (width of 16). The upper nibble is used to store the viewing base. Values here are 0 for BINARY, 1 for OCTAL, 2 for DECIMAL, 3 for HEXADECIMAL, or 4 for ASCII. As an example, if the symbol width variables contain 38h (hex), then the viewing width is 9 and the viewing base is hexadecimal.

111 418 bytes - Labels 2 thru 20 Information (19 * 22 bytes)

18 bytes - Clock Definition

529 1 byte - "J" Clock Master definition
(0=OFF , 1=FALLING EDGE, 2=RISING EDGE,
3=EITHER EDGE)
530 4 bytes - "K" through "N" Clock Master definition
534 1 byte - Reserved
535 1 byte - "J" Clock Slave definition
(0=OFF, 1=FALLING EDGE, 2=RISING EDGE,
3=EITHER EDGE)
536 4 bytes - "K" through "N" Clock Slave definition
540 1 byte - Reserved
541 1 byte - "J" Clock Level definition
(0=Clock not used as a level, 2=Low level,
3=High level)
542 4 bytes - "K" through "N" Clock Level definition
546 1 byte - Reserved

547 460 bytes - Format specification Analyzer2

1330 bytes - TRACE information

8 bytes - Arming specification

1007 1 byte - Reserved

1008 1 byte - Level 1 Master
(1=MACHINE A, 2=MACHINE B, 3=BNC_IN)

1009 1 byte - Level 2 Master
(1=MACHINE A, 2=MACHINE B, 3=BNC_IN)

1010 1 byte - Reserved

1011 1 byte - Level 0 Slave

1012 1 byte - Level 1 Slave

1013 1 byte - Level 2 Slave

1014 1 byte - Reserved

Arming works as follows :

The slave(s) for a given level are defined by the value of the byte for that level. The slave definition bytes are :

Bits 8 thru 4 unused	BNC OUT	MACH B	MACH A

If the slave for a given level is 0, then ignore the master for that level.

Level 0 Master is RUN.

Example 1 :

level	Master	Slave
0	RUN	Analyzer1, Analyzer2
1	Analyzer1	0
2	Analyzer2	BNC out

In example 1, analyzers 1 and 2 are armed by RUN. Analyzer2 then arms the BNC out.

Example 2 :

level	Master	Slave
0	RUN	0
1	BNC_IN	Analyzer 1
2	Analyzer 1	Analyzer 2

In example 2, since RUN has no slaves, it is ignored. The BNC_IN arms Analyzer 1 which then in turn arms Analyzer 2.

---

660 bytes - Trace specification Analyzer 1

82 bytes - Timing trace specification

1015 2 bytes - Reserved
1017 4 bytes - Time per Div
(IEEE floating point representation of Time per Division)
1021 4 bytes - Reserved
1025 4 bytes - Delay
(IEEE floating point representation of Delay)
1029 4 bytes - Reserved
32 bytes - Timing trace trigger pattern
1033 2 bytes - Reserved
1035 10 bytes - Channel mask -
80 bit value that must be set to match the specified trigger pattern
1045 10 bytes - Don't care mask -
80 bit value. A "1" in a bit location means that we should ignore the corresponding bit of the channel mask described above.

1055 1 byte - Duration mode
(1=LESS THAN, 2=GREATER THAN)
1056 1 byte - Reserved
1057 4 bytes - Duration
IEEE floating point representation of duration. Limits are as follows:

"<" duration:
332BCC77h(40 ns) to 3C23D70Ah(10 ms)

">" duration:
3300D959h(30 ns) to 3C23D70Ah(10 ms)

1061 4 bytes - Reserved
20 bytes - Timing trace edge pattern
1065 10 bytes - Up_edge mask -
Set of 80 channels which are looking for a rising edge prior to trigger (a "1" in a bit location means to look for that edge).
1075 10 bytes - Down edge mask
12 bytes - Timing trace glitch pattern
1085 1 byte - Glitch mode ON ?
(0=NO, 1=YES)
1086 1 byte - Reserved
1087 10 bytes - Glitch mask -
Set of 80 channels which are looking for a glitch prior to trigger (a "1" in a bit location means that glitch triggering should be used for this particular channel).

518 bytes - State trace specification

1097 2 bytes - Reserved
10 bytes - Arming specification
1099 1 byte - Arming used
(0=NO, 1=YES)
1100 9 bytes - Reserved
10 bytes - Tagging specification
1109 1 byte - Tagging type
(0=OFF, 1=TIME tagging, 2=STATE tagging)
1110 1 byte - Reserved
1111 8 bytes - Store qualifier for tagging
(if tagging states)
See note at end of this appendix for complete description of qualifier specification
34 bytes - Sequence level 1 specification

NOTE: The chip levels 1 through 8 described here are NOT the same as the levels 1 through 8 shown on the HP 16500A screen. These 8 level numbers are internal level numbers used by the HP 16500A software. To determine the actual sequence of the levels (as seen on screen), start at the first sequence level variable. Then proceed thru the primary branch destinations. Following this path will produce the "visible" order of the sequence levels.

