DESIGNED FOR HP-IB SYSTEMS

Programming Note

8566A,8568A/9835,9845-52 NOVEMBER 1981

SUPERCEDES: NONE

Subprogram Library
for the 8566A or 8568A Spectrum Analyzer with the 9835 or 9845 Desktop Computer

Volume 2 **Functions**
State

HEWLETT PACKARD

Required Reading . . .

8566A,8568A/9835,9845-1 Introductory Operating Guide
for the 8566A/8568A Spectrum Analyzers
with the 9835/9845 Desktop Computers
(P/N 5952-9356)

8566A,8568A/9835,9845-51 Subprogram Library
for the 8566A or 8568A Spectrum Analyzer
with the 9835 or 9845 Desktop Computer, Volume 1
(P/N 5952-9365)

Also of Interest . . .

Subprogram Library Tape Cartridge for the HP 8566A or 8568A
with the 9835 or 9845 (P/N 08566-10002 REV A)[1]

8566A Spectrum Analyzer Remote Operation (P/N 08566-90003)

8568A Spectrum Analyzer Remote Operation (P/N 08568-90003)[2]

8568A/9825A-99 Program Execution Time Information
for the 8568A/9825A Automatic Spectrum Analyzer System[3]

INTRODUCTION

The HP 8566A or 8568A Spectrum Analyzer can be controlled over HP-IB* with a computing controller such as the 9835 or 9845 Desktop Computer. Basic operation and programming of such automated spectrum analyzers is described in the 8566A,8568A/9835,9845-1 Introductory Operating Guide. The present series of Programming Notes, continuing here with Volume 2, describes a number of subprograms which extend the value of the spectrum analyzer's built-in firmware and make it easier for you to develop your own custom software.

The subprograms contained in Volume 2 allow a program to interrogate the spectrum analyzer to determine control settings and display modes, without disturbing the state of the analyzer. **Functions** can obtain information about the FUNCTION and COUPLED FUNCTION keys. **State** returns information about Trace, Marker, Sweep, Trigger, and various other states.

Equipment Required

Operation of the subprograms described in this library requires the following equipment:

1. 8566A or 8568A Spectrum Analyzer
2. 9835A/B Desktop Computer with 98332A I/O ROM, or 9845B/T Desktop Computer with 98412A I/O ROM (Opt. 312)
3. 98034A/B HP-IB Interface

Getting Started

The first step is to assemble the automatic spectrum analyzer components. Use the Introductory Operating Guide to help you set up and check-out the system, and begin programming.

To start programming for your specific signal analysis application, study the section entitled "Writing Programs" in Volume 1, which provides details on constructing programs with the subprograms presented in this series. It is assumed that you are familiar with the 8566A or 8568A Spectrum Analyzer Operation and Remote Operation manuals.

[1]REV A includes the Subprogram Library subroutines documented here in Volume 2, along with those previously documented in Volume 1.

[2]The 8568A Learn String is documented only in 8568A Remote manuals dated October, 1981, and later.

[3]Contains execution time data for the 8568A alone.

*Hewlett-Packard Interface Bus, the Hewlett-Packard implementation of IEEE STD 488-1978 and ANSI STD MC 1.1, "Digital Interface for Programmable Instrumentation".

Refer to the section in this volume entitled "Subprogram Descriptions" for detailed definitions of parameters and error codes, and examples of subprogram usage.

For advanced programming, the operation of each subprogram is discussed on a line-by-line basis. Also refer to the 8566A or 8568A Spectrum Analyzer Remote Operation manual for a summary of the contents of the analyzer's Learn String**.

Although the subprograms can be entered into the controller from the keyboard, it is recommended that you obtain the Subprogram Library Tape Cartridge P/N 08566–10002 which contains listings of the subprograms in the Subprogram Library. If you have this "master" cartridge, duplicate it onto a blank cartridge. Save the master cartridge as a backup copy and use the new copy as a "working" cartridge.

The subprograms and their respective file names as they appear on the Subprogram Library Tape Cartridge, are listed in the following table:

Table 1. Subprogram Library Files

File Name	Subprogram	Type	Externals*
FUNCTS	Functions	CALL	Error; Save
STATE	State	CALL	Error; Save

*EXTERNALS are other subprograms which may be called by this subprogram and which therefore must also be appended.

SUBPROGRAM DESCRIPTIONS

Functions
State

CALLS
Returns detailed information about analyzer's state.
Files: FUNCTS, STATE

Description

Frequently it is necessary for a piece of software to know one or more elements of the current state of the spectrum analyzer. For example, a subprogram to calculate the sensitivity (average noise level) of the analyzer must know the current values of frequency, attenuation and resolution bandwidth. To obtain these values, we must interrogate the analyzer. A human operator does this simply by reading the annotations displayed on the CRT. A program must issue a request via the controller, however. The general procedure is to activate the desired function, output an "OA", then read the value returned by the analyzer. This procedure cannot always be used, however. For example, activating one of the coupled functions (such as RB) will put that function in uncoupled operation. As the subprogram would not know whether the function was coupled or uncoupled when it was activated, it would not know whether or not to restore coupled operation after reading the value. Not all states of interest are available via the "OA" instruction, anyway. For example, in reading a value for RL, you will also need to know the units (dBm, dBmV, dBμV, or Volts).

The **Functions** and **State** subprograms are designed to provide this kind of information without disturbing the state of the analyzer. **Function** can obtain information about the FUNCTION and COUPLED FUNCTION keys. The information is organized in four groups, specified by the value of **Code**. See the description in the **Functions** parameter table.

Subprogram **State** returns information about graphics functions (such as Trace and Marker States), Sweep and Trigger states, and various miscellaneous states. See the description in the **State** parameter table.

**The 8568A Learn String is documented only in 8568A Remote manuals dated October, 1981, or later.

Parameters for Functions

Syntax: `CALL Functions(Code,P2,P3,P4,P5,P6,P7)`
Passed: **Code** selects category of function
Returned: P2,P3,P4,P5,P6,P7

Code

(passed) (returned)

1 FREQUENCY SCALE:
P2 = Center Frequency, Hz
P3 = Span, Hz
P4 = Start, Hz
P5 = Stop, Hz
P6 = Frequency Offset, Hz
P7 = Frequency Display Mode (0=CF/Span; 1=Start/Stop)

2 AMPLITUDE SCALE:
P2 = Ref Level Units (1=dBm; 2=dBmV; 3=dBμV; 4=Volts)
P3 = Reference Level Value
P4 = Reference Level Offset, dB
P5 = Amplitude Scale (0=Linear; 1,2,5 or 10=dB/div LOG)
P6 = Detection Mode (1=Normal; 2=Pos Peak; 3=Neg Peak; 4=Sample)
P7 = not used

3 COUPLED FUNCTIONS' VALUES:
P2 = Resolution Bandwidth, Hz
P3 = Video Bandwidth, Hz
P4 = Sweep Time, seconds
P5 = RF Attenuation, dB
P6 = Center Frequency Step Size, Hz
P7 = not used

4 COUPLED FUNCTIONS' STATES:
P2 = Resolution Bandwidth (0=Auto; 1=Uncoupled)
P3 = Video Bandwidth (0=Auto; 1=Uncoupled)
P4 = Sweep Time (0=Auto; 1=Uncoupled)
P5 = RF Attenuation (0=Auto; 1=Uncoupled)
P6 = Step Size (0=Auto; 1=Uncoupled)
P7 = not used

Example

The following program reads the center frequency (just set):

PROGRAM

```
170  OUTPUT 718;"CF222MZ"
180  CALL Functions(1,F,0,0,0,0,0)
190  DISP F
```

DISPLAY

```
222000000
```

Memory Required

2964 bytes

Error Codes

Functions-1: Parameter **Code** must be in range 1-4.