1119 1 byte - Level in use (0=NO, 1=YES)
1120 1 byte - Reserved
1121 8 bytes - Store qualifier
See note at end of this appendix for complete description of qualifier specification
1129 8 bytes - "Find" qualifier
1137 1 byte - Primary branch destination
Level to branch to if the "find" qualifier is satisfied.
1138 1 byte - Reserved
1139 2 bytes - Occurrence counter
Number of occurrences of "find" qualifier that must be found before proceeding to next level
1141 1 byte - Secondary branch in use ?? (0=NO, 1=YES)
1142 1 byte - Reserved
1143 8 bytes - Secondary branch qualifier
1151 1 byte - Secondary branch destination
1152 1 byte - Reserved
1153 238 bytes - Sequence levels 2 through 8 (34 * 7 levels)
1391 1 byte - First level in sequence
This variable gives the level number (in terms of chip levels which corresponds to level 1 on the HP 16500A screen.
1392 1 byte - Trigger level
1393 1 byte - Post-trigger level
Describes which level is the first level AFTER trigger
1394 1 byte - Second-to-last level
Describes which level is the second to last level in the sequence. This is important since it is this level on which ENABLE/DISABLE can be invoked.
1395 1 byte - ENABLE/DISABLE used (0=NO, 1=YES)
1396 1 byte - Reserved
10 bytes - Prestore information
1397 1 byte - Prestore ON ?? (0=NO, 1=YES)
1398 1 byte - Reserved
1399 8 bytes - Prestore qualifier
10 bytes - Restart information
1407 1 byte - Restart ON ?? (0=NO, 1=YES)
1408 1 byte - Reserved
1409 8 bytes - Restart qualifier

22 bytes - State Pattern "A" Description

1417 2 bytes - Reserved

1419 10 bytes - Channel mask -
80 bit value that must be set to match the specified trigger pattern

1429 10 bytes - Don't care mask -
80 bit value. A "1" in a bit location means that we should ignore the corresponding bit of the channel mask described above.

1439 154 bytes - State Patterns "B" through "H" description (7 * 22)

22 bytes - Range term description

1593 1 byte - Range label -
Label number of label which has been selected to perform a range on.

1594 1 byte - Reserved

1595 10 bytes - Upper range value -
80 bit value that represents the upper range value. All bits which are unassigned to this label should be set to "1".

1605 10 bytes - Lower range value -
80 bit value that represents the lower range value. All bits which are unassigned to this label should be set to "0".

60 bytes - Run-until specification

1615 1 byte - Run until condition
(0=OFF, 1=LESS THAN, 2=GREATER THAN, 10h=OUT OF RANGE, 20h=IN RANGE)

1616 1 byte - Reserved

1617 4 bytes - Lower limit value -
IEEE floating point representation of the run until value (less than, greater than) or of the lower limit of the range value (out of range, in range)

1621 4 bytes - Upper limit value -
IEEE floating point representation of the upper limit of the range value (out of range and in range only)

50 bytes - Marker placement information

1625 2 bytes - Reserved

1627 10 bytes - X marker pattern mask -
80 bit value that represents the pattern mask for the X marker

1637 10 bytes - X marker don't care mask -
80 bit value. A "1" in a bit location means that we should ignore the corresponding bit of the pattern mask described above.

1647 2 bytes - Reserved

1649 10 bytes - 0 marker pattern mask -
80 bit value that represents the pattern mask for the 0 marker

1659 10 bytes - 0 marker don't care mask -
80 bit value. A "1" in a bit location means that we should ignore the corresponding bit of the pattern mask described above.

1669 1 byte - X marker criterion -
the X marker to be placed ENTERing (value of 0) or EXITing (value of 1) the pattern (timing machines only)

1670 1 byte - 0 marker criterion -
the 0 marker to be placed ENTERing (value of 0) or EXITing (value of 1) the pattern (timing machines only)

1671 2 bytes - X marker occurrence counter -
the number of occurrences of pattern to be found before placing X marker

1673 2 bytes - 0 marker occurrence counter -
the number of occurrences of pattern to be found before placing 0 marker

1675 660 bytes - Trace specification Analyzer 2

2335 1 byte - Run Mode
(0=SINGLE, 1=CONTINUOUS)

2336 1 byte - Reserved

200 bytes - Analyzer1 Channel Specification

2337 10 bytes - Label 1 Channel Mask -
for each of the 5 pods, a mask which provides information concerning which channels are assigned to this label. Bit assignments are set up as in the format menu for this machine.

POD 5	POD 4	POD 3	POD 2	POD 1

2347 190 bytes - Labels 2 through 20 Channel Mask (10 * 19 labels)

2537 200 bytes - Analyzer2 Channel Specification

Description of State Trace Qualifiers

All of the qualifiers used in the state trace descriptors (i.e. prestore, sequence levels, tagging, etc.) are 8 bytes long. The following section describes these 8 bytes which are used for ALL state trace qualifiers:

1. Reserved -- 1 byte
2. Qualifier mode -- 1 byte
   -- This variable is used to determine the mode of the qualifier specified. Depending on the qualifier, the mode is either always satisfied (value of 1), never satisfied (value of 2), or undetermined (value of 3). Undetermined qualifiers can not be evaluated until the values of the qualifier terms are known.
3. Global operator -- 1 byte
   -- In the qualifier picture given below, the value of the operator in the "G" box. This represents the operator performed between the two sets. A value of 0 is used for "AND", a value of "1" is "OR".
4. Reserved -- 1 byte

5. Set 1 operator -- 1 byte
   -- In the qualifier picture given below, the value of the operator in the "S1" box.
6. Set 1 elements -- 1 byte
   -- List of the elements used in Set 1 term as follows :

unused Bit 8	unused Bit 7	In Range Bit 6	Out of Range Bit 5	Pattern D Bit 4	Pattern C Bit 3	Pattern B Bit 2	Pattern A Bit 1

Where a "1" means an element is in the set, and a "0" means the the element is not.

7. Set 2 operator -- 1 byte
   -- In the qualifier picture given below, the value of the operator in the "S2" box.
8. Set 2 elements -- 1 byte
   -- List of the elements used in Set 2 term as follows :

unused Bit 8	unused Bit 7	unused Bit 6	unused Bit 5	Pattern H Bit 4	Pattern G Bit 3	Pattern F Bit 2	Pattern E Bit 1

Where a "1" means an element is in the set, and a "0" means the the element is not.