Parameters for State

Syntax: `CALL State(Code,P2,P3,P4,P5,P6,P7)`
Passed: **Code** selects category of state
Returned: P2,P3,P4,P5,P6,P7

Code

(passed) (returned)

1 MARKER STATES:
P2 = Marker mode (1 = Off; 2 = Normal; 3 = Delta; 4 = Zoom)
P3 = Signal Track (0 = Off; 1 = On)
P4 = Noise Level (0 = Off; 1 = On)
P5 = Counter Mode (0 = Off; 1 = On) (8568A ONLY!)
P6 = not used
P7 = not used

2 TRACE STATES:
P2 = A-Trace state (1 = CLR/WRT; 2 = MAX HOLD; 3 = VIEW; 4 = BLANK)
P3 = B-Trace state (1 = CLR/WRT; 2 = MAX HOLD; 3 = VIEW; 4 = BLANK)
P4 = C-Trace State (3 = View; 4 = Blank)
P5 = A-B Mode (0 = Off; 1 = On)
P6 = Video Avg. (0 = Off; non-zero = On, = # of samples avg'd)
P7 = Display Line (0 = Off; 1 = On)

3 MISCELLANEOUS:
P2 = Sweep Mode (1 = CONT; 2 = SINGLE)
P3 = Trigger Mode (1 = FREE RUN; 2 = LINE; 3 = EXT; 4 = VIDEO)
P4 = CORR'D State (0 = Off; 1 = On)
P5 = Instrument Check Lights (0 = Off; 1,2,3 = I,II,I + II On)
P6 = not used
P7 = not used

4 INPUT STATES (8568A ONLY!)
P2 = Input # (1; 2)
P3 = Input impedance, ohms (50; 75)
P4 = Preamp Gain, dB (for current input)
P5 = not used
P6 = not used
P7 = not used

Example

Use **State** to determine if any INSTRUMENT CHECK lights are on.

PROGRAM	DISPLAY
`180 CALL State(3,0,0,0,C,0,0)`	
`190 DISP C`	2

The value 2 indicates that the second INSTR CHK light is on.

Memory Required

2452 bytes

Error Code

State-1: Parameter Code must be in range 1-4 (8568A) or 1-3 (8566A).

Listing and Annotation for FUNCTIONS

```
101    ! Functions (File: FUNCTS)
102    ! HP 08566-10002, 810702
103    ! For HP 8566A or 8568A with System 35 or 45
104    ! CALL: Returns 8566A/8568A Function Values
105    !      Code=request code
106    !          (1-4 for freq, ampl, coupled fn vals, coupled fn states)
107    !      P2-P7=requested values (R)
108    ! EXTERNALS: Error; Save
109    !
110    SUB Functions(Code,P2,P3,P4,P5,P6,P7)
111        OPTION BASE 1
112        COM R,Sa,E$[20],L$[80]
113        DIM S$[15],P(2:7)
114        Code=INT(Code)
115        Ern=1
116        IF (Code<1) OR (Code>4) THEN Err
117        CALL Save(Analyzer)
118        Led_bytes=FNLstr(19,0)+FNLstr(18,-8)
119        S_ctr=1
120        IF (Code=2) OR (Code=4) THEN Code2
121        S$="CF SP FA FB KSV"
122           FOR S=2 TO 6*(Code=1)+3*(Code=3)
123           P(S)=FNRd_act(S_ctr,S$)
124           NEXT S
125        Step_size=P(3)/10
126        OUTPUT Sa;"FA"
127        P7=Start_stop=BIT(Led_bytes,0)
128        IF NOT Start_stop THEN OUTPUT Sa;"CF"
129 Code2: IF Code<>2 THEN Code3
130        IF Analyzer=8568 THEN C2_68
131 C2_66:  IF FNLstr(26,7) THEN Log
132           P(2)=1+BINAND(FNLstr(65,2),3)
133           GOTO Rf_lvl
134 Log:   P(2)=1+FNLstr(64,6)
135        S$="LG "
136        P(5)=FNRd_act(1,S$)
137 Rf_lvl: S$="RL KSZ"
138           FOR S=3 TO 4
139           P(S)=FNRd_act(S_ctr,S$)
140           NEXT S
141        GOTO C2_p6
142 C2_68: P(2)=1+FNLstr(72,6)
143        S$="RL KSZLG "
144           FOR S=3 TO 5
145           P(S)=FNRd_act(S_ctr,S$)
146           NEXT S
147        P(5)=P(5)*FNLstr(26,7)
148        IF NOT P(5) THEN OUTPUT Sa;"LN"
149 C2_p6: S=1+FNLstr(29,3)
```

101 – 107: Description.
108: Subroutines **Error** and **Save** required.
110: Define subroutine subprogram **Functions** with parameters **Code,P2,P3,P4,P5,P6** and **P7**.
111: Declare array indexing to being at 1.
112: Declare COMmon storage.
113: DIMension **S$** up to 15 characters, then array **P** for indices from 2 to 7.
114: Retain only the integer portion of **Code**.
115: Set error code number **Ern** = 1.
116: If **Code** is illegal, report error.
117: Call **Save** to obtain the current analyzer state in the **L$** string.
118: Function subroutine **Lstr** (in the SA/RE file), returns the decimal value of the byte specified. The Learn String coding is detailed in the Spectrum Analyzer Remote Operation manual.* Learn String bytes 18 and 19 are LED conditions of the RF Section of the analyzer. The variable **Led_bytes** is a 16 bit word with the LED states; this saves the states of the coupled functions so they can be restored later, if necessary.
119: Set **S_ctr,** the **S$** string place counter, to 1.
120: If **Code** is set for amplitude scale (**Code** = 2) or coupled function states (**Code** = 4), go to line 129.
121: To read frequency scale values (**Code** = 1) or coupled functions values (**Code** = 3), start here. Load **S$** string with analyzer command codes for main front panel FUNCTIONS and frequency offset.
122 – 124: Return active function values into **P**—array. If the frequency scale values are requested (**Code** = 1), read CF SP FA FB and KSV values into **P(2)—P(6),** respectively. If coupled functions values are requested (**Code** = 3), read only CF and SP into **P(2)** and **P(3),** respectively. (SP will be needed to determine CF step size if step size is in AUTO).
125: Calculate Center Frequency Step Size for AUTO.
126: Force Start/Stop frequency scale label mode.
127 – 128: Read frequency scale mode from bit 0 of **Led_bytes** (derived from the Learn String in line 117), and save as **P7** and **Start_stop.** If **Start_stop** = 0, scale should be Center Frequency/Span; set this by making Center Frequency the active function.
129: Go to line 152 unless amplitude scale values requested.
130: If **Save** return parameter **Analyzer** (line 117) represents an 8568A, then go to line 142. Amplitude scale values are stored differently in Learn Strings from 8566A's and 8568A's.
131: An 8566A analyzer is assumed. If in LOG scale (bit 7 of byte 26 is 1), go to line 134.
132 – 133: The analyzer is in Linear scale mode. Shift Learn String byte 65 two bits right and extract bits 0 and 1 (originally bits 2 and 3). These are the Ref Level Units bits for Linear scaling. Add one to the value of these bits to create a value in the range 1-4. Go to line 137.
134: The analyzer is in Log scale mode. Shift Learn String byte 64 six bits right and extract the two lower bits (original bits 6 and 7 will be bits 0 and 1). These are the Ref Level Units bits for Log scaling. Add 1 to the value of these bits to create a value in the range 1-4.
135 – 136: Store Log scale command code "LG" in **S$,** and use function subprogram **Rd_act** (see line 180) to make Log Scale the active function; read the current scale value**P(5)**.
137: Store command codes "RL" and "KSZ" in **S$**.
138 – 140: Use function subprogram **Rd_act** (see line 180) to read Ref Level and Ref Level Offset into **P(3)** and **P(4)**.
141: Go to line 149.
142: An 8568A analyzer is assumed. Read amplitude scale values (**Code** = 2). The reference level units code is read from the Learn String (bit 6 of byte 72); add 1 to create a value in the range 1-4.
143: Store command codes "RL", "KSZ", and "LG" in **S$**.
144 – 146: Read Ref Level, Ref Level Offset, and scale value into **P(3)—P(5)**, respectively, using **Rd_act** (see line 180).
147: The Learn String is read (bit 7 of byte 26) to determine whether the analyzer was in Log or Linear scale (1=Log, 0=Linear). If Linear, **P(5)** is set to 0. Otherwise, the current value of **P(5)** represents dB/division and is left intact.
148: If Linear mode, return the analyzer to Linear scaling.
149: Calculate the trace detection mode bit field plus 1, and store in **S** for use as a pointer.