---

2. "1650 DISP " (714 bytes) - Display variables.

318 bytes - Machine A variables.

2753 16 bytes - reserved

66 bytes - machine A list information

2769 1 byte - marker mode (0=OFF*, 1=PATTERNS, 2=STATES, 3=TIMES, 4=STATISTICS)

2770 1 byte - reserved

2771 2 bytes - number of states between X marker and trigger

2773 2 bytes - number of states between O marker and trigger

2775 1 byte - currently displayed marker (0=X marker, 1=O marker)

2776 3 bytes - reserved

2779 2 bytes - number of states between X marker and trigger in mixed display

2781 2 bytes - number of states between O marker and trigger in mixed display

2783 52 bytes - reserved

170 bytes - machine A waveform information

2835 1 byte - wavefrom display mode (0=0..8 waveforms, (1=9..16 waveforms, 2=17..24 waveforms)

2836 1 byte - accumulate mode (0=ON, 1=OFF)

2837 1 byte - number of displayed waveforms (0..24)

2838 1 byte - reserved

2839 4 bytes - IEEE floating point value of time X to trigger

2843 4 bytes - IEEE floating point value of time O to trigger

2847 1 byte - timing marker mode (0=OFF*, 1=TIME, 2=PATTERNS, 3=STATISTICS)

2848 1 byte - displayed timing marker (0=X marker, 1=O marker)

2849 1 byte - index of label to show value under marker (0..19)

2850 1 byte - reserved

2851 4 bytes - IEEE float of time X to trigger in mixed display

2855 4 bytes - IEEE float of time O to trigger in mixed display

2859 2 bytes - reserved

2861 144 bytes - description of displayed waveforms

For each of 24 possible waveforms:

1 byte - waveform status (-1=deleted, (0=OFF, 1=ON DISPLAY)
1 byte - label index (-1=label was turned off, 0..19)
1 byte - which bit of above label
1 byte - overlay on/off (0=OFF, 1=ON)
1 byte - source module slot for waveform on HP 16510A (4-8 for slot A-E)
1 byte - reserved

24 bytes - trace information for machine A

3005 1 byte - state trace branching mode (0=OFF*, 1=RESTART, 2=PER LEVEL)

3006 1 byte - next free level in configuration variable structure

3007 8 bytes - mapping array between screen sequence levels and configuration variable levels. (i.e. first byte gives the configuration variable level of sequence level 1)

3015 3 bytes - 3 sequence levels visible on screen

3018 1 byte - reserved

3019 1 byte - which trace resources on display (0=a-d, 4=e-h, 8=range)

3020 9 bytes - reserved

10 bytes - machine A configuration information

3029 1 byte - label index of left-most visible label in state trace menu

3030 1 byte - x location of left edge of left most visible label in the trace menu (12=fully on screen, <12=partially on screen, never >12)

3031 2 bytes - reserved

3033 1 byte - label index of top-most visible label in format menu

3034 5 bytes - reserved

32 bytes - machine A statistics information

3039 4 bytes - number of runs

3043 4 bytes - number of hits

3047 4 bytes - IEEE float minimum time X to O

3051 4 bytes - IEEE float maximum time X to O

3055 4 bytes - IEEE float total time X to O of all runs

3059 1 byte - X marker placement (for state: 0=from Trigger, 1=from Start; for timing: 0)

3060 1 byte - O marker placement (for state: 0=from Trigger, 1=from Start, 2=from x Marker; for timing: 0=from Trigger, 1=from x Marker)

3061 10 bytes - reserved

3071 318 bytes - Machine B display variables (same as A)

48 bytes - state listing column descriptors

16 bytes - Machine A

3389 1 byte - machine that label is from -
for HP 16510A:
0=slot A machine A
1=slot A machine B
2=slot B machine A
3=slot B machine B
4=slot C machine A
5=slot C machine B
6=slot D machine A
7=slot D machine B
8=slot E machine A
9=slot E machine B
(for HP 165X, 0=machine A, 1=machine B)

3390 1 byte - label index of column 1's label (0..19=labels, 99=TIME, 100=STATE)

3391 14 bytes - column 2..8 machine and label index
For each column 2 through 8:
1 byte - machine number
1 byte - label index

3405 16 bytes - Machine B (see above)

3421 16 bytes - Mixed Display (see above)

30 bytes - state listing display parameters

3437 8 bytes - reserved

3445 2 bytes - line number of center line in state listing for machine A

3447 2 bytes - line number of center line in state listing for machine B

3449 2 bytes - line number of center line in state listing for mixed state-state. A value greater than 5000 means current line from machine B. Actual value is obtained by subtracting 10000.

3451 6 bytes - reserved

3457 1 byte - x location of left edge of leftmost visible label in machine A listing. (12=fully on screen, left justified; <12=partially on screen; never >12)

3458 1 byte - (same as above for machine B listing)

3459 1 byte - (same as above for mixed state-state listing)

3460 1 byte - reserved

3461 1 byte - index of first display column in state listing for machine A (0-7)

3462 1 byte - (same as above for machine B)

3463 1 byte - (same as above for mixed state-state listing)

3464 1 byte - reserved

3465 1 byte - relative or absolute time (or states) column (6=RELATIVE, 7=ABSOULTE)

3466 1 byte - current module (for HP 16510A: slot number 4-8 for slot A-E) (-1 for HP 165X)

3. "DATA " (14506 bytes)

The data section consists of 14506 bytes of 8 bit data, organized in one of four ways depending on the type of data. The four data types are:

1. State (no tagging)
2. State (with tagging)
3. Glitch timing
4. Transitional timing

In general, the data section is set up as 160 bytes of header information, followed by 1024 lines of 14 bytes of information each, followed by ten reserved bytes. The header gives information for each analyzer concerning the amount and type of data captured, where the trace point occurred in the data, which pods are assigned to which analyzer, etc. Each line consists of two bytes (16 bits) of status for Analyzer 1, two bytes (16 bits) of status for Analyzer 2, then five sets of two bytes (16 bits) of information (either data, count, or glitch) corresponding to the five 16 bit pods of the HP 16510A.