*All 8566A Remote manuals, and 8568A Remote manuals dated October, 1981, and later.

```
150        S$="324_1"
151        P(6)=VAL(S$[S;1])
152 Code3:    IF Code<>3 THEN Code4
153        S$="RB VB ST AT SS "
154        S_ctr=1
155           FOR S=2 TO 6
156           P(S)=FNRd_act(S_ctr,S$)
157           NEXT S
158      IF BIT(Led_bytes,10)=0 THEN P(6)=Step_size
159        S$="RVTAS"
160           FOR S=1 TO 5
161           IF BIT(Led_bytes,S+5)=0 THEN OUTPUT Sa;"C"&S$[S;1]
162           NEXT S
163 Code4:    IF Code<>4 THEN Exit
164           FOR S=2 TO 6
165           P(S)=BIT(Led_bytes,S+4)
166           NEXT S
167 Exit: OUTPUT Sa;"HD"
168        P2=P(2)
169        P3=P(3)
170        P4=P(4)
171        P5=P(5)
172        P6=P(6)
173        SUBEXIT
174 Err:    E$="Functions"
175         CALL Error(Ern)
176         PAUSE
177     SUBEND
178     !
179     !
180     DEF FNRd_act(Index,S$)
181         OPTION BASE 1
182         COM R,Sa,E$[20],L$[80]
183 Rd_act: OUTPUT Sa;S$[Index,Index+2]&"OA"
184         ENTER Sa;Val
185         Index=Index+3
186         RETURN Val
187     FNEND
188     ! End Functions
189     !
190     !
191     ! END OF PROGRAM
```

150–151: Load "324—1" into string **S$** as a look-up table, and convert the **S** value to the correct value for the trace detection code.
152: Go to line 163 unless coupled function values requested.
153: Put coupled function commands into string **S$**.
154: Reset **S_ctr** to 1.
155–157: Put RB, VB, ST, AT, and SS values into **P(2)—P(6)** by reading the active function with **Rd_act** (see line 180).
158: If center frequency step size is in AUTO mode (BIT(**Led_bytes,** 10)=0), put **Step_size** into **P(6)**.
159: Put command codes for coupled function AUTO conditions into **S$**. (Each letter in the string will be preceded by a "C" in the command.)
160–162: The coupled functions are returned to their original states as recorded in **Led_bytes** (from line 118).
163: Go to line 167 unless coupled function states requested.
164–166: Read each of the coupled function LED bits in **Led_bytes** into **P(2)—P(6)**.
167: Disable the active function.
168–172: Load **P(2)—P(6)** values into return parameters **P2—P6**.
173: Return to calling program segment.
174–176: Send error message number **Ern** and halt program execution.
177: End of subroutine subprogram **Functions**.
180: Define system function subprogram **Rd_act** which reads the function activated by sending **Index**.
181: Declare array indexing to begin at 1.
182: Declare COMmon storage.
183: Transmit the indicated 3 character substring followed by "OA".
184: Read the active function value into **Val**.
185: Increment **Index** so that repeated calls to **Rd_act** will return values for the sequence of functions stored in **S$**.
186: Return **Val** to calling program segment.
187: End of system function subprogram **Rd_act**.

Listing and Annotation for STATE

```
101    ! State  (File: STATE)
102    ! HP 08566-10002, 810702
103    ! For HP 8566A or 8568A with System 35 or 45
104    ! CALL: returns 8566A/8568A Control State
105    !    Code = request code (1,2,3 = mrkr fns, traces, misc)
106    !                8568A ONLY: (4 = input states)
107    !   P2-P7 = requested values (R)
108    ! EXTERNALS: Error; Save
109    !
110    SUB State(Code,P2,P3,P4,P5,P6,P7)
111         OPTION BASE 1
112         COM R,Sa,E$[20],L$[80]
113         DIM S$[8]
114         CALL Save(Analyzer)
115         Code=INT(Code)
116         Ern=1
117         IF Code<1 THEN Err
118         IF (Analyzer=8566) AND (Code>3) THEN Err
119         IF (Analyzer=8568) AND (Code>4) THEN Err
120         Rf_leds=FNLstr(19,0)+FNLstr(18,-8)
121         Ds_leds=FNLstr(21,0)+FNLstr(20,-8)
122 Code_1:  IF Code<>1 THEN Code_2
123            P2=MAX(1,FNLstr(63,0)-16)
124            IF P2>4 THEN P2=P2-3
125            P3=BIT(Rf_leds,14)
126            P4=BIT(Rf_leds,3)
127            IF Analyzer=8568 THEN P5=BIT(Rf_leds,13)
128 Code_2:  IF Code<>2 THEN Code_3
129          S$="70128345"
130          FOR S=1 TO 8
131            IF NOT BIT(Ds_leds,VAL(S$[S,S])) THEN Next_s
132            IF S<5 THEN P2=S
133            IF S>4 THEN P3=S-4
134 Next_s:  NEXT S
135          OUTPUT Sa;"DA307201DR"
136          ENTER Sa;P4
137          P4=3+(P4=1044)
138          P5=BIT(Ds_leds,6)
139          P7=BIT(Rf_leds,4)
140          P6=BIT(FNLstr(73,0),2)
141          IF NOT P6 THEN Code_3
142          OUTPUT Sa;"KSGOA"
143          ENTER Sa;P6
144 Code_3:  IF Code<>3 THEN Code_4
145          P2=BIT(Ds_leds,10)+1
146          P3=1
147          FOR S=1 TO 3
148            P3=P3+S*BIT(Ds_leds,S+10)
149          NEXT S
150          OUTPUT Sa;"DA219201DR"
151          ENTER Sa;P4
152          P4=(P4<>145)
```

101–107: Description.
108: Subroutines **Error** and **Save** required.
110: Define subroutine subprogram **State** with parameters **Code,P2,P3,P4,P5,P6** and **P7**.
111: Declare array indexing to begin at 1.
112: Declare COMmon storage.
113: DIMension **S$** up to 8 characters.
114: Call **Save** to obtain the current analyzer state in the **L$** string.
115: Retain only the integer portion of **Code**.
116: Set error code number **Ern** = 1.
117–119: If **Code** is illegal, report error. (The legal range for **Code** is different for 8566A's and 8568A's.)
120: Function subroutine **Lstr** (in the SA/RE file) returns the decimal value of the byte specified. The Learn String coding is detailed in the Spectrum Analyzer Remote Operation manual‡. Learn String bytes 18 and 19 are LED conditions of the RF section of the analyzer. The variable **Rf_leds** is a 16 bit word with the LED states; this saves the states of the coupled functions so they can be restored later if necessary.
121: Bytes 20 and 21 (LED conditions of the Display Section) are packed into a second 16 bit word, **Ds_leds**.
122: If **Code** is not set for Marker states, go to line 128.
123: Get the marker state byte from the learn string (byte 63). The possible values for this byte are 0, and 18-20. (The 8568A also allows values 21-23 which represent marker counter states.) Subtract 16 to convert the 18-20 range to 2-4 (and the 21-23 range to 5-7); replace 0 with 1, using the max function.
124: In the case of the 8568A, **P2** may now be in the range 2-7. if **P2** is greater than 4, a marker counter state is indicated. Replace **P2** with (**P2**-3) to shift the value into the 2-4 range.
125: Bit 14 of **Rf_leds** is the SIGNAL TRACK state; enter into **P3**.
126: Bit 3 of **Rf_leds** is the NOISE MARKER state; enter into **P4**.
127: In case of the 8568A, bit 13 of **Rf_leds** is the FREQ COUNT state; enter into **P5**.
128: If **Code** is not set for Trace states, go to line 144.
129: Construct a bit pointer table in string **S$**. The table consists of two fields of four numbers each. The first four numbers specify the location of the proper bit in **Ds_leds** for each of the four Trace A states: CLEAR-WRITE A, MAX HOLD A, VIEW A, BLANK A. The numbers in the second field similarly specify the bit positions in **Ds_leds** for Trace B's state data.
130–134: Test the contents of the bit positions specified by the pointer table in **S$**. One of the four Trace A bits, and one of the four Trace B bits will be set; the other bits will be 0. A Value of **S** from 1 to 4 is the correct value for Trace A, and the one for which the bit is set is returned as **P2**. A Value of **S** from 5 to 8 corresponds to Trace B; subtract 4 from the one for which the bit is set, and return as **P3**.
135–136: Set the display address to the control word for Trace C (first word of page 4 of the trace memory). Specify O1 format, and read the word into **P4**.
137: Set **P4** = 4 if the control word is 1044 (End of Memory), indicating Trace C is in BLANK. Otherwise, the trace is assumed in VIEW, so **P4** = 3.
138: Return bit 6 of **Ds_leds** (A-B Mode) as **P5**.
139: Return bit 4 of **Rf_leds** (Display Line state) as **P7**.
140–143: Return bit 2 of byte 73 (Video Averaging state) as **P6**. If Video Averaging is on, read the number of samples averaged and return as **P6** instead.
144: If **Code** is not set for Miscellaneous states, go to line 154.
145: Read bit 10 of **Ds_leds** (Single Sweep bit). Add 1 so that 1,2, correspond to S1, S2 programming codes. Return as **P2**.
146: Assume Free Run TRIGGER mode: **P3** = 1.
147–149: If bit 11, 12, or 13 of **Ds_leds** is set, increment **P3** by 1, 2, or 3 to make **P3** correspond to T1, T2, T3, T4 programming codes.
150–151: Set the display address to the display control word in page 3 of the trace memory that enables the CORR'D message. Read the word in O1 format into **P4**.
152: Set **P4** = 0 (CORR'D off) if the word is 145 (control word to skip to next 16 block), else **P4** = 1.

‡All 8566A Remote manuals, and 8568A Remote manuals dated October, 1981, and later.

```
153              P5=BIT(Rf_leds,12)+2*BIT(Rf_leds,11)
154 Code_4:      IF Code<>4 THEN Clr_act_fnct
155              P2=BIT(FNLstr(23,0),3)+1
156              S$="KS<"
157              IF P2=2 THEN S$="KS>"
158              OUTPUT Sa;S$,"OA"
159              ENTER Sa;P4
160              P3=50+25*BIT(FNLstr(72,0),4)
161 Clr_act_fnct: OUTPUT Sa;"HD"
162              SUBEXIT
163 Err:     E$="State"
164          CALL Error(Ern)
165          PAUSE
166      SUBEND
167      ! End State
168      !
169      !
170      ! END OF PROGRAM
```

153: Merge the contents of bit 12 (Instrument Check Light I) and bit 11 (Instrument Check Light II) of **Rf_leds**, and return Instrument Check Light status as **P5**.
154: If **Code** is not set for Input state (8568A only!), go to line 161.
155: Read bit 3 of byte 23 (Input number); add 1 to convert the 0,1 to 1,2 and return as **P2**.
156–157: Depending on **P2**, set either "KS<" or "KS>" into **S$**.
158–159: Output the selected string, and read the preamp gain into **P4**.
160: Read bit 4 by byte 72 (input impedance bit). If the bit is 0, set **P3** to 50 (ohms); if the bit is 1, set **P3** to 75 (ohms).
161: Clear "active function" display area.
162: Return to calling program segment.
163–165: Send error message number **Ern** and halt program execution.
166: End of subroutine subprogram **State**.

For more information, call your local HP Sales Office or nearest Regional Office: Eastern (201) 265-5000; Midwestern (312) 255-9800; Southern (404) 955-1500; Western (213) 970-7500; Canadian (416) 678-9430. Ask the operator for instrument sales. Or write Hewlett-Packard, 1501 Page Mill Road, Palo Alto, CA 94304. In Europe: Hewlett-Packard S.A., 7, rue du Bois-du-Lan, P.O. Box, CH 1217 Meyrin 2, Geneva, Switzerland. In Japan: Yokogawa-Hewlett-Packard Ltd., 29-21, Takaido-Higashi 3-chome, Suginami-ku, Tokyo 168.