Note that one analyzer's information is independent of the other analyzer's information. In other words, on any given line, one analyzer may contain data information for a timing machine, while the other analyzer may contain count information for a state machine with time tags enabled. The status bytes for each analyzer describe what the information for that line contains. Therefore, when describing the different formats that data may contain below, keep in mind that this format pertains only to those pods that are assigned to the analyzer of the specified type. The other analyzer's data is TOTALLY independent and conforms to its own format.

160 bytes - Description of data header

2 bytes - Instrument ID -
for the HP 16510A, this value is always 16500
(for the HP 165X, this value is always 1650)

2 bytes - Revision Code

78 bytes - Analyzer 1 Data Information

NOTE: The values stored here in the data header represent the captured data currently stored in this structure and not what the current configuration of the analyzer is. As an example, the mode of the data may be STATE with tagging, while the current setup of the analyzer is TIMING.

1 byte - Machine data mode -
(0=IS_OFF, 1=STATE data with tagging, 2=STATE data no tagging, 3=TIMING data Glitch Mode, 4=TIMING data Transitional Mode)

1 byte - List of pods in this analyzer

unused	unused	Pod 1	Pod 2	Pod 3	Pod 4	Pod 5	unused

where a "1" in a given location means that this pod is assigned to this analyzer.

1 byte - Master chip in this analyzer -
When several chips are grouped together in a single analyzer one chip is designated as a master chip. This variable is used to hold this master value. A value of 4 represents POD 1, 3 for POD 2, 2 for POD 3, 1 for POD 4, and 0 for POD 5.

1 byte - Reserved

10 bytes - Number of rows of valid data for this analyzer -
for each of the five pods, the number of rows of valid data. Two bytes are used to store the value, and the first 2 bytes are used to hold POD 5 value, the next 2 for POD 4 value, etc.

1 byte - Trace point seen in this analyzer -
Was a trace point seen (value=1) or forced (value=0)

1 byte - Reserved

10 bytes - Trace point location for this analyzer -
for each of the five pods, the row number in which the trace point was found. Two bytes are used to store the value in a manner similar to the number of valid rows described above.

4 bytes - Time from arm to trigger for this analyzer -
Time elapsed from the arm of this machine to the trigger of this machine (in 40 ns units). A value of -1 indicates counter overflow.

1 byte - Armer of this analyzer -
Indicator of what armed this analyzer (0=RUN, 1=BNC, 2=other machine)

1 byte - Devices armed by this analyzer

Set of devices armed by this machine

-- unused --	BNC OUT	MACHINE B	MACHINE A
Bits 8 through 4	Bit 3	Bit 2	Bit 1

A "1" in a given bit position implies that this analyzer arms that device, while a "0" means the device is not armed by this analyzer.

4 bytes - Sample period for this analyzer (timing only) - Sample period at which data was acquired. Value represents the number of nanoseconds per sample.

4 bytes - Delay for this analyzer (timing only) - Delay at which data was acquired. Value represents the amount of delay in nanoseconds.

1 byte - Time tags on ?? (state with tagging only) - In state tagging mode, was the data captured with time tags (value = 1) or state tags (value = 0).

1 byte - Reserved

5 bytes - Demultiplexing (state only) - For each of the five pods (first byte is POD 5, fifth byte is POD 1) in a state machine, this describes the multiplexing of each of the five pods. (0=NO DEMUX, 1=TRUE DEMUX, 2=MIXED CLOCKS).

1 byte - Reserved

4 bytes - POD 5 trace point adjustment - For POD 5, the number of nanoseconds that are to be subtracted from the trace point described above to get the actual trace point value.

16 bytes - Pods 4 through 1 trace point adjustment (4 * 4 pods)

10 bytes - Reserved

78 bytes - Analyzer 2 Data Information

DATA SECTION DESCRIPTION

A. Description of Data Section -- 14336 bytes (14 bytes * 1024 rows)

The DATA contained in the data section of the data learn string will appear in one of four forms depending on the mode in which it was acquired. The four modes are:

1. State Data without tags
2. State Data with either time or state tags
3. Timing Glitch Data
4. Transitional Timing Data

The following four sections describe the four data modes that may be encountered.

1. State Data (no tags)

A. Status Bytes
-- In normal state mode, the status bytes provide no information.

B. Information Bytes
-- State acquisition (no tags) works as follows :

1) With each clock, data is obtained from the target system and checked with the sequencer. If the state matches a state that should be stored, it is placed into the memory.

C. Example

Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data

2. State Data (with either time or state tags)

A. Status Bytes

-- In state tagging mode, the tags allow the user to determine whether a given row of the data is a data line, a count (tag) line, or a prestore line.

1) Bit 2 -- Data vs. Count bit
If Bit 2 is set, then this row of information contains tags. If Bit 2 is clear, then this row of information contains actual acquisition data as obtained from the target system. The counts are relative counts from one state to the one previous. Therefore, the count for the first state in the data structure is invalid, and should be ignored. The count is stored in floating point format with 11 bits of mantissa and 5 bits of exponent (EEEEEMMMMMMMMMMM). The actual value of the count is given by the equation :

Count = mantissa * (2 ** exponent)

where mantissa = MMMMMMMMMMM
exponent = EEEEE

If time tagging is on, then this count value represents the number of 40 nanosecond units that have elapsed between the two stored states. In the case of state tagging, the count represents the number of qualified states that were encountered between the stored states.

2) Bit 3 -- Prestore vs. Tag bit
If Bit 3 is set, then this data row and its corresponding count row represent prestored information. The count, therefore, should be ignored.

B. Information Bytes

-- State acquisition (with tags) works as follows :

1) With each clock, data is obtained from the target system and checked with the sequencer. If the state does not match the sequencer qualifier, it is checked against the prestore qualifier. If it matches the prestore qualifier, then it it is placed into the prestore buffer. If the state does not match either the sequencer qualifier OR the prestore qualifier, it is discarded.

2) If a state matches the sequencer qualifiers, then the prestore buffer is checked. If there are any states in the prestore buffer at this time, then these prestore states are first placed into memory, along with a dummy count row. After this check, then the qualified state is placed into memory, followed by the count row which specified how many states (or 40 ns units) have elapsed since the last stored state. If this is the first stored state in memory, then the count information that is stored should be discarded.

C. Example

Status	Status	Data	Data	Data	Data	Data
Status	Status	Count	Count	Count	Count	Count
Status	Status	Prestore	Prestore	Prestore	Prestore	Prestore
Status	Status	*	*	*	*	*
Status	Status	Data	Data	Data	Data	Data
Status	Status	Count	Count	Count	Count	Count

* = Invalid data

3. Timing Glitch Data

A. Status Bytes
-- In glitch timing mode, the status bytes signal whether a given row in the data contains actual acquisition data information or glitch information.

1) Bit 1 -- Data vs. Glitch bit
If Bit 1 is set, then this row of information contains glitch information. If Bit 1 is clear, then this row contains actual acquisition data as obtained from the target system.

B. Information Bytes
-- Glitch timing works as follows :

1) At every sample period, the target system is sampled. The data is then stored into memory. The glitch detectors are then checked and if a glitch has been detected between the previous sample and the current sample, the corresponding glitch bits are set. The glitch information is then stored. If this is the first stored sample in memory, then the glitch information that is stored should be discarded.

C. Example

Status$_0$	Status$_0$	Data	Data	Data	Data	Data
Status$_1$	Status$_1$	Glitch	Glitch	Glitch	Glitch	Glitch
Status$_0$	Status$_0$	Data	Data	Data	Data	Data
Status$_1$	Status$_1$	Glitch	Glitch	Glitch	Glitch	Glitch

4. Transitional Timing Data

A. Status Bytes
-- In transitional timing mode, the status bytes signal whether a given row in the data contains acquisition information or transition count information.

1) Bits 10 and 9 -- Data vs. Count Bits Pod 5
Bits 10 and 9 of the status for transitional timing provide the user with enough information to determine whether the information for Pod 5 on a given row is acquisition data, the start of a count block, or the middle of a count block. There are three possible values these two bits may take on :

-- 00 - This row contains part of a count, but is not the first row of a count.
-- 01 - This row contains the first word of a count. 9 ENTRIES
-- 10 - This row contains actual acquisition data as obtained from the target system.

2) Bits 8 and 7 -- Data vs. Count Bits Pod 4

3) Bits 6 and 5 -- Data vs. Count Bits Pod 3

4) Bits 4 and 3 -- Data vs. Count Bits Pod 2

5) Bits 2 and 1 -- Data vs. Count Bits Pod 1

B. Information Bytes
-- Transitional timing works as follows :

1) Four samples of data are taken at 10 nanosecond intervals. The data is stored and the value of the last sample is retained.

2) Four more samples of data are taken. If any of these four samples differs from the last sample of the previous step, then these four samples are stored and the last value is once again retained.

3) If all four samples of this step are the same as the last sample taken in step 1, then no data is stored. Instead, a counter is incremented. This process will continue until a group of four samples is found which differs from the retained sample. At this time, the count will be stored in the memory, the counters reset, the current data stored, and the last sample of the four once again retained for comparison. Note that this stored count indicates the number of 40 nanosecond intervals that have elapsed between the old data and the new data.

4) The rows of the acquisition data may, therefore, be either four rows of data followed by four more rows of data, or four rows of data followed by four rows of count. Rows of count will always be followed by four rows of data except for the last row, which may be either data or count.

5) Note that this process occurs on a pod by pod basis. The individual status bits will indicate what each pod is doing.

B. Example .

Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Count	Count	Data	Data
Status	Status	Data	Count	Count	Data	Data
Status	Status	Data	Count	Count	Data	Data
Status	Status	Data	Count	Count	Data	Data
Status	Status	Count	Data	Data	Count	Data
Status	Status	Count	Data	Data	Count	Data
Status	Status	Count	Data	Data	Count	Data
Status	Status	Count	Data	Data	Count	Data
Status	Status	Data	Data	Count	Data	Data
Status	Status	Data	Data	Count	Data	Data
Status	Status	Data	Data	Count	Data	Data
Status	Status	Data	Data	Count	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data
Status	Status	Data	Data	Data	Data	Data

---

4. "SYMBOLS A " (variable length) Symbols for machine A. See Appendix B for description of symbol section format.

5. "SYMBOLS B " (variable length) Symbols for machine B. See Appendix B for description of symbol section format.

6. "INVASM A " (11 bytes) Inverse assembler file name for machine A. This section is only returned if an inverse assembler has been loaded for machine A. If it is sent in as part of the SETUP command, the inverse assembler file will be loaded from the disc at the completion of the command.

   10 bytes - Inverse assembler file name.

   1 byte - reserved (always 0)

7. "INVASM B " (11 bytes) Inverse assembler file name for machine B. See "INVASM A" above.

Appendix E -- HP 16520A PATTERN GENERATOR SETUP Command Learn String

The SETUP command for the PATTERN GENERATOR module is used to configure system parameters such as the pod and bit assignments, input thresholds, strobe values, clock rates, etc.

As with all HP 16500A SETUPs, the pattern generator configuration consists of a block of configuration data. The block of configuration data contains one or more sections and is defined in Appendix A. The ID code for the pattern generator module is 21.

The block of configuration data for the pattern generator module varies depending on the number of sections in it and the number of extender boards connected to the master. All sections are optional and may be omitted when sending the SETUP command. Version V01.00 will send all sections (except the symbol section if no symbols have been specified) when the query is requested. Later versions will only send the configuration and symbol (if applicable) sections for the query.

A full pattern generator configuration is made up of 7 sections. Default values are indicated with '*' whenever applicable and numbers are listed in the left hand column to specify the byte in the learn string that the following value starts in.

The seven sections are:

1. "CONFIG " (1128 bytes) - Main control parameters

17	2 bytes - number of pods in module (including slaves)
19	2 bytes - clock source (0=internal*, 1=external)
21	2 bytes - input threshold type (0=TTL*, 1=ECL, 2=user)
23	2 bytes - threshold value if 2 above (-99=-9.9V..99=9.9V)
25	4 bytes - reserved
29	2 bytes - clock period (0=20ns, 1=50ns, 2=100ns, 3=200ns*, 4=500ns, 5=1ms, 6=2ms, 7=5ms, 8=10ms, 9=20ms,, 10=50ms, 11=100ms, 12=200ms)
31	2 bytes - external clock divider (0= /1*, 1= /5, 2= /10)
33	12 bytes - strobe edge positions For each one of the three strobe lines: 2 bytes - left edge position (0..10, 2*) For each one of the three strobe lines: 2 bytes - right edge position (0..10, 6*)
45	6 bytes - strobe width For each one of the three strobe lines: 2 bytes - width (0..10, 4*)
51	1 byte - load strobes on 1/5th boundaries (0=NO*, 1=YES)
52	1 byte - strobe load qualifier (0=invert*, 1=don't invert, 2=load all zeroes)

```
53        6 bytes - strobe polarities
                    For each one of the three strobe lines:
                    2 bytes - polarity (0=NEGATIVE, 1=POSITIVE*)
59        28 bytes - reserved
87        1 byte - first label on screen
88        1 byte - reserved
89        462 bytes - label data structure for 21 labels
               For each one of 21 labels:
               6 bytes - label name ("A     ".."T     "&"Instr " *)
               2 bytes - reserved (always 0)
               1 byte - display base (0=BINARY, 1=OCTAL, 2=DECIMAL,
                        3=HEXADECIMAL*, 4=ASCII, 5=SYMBOL)
               1 byte - label polarity (0=NEGATIVE, 1=POSITIVE*)
               8 bytes - reserved
               1 byte - number of channels assigned (1..32)
               1 byte - label visibility (0=OFF, 1=ON*)
               1 byte - symbol display base and width (37h*)
551       546 bytes - bit assignment for 21 labels
               For each one of 21 labels:
               For each one of 26 possible pods:
               1 - byte bit assignment
1097      42 bytes - reserved
1139      2 bytes - left-most label displayed (20*)
1141      2 bytes - right-most label displayed (19*)
1143      2 bytes - single step counter (1*)
```

2. "MAINPROG " (variable length depending on slaves connected and length of program listing.) - Contains the main pattern generator listing.

```
2 bytes - number of pods (width of program)@
2 bytes - length of program (with macros expanded)
2 bytes - current line number on center of display
6 bytes - reserved
2 bytes - number of lines in program (macros not expanded)%
```

NOTE: Macro calls require 2 program lines. The first line contains the macro opcode and the values for parameter 1 for each label. The second line contains the opcode for the macro's parameter and the values for parameter 2 for each label.

```
<one byte for each line(%)> list of opcodes (0=None*,
          1=WAIT IMB, 2=WAIT, 3=REPEAT, 4=SIGNAL IMB,
          8=BREAK, 16=MACRO1, 17=MACRO2, 18=MACRO3,
          19=MACRO4, 20=MACRO1 Parameter, 21=MACRO2
          Parameter, 22=MACRO3 Parameter, 23=MACRO4
          Parameter)
```

<one byte for each line(%)> list of parameters for REPEAT and WAIT opcodes (for REPEAT: 1*..255, for WAIT 0..255* where a zero in a bit position means CONT and 1 means WAIT defined as follows:

WAIT Parameter BITS 2 1 0	Bit Position
0 0 0	0
0 0 1	4
0 1 0	2
0 1 1	6
1 0 0	1
1 0 1	5
1 1 0	3
1 1 1	7

<one byte for each pod of each line(% times @)> pod data
For each line in program (%):
<"width" bytes (@)> program data
<one byte for each pod of each line(% times @)> autofill
For each line in program (%):
<"width" bytes(@)> autofill data

3. "MACRO1 " (variable length depending on slaves and length of macro 1 listing.)

1 byte - number of pods (program width) $
1 byte - number of lines in macro &
1 byte - number of times macro is referenced
7 bytes - macro name
1 byte - current macro number (0..3)
280 bytes - macro parameter names
For 40 (2 per label) parameters:
7 bytes - parameter name
<6 bytes for each line in macro(&)+2> parameter names
For each line in macro(&)+2:
6 bytes - parameter name
<one byte for each macro line(&)> list of opcodes (see "MAINPROG " above)
<one byte for each macro line(&)> list of parameters (see "MAINPROG ")
<one byte for each pod of each line(& times $)> pod data
For each line in program (&):
<"width" bytes ($)> program data
<one byte for each pod of each line(& times $)> autofill
For each line in program (&):
<"width" bytes($)> autofill data

4. "MACRO2 " (variable length depending on slaves and length of macro 2 listing.)
   See "MACRO1 " section definition.

5. "MACRO3 " (variable length depending on slaves and length of macro 3 listing.)
   See "MACRO1 " section definition.

6. "MACRO4 " (variable length depending on slaves and length of macro 4 listing.)
   See "MACRO1 " section definition.

7. "SYMBOLS " (optional section only used if symbols have been specified. The length is variable depending on number of symbols.)

   The contents for symbol sections are described in Appendix B.

Appendix F -- HP 16530A OSCILLOSCOPE SETUP Command Learn String

The SETUP command for the OSCILLOSCOPE module is used to configure oscilloscope parameters such as the input channel setups, timebase setups, trigger specification, waveform data, etc.

The oscilloscope SETUP consists of two main sections:

1. The setup specifications which configure the oscilloscope
2. The waveform data

NOTE: Most variables in the setup specification section are related to one or more variables in the same subsection and/or in the other subsection. For example, if you change the trigger source, the trigger mode port and the vertical port contents of the old and new source must be modified accordingly. These variables are indicated with '**' and a reference number in square brackets whenever applicable. Thus, when these variables are encounted, look at the same reference numbers in this appendix for the related variables that must also be changed. It is cumbersome to explain the relationships and the associated values between these related variables, so we recommend that you use the previously sent out values in the other learn strings.

To change s/div or delay, you might want to use the command:

```
TIMebase:RANGe <nrf>
TIMebase:DELay <nrf>.
```

The two main sections are:

1. "SETUP " (1573 bytes) - oscilloscope configurations
   The ocilloscope setup specifications consist of up to 3 subsections (1573 bytes). Default values are indicated with '*' whenever applicable. The ID code for the HP 16530A is 31.

   The SETUP section consists of the following three subsections:

   140 bytes - Current ocilloscope screen configuration

   - **4 bytes - x to o (-2500s...2500s, 0s*) [23]
   - **4 bytes - trigger to x (-2500s...2500s, 0s*) [23]
   - **4 bytes - trigger to o (-2500s...2500s, 0s*) [23]
   - **4 bytes - s/div (5ns...10s, 10us*) [1]
   - **4 bytes - delay (-2500s... 2500s, 0.0s*) [2]
   - **2 bytes - channel choice in channel submenu (0...2n-1, where n is the number of acquisition boards in the oscilloscope, 0*) [3]

**4 bytes - v/div (10mV...4V, 1.5V*)
[4][7]
**4 bytes - offset (v/div < 200mV, the limit is +/- 800mV; v/div >= 200mV, the limit is +/-16V, 2.5V*)
[4][5][7]
**2 bytes - probe setting choice (0...1000, 10*)
[8]
**2 bytes - impedance choice (0=1Mohm*, 1=50ohms)
[6]
**2 bytes - preset choice (0=TTL*, 1=ECL, 2=User Defined)
[7]
2 bytes - marker choice (0=Off*, 1=On, 2=Auto)
10 bytes - reserved
**2 bytes - trigger "when" in pattern mode (0=Entered*, 1=Exited)
[9]
**4 bytes - trigger level for the current trigger source (1.4V*)
[4][5][7][10]
**2 bytes - trigger source (0...2n-1, where n is the number of acquisition boards in the oscilloscope, 0*)
[11]
**2 bytes - trigger slope choice (0=Positive*, 1=Negative)
[12]
**2 bytes - trigger counts (1...32000, 1*)
[17]
**2 bytes - trigger mode (0=Edge*, 1=Pattern, 2=Immediate)
[13]
**2 bytes - auto-trigger choice (0=Off, 1=On*)
[14]
**2 bytes - display mode (0=Normal*, 1=Average, 2=Accumulate)
[15]
**2 bytes - average number (2...256, 8*)
[16]
2 bytes - connect dots choice (0=Off*, 1=On)
2 bytes - reserved
2 bytes - auto-meas channel choice (0...2n-1, where n is the number of acquisition boards in the oscilloscope, 0*)
2 bytes - channel choice for X-marker in auto marker placement (0...2n-1, where n is the number of acquisition boards in the oscilloscope, 0*)
2 bytes - level for X-marker in auto marker placement (10...90, 50*)
2 bytes - slope for X-marker in auto marker placement (0=Positive*, 1=Negative)
2 bytes - slope occurrence for X-marker in auto marker placement (1...100, 1*)
2 bytes - statistics choice (0=Off, 1=On*)
2 bytes - channel choice for O-marker in auto marker placement,(0...2n-1, where n is the number of acquisition boards in the oscilloscope, 0*)

2 bytes - level for 0-marker in auto marker placement (10....90, 50*)
2 bytes - slope for 0-marker in auto marker placement (0=Positive*, 1=Negative)
2 bytes - slope occurrence for 0-marker in auto marker placement (1...100, 1*)
2 bytes - run until time X-0 choice (0=Off*, 1=Less Than, 2=Greater than, 3=In Range, 4=Not in Range)
8 bytes - reserved
4 bytes - low end for run until time X-0 specification (-100Ms...+100Ms, 0s*)
4 bytes - high end for run until time X-0 specification (-100Ms...+100Ms, 0s*)
2 bytes - calibration choice (0=Vertical*, 1=Trigger Level, 2=Delay, 3=Ch-ch-skew, 4=Balance Adjust, 5=Set to Default)
4 bytes - ch to ch skew for ch2 (-100ns...100ns, 0s*)

NOTE: All ch-ch-skew is based on ch1, the first channel in the topmost acq. board of the scope.

4 bytes - ch to ch skew for ch3 (2nd acq board)
4 bytes - ch to ch skew for ch4 (")
4 bytes - ch to ch skew for ch5 (3rd acq board)
4 bytes - ch to ch skew for ch6 (")
4 bytes - ch to ch skew for ch7 (4th acq board)
4 bytes - ch to ch skew for ch8 (")

710 bytes - used for internal computations

36 bytes - reserved
4 bytes - trigger count delay skew [17]
516 bytes - reserved
1 byte - flag indicates that v/div, offset, and trigger level settings changed due to the preset change [7]
1 byte - reserved
1 byte - flag which indicates if v/div is on 1/2/4 sequence or value in between [4]
1 byte - flag indicates that s/div or delay changed [1][2]
1 byte - flag which indicates if s/div is on 1/2/5 sequence or value in between [1]
1 byte - reserved
1 byte - flag which indicates if all patterns in pattern trigger mode are don't cares [20][22]

1 byte - display type
[15]
2 bytes - persist type
[15]
144 bytes - reserved

723 bytes - variables that are affected either by the "SCREEN CONFIGURATION" changes or by commands sent over the bus. For the allowed range and the default value of the same variables, see the "SCREEN CONFIGURATION" subsection.

600 bytes - these are channel specific variables and are allocated for 8 channels. When there are less than 8 channels in the oscilloscope, the first bytes should be used.

When you change the screen related variables in this subsection such as v/div, offset, trigger level, you should change the related variables in the "SCREEN CONFIGURATION" subsection only if current channel in the channel submenu or the trigger source (if trigger mode is EDGE) is the same as the source of the variable you are changing in this subsection.

**32(8*4) bytes - v/div of ch0...ch7
[4][7]
**8(8*1) bytes - v/div index
[4][7]
**8(8*1) bytes - voltage low range
[4][7]
**32(8*4) bytes - offset
[4][5][7]
**32(8*4) bytes - offset knob resolution
[4][7]
**32(8*4) bytes - offset resolution for offset DAC
[4][7]
**16(8*2) bytes - probe setting in integer
[8]
**32(8*4) bytes - probe setting in floating point
[8]
**8(8*1) bytes - preset choice
[7]
**32(8*4) bytes - v/div setting before TTL/ECL
[7]
**32(8*4) bytes - offset setting before TTL/ECL
[7]

**32(8*4) bytes - trigger level setting before TTL/ECL
[7]
**8(8*1) bytes - impedance choice
[6]
**8(8*1) bytes - Preamp Range Number
[4][7]
**32(8*4) bytes - trigger level
[4][5][7][10]
**32(8*4) bytes - trigger level resolution
[4][7]
**8(8*1) bytes - trigger slope
[12]
**8(8*1) bytes - trigger pattern (0=Low, 1=High,
2=Don't Care)
[22]
**8(8*1) bytes - trigger sensitivity (0=Low, 1=High)
[4][7]
**8(8*1) bytes - vertical port contents
[4][6][7][11][12][13][22]
**8(8*1) bytes - gain vernier port contents
[4][7]
**8(8*1) bytes - trigger level port contents
[4][5][7][10]
**8(8*1) bytes - offset port MSB contents
[4][5][7]
**8(8*1) bytes - offset port LSB contents
[4][5][7]
160 bytes - reserved

123 bytes - timebase board specific variables

1 byte - reserved
**1 byte - flag to offset/trigger action routine that
v/div changed
[4]
2 bytes - reserved
**4 bytes - s/div
[1]
4 bytes - reserved
**2 bytes - s/div index for the knob
[1]
**4 bytes - sample rate
[1][2]
**4 bytes - sample period
[1][2]
4 bytes - reserved
**4 bytes - delay times sample rate
[1][2]

**4 bytes - delay
[2]
4 bytes - reserved
**4 bytes - delay resolution
[1]
**4 bytes - post store
[1][2]
**1 byte - auto trigger choice
[14]
**1 byte - trigger mode
[13]
**2 bytes - trigger source
[11]
**4 bytes - trigger counts
[17]
**2 bytes - trigger "when" in pattern mode
[9]
2 bytes - probe setting of the external trigger source
**2 bytes - impedance choice of the external trigger source
[19]
**4 bytes - trigger level of the external trigger source
[18]
**1 byte - pattern setting of the external trigger source
[20]
**1 byte - trigger slope of the external trigger source
[21]
**2 bytes - display mode
[15]
**2 bytes - average number
[16]
**4 bytes - the number of samples displayed on screen
[1][2]
8 bytes - reserved
2 bytes - the whole number portion of the number of points per column
[1][2]
2 bytes - the fraction portion of the number of points per column
[1][2]
**1 byte - trigger level LSB port contents for the external trigger source
[18]
**1 byte - trigger level MSB port contents for the external trigger source
[18]
**1 byte - sample rate port 7 content
[1][2]
**1 byte - sample rate port 15 content
[1][2]
**1 byte - interpolator clock source port content
[1][2]

**1 byte - timebase port content
[1][2][6][19]
**1 byte - trigger mode port content
[9][11][12][13][20][21]
**1 byte - prestore LSB port
[1][2]
**1 byte - prestore MSB port
[1][2]
**1 byte - poststore LSB port
[1][2]
**1 byte - poststore MID port
[1][2]
**1 byte - poststore MSB port
[1][2]
**2 bytes - current channel in channel menu
[3]
23 bytes - reserved

2. "DATA " (24 times the number of acquisition boards) - The waveform data section cosists of the raw data (4K per channel) and the display data (8K per channel). Therefore, the data block size is 12K times the number of channels in the oscilloscope.

The raw data always contains the 4K valid data, whereas, the display data might not have all of the 4K valid data. The display data includes the data shown only on the oscilloscope screen (the latter part would be invalid). To find out how many display data are valid, look at the number of samples in the above "INTERNAL CONFIGURATION" subsection.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP