Programming Reference Guide

HP 8923B DECT Test Set
Notices

Subject Matter
Information contained in this document is subject to change without notice.
All Rights Reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

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

Warranty
This Hewlett-Packard instrument product is warranted against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by HP. Buyer shall prepay shipping charges to HP and HP shall pay shipping charges, duties, and taxes for products returned to HP from another country.

HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

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

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
Exclusive Remedies
THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance
Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.
This material may be reproduced by or for the U.S. Government pursuant to the Copyright License under the clause at DFARS 52.227-7013 (APR 1988).

Hewlett-Packard Company
Learning Products Department
Station Road,
South Queensferry
West Lothian
EH30 9TG
Scotland, UK

Ordering Information
To order this manual, call or write your nearest Hewlett-Packard Sales office. Within the USA, it is better to order directly from the HP Support Materials Organization in Roseville, California. Ask your nearest HP office for information and forms for the “Direct Order System.”
# Sales and Service Offices

<table>
<thead>
<tr>
<th>REGIONAL SALES AND SERVICE OFFICES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SOUTH EAST EUROPE</strong></td>
</tr>
<tr>
<td>Hewlett-Packard S.A.</td>
</tr>
<tr>
<td>World Trade Center</td>
</tr>
<tr>
<td>110 Avenue Louis-Casai</td>
</tr>
<tr>
<td>1215 Cointrin, GENEVA</td>
</tr>
<tr>
<td>Switzerland</td>
</tr>
<tr>
<td>Tel: (022) 98 96 51</td>
</tr>
<tr>
<td>Telex: 27225 hpser</td>
</tr>
<tr>
<td>Mail Address:</td>
</tr>
<tr>
<td>P.O. Box</td>
</tr>
<tr>
<td>CH-1217 Meyrin 1</td>
</tr>
<tr>
<td>GENEVA</td>
</tr>
<tr>
<td>Switzerland</td>
</tr>
<tr>
<td><strong>MIDDLE EAST AND</strong></td>
</tr>
<tr>
<td><strong>CENTRAL AFRICA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard S.A.</td>
</tr>
<tr>
<td>Middle East/Central</td>
</tr>
<tr>
<td>Africa Sales H.Q.</td>
</tr>
<tr>
<td>T. rue du Bois-du-Lan</td>
</tr>
<tr>
<td>P.O. Box 364</td>
</tr>
<tr>
<td>CH-1217 Meyrin 1</td>
</tr>
<tr>
<td>GENEVA, Switzerland</td>
</tr>
<tr>
<td>Tel: (022) 7807111</td>
</tr>
<tr>
<td>Telex: 27833 hmeac ch</td>
</tr>
<tr>
<td><strong>ASIA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Asia Ltd.</td>
</tr>
<tr>
<td>17-21/F Shell Tower, Time Square,</td>
</tr>
<tr>
<td>1 Matheson Street, Causeway Bay,</td>
</tr>
<tr>
<td><strong>HONG KONG</strong></td>
</tr>
<tr>
<td>Tel: (852) 556 7070</td>
</tr>
<tr>
<td>Cable: HFASIAL TD</td>
</tr>
<tr>
<td><strong>NORTHERN EUROPE</strong></td>
</tr>
<tr>
<td>Hewlett-Packard S.A.</td>
</tr>
<tr>
<td>V.D. Hooplaan 241</td>
</tr>
<tr>
<td>NL-1105 LN AMSTELVEEN</td>
</tr>
<tr>
<td>The Netherlands</td>
</tr>
<tr>
<td>Tel: 20 3473632</td>
</tr>
<tr>
<td>Telex: 15819 hpser</td>
</tr>
<tr>
<td><strong>UNITED KINGDOM</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Ltd.</td>
</tr>
<tr>
<td>Nine Mile Ride</td>
</tr>
<tr>
<td>WOKINGHAM</td>
</tr>
<tr>
<td>Berkshire, RG113LL</td>
</tr>
<tr>
<td>Tel: 0344 770100</td>
</tr>
<tr>
<td><strong>OTHER INTERNATIONAL AREAS</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Co.</td>
</tr>
<tr>
<td>Intercontinental</td>
</tr>
<tr>
<td>Headquarters</td>
</tr>
<tr>
<td>3485 Deer Creek Road</td>
</tr>
<tr>
<td>PALO ALTO, CA 94304</td>
</tr>
<tr>
<td>Tel: (415) 857-5027</td>
</tr>
<tr>
<td>Telex: 034 S800</td>
</tr>
<tr>
<td>Cable: HEPACK</td>
</tr>
<tr>
<td><strong>CANADA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard (Canada) Ltd.</td>
</tr>
<tr>
<td>6577 Gerey Drive</td>
</tr>
<tr>
<td>MISSISSAUGA, Ontario L4V 1MS</td>
</tr>
<tr>
<td>Tel: (416) 676-0430</td>
</tr>
<tr>
<td>V Mail: (416) 678-8533</td>
</tr>
<tr>
<td><strong>EASTERN EUROPE</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Ges. m.b.h.</td>
</tr>
<tr>
<td>Liebigasse 1</td>
</tr>
<tr>
<td>P.O. Box 72</td>
</tr>
<tr>
<td>A-1222 VIENNA, Austria</td>
</tr>
<tr>
<td>Tel: (222) 2500-0</td>
</tr>
<tr>
<td>Telex: 1 3 4425 HEPA A</td>
</tr>
<tr>
<td><strong>EASTERN USA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Co.</td>
</tr>
<tr>
<td>4 Choice Cherry Road</td>
</tr>
<tr>
<td>ROCKVILLE, MD 20850</td>
</tr>
<tr>
<td>Tel: (301) 670-4300</td>
</tr>
<tr>
<td><strong>MIDWESTERN USA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Co.</td>
</tr>
<tr>
<td>6201 Tidview Drive</td>
</tr>
<tr>
<td>ROLLING MEADOWS IL, 6000S</td>
</tr>
<tr>
<td>Tel: (708) 255-9800</td>
</tr>
<tr>
<td><strong>SOUTHERN USA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Co.</td>
</tr>
<tr>
<td>2015 South Park Place</td>
</tr>
<tr>
<td>ATLANTA, GA 30339</td>
</tr>
<tr>
<td>Tel: (404) 955-1500</td>
</tr>
<tr>
<td><strong>SOUTHERN USA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Co.</td>
</tr>
<tr>
<td>2015 South Park Place</td>
</tr>
<tr>
<td>ATLANTA, GA 30339</td>
</tr>
<tr>
<td>Tel: (404) 955-1500</td>
</tr>
<tr>
<td><strong>WESTERN USA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Co.</td>
</tr>
<tr>
<td>5161 Lancershims Blvd.</td>
</tr>
<tr>
<td>NORTH HOLLYWOOD, CA 91601</td>
</tr>
<tr>
<td>Tel: (818) 505-5875</td>
</tr>
<tr>
<td><strong>WESTERN USA</strong></td>
</tr>
<tr>
<td>Hewlett-Packard Co.</td>
</tr>
<tr>
<td>5161 Lancershims Blvd.</td>
</tr>
<tr>
<td>NORTH HOLLYWOOD, CA 91601</td>
</tr>
<tr>
<td>Tel: (818) 505-5875</td>
</tr>
<tr>
<td><strong>UNITED STATES OF</strong></td>
</tr>
<tr>
<td>AMERICA</td>
</tr>
<tr>
<td>Customer Information Center</td>
</tr>
<tr>
<td>Tel: (800) 752-0900</td>
</tr>
<tr>
<td>6:00 AM to 5:00 PM</td>
</tr>
<tr>
<td>Pacific Time</td>
</tr>
</tbody>
</table>
## Contents

1. **Introduction**  
   - Relevant Documents ........................................ 1-1  
   - Configuration .............................................. 1-1  

2. **Command Guidelines**  
   - Command Names .............................................. 2-1  
   - Optional Parts of Commands .............................. 2-1  
   - Command Punctuation ..................................... 2-2  
   - Using Quotes for String Entries ......................... 2-2  
   - Using Spaces ............................................... 2-2  
   - Example ...................................................... 2-2  
   - Command Structure ......................................... 2-2  
   - Using Semicolons for Multiple Commands ................ 2-2  
   - Using Question Marks to Query ........................... 2-3  
   - Specifying Units of Magnitude ............................ 2-3  
   - Syntax Diagrams ............................................. 2-3  

3. **Equivalent Front Panel Key Commands**  
   - Front Panel Keys ............................................ 3-1  
   - (SHIFT), (CANCEL), Cursor Knob .......................... 3-1  
   - DATA Keys ................................................... 3-1  
   - DATA FUNCTION Keys ....................................... 3-1  
   - INSTRUMENT STATE Keys .................................. 3-3  
   - SCREEN CONTROL Keys .................................... 3-4  
   - USER Keys .................................................... 3-4  

4. **Command Summary**

5. **IEEE 488.2 Common Commands**  
   - *CLS (Clear Status) ......................................... 5-2  
   - *ESE (Event Status Enable) ............................... 5-3  
   - *ESR? (Event Status Register) ............................ 5-5  
   - *IDN? (Identification Number) ............................ 5-6  
   - *OPC (Operation Complete) ............................... 5-7  
   - *OPT? ......................................................... 5-8  
   - *PCB ............................................................ 5-9  
   - *RCL (Recall) ............................................... 5-10  
   - *RST (Reset) ................................................ 5-11  
   - *SAV (Save) .................................................. 5-12  
   - *SRE (Service Request Enable) ............................ 5-13  
   - *STB? (Status Byte) ...................................... 5-14  
   - *TRG (Trigger) ............................................... 5-15  
   - *TST? (Test) ................................................. 5-16  
   - *WAI (Wait) .................................................. 5-17
6. **Status Registers**
   - Status Byte ............................................ 6-3
   - Status Registers ....................................... 6-4
     - OPERAtion Status Register ......................... 6-4
     - Standard Event Status Register .................. 6-5
     - QUEStionable Status Register .................... 6-6
     - CALibration Status Register ........................ 6-7
     - HARDware1 Status Register ......................... 6-8
     - HARDware2 Status Register ......................... 6-9
     - COMMunicate Status Register ...................... 6-10

7. **AFAalyzer Subsystem**
   - AFAalyzer:INPut ....................................... 7-2
   - AFAalyzer:VOLTage ..................................... 7-3

8. **AGenerator Subsystem**
   - AGGenerator[:STATE] .................................. 8-2
   - AGGenerator:AMPLitude ............................... 8-3
   - AGGenerator:FREQuency ............................... 8-4
   - AGenerator:TRANsmit ................................. 8-5
   - AGenerator:TX ......................................... 8-6
   - AGenerator:VARIABLE:FREQuency ..................... 8-7

9. **BETest Subsystem**
   - BETest:BITS .............................................. 9-2
   - BETest:BITS:INCREment ............................... 9-3
   - BETest:WERRor:CRITerion ............................. 9-4

10. **CONFigure Subsystem**
    - CONFigure:BADDress ................................ 10-4
    - CONFigure:BEEPer .................................... 10-5
    - CONFigure:BMODE ..................................... 10-6
    - CONFigure:DATE ...................................... 10-7
    - CONFigure:DATE:YEAR ................................ 10-8
    - CONFigure:DATE:MONTH ................................ 10-9
    - CONFigure:DATE:DAY .................................. 10-10
    - CONFigure:DECT:MSYNc ............................... 10-11
    - CONFigure:EDISk ..................................... 10-12
    - CONFigure:INTensity ................................ 10-13
    - CONFigure:PRINT:ADDress ......................... 10-14
    - CONFigure:PRINT:FFENd .............................. 10-15
    - CONFigure:PRINT:FFSStart ......................... 10-16
    - CONFigure:PRINT:LINES ................................ 10-17
    - CONFigure:PRINT:PORTs .............................. 10-18
    - CONFigure:PRINT:DESTination ...................... 10-19
    - CONFigure:PRINT:TITLe .............................. 10-20
    - CONFigure:SPORT:BAUD ............................... 10-21
    - CONFigure:SPORT:DATA ................................ 10-22
    - CONFigure:SPORT:IBECho ........................... 10-23
    - CONFigure:SPORT:IECHO ............................. 10-24
    - CONFigure:SPORT:PArity ............................ 10-25
    - CONFigure:SPORT:RPACE ............................. 10-26
    - CONFigure:SPORT:SNPut ............................ 10-27
    - CONFigure:SPORT:STOP ............................... 10-28
    - CONFigure:SPORT:XPACe ............................. 10-29
11. DECT Subsystem

DECT:DUMMy[:STATE] .................................................. 11-4
DECT:DUMMy:SYNC:ABOrt ........................................... 11-5
DECT:DUMMy:SYNC[:IMMediate] ................................. 11-6
DECT:EUT .................................................................. 11-7
DECT:EUT:PAri? .......................................................... 11-8
DECT:EUT:PMID? .......................................................... 11-9
DECT:FP:DUMMy:CARrIer? ........................................... 11-10
DECT:FP:DUMMy:SLOT? ................................................. 11-11
DECT:FP:DUMMy:SYNC:ABOrt .................................... 11-12
DECT:FP:TRAffIc:CARrIer ............................................ 11-14
DECT:FP:TRAffIc:CARrIer:INCemenT ........................ 11-15
DECT:FP:TRAffIc:CONNect .......................................... 11-16
DECT:FP:TRAffIc:REIease ........................................... 11-17
DECT:FP:TRAffIc:SLOT ............................................... 11-18
DECT:FP:TRAffIc:SLOT:INCemenT ................................ 11-19
DECT:LOGIing[:STATE] ............................................... 11-20
DECT:LOGIing:SPORt:BAUD ....................................... 11-21
DECT:LOGIing:SPORt:HANDshakE ......................... 11-22
DECT:PAri ................................................................. 11-23
DECT:PMID ................................................................. 11-24
DECT:PP:DUMMy[:STATE] ............................................ 11-25
DECT:PP:DUMMy:CARrIer ........................................... 11-26
DECT:PP:DUMMy:CARrIer:INCemenT ....................... 11-27
DECT:PP:DUMMy:SLOT ............................................... 11-28
DECT:PP:DUMMy:SLOT:INCemenT ............................ 11-29
DECT:PP:TRAffIc:CARrIer ............................................ 11-30
DECT:PP:TRAffIc:CARrIer:INCemenT ....................... 11-31
DECT:PP:TRAffIc:CONNect .......................................... 11-32
DECT:PP:TRAffIc:REIease ........................................... 11-33
DECT:PP:TRAffIc:SLOT ............................................... 11-34
DECT:PP:TRAffIc:SLOT:INCemenT ............................ 11-35
DECT:STATUs? .......................................................... 11-43
DECT:SYNC[:IMMediate] ............................................. 11-44
DECT:SYNC:ABOrt ..................................................... 11-45
DECT:TRAffIc:CONNect ............................................. 11-46
DECT:TRAffIc:REIease .............................................. 11-47
DECT:VOIcE:DESTInation ........................................... 11-48
12. DISPLAY Subsystem
DISPLAY:SCREEN ........................................ 12-2
DISPLAY:BETest ........................................ 12-3
DISPLAY:BETest:VIEW ................................... 12-4
DISPLAY:FREQuency .................................... 12-5
DISPLAY:OSCilloscope .................................. 12-6

13. ESource Subsystem
ESource:POWer:ADVance ................................. 13-2
ESource:POWer:OPosition ............................... 13-3
ESource:SLOT ........................................... 13-4
ESource:SLOT:INCrement ............................... 13-5
ESource:STATe .......................................... 13-6
ESource:PATTern ....................................... 13-7

14. MEASure Subsystem
MEASure:AUdio:ACVolts? .............................. 14-5
MEASure:AUdio:DCVolts? .............................. 14-6
MEASure:AUdio:FREQuency? ......................... 14-7
MEASure:AUdio:OSCilloscope:MARk:LEVel:VOLts? .... 14-8
MEASure:AUdio:OSCilloscope:TRACE? .................. 14-10
MEASure:BETest:BERRor:COUNt? .................... 14-11
MEASure:BETest:BERRor:ICount? ...................... 14-12
MEASure:BETest:BERRor:IRATio? ..................... 14-13
MEASure:BETest:BERRor:RATio? ...................... 14-14
MEASure:BETest:BITest? ............................ 14-16
MEASure:BETest:ITWTested? ........................ 14-17
MEASure:BETest:WERRor:COUNt? .................... 14-18
MEASure:BETest:WERRor:ICount? ...................... 14-19
MEASure:BETest:WERRor:IRATio? ..................... 14-20
MEASure:BETest:WERRor:RATio? ...................... 14-21
MEASure:MODE ........................................ 14-23
MEASure:PACKet ..................................... 14-24
MEASure:PATTern .................................. 14-25
MEASure:PATTern:DBField ............................ 14-26
MEASure:RF:FREQuency:ACCuracy? ........................ 14-27
MEASure:RF:FREQuency:COMposite? ........................ 14-28
MEASure:RF:FREQuency:DEViation:ONE:BAverage? ........................ 14-29
MEASure:RF:FREQuency:DEViation:ZERO:BAverage? ........................ 14-32
MEASure:RF:FREQuency:DRIFTy? ........................ 14-35
MEASure:RF:NTPower? ................................ 14-36
MEASure:RF:PTRm:MARker:LEVel:FALL? ........................ 14-37
MEASure:RF:PTRm:MARker:LEVel:MID? ........................ 14-38
MEASure:RF:PTRm:MARker:LEVel:RISE? ........................ 14-39
MEASure:RF:PTRm:MARker:TIMe:FALL? ........................ 14-40
MEASure:RF:PTRm:MARker:TIMe:MID? ........................ 14-41
MEASure:RF:PTRm:MARker:TIMe:RISE? ........................ 14-42
MEASure:RF:PTRm:MASK:FALL? ........................ 14-43
15. OSCilloscope Subsystem
   OSCilloscope:MARKer:NPEak .............................. 15-3
   OSCilloscope:MARKer:PPEak .............................. 15-4
   OSCilloscope:MARKer:POSITION ........................... 15-5
   OSCilloscope:SCALE:TIME ................................. 15-6
   OSCilloscope:SCALE:VERTical:OFFSet .................... 15-7
   OSCilloscope:SCALE:VERTical:VOLts ...................... 15-8
   OSCilloscope:TRIGger:LEVEL .............................. 15-9
   OSCilloscope:TRIGger:LEVEL:INCRement ................. 15-10
   OSCilloscope:TRIGger:MODE ............................... 15-11
   OSCilloscope:TRIGger:PRETrigger ....................... 15-12
   OSCilloscope:TRIGger:RESet ............................. 15-14
   OSCilloscope:TRIGger:SENSE ............................. 15-15
   OSCilloscope:TRIGger:SOURce ............................ 15-16
   OSCilloscope:TRIGger:TYPE .............................. 15-17

16. PTIME Subsystem
   PTIME:MARKer:POSITION:FALL ........................... 16-2
   PTIME:MARKer:POSITION:MID ............................. 16-3
   PTIME:MARKer:POSITION:RISE ........................... 16-4
   PTIME:MASK ............................................. 16-5
   PTIME:ZField .......................................... 16-6

17. REGister Subsystem
   [REGister:]CLEAR ........................................... 17-2
   [REGister:]CLEAR:ALL .................................... 17-3
   [REGister:]RECALL ....................................... 17-4
   [REGister:]SAVE .......................................... 17-5

18. RFAAnalyzer Subsystem
   RFAAnalyzer:AMPLitude .................................. 18-2
   RFAAnalyzer:AMPLitude:CORRection:LOSS ................ 18-3
   RFAAnalyzer:CARRier .................................... 18-4
   RFAAnalyzer:CARRier:INCRement ........................ 18-5
   RFAAnalyzer:COUPLing ................................... 18-6
   RFAAnalyzer:COUPLing:INPut ............................ 18-7
   RFAAnalyzer:FREQuency .................................. 18-8
   RFAAnalyzer:PMETer:ZERO ................................ 18-9
19. RFGenerator Subsystem
   RFGenerator:AMPLitude ........................................ 19-2
   RFGenerator:ATTenuator ....................................... 19-3
   RFGenerator:ATTenuator:AUTO .................................. 19-4
   RFGenerator:CW:CARRIer ....................................... 19-5
   RFGenerator:CW:PATrern ........................................ 19-6
   RFGenerator:MODE ............................................. 19-7

20. STATus Subsystem
   STATus:CALibration .......................................... 20-2
   STATus:COMMunicate ......................................... 20-3
   STATus:HARDware1 .......................................... 20-4
   STATus:HARDware2 .......................................... 20-5
   STATus:OPERation ........................................... 20-6
   STATus:PRESet .............................................. 20-7
   STATus:QUEStionable ........................................ 20-8
   Additional Commands in the STATus Subsystem .............. 20-9
   :CONDition? ............................................... 20-9
   :ENABle ................................................... 20-9
   [:EVENt]? ................................................. 20-9
   :NTRansition ................................................ 20-9
   :PNTransition .............................................. 20-9

21. SYStem Subsystem
   SYStem[:ERRror]? ........................................... 21-2

22. TRIGger Subsystem
   TRIGger[:IMMediate] .................................... 22-2
   TRIGger:ABORt ............................................... 22-3
   TRIGger:BETest ............................................. 22-4
   TRIGger:BETest:MODE .................................. 22-5
   TRIGger:DELay ............................................. 22-6
   TRIGger:MODE:RETRigger .................................. 22-7
   TRIGger:SOURce ............................................ 22-8

23. Example Programs
   Example Program 1 ........................................... 23-1
   Example Program 2 ........................................... 23-5
   Example Program 3 ........................................... 23-9
   Example Program 4 ........................................... 23-14

A. Appendix A
   Definition of the Units Used in HP-IB Programming ........ A-1
   Display Units (DUNits) ..................................... A-1
   Changing Display Units ................................... A-1
   Reading-Back Display Units ................................ A-1
   Guidelines for Display Units ............................... A-2
   HP-IB UNIts ................................................. A-2
   Reading-Back HP-IB Units .................................. A-2
   Guidelines for HP-IB UNIts ................................ A-2
   Attribute Units (AUNits) .................................. A-3
   Changing Attribute Units ................................ A-3
   Reading-back Attribute Units .............................. A-3
   Using STATE Commands .................................... A-4
   State Command Guidelines ................................ A-4
B. Appendix B
Introduction ....................................................... B-1
:STAtE ....................................................... B-1
:FUNctions ..................................................... B-1
:UNITs ......................................................... B-1
:AUNits ........................................................ B-1
:AVerage[:VAlue] ............................................... B-1
:AVerage:RESet ............................................... B-1
:AVerage:STATE ............................................... B-2
:REFERENCE[:VAlue] ............................................. B-2
:REFERENCE:FUNctions ....................................... B-2
:REFERENCE:STATE ............................................. B-2

Programming Command Guidelines .............................. B-3

C. Appendix C
Introduction ....................................................... C-1
:FUNctions ..................................................... C-1
:UNITs ......................................................... C-1
:INCREMENT .................................................... C-1
:INCREMENT:FUNctions ....................................... C-1
:INCREMENT:MODE .............................................. C-1
:INCREMENT:MULTIply ......................................... C-2
:INCREMENT:DIVide ............................................. C-2

Programming Command Guidelines .............................. C-2

D. Appendix D
Introduction ....................................................... D-1
:INCREMENT:MULTIply ......................................... D-1
:INCREMENT:DIVide ............................................. D-1

E. Appendix E
Using Different Numerical Bases ............................... E-1
Hexadecimal ..................................................... E-1
Octal .......................................................... E-1
Binary .......................................................... E-1

Index
Figures

6-1. Status Registers ...................................................... 6-2
11-1. DECT Syntax Diagram .............................................. 11-1
11-2. DECT Syntax Diagram .............................................. 11-2
11-3. DECT Syntax Diagram .............................................. 11-3
12-1. Display Syntax Diagram ........................................... 12-1
17-1. REGISTER Syntax Diagram ........................................ 17-1
18-1. RFAnalyzer Syntax Diagram ....................................... 18-1
19-1. RFGenerator Syntax Diagram ..................................... 19-1
20-1. STATUS Syntax Diagram .......................................... 20-1
20-2. Additional Commands Syntax Diagram ......................... 20-10
21-1. SYSTEM Syntax Diagram .......................................... 21-1

Tables

4-1. Command Summary .................................................. 4-1
5-1. Event Status Enable Register .................................... 5-4
5-2. Service Request Enable Settings ................................. 5-13
5-3. Test Failures ....................................................... 5-16
6-1. Status Byte Register .............................................. 6-3
6-2. OPERAtion Status Register ...................................... 6-4
6-3. Standard Event Status Register ................................. 6-5
6-4. QUESTIONable Status Register ................................... 6-6
6-5. CALibration Status Register ..................................... 6-7
6-6. HARDware1 Status Register ..................................... 6-8
6-7. HARDware2 Status Register ..................................... 6-9
6-8. COMMunicate Status Register ................................... 6-10
A-1. Units for HP-IB Programming ................................... A-2
B-1. Programming Command Guidelines .............................. B-3
C-1. Programming Command Guidelines .............................. C-2
Introduction

The HP 8923B DECT Test Set tests DECT Fixed and Portable Parts. The instrument can be operated manually using the front-panel keys or remotely using the Hewlett-Packard Interface Bus (HP-IB).

Relevant Documents

The HP-IB conforms to IEEE 488.1 and IEEE 488.2 standards providing bi-directional data transfer between devices. The HP 8923B also follows the Standard Commands for Programmable Instrument, (SCPI), Volume 2 (1994) standard with respect to its command structure. This allows several functions:

- Programs running in the HP 8923B's IBASIC Controller can control all the HP 8923B's functions using its internal HP-IB. This controller provides a single-instrument automated test system.
- Programs running in the HP 8923B's IBASIC Controller can control other instruments connected to the bus.
- A connected controller can remotely control the HP 8923B.

For instructions on connecting the HP 8923B to the instrument bus, the printer, and the controller, refer to the *HP 8923B DECT Test Set User's Manual*.

Configuration

This chapter tells you how to configure a computer to your HP 8923B for HP-IB control.

1. Attach an HP-IB cable from the rear panel HP-IB connector to any instruments/ controller in the test system.
2. Access the CONFIG screen.
3. Set the HP-IB Addr.
4. Set the HP-IB Mode.
   a. Talk&Lstn should be used unless you are using the HP 8923B to control the connected HP-IB devices.
   b. Control configures the HP 8923B to be a system controller. Use this setting for controlling connected HP-IB devices.
Command Guidelines

Rules and guidelines for HP-IB programming are contained in this chapter and Chapter 3, “Equivalent Front Panel Key Commands”. These chapters inform you of how to operate the HP 8923B using HP-IB.

Command Names

All commands of greater than four characters have an alternate abbreviated form using only the upper case letters and number (if used).

Upper or lower case characters may be used for all commands.

For example, to set the amplitude of RF Generator, you could use any of the following commands:

RFGENERATOR:AMPLITUDE -10DBM
or
rgenerator:amplitude -10DBM
or
rf:ampl -10DBM
or
RFG:AMPL -10DBM

Optional Parts of Commands

In the description of a command, any part enclosed in square brackets [ ] is optional. It may be omitted for brevity or included for completeness. The brackets are symbolic and must not be sent as part of the command. In the description of a command which can be set to one value from a list are represented with the list elements separated by a |.

Example

DECT:DUMMy[:STATe] ON | OFF | 0 | 1

may be sent as

DECT:DUMMy ON

or

DECT:DUMMy:STATe ON
Command Punctuation

Note The punctuation for HP 8923B HP-IB commands conforms to the IEEE 488.2 standards. It is possible that the programming language you are running on your controller will not accept some of the punctuation used in the BASIC examples. It is therefore necessary that you understand the punctuation and language equivalents required by your language for HP-IB operation.

Using Quotes for String Entries
Quotation marks (" and ") are used to select a string field setting, for example Portable in the DECT:EUT field. The value is entered into the command line as a quoted alphanumeric string.
Quotes are used with all Underlined (toggle) and menu choice (One-of-many fields).
For example, if you need to set the BER screen's Display to Ratio, you would enter the menu choice, 'Ratio'.

```
DISP:BERT 'Ratio'
```

Using Spaces
When changing a field setting, a space must always precede the value in the command (command <space> value), regardless of field type.

Example
```
RFG:AMPL SPACE -20DBM
```

Command Structure
HP-IB command syntax is arranged by a control hierarchy that is analogous to front panel operation.

When you want to configure the instrument using front-panel controls, access the screen first, select the desired field, and make the appropriate setting. HP-IB commands are used in the same way, using a colon (:) to separate the command headers.

For example, if you have already accessed the CALL SETUP screen using the "DISP CALL" command, and you want to drive the HP 8923 to test the "Fixed" part, enter the command:

```
DECT:EUT 'Fixed'
```

Using Semicolons for Multiple Commands
You can output multiple commands from one program line by separating the commands with a semicolon (;). Commands which follow a semi-colon are interpreted as existing at the same hierarchical level as the previous command unless they begin with a colon. If they do begin with a colon they are interpreted as a complete command path.

For example, on one command line, you can:

1. Access the CALL setup screen,
2. Set the dummy bearer carrier,
3. Set the dummy bearer timeslot,
4. Start the dummy bearer transmitting.
For example,

```
DISP CALL;:DECT:PP:DEMMY:CARR 3;SLOT 8;STATE ON
```

which is equivalent to

```
```

and is equivalent to each command on individual lines.

### Using Question Marks to Query

The question mark (?) is used to query (read-back) an instrument setting or measurement value. It is entered immediately at the end of a command line query. Information must be read into a variable before it can be displayed, printed, or used as a numeric value in your program.

Queried information is returned in the same format used to set the value. A queried numeric entry function returns numeric data; quoted string functions return quoted string information.

For example, the BASIC language commands ...

```
OUTPUT 714;"DISP:BET?"  !Query the BER measurement display field.
ENTER 714;Bet$           !Enter queried value into a variable.
PRINT Bet$               !Print the queried value.
```

... print the string value of the BER Measurement display field. The printed value for this example is either CNT, or Ratio (depending on the current field setting).

### Specifying Units of Magnitude

Many of the HP 8923B's numeric settings and measurements have one or more associated units (V, mV, µV; Hz, kHz, MHz, GHz). Using manual operation, the units can be easily changed to the most convenient format by pressing the appropriate units key on the front panel.

HP-IB operation is similar to manual operation in that allows any units which are appropriate to that field to be employed. However, using HP-IB, numeric values are returned in the units defined by the appropriate :UNIT subsystem, see Appendix A.

### Syntax Diagrams

The syntax diagrams display at a glance the commands relevant to the subsystem. In the diagrams, "Sp" indicates a space between commands.
Equivalent Front Panel Key Commands

Most front-panel keys have an equivalent HP-IB command. All command examples that follow in this document are in HP BASIC language.

Front Panel Keys

**SHIFT**, **CANCEL**, **Cursor Knob**

These front-panel features are not required for HP-IB use, and have no equivalent HP-IB commands.

**DATA Keys**

Apart from the number keys, the DATA keys contain the units keys, **ON/CFF**, **YES**, **NO**, and **ENTER**. Data Functions are turned on and off using the **STATE** command. **YES**, **NO**, and **ENTER** have no equivalent HP-IB command.

**DATA FUNCTION Keys**

The Data Functions enable you to change the way measurements are calculated and displayed. These keys are **AVG**, **INCR SET**, **REF SET**, **INCR X10**, and **INCR ÷10**.

**Guidelines for Using Data Functions**

- Attribute units (AU/Nits) are used when Data Functions (such as REF SET) are enabled. This allows you to discriminate between the units used for the measurement result and the units used for the Data Function being used.

- Data Function values, such as the number of Averages or the Reference value, are retained for later use if the function is turned off. However, the values are lost under the following conditions:
  - The HP 8923B is turned off.
  - The HP 8923B is PRESET.
  - A saved register is recalled.

**AVG**

This key sets the number of measurements for the averaging function.

**Syntax**: `AVERage:VALue <value>`

**Example** - Set an averaging value of 10 for the output power from the Portable Part.

```
OUTPUT 714; "MEAS:RF:NTP: AVER 10"
```
To Reset Averaging

Use the RES command to reset the averaging algorithm used to calculate an averaged measurement.

**Syntax** - :AVERAGE:REST

**Example** - OUTPUT 714;"MEAS:RF:NTP:AVER:RES"

**INCR SET**

This key specifies an increment value for appropriate numeric fields.

**Syntax** - :INCR <value>

**Example** - Set the increment value for the Amplitude field to 2.5.

OUTPUT 714;"RFG:AMPL:INCR 2.5 DBM"

**Note**

HP-IB units are assumed for the increment value unless you specify a Display unit.

---

Specifying the INCRement Mode

The INCR MODE command specifies logarithmic or linear increments.

**Syntax** - :INCR:MODE <LINEar or LOGarithm>

**Example** - OUTPUT 714;"RFG:FREQ:INCR:MODE LINEar"

**INCR ÷10**

This key reduces the current increment value for a setting by a factor of ten.

**Syntax** - :INCR:DIVide

**Example** - Reduce the increment value from 10 dBm to 1 dBm for the RF Generator Amplitude field.

OUTPUT 714;"RFG:AMPL:INCR:DIV"

**INCR ×10**

This key increases the current increment value for a setting by a factor of ten.

**Syntax** - :INCR:MULTiply

**Example** - Increase the increment value from 1 dBm to 10 dBm for the RF Generator Amplitude field.

OUTPUT 714;"RFG:AMPL:INCR:MULT"

**Increment Up/Down (Arrow Keys ▼▼)**

This keys change the setting’s value by one increment value (up or down). The increment value is determined by the INCR SET (INCR) function.

**Syntax** - :INCR <UP or DOWN>

**Example** - Increment the RF Generator Amplitude field’s value.

OUTPUT 714;"RFG:AMPL:INCR UP"

**REF SET**

This key defines the level when making a reference measurement.

**Syntax** - :REFERENCE <value>

**Example** - OUTPUT 714;"MEAS:RF:FREQ:DEV:ONE:REF 10"

3.2 Equivalent Front Panel Key Commands
Unless you specify a different unit, the HP-IB unit for the measurement is assumed when setting the reference value.

**INSTRUMENT STATE Keys**

**LOCAL**
This key returns control of the instrument to the front-panel keys after using HP-IB.

**Syntax** - LOCAL <address>
**Example** - Restore manual control to the instrument at HP-IB address 714.
  LOCAL 714

**PRESET**
This key resets the HP 8923B to its power-up state. This is an IEEE 488.2 Common Command.

**Syntax** - *RST
**Example** - OUTPUT 714;"*RST"

**RECALL**
This key recalls an instrument setup that has been saved. The REGister command in the syntax is optional.

**Syntax** - [REGister:]RECall '<quoted string>' or <numeric>
**Example** - Two ways of recalling register named SETUP1.
  OUTPUT 714;"REG:REC 'SETUP1'"
  OUTPUT 714;"REC 'SETUP1'"

**See Also.** Refer to the *SAV and *RCL Common Commands described later in the Common Command Section.

**SAVE**
This key saves the instrument setup. The REGister command in the syntax is optional.

**Syntax** - [REGister:]SAVE '<quoted string>' or <numeric>
**Example** - Two ways of saving register SETUP1.
  OUTPUT 714;"REG:SAVE 'SETUP1'"
  OUTPUT 714;"SAVE 'SETUP1'"

**Removing Saved Registers**
One or all instrument setups you have saved can be removed from memory. The REGister command in the syntax is optional.

**Syntax** - [REGister:]CLEar '<quoted string>' or <numeric> or :ALL

**Examples**
  OUTPUT 714;"REG:CLE 'SETUP2'" - Removes register SETUP2.
  OUTPUT 714;"CLE:ALL" - Removes all saved registers.

**See Also.** Refer to the *SAV and *RCL Common Commands described later in the Common Command Section.
SCREEN CONTROL Keys

**MSG**, **HELP**, **CONFIG**

The DISPlay command is used to access any screen, including those manually accessed using these keys.

**Syntax.**

DISPlay <screen>

**Example** - OUTPUT 714;"DISP CONFIGure"

**HOLD**

There is no equivalent HP-IB command for the measurement HOLD key function. However, you can imitate this function using Single Triggering of measurements. Refer to the Triggering Measurements information later in this section.

**PREV**

There is no equivalent HP-IB command for the PREV key function.

**PRINT**

The PRINT function is used to print a 'pixel dump' of the displayed screen, and does not have an equivalent HP-IB command. To print measurement results using HP-IB, you must query the measurement and print the value in a format determined by your program.

**USER Keys**

There are no equivalent HP-IB commands for the USER keys.
# Command Summary

## Table 4-1. Command Summary

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Commands (in alphabetical order)</th>
</tr>
</thead>
</table>
| **IEEE 488.2**<br>Common Commands | `*CLS`  
|          | `*ESE`  
|          | `*ESR?`  
|          | `*IDN?`  
|          | `*OPC`  
|          | `*OPT?`  
|          | `*PCB`  
|          | `*RCL`  
|          | `*RST`  
|          | `*SAV`  
|          | `*SRE`  
|          | `*STB?`  
|          | `*TRG`  
|          | `*TST?`  
|          | `*WAI`  |
| **AFAnalyzer Subsystem** | AFAnalyzer:INPut  
|          | AFAnalyzer:VOLTage  |
| **AFGenerator Subsystem** | AFGenerator[:STATE]  
|          | AFGenerator:AMPLitude  
|          | AFGenerator:FREQuency  
|          | AFGenerator:TRANmit  
|          | AFGenerator:TX  
|          | AFGenerator:VARIABLE:FREQuency  
|          | AFGenerator:VARIABLE:FREQuency:DUUnits  
|          | AFGenerator:VARIABLE:FREQuency:INCRement  
|          | AFGenerator:VARIABLE:FREQuency:INCRement:DIVIDe  
|          | AFGenerator:VARIABLE:FREQuency:INCRement:DUUnits  
|          | AFGenerator:VARIABLE:FREQuency:INCRement:MODE  
|          | AFGenerator:VARIABLE:FREQuency:INCRement:MULTiply  
<p>|          | AFGenerator:VARIABLE:FREQuency:UNITS  |</p>
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Commands (in alphabetical order)</th>
</tr>
</thead>
</table>
| **BETest Subsystem:** | BETest:BITS  
|                  | BETest:BITS:INCrement  
|                  | BETest:WEIRror:CRIterion  |
| **CONFigure Subsystem:** | CONFigure:BADdress  
|                  | CONFigure:BEEPer  
|                  | CONFigure:BMODE  
|                  | CONFigure:DATE  
|                  | CONFigure:DECT:MSYNc  
|                  | CONFigure:EDISK  
|                  | CONFigure:INTensity  
|                  | CONFigure:PRINT:ADDRess  
|                  | CONFigure:PRINT:FFEND  
|                  | CONFigure:PRINT:FPStart  
|                  | CONFigure:PRINT:LINES  
|                  | CONFigure:PRINT:PORTb  
|                  | CONFigure:PRINT:TITLE  
|                  | CONFigure:SPORT:BAUD  
|                  | CONFigure:SPORT:DATA  
|                  | CONFigure:SPORT:1BCcho  
|                  | CONFigure:SPORT:1ECHO  
|                  | CONFigure:SPORT:PARity  
|                  | CONFigure:SPORT:RACE  
|                  | CONFigure:SPORT:SPINPut  
|                  | CONFigure:SPORT:STOP  
|                  | CONFigure:SPORT:XPACE  
|                  | CONFigure:SRLocation  
|                  | CONFigure:TIME  |
| **DECT Subsystem:** | DECT:DUMMy[:STATE]  
|                  | DECT:DUMMy:SYNC:ABORt  
|                  | DECT:DUMMy:SYNC[:IMMediate]  
|                  | DECT:EUT  
|                  | DECT:EUT:PAR?  
|                  | DECT:EUT:PMID?  
|                  | DECT:FP:DUMMy:CARRier?  
|                  | DECT:FP:DUMMy:SLOT  
|                  | DECT:FP:DUMMy:SYNC:ABORt  
|                  | DECT:FP:DUMMy:SYNC[:IMMediate]  |
Table 4-1. Command Summary (continued)

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Commands (in alphabetical order)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECT:FP:TRAFFic:CARRier</td>
<td></td>
</tr>
<tr>
<td>DECT:FP:TRAFFic:CARRier:INCRement</td>
<td></td>
</tr>
<tr>
<td>DECT:FP:TRAFFic:CONNect</td>
<td></td>
</tr>
<tr>
<td>DECT:FP:TRAFFic:RELease</td>
<td></td>
</tr>
<tr>
<td>DECT:FP:TRAFFic:SLOT</td>
<td></td>
</tr>
<tr>
<td>DECT:FP:TRAFFic:SLOT:INCRement</td>
<td></td>
</tr>
<tr>
<td>DECT:LOGging:STATE</td>
<td></td>
</tr>
<tr>
<td>DECT:LOGging:SPORT:BAUD</td>
<td></td>
</tr>
<tr>
<td>DECT:LOGging:SPORT:HANDshake</td>
<td></td>
</tr>
<tr>
<td>DECT:PARI</td>
<td></td>
</tr>
<tr>
<td>DECT:PMID</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:DUMMy:STATE</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:DUMMy:CARRier</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:DUMMy:CARRier:INCRement</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:DUMMy:SLOT</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:DUMmy:SLOT:INCRement</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:TRAFFic:CARRier</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:TRAFFic:CARRier:INCRement</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:TRAFFic:CONNect</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:TRAFFic:RELease</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:TRAFFic:SLOT</td>
<td></td>
</tr>
<tr>
<td>DECT:PP:TRAFFic:SLOT:INCRement</td>
<td></td>
</tr>
<tr>
<td>DECT:PROPrirty:RX:AField:MTAil:TEST:ESCape?</td>
<td></td>
</tr>
<tr>
<td>DECT:PROPrirty:RX:AField:MTAil:TEST:ESCape:STATus?</td>
<td></td>
</tr>
<tr>
<td>DECT:PROPrirty:TX:AField:MTAil:TEST:ANTenna</td>
<td></td>
</tr>
<tr>
<td>DECT:PROPrirty:TX:AField:MTAil:TEST:ANTenna:INCRement</td>
<td></td>
</tr>
<tr>
<td>DECT:PROPrirty:TX:AField:MTAil:TEST:SEND</td>
<td></td>
</tr>
<tr>
<td>DECT:PROPrirty:TX:AField:MTAil:TEST:ESCape</td>
<td></td>
</tr>
<tr>
<td>DECT:PROPrirty:TX:AField:MTAil:TEST:ESCape:SEND</td>
<td></td>
</tr>
<tr>
<td>DECT:STATus?</td>
<td></td>
</tr>
<tr>
<td>DECT:SYNC:[IMMediate]</td>
<td></td>
</tr>
<tr>
<td>DECT:SYNC:ABOrt</td>
<td></td>
</tr>
<tr>
<td>DECT:TRAFFic:CONNect</td>
<td></td>
</tr>
<tr>
<td>DECT:TRAFFic:RELease</td>
<td></td>
</tr>
<tr>
<td>DECT:VOICE:DESTination</td>
<td></td>
</tr>
<tr>
<td>Subsystem</td>
<td>Commands (in alphabetical order)</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>DISPlay Subsystem:</strong></td>
<td>DISPlay</td>
</tr>
<tr>
<td></td>
<td>DISPlay[:SCReen]</td>
</tr>
<tr>
<td></td>
<td>DISPlay:BETest</td>
</tr>
<tr>
<td></td>
<td>DISPlay:BETest:VIEW</td>
</tr>
<tr>
<td></td>
<td>DISPlay:FREQuency</td>
</tr>
<tr>
<td></td>
<td>DISPlay:OSCiloscope</td>
</tr>
<tr>
<td><strong>ESource Subsystem:</strong></td>
<td>ESource:POWer:ADVance</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:ADVance:DUNIts</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:ADVance:INCRe ment</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:ADVance:INCRe ment:DIVide</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:ADVance:INCRe ment:DUNits</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:ADVance:INCRe ment:MODE</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:ADVance:INCRe ment:MULTiply</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:ADVance:UNIt s</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition:DUNits</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition:INCRe ment</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition:INCRe ment:DIVide</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition:INCRe ment:DUNits</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition:INCRe ment:MODE</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition:INCRe ment:MULTiply</td>
</tr>
<tr>
<td></td>
<td>ESource:POWer:OPOsition:UNIt s</td>
</tr>
<tr>
<td></td>
<td>ESource:SLOT</td>
</tr>
<tr>
<td></td>
<td>ESource:SLOT:INCRe ment</td>
</tr>
<tr>
<td></td>
<td>ESource:STATes</td>
</tr>
<tr>
<td></td>
<td>ESource:PATtern</td>
</tr>
<tr>
<td><strong>MEASure Subsystem:</strong></td>
<td>MEASure:AUDio:ACVolts?</td>
</tr>
<tr>
<td></td>
<td>MEASure:AUDio:DCVolts?</td>
</tr>
<tr>
<td></td>
<td>MEASure:AUDio:FREQuency?</td>
</tr>
<tr>
<td></td>
<td>MEASure:AUDio:OSCiloscope:MARKer:LEVel:VOLts?</td>
</tr>
<tr>
<td></td>
<td>MEASure:AUDio:OSCiloscope:MARKer:TIME?</td>
</tr>
<tr>
<td></td>
<td>MEASure:AUDio:OSCiloscope:TRACE?</td>
</tr>
</tbody>
</table>
Table 4-1. Command Summary (continued)

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Commands (in alphabetical order)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEASURE Subsystem:</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:BERRor:COUNT?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:BERRor:CCount?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:BERRor:ICount?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:BERRor:IRatio?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:BERRor:Ratio?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:BTESted?:</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:IBTested?:</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:JWTested?:</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:WERRor:COUNT?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:WERRor:CCount?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:WERRor:IRatio?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:WERRor:Ratio?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:BETest:WTESTed?:</td>
<td></td>
</tr>
<tr>
<td>MEASURE:MODE</td>
<td></td>
</tr>
<tr>
<td>MEASURE:PACKet</td>
<td></td>
</tr>
<tr>
<td>MEASURE:PATTern</td>
<td></td>
</tr>
<tr>
<td>MEASURE:PATTern:DBField</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:ACCuracy?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:COMposite?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:DEViation:ONE:BAverage?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:DEViation:ONE:BMAXimum?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:DEViation:ONE:BMINimum?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:DEViation:ZERO:BAverage?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:DEViation:ZERO:BMAXimum?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:DEViation:ZERO:BMINimum?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:FREQuency:DRIFT?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:NPower?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MARKer:LEVEL:FALL?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MARKer:LEVEL:MID?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MARKer:LEVEL:RISE?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MARKer:TIME:FALL?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MARKer:TIME:MID?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MARKer:TIME:RISE?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MASK:FALL?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MASK:MID</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:MASK:RISE?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:TRACe:FALL?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:TRACe:MID?</td>
<td></td>
</tr>
<tr>
<td>MEASURE:RF:PTI:TRACe:RISE?</td>
<td></td>
</tr>
<tr>
<td>Subsystem</td>
<td>Commands (in alphabetical order)</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>MEASure</strong></td>
<td>MEASure:RF:TIMing:JITTER:</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:AUUnits</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:AVErage:VALue</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:AVErage:RESet</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:AVErage:STATe</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:DUUnits</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:REFERence:VALue</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:REFERence:DUUnits</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:REFERence:STATe</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:STATe</td>
</tr>
<tr>
<td></td>
<td>MEASure:RF:TIMing:JITTER:UNITs</td>
</tr>
<tr>
<td></td>
<td>MEASure:SYNC?</td>
</tr>
<tr>
<td><strong>OSCiloscope</strong></td>
<td>OSCiloscope:MARKer:NPEak</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:MARKer:PPEak</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:MARKer:POSition</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:SCALE:TIME</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:SCALE:VERTical:OFFSet</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:SCALE:VOLts</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:LEVel</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:LEVel:INCrement</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:MODE</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:PRETrigger</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:PRETrigger:INCrement</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:RESet</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:SENSe</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:SOURce</td>
</tr>
<tr>
<td></td>
<td>OSCiloscope:TRIGger:TYPE</td>
</tr>
<tr>
<td><strong>PT1Me</strong></td>
<td>PT1Me:Marker:POSition:FALL</td>
</tr>
<tr>
<td></td>
<td>PT1Me:Marker:POSition:MD</td>
</tr>
<tr>
<td></td>
<td>PT1Me:Marker:POSition:RISE</td>
</tr>
<tr>
<td></td>
<td>PT1Me:MASK</td>
</tr>
<tr>
<td></td>
<td>PT1Me:ZField</td>
</tr>
<tr>
<td><strong>REGister</strong></td>
<td>[REGister]CLEar</td>
</tr>
<tr>
<td></td>
<td>[REGister]CLEar:ALL</td>
</tr>
<tr>
<td></td>
<td>[REGister]RECAll</td>
</tr>
<tr>
<td></td>
<td>[REGister]SAVE</td>
</tr>
<tr>
<td>Subsystem</td>
<td>Commands (in alphabetical order)</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>RFANalyzer Subsystem:</td>
<td>RFANalyzer:AMPLitude</td>
</tr>
<tr>
<td></td>
<td>RFANalyzer:AMPLitude:CORRection:LOGS</td>
</tr>
<tr>
<td></td>
<td>RFANalyzer:CARRier</td>
</tr>
<tr>
<td></td>
<td>RFANalyzer:CARRier:INCReement</td>
</tr>
<tr>
<td></td>
<td>RFANalyzer:COUPling</td>
</tr>
<tr>
<td></td>
<td>RFANalyzer:COUPling:INPut</td>
</tr>
<tr>
<td></td>
<td>RFANalyzer:FREquency</td>
</tr>
<tr>
<td></td>
<td>RFANalyzer:PMETER:ZERO</td>
</tr>
<tr>
<td>RFGenerator Subsystem:</td>
<td>RFGenerator:AMPLitude</td>
</tr>
<tr>
<td></td>
<td>RFGenerator:ATTenuator</td>
</tr>
<tr>
<td></td>
<td>RFGenerator:ATTenuator:AUTO</td>
</tr>
<tr>
<td></td>
<td>RFGenerator:CW:CARRier</td>
</tr>
<tr>
<td></td>
<td>RFGenerator:CW:PATTern</td>
</tr>
<tr>
<td></td>
<td>RFGenerator:MODE</td>
</tr>
<tr>
<td>STATus Subsystem:</td>
<td>STATus:CALibration</td>
</tr>
<tr>
<td></td>
<td>STATus:COMMunicate</td>
</tr>
<tr>
<td></td>
<td>STATus:HARDware1</td>
</tr>
<tr>
<td></td>
<td>STATus:HARDware2</td>
</tr>
<tr>
<td></td>
<td>STATus:OPERation</td>
</tr>
<tr>
<td></td>
<td>STATus:PRESet</td>
</tr>
<tr>
<td></td>
<td>STATus:QUESTionable</td>
</tr>
<tr>
<td></td>
<td>(Five additional sub-commands at end of STATus Subsystem</td>
</tr>
<tr>
<td></td>
<td>Chapter)</td>
</tr>
<tr>
<td>SYStem Subsystem:</td>
<td>SYStem:ERRor[]</td>
</tr>
<tr>
<td>TRIGger Subsystem:</td>
<td>TRIGger:[IMMediate]</td>
</tr>
<tr>
<td></td>
<td>TRIGger:ABORT</td>
</tr>
<tr>
<td></td>
<td>TRIGger:BETest</td>
</tr>
<tr>
<td></td>
<td>TRIGger:BETest:MODE</td>
</tr>
<tr>
<td></td>
<td>TRIGger:DElay</td>
</tr>
<tr>
<td></td>
<td>TRIGger:MODE:RETigger</td>
</tr>
<tr>
<td></td>
<td>TRIGger:SOURce</td>
</tr>
</tbody>
</table>
IEEE 488.2 Common Commands

IEEE 488.2 mandates the use of some common commands. These commands have a special syntax (beginning with a "+"), which is not legal for other commands. The common commands control some of the basic instrument functions:

- Instrument identification and reset
- Status reading and clearing
- Receiving and processing of commands and queries by the instrument
*CLS
(Clear Status)

The *CLS (clear status) common command clears the status data structures, including the
device defined error queue. This command also aborts the *OPC. If the *CLS command
immediately follows a PROGRAM MESSAGE TERMINATOR, the output and the MAV (message
available) bit will be cleared.

Command Syntax

*CLS

Example

OUTPUT 714; "*CLS"
*ESE
(Event Status Enable)

The *ESE command sets the Standard Event Status Enable Register bits. The Standard Event Status Enable Register contains a mask value for the bits to be enabled in the Standard Event Status Register. A "one" in the Standard Event Status Enable Register will enable the corresponding bit in the Standard Event Status Register, a logic zero will disable the bit. Refer to Figure 20-2 for the information about the Standard Event Status Enable Register bits. The *ESE query returns the contents of the Standard Event Status Enable Register.

Command Syntax

*ESE? <mask>
Where <mask> = 0 to 255

Example

OUTPUT 714;++ESE I

In this example, the *ESE I command will enable the OPC (operation complete) bit 6 of the Standard Event Status Enable Register.

Query Syntax

*ESE?

Returned Format

<mask><NL>
Where <mask> = 0 to 255

Example

OUTPUT 714;++ESE?
ENTER 714;Event
PRINT Event
<table>
<thead>
<tr>
<th>Bit</th>
<th>Weight</th>
<th>Event Status Enable Register Enables</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>OPC - Operation Complete</td>
<td>0 = operation is not complete</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>RQC - Request Control</td>
<td>0 = request control - NOT used - always 0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>QYE - Query Error</td>
<td>0 = no query errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = a query error has been detected</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>DDE - Device Dependent Error</td>
<td>0 = no device dependent errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = a device dependent error has been detected</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>EXE - Execution Error</td>
<td>0 = no execution error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = an execution error has been detected</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>CME - Command Error</td>
<td>0 = no command errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = a command error has been detected</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>URQ - User Request</td>
<td>not used in the HP 8923B</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>PON - Power On</td>
<td>1 = an OFF to ON transition has occurred</td>
</tr>
</tbody>
</table>
*ESR?
(Event Status Register)
The *ESR? query returns the contents of the Standard Event Status Register.

Note  Reading the Standard Event Status Register clears the contents of the register.

Query Syntax:

*ESR?

Returned Format
<status><NL>
Where <status> = 0 to 255

Example
OUTPUT 714;"*ESR?"
ENTER 714;Event
PRINT Event

Table 5-1 shows the Event Status Register. The table shows each bit in the Event Status Register and the bit weight. When you read the Event Status Register, the value returned is the total bit weights of all bits that are true at the time you read the byte.
*IDN?
(Identification Number)
The *IDN? query allows the instrument to identify itself. It returns the string:

"Hewlett-Packard,8923B, XXXXUYYYY,B.UU,VV"

Where:

XXXXUYYYY = the serial number of this instrument

B.UU,VV = the firmware revision of this instrument.

An *IDN? query must be the last query in a message. Any queries after the *IDN? query in this program message will be ignored.

Query Syntax

*IDN?

Returned Format

Hewlett-Packard,8923B, XXXXUYYYY,B.UU,VV</NL>

Example

DIM Id$[100]
OUTPUT 714;"*IDN?"
Enter 714; Id$
PRINT Id$
*OPC
(Operation Complete)

The *OPC (operation complete) command will cause the instrument to set the operation complete bit in the Standard Event Status Register only when all pending operations are complete. The *OPC? query places an ASCII "1" in the output queue when all pending device operations are complete. There is a one second minimum delay between the query and the response. A pending operation in the HP 8923B is any measurement which is armed but not complete. When in remote operation with repetitive triggering all measurements, apart from BER, are self-arming. When in remote operation with single triggering all measurements, apart from BER, are armed by sending the TRIGger[:IMMediate] command or *TRG. The BER measurement is armed by sending the TRIGger:BETest:RUN command.

Command Syntax

*OPC

Example
OUTPUT 714;"*OPC"

Query Syntax

*OPC?

Returned Format
1<NL>

Example
OUTPUT 714;"*OPC?"
ENTER 714:0p
PRINT 0p
*OPT?
The *OPT? query will return a string containing the instrument options that are installed in the HP 8923B.

Query Syntax

*OPT?

Return Syntax

Where <string> = "0,0,0"

Note

Currently, there are no reportable options available in the HP 8923B.

Example

DIM Value$[100]
OUTPUT 714;"*OPT?"
ENTER 714;Value$
PRINT Value$
*PCB

The *PCB command tells the instrument the address of the controller to which control is to be passed back when the instrument sends a "take control" message.

Command Syntax

*PCB <primary address><secondary address>

Where

<primary address> = integer 0 to 30
<secondary address> = integer 0 to 30

Example

OUTPUT 714;"*PCB 7,0 "
**RCL**  
(Recall)

The *RCL* command restores the state of the instrument from the specified internal save/recall register. An instrument setup must have been stored previously in the specified register. Registers 0 through 99 are general purpose and can be used with the *SAVE* command.

---

**Note**  
An error message will appear on the screen if nothing has been previously saved in the specified register.

---

**Command Syntax**

*RCL* `<rd_register>`

Where `<rd_register>` = 0 through 99 though the total number of registers used may be limited by the amount of memory available.

**Example**

```
OUTPUT 714; "*RCL 75"
```

An instrument state stored using [REGister:]SAVE may be recalled using *RCL or [REGister:]RECall. If the [REGister:]SAVE uses an alphanumeric string as the register name, the *RCL* command will not work. *RCL* only works with registers named using an integer from 0 through 99.
*RST
(Reset)

The *RST command places the instrument in a known state. It is equivalent to pressing the front panel [Preset] key.

Command Syntax

*RST

Example

OUTPUT 714; "*RST"
*SAV
(Save)

The *SAV command stores the current state of the instrument in an internal save register. The data parameter is the number of the save register where the data will be saved. Internal registers 0 through 99 are valid for this command. The total number of registers which can be saved is limited by the number of settings which differ from their preset condition and the memory available.

Command Syntax

*SAV <number>
Where <number> = 0 through 99

Example
OUTPUT 714;"*SAV 85"

The [REGISTER:]RECALL command may be used to return the instrument to the state at which the instrument was saved using *SAV. The [REGISTER:]RECALL must use the same integer to return to this state. Strings are not accepted.
*SRE
(Service Request Enable)

The *SRE command sets the Service Request Enable Register bits. The Service Request Enable Register contains a mask value for the bits to be enabled in the Status Byte Register. A logic one in the Service Request Enable Register will enable the corresponding bit in the Status Byte Register, a logic zero will disable the bit. Refer to Table 5-2 for the definition of the bits in the Service Request Enable Register. For additional information, refer to the STATus Subsystem in Chapter 20.

The *SRE query returns the current setting.

Command Syntax

*SRE <mask>
Where <mask> = 0 through 255

Example
OUTPUT 714;"*SRE 16"

Note
This example enables a service request to be generated when a message is available in the output queue. When a message is available, the MAV bit will be high.

Query Syntax

*SRE?
<mask><NL>
Where <mask> = sum of all the bits that are set, 0 through 255.

Example
OUTPUT 714;"*SRE?"
ENTER 714;Value
PRINT Value

Table 5-2. Service Request Enable Settings

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Used in the HP 8923B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HARDware1 Status Register summary bit</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>HARDware2 Status Register summary bit</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>COMMunicate Status Register summary bit</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>QUESTIONable Status Register summary bit</td>
<td>SCPI</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>MAV</td>
<td>IEEE 488.2</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>Standard Event Register summary bit</td>
<td>IEEE 488.2</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>RQS</td>
<td>IEEE 488.1</td>
<td>yes</td>
</tr>
<tr>
<td>7</td>
<td>OPERation Status Register summary bit</td>
<td>SCPI</td>
<td>yes</td>
</tr>
</tbody>
</table>
*STB?  
(Status Byte)

The *STB? query returns the current value of the instrument’s status byte. The RQS (request service) bit is reported on bit 6. The RQS indicates whether or not the device has at least one reason for requesting service.

Note To read the instrument’s status byte with RQS reported on bit 6, use the HP-IB Serial Poll.

Query Syntax

*STB?

Query Syntax

<value><NL>
Where <value> = 0 through 255

Example

OUTPUT 714;"*STB?"
Enter 714;Value
PRINT Value
*TRG
(Trigger)

The "TRG command has the same effect as the Trigger[:IMMediate] command and instructs the instrument to trigger.

Command Syntax

*TRG

Example

OUTPUT 714;"*TRG"
*TST?  
(Test)  
The *TST query causes the instrument to perform a self-test. The result of the test will be placed in the output queue.

Note  Prior to sending this command, all front panel inputs must be disconnected.

A zero indicates the test passed and a non-zero value indicates the test failed.

Command Syntax

*TST?

Returned Format

<result><NL>
Where <result> = 0 or a non-zero value.
0 indicates the test has passed.
Non-zero indicates the test has failed.

The non-zero values indicate the following failures:

<table>
<thead>
<tr>
<th>Returned Value</th>
<th>Failed Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Host CPU failure</td>
</tr>
<tr>
<td>2</td>
<td>ROM Checksum failure</td>
</tr>
<tr>
<td>4, 8, or 16 (depending on the address of the failure block)</td>
<td>ROM Test failure</td>
</tr>
<tr>
<td>32</td>
<td>Timer Chip failure</td>
</tr>
<tr>
<td>64</td>
<td>Real-time Clock failure</td>
</tr>
<tr>
<td>128</td>
<td>Keyboard failure</td>
</tr>
<tr>
<td>256</td>
<td>CRT failure</td>
</tr>
<tr>
<td>512</td>
<td>RF Hardware failure</td>
</tr>
<tr>
<td>1024</td>
<td>RS-232 Port A failure</td>
</tr>
<tr>
<td>2048</td>
<td>RS-232 Port B failure</td>
</tr>
<tr>
<td>4096</td>
<td>DSP or D-Board communications failure</td>
</tr>
</tbody>
</table>
*WAI
(Wait)

The *WAI command pauses the instrument, preventing it from executing any further HP-IB commands of queries until no operations are pending.

Command Syntax

*WAI

Example

OUTPUT 714;"MEAS:_PATTERN 'Facc'"
OUTPUT 714;"TRIG:MODE:RETRIGGER SINGLE"
OUTPUT 714;"*TRG"
OUTPUT 714;"*WAI"

! The following command will not execute until the trigger has occurred
! and is a valid measurement result.
OUTPUT 714;"MEAS:RF:FREQ:ACC?"
ENTER 714;Freq_acc
PRINT Freq_acc
Status Registers

This overview details the status reporting system of the HP 8923B. It describes the contents and meaning of the various registers, bits, and queues in the status reporting structure. The status reporting structure used by the HP 8923B complies with the IEEE 488.2 (1987) standard.

The HP 8923B also complies with SCPI status reporting requirements. This subsystem covers errors that result from broken hardware, executing a command or attempting to execute a command over the HP-IB, as well as progress from outstanding operations. For additional information, refer to the STATus Subsystem in Chapter 20.
Figure 6.1. Status Registers

6-2 Status Registers
Status Byte

The Status Byte describe is returned in response to a Serial Poll, or a direct read of the status register.

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HARDware1 Status Register summary bit</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>HARDware2 Status Register summary bit</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>COMMunicate Status Register summary bit</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>QUESTionable Status Register summary bit</td>
<td>SCPI</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>MAV</td>
<td>IEEE 488.2</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>Standard Event Register summary bit</td>
<td>IEEE 488.2</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>RQS</td>
<td>IEEE 488.1</td>
<td>yes</td>
</tr>
<tr>
<td>7</td>
<td>OPERation Status Register summary bit</td>
<td>SCPI</td>
<td>yes</td>
</tr>
</tbody>
</table>

All of these bits are summary bits that indicate there is some information existing in this or another register or message queue. The Status Register itself is defined in IEEE 488.2 (1987) standard.

Bit 0: This is a summary bit that indicates a bit is set (TRUE) in the HARDware1 Status Register.

Bit 1: This is a summary bit that indicates a bit is set (TRUE) in the HARDware2 Status Register.

Bit 2: This is a summary bit that indicates a bit is set (TRUE) in the COMMunicate Status Register.

Bit 3: This is a summary bit for the QUESTionable Data/Signal Status Register.

Bit 4: This is a summary bit that indicates that there is information in the HP-IB output queue, when set (TRUE).

Bit 5: This is a summary bit for the Standard Event Status Register.

Bit 6: Unlike all the other bits in all the status registers, it cannot be masked. This is the bit that is responsible for causing the Service Request to be sent to the controller (SRQ).

Bit 7: This is a summary bit for the Operation Status Register defined by SCPI.
Status Registers

The registers are summarized in the Status Byte. All of the registers follow the requirements of IEEE 488.2 (1987), having enable (mask) registers. The registers also have associated transition registers and condition registers. The enable, transition, and condition registers are not shown but are assumed to have a one-to-one correspondence with the status registers.

**OPERation Status Register**

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CALibrating</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>SETTing</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>RANGing</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>SWEeping</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>MEASuring</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>Waiting for TRIGger summary</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>Waiting for ARM summary</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>CORRecting</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>Availble to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>Availble to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>Availble to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>Availble to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>Availble to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>13</td>
<td>INSTRument summary</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>14</td>
<td>PROgram Running</td>
<td>SCPI</td>
<td>yes</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>SCPI</td>
<td>no</td>
</tr>
</tbody>
</table>

Bit 14 is asserted whenever an HP 8923B IBASIC programming is in the running state.
Standard Event Status Register

Table 6-3. Standard Event Status Register

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation Complete</td>
<td>IEEE</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>Request Control</td>
<td>IEEE</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Query Error</td>
<td>IEEE</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>Device Dependant Error</td>
<td>IEEE</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>Execution Error</td>
<td>IEEE</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>Command Error</td>
<td>IEEE</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>User Request</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>Power On</td>
<td>IEEE</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>13</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>14</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
<tr>
<td>15</td>
<td>Reserved by IEEE</td>
<td>IEEE</td>
<td>no</td>
</tr>
</tbody>
</table>

Execution errors in the HP 8923B are caused by attempting an operation that is not possible due to the current state of the instrument. Command and query errors are detected by the HP-IB language parser and are associated with the syntax of the command or query.

An entry will be placed into the System Error Queue whenever a Command, Execution, Query, or Device Dependant Error is reported.

Device Dependant Errors are used with pending entries.

Bit 0 is set whenever all active measurements have made one complete valid measurement and all outgoing signals are stable. Bit 7 is set to a logic one every time the power has been removed since the last HP-IB command.
**QUESTionable Status Register**

Table 6-4. **QUESTionable Status Register**

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>VOLTage</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>CURRRent</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>TIME</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>POWer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>TEMPerature</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>FREQuency</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>PHASE</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>MODulation</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>CALibration Status Register summary bit</td>
<td>SCPI</td>
<td>yes</td>
</tr>
<tr>
<td>9</td>
<td>Available to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>Available to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>Available to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>Available to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>13</td>
<td>Available to the designer</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>14</td>
<td>Command warning</td>
<td>SCPI</td>
<td>no</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>SCPI</td>
<td>no</td>
</tr>
</tbody>
</table>

Bit 8 indicates that calibration is questionable. This acts, in the HP 8923B, as a summary bit for the calibration status register.
### CALibration Status Register

#### Table 6-5. CALibration Status Register

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>Sampler Self calibration failed</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Counter Self calibration failed</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>Voltmeter Self calibration failed</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>13</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>14</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>SCPI</td>
<td>no</td>
</tr>
</tbody>
</table>

Bit 1 is asserted whenever the High Speed Sampler Self Calibration procedure fails.

Bit 2 is asserted whenever the Counter Self Calibration procedure fails.

Bit 3 is asserted whenever the Voltmeter Self Calibration procedure fails.
# HARDware1 Status Register

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>Power-up Self Test Failure</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>13</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>14</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>SCPI</td>
<td>no</td>
</tr>
</tbody>
</table>

Bit 4 is asserted whenever the power-up digital diagnostic self tests fail.
**HARDware2 Status Register**

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>Improper trigger selection</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>Improper pattern selection</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>Improper coupling selection</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>SCPI</td>
<td>no</td>
</tr>
</tbody>
</table>

Bit 12 is asserted whenever an incorrect trigger has been selected for the given instrument state.

Bit 13 is asserted whenever an unsupported measurement pattern has been selected.

Bit 14 is asserted whenever coupling of the analyzer to the dummy is not possible.
## COMMunicate Status Register

**Table 6-8. COMMunicate Status Register**

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Defined</th>
<th>Used in the HP 8923B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DSP communication failure</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>1</td>
<td>Protocol processor communication failure</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>MAC escape test message received</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>FP traffic bearer setting pending</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>PP bearer setting pending</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>Active dummy bearer</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>Active traffic bearer</td>
<td>HP 8923B</td>
<td>yes</td>
</tr>
<tr>
<td>7</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>13</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>14</td>
<td>unused</td>
<td>HP 8923B</td>
<td>no</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>SCPI</td>
<td>no</td>
</tr>
</tbody>
</table>

Bit 0 is asserted whenever there is a failure in communications between the host processor and the DSP processor.

Bit 1 is asserted whenever there is a failure in communications between the host processor and the protocol processor.

Bit 2 is asserted whenever a MAC layer ESCAPE test message is received.

Bit 3 is asserted whenever there is an inconsistency between the fixed part's dummy bearer and traffic bearer carrier/slot settings.

Bit 4 is asserted whenever there is an active traffic bearer setting; traffic bearer connected.

Bit 5 is asserted whenever there is an active dummy bearer in existence; either one being transmitted or when synchronized/locked to a received one.
AFANalyzer Subsystem

This subsystem analyses the audio applied to the INPUT port

AFANalyzer Syntax Diagram
AFANalyzer:INPut
This command Selects the audio analyzer measurement port.

Command Syntax

AFANalyzer:INPut <string>
Where <string> is 'AUDIO IN' | 'Rx Audio'.

Example Statements

OUTPUT 714;"AFANalyzer:INPut 'AUDIO IN '"
OUTPUT 714;"AFANalyzer:INPut 'Rx Audio '"

Query Syntax

AFANalyzer:INPut?

Return Format

String
AFANalyzer:VOLTage
This command toggles the voltage measurement 'AC' | 'DC'.

Command Syntax
AFANalyzer:VOLTage <string>
Where <string> has values 'AC' | 'DC'

Example Statements
OUTPUT 714;"AFANalyzer:VOLTage 'AC'"
OUTPUT 714;"AFANalyzer:VOLTage 'DC'"

Query Syntax
AFANalyzer:VOLTage?

Return Format
String
AFGenerator Subsystem

This subsystem controls the instruments audio frequency generator.

AFGenerator Syntax Diagram
**AFGenerator[:STATE]**

This command switches the audio frequency output on or off.

**Command Syntax**

AFGenerator[:STATE] <boolean>
Where <boolean> = 'ON' | 'OFF'

**Example Statements**

OUTPUT 714:"AFGenerator[:STATE] 'ON'"
OUTPUT 714:"AFGenerator[:STATE] 'OFF'"

**Query Syntax**

AFGenerator[:STATE]?

**Return Format**

Boolean
**AFGenerator:AMPLitude**

Sets the amplitude of the audio signal.

**Command Syntax**

`AFGenerator:AMPLitude <real>`

Where `<real>` = 0 to 2

**Example Statements**

- OUTPUT 714;"AFGenerator:AMPLitude 0.005V"
- OUTPUT 714;"AFGenerator:AMPLitude 1.1V"

**Query Syntax**

`AFGenerator:AMPLitude?`

**Return Format**

Real
**AFGenerator:FREQuency**
This command sets the frequency of the audio generator.

**Command Syntax**

```
AFGenerator:FREQuency <audio_freq>
Where <audio_freq> = '400HZ' | '1KHZ'
```

**Example Statements**

```
OUTPUT 714;"AFGenerator:FREQuency '400HZ'"
OUTPUT 714;"AFGenerator:FREQuency '1KHZ'"
```

**Query Syntax**

```
AFGenerator:FREQuency?
```

**Return Format**

String
AFGenerator::TRANsmit

Connects/disconnects the AP source to/from the CODEC input/mouthpiece.

Command Syntax

AFGenerator::TRANsmit <string> or
AFGenerator::TX <string>
Where <string> is 'Yes' | 'No'.

Example Statements

OUTPUT 714,"AFGenerator::TRANsmit 'Yes'"
OUTPUT 714,"AFGenerator::TRANsmit 'No'"
OUTPUT 714,"AFGenerator::TX 'Yes'"
OUTPUT 714,"AFGenerator::TX 'No'"

Query Syntax

AFGenerator::TRANsmit? or
AFGenerator::TX?

Return Format

String
AFGenerator:TX

Connects/disconnects the AF source to/from the CODEC input/mouthpiece.

**Command Syntax**

AFGenerator:TRANsmit <string> or
AFGenerator:TX <string>

Where <string> is 'Yes' | 'No'.

**Example Statements**

```
OUTPUT 714;'AFGenerator:TRANsmit 'Yes''
OUTPUT 714;'AFGenerator:TRANsmit 'No''
OUTPUT 714;'AFGenerator:TX 'Yes''
OUTPUT 714;'AFGenerator:TX 'No''
```

**Query Syntax**

AFGenerator:TRANsmit? or
AFGenerator:TX?

**Return Format**

String
**AFGenerator:VARiable:FREQuency**

Sets the frequency of the variable source.

**Command Syntax**

```
AFGenerator:VARiable:FREQuency <real>
```

Where `<real>` = 100 to 21000

**Example Statements**

```
OUTPUT 714;"AFGenerator:FREQuency 100HZ"
OUTPUT 714;"AFGenerator:FREQuency 21KHZ"
```

**Query Syntax**

```
AFGenerator:FREQuency?
```

**Return Format**

String
CONFigure Subsystem

This subsystem configures various HP 8923B auxiliary controls.
CONFigure Syntax Diagram
CONFigure Syntax Diagram
CONFigure:BADDress
This command sets the HP 8628B's HP IB address.

Command Syntax

CONFigure:BADDress <value>
  Where <value> = XX
  XX - address 0 through 30

Example Statements

OUTPUT 714;"CONFigure:BADDress 14"

Query Syntax

CONFigure:BADDress?

Return Format

Integer
CONFigure:BEEPer

This command sets the volume of the instrument beeper.

Command Syntax

    CONFigure:BEEPer <beeper_value>
    Where <beeper_value> = 'Off' | 'Quiet' | 'Loud'

Example Statements

    OUTPUT 714;'CONFigure:BEEPer 'Off'"
    OUTPUT 714;'CONFigure:BEEPer 'Quiet'"
    OUTPUT 714;'CONFigure:BEEPer 'Loud'"

Query Syntax

    CONFigure:BEEPer?

Return Format

    String
**CONFfigure:BMODe**

This command sets the instrument's HP-IB to talk & listen mode or control mode.

The talk & listen mode is set when the instrument is being controlled from an external computer. The "controller" mode is set when the HP 8923B I-BASIC is controlling external instruments via HP-IB.

**Command Syntax**

```
CONFigure:BMODe <i-basic_mode>
Where <i-basic_mode> = 'Talk&Lstn' | 'Control'
```

**Example Statements**

```
OUTPUT 714;"CONFigure:BMODe 'Control';"
OUTPUT 714;"CONFigure:BMODe 'Talk&Lstn';"
```

**Query Syntax**

```
CONFigure:BMODe?
```

**Return Format**

String
CONFigure:DATE
This command sets the current date used by the IIP 8923B clock.

Command Syntax

CONFigure:DATE <value>
Where <value> = YYYYMMDD
MM - month
DD - day
YYYY - year

Example Statements

OUTPUT 714;"CONFigure:DATE 19951225" - 25th December 1995

Query Syntax

CONFigure:DATE?

Return Format

Integer
CONFigure:DATE:YEAR
This command sets the current year used by the HP 8923B clock.

Command Syntax
CONFigure:DATE:YEAR <value>
Where <value> = YYYY
YYYY - year

Example Statements
OUTPUT 714;"CONFigure:DATE:YEAR 1996" - 1996

Query Syntax
CONFigure:DATE:YEAR?

Return Format
Integer
**CONFigure:DATE:MONTH**

This command sets the current date used by the HP 8923B clock.

**Command Syntax**

CONFigure:DATE:MONTH <value>
Where <value> = MM
MM - month

**Example Statements**

OUTPUT 714;"CONFigure:DATE:MONTH 12" - December

**Query Syntax**

CONFigure:DATE:MONTH?

**Return Format**

Integer
CONFigure:DATE:DAY
This command sets the current date used by the HP 8923B clock.

Command Syntax

CONFigure:DATE:DAY <value>
Where <value> = DD
DD - day

Example Statements

OUTPUT 714;"CONFigure:DAY 25" - 25th

Query Syntax

CONFigure:DATE:DAY?

Return Format

Integer
CONFigure:DECT:MSYNc

This command sets the HP 8923B to either output a multiframe synchronization signal (Master) or to treat it as an input and synchronize to it (slave). This signal is sent through the SYNC OUT and SYNC IN connectors on the rear panel.

**Note**  
This control is only effective when testing portable parts. When testing fixed parts, the HP 8923B synchronizes to a bearer transmitted by the fixed part.

---

**Command Syntax**

```plaintext
CONFigure:DECT:MSYNc <string>  
Where <string> = 'Master' | 'Slave'
```

**Example Statement**

```plaintext
OUTPUT 714;"CONFigure:DECT:MSYNc 'Master'"
```

**Query Syntax**

```plaintext
CONFigure:DECT:MSYNc?
```

**Return Format**

String
CONFigure:EDISK

This configures the IIP 8923D to interface with an external HP-IB disk drive.

Command Syntax

CONFigure:EDISK <disk_parameter>
Where <disk_parameter> = WXY,Z
  W = 7 (HP-IB select code)
  XY = 0 through 31 (address)
  Z = 0 (left-hand disk drive) or 1 (right-hand drive)

Example Statements

OUTPUT 714:"CONFigure:EDISK:,702,0"

Query Syntax

CONFigure:EDISK?

Return Format

String
CONFigure:INTensity

This command sets the instrument display intensity.

Command Syntax

CONFigure:INTensity <value>
Where <value> = 1 through 8
1 - dim
8 - bright

Example Statements

OUTPUT 714:"CONFigure:INTensity 7"

Query Syntax

CONFigure:INTensity?

Return Format

Integer
CONFigure:PRINt:ADDRess
This command sets the HP-IB address of an external HP-IB printer.

Command Syntax

CONFigure:PRINt:ADDRess <address>
Where the <address> = 1 through 30

Example Statements

OUTPUT 714;"CONFigure:PRINt:ADDRess 1"

Query Syntax

CONFigure:PRINt:ADDRess?

Return Format

Integer
CONFigure:PRINt:FFEND

This command activates/deactivates the form-feed at the end of a printout.

Command Syntax

CONFigure:PRINt:FFEND <string>
Where <string> = 'Yes' | 'No'

Example Statements

OUTPUT 714;"CONFigure:PRINt:FFEND 'Yes'"

Query Syntax

CONFigure:PRINt:FFEND?

Return Format

String
CONFigure:PRINt:FFSTart

This command activates/deactivates the form-feed at the start of a printout.

Command Syntax

    CONFigure:PRINt:FFSTart <string>
    Where <string> = 'Yes' | 'No'

Example Statements

    OUTPUT 714:"CONFigure:PRINt:FFSTart 'Yes'"

Query Syntax

    CONFigure:PRINt:FFSTart?

Return Format

    String
**CONFigure:PRINt:LI Nes**

This command sets the number of lines per page for the print out.

**Command Syntax**

```
CONFigure:PRINt:LI Nes <value>
```

Where `<value>` = 20 through 120

**Example Statements**

```
OUTPUT 714;"CONFigure:PRINt:LI Nes 66"
```

**Query Syntax**

```
CONFigure:PRINt:LI Nes?
```

**Return Format**

Integer
**CONFigure:PRINt:PORTs**

This command sets the destination output port for the printer.

**Command Syntax**

```
CONFigure:PRINt:PORTs <string>
```

Where the `<string>` is 'Serial' | 'HP-IB'

**Example Statements**

```
OUTPUT 714:'CONFigure:PRINt:PORTs 'Serial''
```

**Query Syntax**

```
CONFigure:PRINt:PORTs?
```

**Return Format**

String
CONFigure:PRINt:DESTination
This command sets the destination output port for the printer.

Command Syntax
CONFigure:PRINt:DESTination <string>
  Where the <string> is 'Serial' | 'HP-IB'

Example Statements
OUTPUT 714:"CONFigure:PRINt:DESTination 'Serial'"

Query Syntax
CONFigure:PRINt:DESTination?

Return Format
String
CONFigure:PRINT:TITLE
This command sets up the one line title that will appear on the first line of a print out.

Command Syntax

CONFigure:PRINT:TITLE <string>
Where <string> is a maximum of 50 characters long.

Example Statements

OUTPUT 714:"CONFigure:PRINT:TITLE 'BER Results'"

Query Syntax

CONFigure:PRINT:TITLE?

Return Format

String
CONFigure:SPORt:BAUD
This command sets the baud rate for the rear panel I-BASIC/print serial port.

Command Syntax

CONFigure:SPORt:BAUD <string>
Where <string> is '19200' | '9600' | '4800' | '2400' | '1200' |
'600' | '300' | '150''

Example Statements

OUTPUT 714;"CONFigure:SPORt:BAUD '19200'"
OUTPUT 714;"CONFigure:SPORt:BAUD '2400'"

Query Syntax

CONFigure:SPORt:BAUD?

Return Format

String
CONFigure:SPORt:DATA
This command sets the number of data bits for the rear panel I-BASIC/print serial port.

Command Syntax

CONFigure:SPORt:DATA <data_length>
Where <data_length> = '7 bits' | '8 bits'

Example Statements

OUTPUT 714;"CONFigure:SPORt:DATA '8 bits'"

Query Syntax

CONFigure:SPORt:DATA?

Return Format

String
CONFigure:SPORt:IBECgo

This command activates/deactivates echoing of received I-BASIC commands to the I-BASIC/print serial port.

**Command Syntax**

```
CONFigure:SPORt:IBECgo <string>
Where <string> = 'On' | 'Off'
```

**Example Statements**

```
OUTPUT 714;"CONFigure:SPORt:IBECgo 'On'"
OUTPUT 714;"CONFigure:SPORt:IBECgo 'Off'"
```

**Query Syntax**

```
CONFigure:SPORt:IBECgo?
```

**Return Format**

String
**CONFigure:SPORt:IECHO**

This command activates/deactivates echoing of received instrument commands to the I-BASIC/print serial port.

**Command Syntax**

```plaintext
CONFigure:SPORt:IECHO <string>
Where <string> = 'On' | 'Off'
```

**Example Statements**

```plaintext
OUTPUT 714;"CONFigure:SPORt:IECHO 'On'"
OUTPUT 714;"CONFigure:SPORt:IECHO 'Off'"
```

**Query Syntax**

```plaintext
CONFigure:SPORt:IECHO?
```

**Return Format**

String
CONFigure:SPORT:PARity
This command selects the parity used for the I-DASIC/print serial port.

Command Syntax
CONFigure:SPORT:PARity <string>
Where <string> = 'None' | 'Odd' | 'Even' | 'Always 1' | 'Always 0'

Example Statements
OUTPUT 714:"CONFigure:SPORT:PARity 'None'"

Query Syntax
CONFigure:SPORT:PARity?

Return Format
String
**CONFigure:SPORt:RPACe**

This command sets the receiver pace for the I-BASIC/print serial port data transfer.

**Command Syntax**

CONFigure:SPORt:RPACe <string>

Where <string> = 'None' | 'Xon/Xoff'

**Example Statements**

OUTPUT 714;"CONFigure:SPORt:RPACe 'Xon/Xoff'"

**Query Syntax**

CONFigure:SPORt:RPACe?

**Return Format**

String
CONFigure:SPORt:SINPut

This command sets up the destination of the commands sent to the I-DASIC/print serial port. Commands can be input to the instrument or to I-BASIC.

**Note**

Another method of writing this command exists. This command is completely interchangeable with CONFigure:SPORt:SINPut. The command is:

CONFigure:SPORt:SIN

**Command Syntax**

CONFigure:SPORt:SINPut <string>

or

CONFigure:SPORt:SIN <string>

Where <string> = 'Inst' | 'IBASIC'

**Example Statements**

OUTPUT 714;'CONFigure:SPORt:SINPut 'Inst''

OUTPUT 714;'CONFigure:SPORt:SIN 'IBASIC''

**Query Syntax**

CONFigure:SPORt:SINPut?

or

CONFigure:SPORt:SIN?

**Return Format**

String
CONFigure:SPORt:STOP
This command sets the number of stop bits for the I-DASIC/print serial port data transfer.

Command Syntax

CONFigure:SPORt:STOP <stop_length>
Where <stop_length> = '1 bit' | '2 bits'

Example Statements

OUTPUT 714;"CONFigure:SPORt:STOP '1 bit'"

Query Syntax

CONFigure:SPORt:STOP?

Return Format

String
CONFigure:SPORT:XPACe

This command sets up the transmitted pace of the I-BASIC/PRINT serial port for data transfer.

**Command Syntax**

CONFigure:SPORT:XPACe <string>
Where <string> = 'None' | 'Xon/Xoff'

**Example Statements**

```
OUTPUT 714;"CONFigure:SPORT:XPACe 'Xon/Xoff'
```

**Query Syntax**

CONFigure:SPORT:XPACe?

**Return Format**

String
**CONFigure:SRLocation**

This command sets location of the save/recall register.

**Command Syntax**

```
CONFigure:SRLocation <string>
```

Where `<string>` = 'Internal' | 'Card' | 'RAM' | 'Disk'

**Example Statements**

```
OUTPUT 714; "CONFigure:SRLocation 'Card'"
```

**Query Syntax**

```
CONFigure:SRLocation?
```

**Return Format**

String
CONFigure:TIME
This command sets the HP 8923B clock.

Command Syntax

CONFigure:TIME <value>
Where <value> = HH.MM
HH - hours
MM - minutes

Example Statements

OUTPUT 714;"CONFigure:TIME 17.05"

Query Syntax

CONFigure:TIME?

Return Format

Real
BETest Subsystem

This subsystem controls the bit error ratio measurements.

BETest Syntax Diagram
**BETest:BITS**

Sets the number of bits to be tested in a BER measurement.

**Command Syntax**

BETest:BITS <bit_number>

Where <bit_number> is an integer from 320 through $10^6 - 1$

**Example Statements**

OUTPUT 714;"BETest:BITS 100000"

**Query Syntax**

BETest:BITS?

**Return Format**

Integer
**BETest:BITS:INCRement**

Sets the increment of the amount of bits that are to be tested in a BER measurement.

**Command Syntax**

BETest:BITS:INCRement <increment_parameter> or <integer>

Where <increment_command> = DOWN | UP

Where <integer> = 0 to 10^6 − 1

**Example Statements**

OUTPUT 714;"BETest:BITS:INCRement DOWN"
OUTPUT 714;"BETest:BITS:INCRement UP"
OUTPUT 714;"BETest:BITS:INCRement 100"

**Query Syntax**

BETest:BITS:INCRement?

**Return Format**

Integer

Appendix D for additional related commands.
**BETest:WERRor:CRIterion**

This command selects the detection method for a word error. See “Making Measurements” section in the HP 8923B DECT Test Set User’s Guide for details of the methods “Threshold” and “No-B-field”.

**Command Syntax**

\[ \text{BETest:WERRor:CRIterion: <string>} \]

Where <string> = 'Threshold' | 'No-B-field'

**Example Statements**

\[ \text{OUTPUT 714;"BETest:WERRor:CRIterion 'No-B-field'"} \]
\[ \text{OUTPUT 714;"BETest:WERRor:CRIterion 'Threshold'"} \]

**Query Syntax**

\[ \text{BETest:WERRor:CRIterion?} \]

**Return Format**

String
DECT Subsystem

This chapter describes all the commands related to establishing and maintaining a communication link with a DECT Fixed Part or Portable Part.

Figure 11-1. DECT Syntax Diagram
Figure 11.2. DECT Syntax Diagram
Figure 11-3. DECT Syntax Diagram
DECT: DUMMY[ : STATE ]

This command activates/de-activates the transmission of a dummy bearer.

Command Syntax

DECT: DUMMY <boolean>

or

DECT: DUMMY: STATE <boolean>

Where <boolean> = 0 | 1 | OFF | ON

0 or OFF switches off the dummy bearer

1 or ON switches on the dummy bearer

Example Statements

OUTPUT 714 ; "DECT: DUMMY 1"

or

OUTPUT 714 ; "DECT: DUMMY: STATE 1"

Query Syntax

DECT: DUMMY?

or

DECT: DUMMY: STATE?

Return Format

Boolean
DECT:DUMMy:SYNC:ABORt

This command aborts the IIP 8923D’s synchronization to the dummy bearer (which is transmitted by the FP).

Note

The following commands are equivalent:

- DECT:SYNC:ABORt
- DECT:DUMMy:SYNC:ABORt
- DECT:FP:DUMMy:SYNC:ABORt
- DECT:FIXed:DUMMy:SYNC:ABORt

Command Syntax

DECT:DUMMy:SYNC:ABORt

Example Statements

OUTPUT 714;"DECT:DUMMy:SYNC:ABORt"
DECT:DUMMy:SYNC[:IMMediate]

This command instructs the HP 8923D to synchronize to the dummy bearer. The command should only be used when the EUT is a fixed part.

**Note**

The following commands are equivalent:

- DECT:SYNC
- DECT:SYNC:IMMediate
- DECT:DUMMy:SYNC
- DECT:DUMMy:SYNC:IMMediate
- DECT:PP:DUMMy:SYNC[:IMMediate]
- DECT:FIXed:DUMMy:SYNC[:IMMediate]

**Command Syntax**

- DECT:DUMMy:SYNC
- or
- DECT:DUMmy:SYNC:IMMediate

**Example Statements**

```
OUTPUT 714;"DECT:DUMmy:SYNC"
or
OUTPUT 714;"DECT:DUMmy:SYNC:IMMediate"
```
DECT:EUT

This command configures the HP 8623B to test either a Portable Part or a Fixed Part.

Command Syntax

DECT:EUT <string>
  Where <string> = 'Portable' | 'Fixed'

Example Statements

OUTPUT 714;"DECT:EUT 'Portable'"
OUTPUT 714;"DECT:EUT 'Fixed'"

Query Syntax

DECT:EUT?

Return Format

String
DECT:EUT:PARI?

This query returns an 8 or 9 character hex string; the PARI of the EUT (fixed part) being tested.

Query Syntax

DECT:EUT:PARI?

Return Format

String

Example Return String

000080C64

Note

If there is no EUT connected then the string, ----, is returned.
DECT:EUT:PMID?

Returns a 5 character hex string; the PMID of the EUT (portable part) being tested.

Query Syntax

DECT:EUT:PMID?

Return Format

String

Example Return String

49D3A

Note

If there is no EUT connected then the string, ----, is returned.
DECT:FP:DUUy:CARRier?

This query returns the RF carrier of the dummy bearer when testing a fixed part. There is no HP S923B command to set the dummy bearer carrier when testing. The fixed part initiates the dummy bearer.

Note

The following commands are equivalent:

DECT:FP:DUUy:CARRier
DECT:FIXed:DUUy:CARRier

Query Syntax

DECT:FIXed:DUUy:CARRier?
or
DECT:FP:DUUy:CARRier?

Return Format

String
DECT:FP:DUMMy:SLOT?

This query returns the timeslot for the dummy bearer when testing a fixed part. There is no HP 8823B command to set the dummy bearer timeslot when testing. The fixed part initiates the dummy bearer.

**Note**  
The following commands are equivalent:

```
DECT:FP:DUMMy:SLOT
DECT:FIXed:DUMMy:SLOT
```

**Query Syntax**

```
DECT:FIXed:DUMMy:SLOT?
or
DECT:FP:DUMMy:SLOT?
```

**Return Format**

String
DECT:FP:DUMMy:SYNC:ABORt

This command aborts the synchronization to the dummy bearer when testing a fixed part.

Note

The following commands are equivalent:

- DECT:SYNC:ABORt
- DECT:DUMMy:SYNC:ABORt
- DECT:FP:DUMMy:SYNC:ABORt
- DECT:FIXed:DUMMy:SYNC:ABORt

Command Syntax

- DECT:FP:DUMMy:SYNC:ABORt

Example Statements

- OUTPUT 714;"DECT:FP:DUMMy:SYNC:ABORt"
DECT:FP:DUmMy:SYNC[:IMMediate]

This command instructs the FIP 8923B to synchronize to the dummy bearer. The command should only be used when the EUT is a fixed part.

Note
The following commands are equivalent:

- DECT:SYNC
- DECT:SYNC:IMMediate
- DECT:FP:DUmMy:SYNC
- DECT:FP:DUmMy:SYNC:IMMediate
- DECT:FIXed:DUmMy:SYNC
- DECT:FIXed:DUmMy:SYNC:IMMediate
- DECT:DUmMy:SYNC:IMMediate
- DECT:DUmMy:SYNC:IMMediate

Command Syntax

- DECT:FP:DUmMy:SYNC
- or
- DECT:FP:DUmMy:SYNC:IMMediate
- or
- DECT:FIXed:DUmMy:SYNC
- or
- DECT:FIXed:DUmMy:SYNC:IMMediate

Example Statements

- OUTPUT 714;"DECT:FP:DUmMy:SYNC"
- OUTPUT 714;"DECT:FP:DUmMy:SYNC:IMMediate"
- OUTPUT 714;"DECT:FIXed:DUmMy:SYNC"
- OUTPUT 714;"DECT:FIXed:DUmMy:SYNC:IMMediate"
DECT:FP:TRAFFfic:CARRier
This command sets the RF carrier position for the traffic bearer when testing a fixed part.

**Note**
The following commands are equivalent:
- DECT:TRAFFfic:CARRier
- DECT:FP:TRAFFfic:CARRier
- DECT:FIXed:TRAFFfic:CARRier

**Command Syntax**
- DECT:FIXed:TRAFFfic:CARRier <integer>
- DECT:FP:TRAFFfic:CARRier <integer>
  Where <integer> = 0 through 9

**Example Statements**
- OUTPUT 714:"DECT:FIXed:TRAFFfic:CARRier 7"
- OUTPUT 714:"DECT:FP:TRAFFfic:CARRier 7"

**Query Syntax**
- DECT:FIXed:TRAFFfic:CARRier?
- DECT:FP:TRAFFfic:CARRier?

**Return Format**
- Integer
DECT:FP:TRAFFfic:CARRier:INCRement

This command controls the increment of the traffic bearer’s carrier when testing a fixed part. The command can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

Note

The following commands are equivalent:

- DECT:FP:TRAFFfic:CARRier:INCRement
- DECT:FIXed:TRAFFfic:CARRier:INCRement

Command Syntax

DECT:FIXed:TRAFFfic:CARRier:INCRement <increment_parameter> | <value>

or

DECT:FP:TRAFFfic:CARRier:INCRement <increment_parameter> | <value>

Where <increment_parameter> = UP | DOWN

Where <value> is an integer.

Example Statements

- OUTPUT 714;"DECT:FIXed:TRAFFfic:CARRier:INCRement UP"
- OUTPUT 714;"DECT:FP:TRAFFfic:CARRier:INCRement DOWN"
- OUTPUT 714;"DECT:FIXed:TRAFFfic:CARRier:INCRement 5"
- OUTPUT 714;"DECT:FP:TRAFFfic:CARRier:INCRement 2"

Query Syntax

- DECT:FIXed:TRAFFfic:CARRier:INCRement?
- DECT:FP:TRAFFfic:CARRier:INCRement?

Return Format

- Integer
DECT:FP:TRAFFic:CONNect
This command forces the HP 8923B to initiate a traffic bearer when testing a fixed part.

Note
The following commands are equivalent:

- DECT:TRAFFic:CONNect
- DECT:FIXed:TRAFFic:CONNect

Command Syntax

- DECT:FIXed:TRAFFic:CONNect
  or
- DECT:FP:TRAFFic:CONNect
  or
- DECT:TRAFFic:CONNect

Example Statements

- OUTPUT 714;"DECT:FIXed:TRAFFic:CONNect"
  or
- OUTPUT 714;"DECT:FP:TRAFFic:CONNect"
  or
- OUTPUT 714;"DECT:TRAFFic:CONNect"
DECT:FP:TRAFFic:RELease

This command forces the IIP 8023D to release the current traffic bearer.

Note
The following commands are equivalent:
- DECT:TRAFFic:RELease
- DECT:FP:TRAFFic:RELease
- DECT:FIXed:TRAFFic:RELease

Command Syntax
- DECT:FIXed:TRAFFic:RELease
  or
- DECT:FP:TRAFFic:RELease
  or
- DECT:TRAFFic:RELease

Example Statements
- OUTPUT 714;"DECT:FIXed:TRAFFic:RELease"
  or
- OUTPUT 714;"DECT:FP:TRAFFic:RELease"
  or
- OUTPUT 714;"DECT:TRAFFic:RELease"
DECT:FP:TRAFFic:SLOT
This command sets the timeslot for the traffic bearer when testing a fixed part.

Note
The following commands are equivalent:

DECT:FP:TRAFFic:SLOT
DECT:FIXed:TRAFFic:SLOT

Command Syntax

DECT:FIXed:TRAFFic:SLOT <integer>

or

DECT:FP:TRAFFic:SLOT <integer>
Where <integer> = 0 through 11

Example Statements

OUTPUT 714;"DECT:FIXed:TRAFFic:SLOT 5"
or
OUTPUT 714;"DECT:FP:TRAFFic:SLOT 5"

Query Syntax

DECT:FIXed:TRAFFic:SLOT?
or
DECT:FP:TRAFFic:SLOT?

Return Format

Integer
DECT:FP:TRAFFic:SLOT:INCRement

This command controls the increment of the traffic bearer's timeslot when testing a fixed part. The command can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

Note

The following commands are equivalent:

- DECT:FP:TRAFFic:SLOT:INCRement
- DECT:FIXed:TRAFFic:SLOT:INCRement

Command Syntax

DECT:FIXed:TRAFFic:SLOT:INCRement <increment_parameter> | <value>
or
DECT:FP:TRAFFic:SLOT:INCRement <increment_parameter> | <value>
Where <increment_parameter> = UP | DOWN
Where <value> = an integer from 0 through 11 inclusive.

Example Statements

- OUTPUT 714;"DECT:FIXed:TRAFFic:SLOT:INCRement 4"
or
- OUTPUT 714;"DECT:FP:TRAFFic:SLOT:INCRement 4"

Query Syntax

- DECT:FIXed:TRAFFic:SLOT:INCRement?
or
- DECT:FP:TRAFFic:SLOT:INCRement?

Return Format

- Integer
DECT:LOGGing[:STATE]e
This command activates/deactivates the DECT protocol logging.

Command Syntax
DECT:LOGGing <string>
or
DECT:LOGGing:STATE <string>
Where <string> = 'ON' | 'OFF'

Example Statements
OUTPUT 714:"DECT:LOGGing 'ON'"
or
OUTPUT 714:"DECT:LOGGing:STATE 'ON'"

Query Syntax
DECT:LOGGing?
or
DECT:LOGGing:STATE?

Return Format
String
DECT:LOGGing:SPORt:BAUD

This command sets the baud rate for the protocol logging period.

Command Syntax

DECT:LOGGing:SPORt:BAUD <string>
Where <string> = '19200' | '9600' | '1200' | '300'

Example Statements

OUTPUT 714;"DECT:LOGGing:SPORt:BAUD '9600'"

Query Syntax

DECT:LOGGing:SPORt:BAUD?

Return Format

String
DECT:LOGGing:SPORt:HANDshake

This command sets the receiver pace for the protocol logging serial port.

Command Syntax

DECT:LOGGing:SPORt:HANDshake <string>
Where <string> = 'None' | 'Xon/Xoff'

Example Statements

OUTPUT 714:"DECT:LOGGing:SPORt:HANDshake 'Xon/Xoff"

Query Syntax

DECT:LOGGing:SPORt:HANDshake?

Return Format

String
DECT:PARI
This command sets the Primary Access Rights Identity.

Command Syntax
DECT:PARI <string>
Where <string> = '9 hexadecimal characters' or '8 hexadecimal characters'

Example Statements
OUTPUT 714:"DECT:PARI '000009D3A'"

Query Syntax
DECT:PARI?

Return Format
String
DECT:PMID

This command sets the Portable Part MAC Identity. This is a five character hexadecimal number, which is entered as a string.

Command Syntax

DECT:PMID <string>
Where <string> = '5 hexadecimal digits'

Example Statements

OUTPUT 714;"DECT:PMID 'ABCDE'"

Query Syntax

DECT:PMID?

Return Format

String
DECT:PP:DUMMY[:STATE]
This command sets the dummy bearer state to on or off.

Note
The following commands are equivalent:

DECT:DUMMY
DECT:DUMMY:STATE
DECT:PP:DUMMY
DECT:PP:DUMMY:STATE
DECT:PORTable:DUMMY
DECT:PORTable:DUMMY:STATE

Controlling the position of the dummy bearer is only relevant when testing a DECT portable part.

Command Syntax

DECT:PP:DUMMY:STATE <boolean-parameter>
DECT:PORTable:DUMMY:STATE <boolean-parameter>
DECT:PP:DUMMY <boolean-parameter>
DECT:PORTable:DUMMY <boolean-parameter>

Where <boolean-parameter> = 0 | 1 | OFF | ON

Example Statement

OUTPUT 714;"DECT:PP:DUMMY:STATE ON"
OUTPUT 714;"DECT:PORTable:DUMMY:STATE ON"
OUTPUT 714;"DECT:PP:DUMMY ON"
OUTPUT 714;"DECT:PORTable:DUMMY ON"

Query Syntax

DECT:PP:DUMMY:STATE?
DECT:PORTable:DUMMY:STATE?
DECT:PP:DUMMY?
DECT:PORTable:DUMMY?

Return Format

Boolean
DECT:PP:Dummy:Carrier
This command sets the RF carrier position for the dummy bearer when testing a portable part.

Note
The following commands are equivalent:

- DECT:PP:Dummy:Carrier
- DECT:Portable:Dummy:Carrier

Command Syntax

- DECT:Portable:Dummy:Carrier <integer>
- DECT:PP:Dummy:Carrier <integer>
  Where <integer> = 0 | 9

Example Statements

- OUTPUT 714;"DECT:Portable:Dummy:Carrier 7"
- OUTPUT 714;"DECT:PP:Dummy:Carrier 7"

Query Syntax

- DECT:Portable:Dummy:Carrier?
- DECT:PP:Dummy:Carrier?

Return Format

- Integer
DECT:PP:DUmMy:CARRier:INCRement

This command controls the increment of the traffic bearer’s carrier when testing a portable part. This can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

Note

The following commands are equivalent:

DECT:PP:DUmMy:CARRier:INCRement
DECT:PORTable:DUmMy:CARRier:INCRement

Command Syntax

DECT:PORTable:DUmMy:CARRier:INCRement <increment_parameter> | <value>
or
DECT:PP:DUmMy:CARRier:INCRement <increment_parameter> | <value>
Where <increment_parameter> = UP | DOWN
Where <value> is an integer.

Example Statements

OUTPUT 714;"DECT:PORTable:DUmMy:CARRier:INCRement UP"
OUTPUT 714;"DECT:PP:DUmMy:CARRier:INCRement DOWN"
OUTPUT 714;"DECT:PORTable:DUmMy:CARRier:INCRement 2"
OUTPUT 714;"DECT:PP:DUmMy:CARRier:INCRement 2"

Query Syntax

DECT:PORTable:DUmMy:CARRier:INCRement?
or
DECT:PP:DUmMy:CARRier:INCRement?

Return Format

Integer
DECT:PP:Dummy:SLOT

This command sets the timeslot position for the dummy bearer.

**Note**

The following commands are equivalent:

DECT:PP:Dummy:SLOT
DECT:PORTable:Dummy:SLOT

---

**Command Syntax**

DECT:PORTable:Dummy:SLOT <integer>

or

DECT:PP:Dummy:SLOT <integer>

Where <integer> = 0 through 11 inclusive.

---

**Example Statements**

OUTPUT 714;"DECT:PORTable:Dummy:SLOT 10"
OUTPUT 714;"DECT:PP:Dummy:SLOT 1"

---

**Query Syntax**

DECT:PORTable:Dummy:SLOT?

or

DECT:PP:Dummy:SLOT?

---

**Return Format**

Integer
DECT:PP:Dummy:SLOT:INCREMENT

This command controls the increment of the dummy bearer's timeslot when testing a portable part. The command can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

**Note**

The following commands are equivalent:

DECT:PP:Dummy:SLOT:INCREMENT
DECT:Portable:Dummy:SLOT:INCREMENT

**Command Syntax**

DECT:PORTable:DUMMy:SLOT:INCREMENT <increment_parameter> | <Integer>

or

DECT:PP:DUMMy:SLOT:INCREMENT <increment_parameter> | <value>
Where <increment_parameter> = UP | DOWN
Where <value> is an optional number between 0 and 11.

**Example Statements**

OUTPUT 714;"DECT:PORTable:DUMMy:SLOT:INCREMENT UP"
OUTPUT 714;"DECT:PP:DUMMy:SLOT:INCREMENT DOWN"
OUTPUT 714;"DECT:PORTable:DUMMy:SLOT:INCREMENT 0"
OUTPUT 714;"DECT:PP:DUMMy:SLOT:INCREMENT 1"

**Query Syntax**

DECT:PORTable:DUMMy:SLOT:INCREMENT?

or

DECT:PP:DUMMy:SLOT:INCREMENT?

**Return Format**

Integer
DECT:PP:TRAFFic:CARRier
This command sets the RF carrier of the traffic bearer when testing a portable part.

Note
The following commands are equivalent:
- DECT:PP:TRAFFic:CARRier
- DECT:PORTable:TRAFFic:CARRier

Command Syntax
- DECT:PORTable:TRAFFic:CARRier <integer>
  or
- DECT:PP:TRAFFic:CARRier <integer>
  Where <integer> = 0 through 9

Example Statements
- OUTPUT 714,"DECT:PORTable:TRAFFic:CARRier 7"
  or
- OUTPUT 714,"DECT:PP:TRAFFic:CARRier 7"

Query Syntax
- DECT:PORTable:TRAFFic:CARRier?
  or
- DECT:PP:TRAFFic:CARRier?

Return Format
- Integer
DECT:PP:TRAFFic:CARRier:INCRement

This command controls the increment of the traffic bearer's carrier when testing a portable part. The command can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

Note: The following commands are equivalent:

- DECT:PP:TRAFFic:SLOT:INCRement
- DECT:PORTable:TRAFFic:SLOT:INCRement

Command Syntax

DECT:PORTable:TRAFFic:CARRier:INCRement <increment_parameter> | <integer>

or

DECT:PP:TRAFFic:CARRier:INCRement <increment_parameter> | <value>

Where <increment_parameter> = UP | DOWN

Where <value> = any integer less than 10

Example Statements

OUTPUT 714;"DECT:PORTable:TRAFFic:CARRier:INCRement 'DOWN'"
OUTPUT 714;"DECT:PORTable:TRAFFic:CARRier:INCRement 5"
OUTPUT 714;"DECT:PP:TRAFFic:CARRier:INCRement 'UP'"
OUTPUT 714;"DECT:PP:TRAFFic:CARRier:INCRement 2"

Query Syntax

DECT:PORTable:TRAFFic:CARRier:INCRement?

or

DECT:PP:TRAFFic:CARRier:INCRement?

Return Format

Integer
DECT:PP:TRAFFic:CONNect
This command causes the HP 8923B to initiate a call/traffic bearer.

**Note**
The following commands are equivalent:

- DECT:TRAFFic:CONNect
- DECT:PP:TRAFFic:CONNect
- DECT:PORTable:TRAFFic:CONNect

The dummy bearer must be established before making a traffic bearer connection.

**Command Syntax**

- DECT:PORTable:TRAFFic:CONNect
- DECT:PP:TRAFFic:CONNect
- DECT:TRAFFic:CONNect

**Example Statements**

- OUTPUT 714;"DECT:PORTable:TRAFFic:CONNect"
- OUTPUT 714;"DECT:PP:TRAFFic:CONNect"
- OUTPUT 714;"DECT:TRAFFic:CONNect"
DECT:PP:TRAFFic:RELease
This command causes the HP 8923B to release the current call/traffic bearer.

**Note** The following commands are equivalent:
- DECT:TRAFFic:RELease
- DECT:PP:TRAFFic:RELease
- DECT:PORTable:TRAFFic:RELease

**Command Syntax**
- DECT:PORTable:TRAFFic:RELease
- or
- DECT:PP:TRAFFic:RELease

**Example Statements**
- OUTPUT 714;"DECT:PORTable:TRAFFic:RELease"
- or
- OUTPUT 714;"DECT:PP:TRAFFic:RELease"
DECT:PP:TRAFFic:SLOT

This command sets the timeslot position for the traffic bearer when testing a PP.

### Note
The following commands are equivalent:

- DECT:PP:TRAFFic:SLOT
- DECT:PORTable:TRAFFic:SLOT

### Command Syntax

```
DECT:PORTable:TRAFFic:SLOT <integer>
```

or

```
DECT:PP:TRAFFic:SLOT <integer>
Where <integer> = 0 through 11
```

### Example Statements

```
OUTPUT 714;"DECT:PORTable:TRAFFic:SLOT 5"
```

or

```
OUTPUT 714;"DECT:PP:TRAFFic:SLOT 5"
```

### Query Syntax

```
DECT:PORTable:TRAFFic:SLOT?
```

or

```
DECT:PP:TRAFFic:SLOT?
```

### Return Format

Integer
DECT:PP:TRAFFic:SLOT:INCREMENT

This command controls the increment of the traffic bearer’s timeslot when testing a portable part. The command can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

**Note**

The following commands are equivalent:

- `DECT:PP:TRAFFic:SLOT:INCREMENT`
- `DECT:PORTable:TRAFFic:SLOT:INCREMENT`

**Command Syntax**

```
DECT:PORTable:TRAFFic:SLOT:INCREMENT <increment_parameter> | <value>
```

or

```
DECT:PP:TRAFFic:SLOT:INCREMENT <increment_parameter> | <integer>
```

Where `<increment_parameter>` = UP | DOWN
Where `<value>` = an integer from 0 through 11 inclusive.

**Example Statements**

```
OUTPUT 714;"DECT:PORTable:TRAFFic:SLOT:INCREMENT 4"
```

or

```
OUTPUT 714;"DECT:PP:TRAFFic:SLOT:INCREMENT 4"
```

**Query Syntax**

```
DECT:PORTable:TRAFFic:SLOT:INCREMENT?
```

or

```
DECT:PP:TRAFFic:SLOT:INCREMENT?
```

**Return Format**

Integer
DECT:PROProprietarY:RX:AFIeld:MTAil:TEST:ESCape?
This query reads the escape test message received from the EUT.

**Note**
The following commands are equivalent:

- DECT:PROProprietarY:RX:AFIeld:MTAil:TEST:ESCape?
- DECT:PROProprietarY:RECeive:AFIeld:MTAil:TEST:ESCape?

**Query Syntax**

- DECT:PROProprietarY:RECeive:AFIeld:MTAil:TEST:ESCape?
  or
- DECT:PROProprietarY:RX:AFIeld:MTAil:TEST:ESCape?

**Return Format**

- String
DECT:PROPrietary:RX:AFIeld:MTAil:TEST:ESCape:STATus?

This query reads the status of the received escape message.
The status will be "New" whenever an escape test message has been received but has not been
read from the EUT.
The status will be "Old" if the received escape test message has been read.

Note
The following commands are equivalent:
DECT:PROPrietary:RX:AFIeld:MTAil:TEST:ESCape:STATus?
DECT:PROPrietary:RECeive:AFIeld:MTAil:TEST:ESCape:STATus?

Query Syntax
DECT:PROPrietary:RECeive:AFIeld:MTAil:TEST:ESCape:STATus?
or
DECT:PROPrietary:RX:AFIeld:MTAil:TEST:ESCape:STATus?

Return Format
String
DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna

This command overrides the antenna value and sets a fixed value to be used in the DEFEAT.ANTENNA.DIVERSITY test message.

**Note**

The following commands are equivalent:

- DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna?
- DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ANTenna?

**Command Syntax**

- DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna <integer>
- or
- DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ANTenna <integer>
  - Where <integer> = 0 through 7

**Example Statements**

- OUTPUT 714;"DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna 2"
- or
- OUTPUT 714;"DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ANTenna 2"

**Query Syntax**

- DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna?
- or
- DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ANTenna?

**Return Format**

- Integer
DECT:PROPrrietary:TX:AField:MTail:TEST:ANTenna:INCREMENT

This command controls the increment for the antenna diversity control command.

This can be used to:
1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

Note: The following commands are equivalent:

DECT:PROPrrietary:TRANsmit:AField:MTail:TEST:ANTenna:INCREMENT?
DECT:PROPrrietary:TX:AField:MTail:TEST:ANTenna:INCREMENT?

Command Syntax

or
Where <increment_parameter> = UP | DOWN
Where <value> is a integer with a maximum value of six.

Example Statements

OUTPUT 714:"DECT:PROPrrietary:TRANsmit:AField:MTail:TEST:ANTenna:INCREMENT 5"
OUTPUT 714:"DECT:PROPrrietary:TRANsmit:AField:MTail:TEST:ANTenna:INCREMENT UP"
OUTPUT 714:"DECT:PROPrrietary:TX:AField:MTail:TEST:ANTenna:INCREMENT 2"
OUTPUT 714:"DECT:PROPrrietary:TX:AField:MTail:TEST:ANTenna:INCREMENT DOWN"

Query Syntax

DECT:PROPrrietary:TRANsmit:AField:MTail:TEST:ANTenna:INCREMENT?
DECT:PROPrrietary:TX:AField:MTail:TEST:ANTenna:INCREMENT?

Return Format

String
DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna:SEND

This command sends the DEFEAT_ANTENNA DIVERSITY test message to the part being tested using the value defined in :TX:AFIeld:MTAil:TEST:ANT.

Note
The following commands are equivalent:

DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna:SEND
DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ANTenna:SEND

Command Syntax

DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna:SEND
or
DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ANTenna:SEND

Example Statements

OUTPUT 714;"DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ANTenna:SEND"
or
OUTPUT 714;"DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ANTenna:SEND"
DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ESCape

This command sets the contents of the escape test message, which is to be transmitted to the EUT.

**Note**

The following commands are equivalent:

- DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ESCape?
- DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ESCape?

---

**Command Syntax**

DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ESCape <string>

or

DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ESCape <string>

Where <string> = user definable (8 hexadecimal characters)

**Example Statements**

- OUTPUT 714;"DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ESCape 'ABCDEF78'"
- or
- OUTPUT 714;"DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ESCape 'ABCDEF78'"

**Query Syntax**

- DECT:PROPrietary:TRANsmit:AFIeld:MTAil:TEST:ESCape?
- or
- DECT:PROPrietary:TX:AFIeld:MTAil:TEST:ESCape?

**Return Format**

- String
DECT:PROPrity:TX:AFIeld:MTAil:TEST:ESCaPe:SEND

This command sends the ESCAPE test message to the EUT.

Note

The following commands are equivalent:

DECT:PROPrity:TX:AFIeld:MTAil:TEST:ESCaPe:SEND
DECT:PROPrity:TRANsmit:AFIeld:MTAil:TEST:ESCaPe:SEND

Command Syntax

DECT:PROPrity:TRANsmit:AFIeld:MTAil:TEST:ESCaPe:SEND
or
DECT:PROPrity:TX:AFIeld:MTAil:TEST:ESCaPe:SEND

Example Statements

OUTPUT 714;"DECT:PROPrity:TRANsmit:AFIeld:MTAil:TEST:ESCaPe:SEND"
or
OUTPUT 714;"DECT:PROPrity:TX:AFIeld:MTAil:TEST:ESCaPe:SEND"
DECT:STATus?

This query returns the call status field. The status options are for the portable part are:

- Off
- Idle
- Calling
- Connected

The status options are for the fixed part are:

- Off
- Sync
- Locked
- Connected

Query Syntax

DECT:STATus?

Return Format

String
DECT:SYNC[:IMMediate]

This command starts the synchronization to the dummy bearer when testing a fixed part.

**Note**

The following commands are equivalent:

- DECT:SYNC
- DECT:SYNC:IMMediate
- DECT:DUMMy:SYNC
- DECT:DUMMy:SYNC:IMMediate
- DECT:FP:DUMMy:SYNC
- DECT:FP:DUMMy:SYNC:IMMediate
- DECT:FIXed:DUMMy:SYNC
- DECT:FIXed:DUMMy:SYNC:IMMediate

**Command Syntax**

DECT:SYNC:IMMediate
DECT:SYNC:ABORt

This command aborts the synchronization to the dummy bearer.

**Note**

The following commands are equivalent:

DECT:SYNC:ABORt
DECT:DUMMY:SYNC:ABORt
DECT:FP:DUMMY:SYNC:ABORt
DECT:FIXed:DUMMY:SYNC:ABORt

**Command Syntax**

DECT:SYNC:ABORt
DECT:TRAFfic:CONNect
This command forces the HP 8923B to initiate a traffic bearer.

Note
The following commands are equivalent:

DECT:TRAFfic:CONNect
DECT:FP:TRAFfic:CONNect
DECT:FIXed:TRAFfic:CONNect
DECT:PP:TRAFfic:CONNect
DECT:PORTable:TRAFfic:CONNect

Command Syntax

DECT:TRAFfic:CONNect
DECT:TRAFFic:RELSease

This command forces the HP 8923B to release the current traffic bearer.

**Note**

The following commands are equivalent:

- DECT:TRAFFic:RELSease
- DECT:FP:TRAFFic:RELSease
- DECT:FIXed:TRAFFic:RELSease
- DECT:PP:TRAFFic:RELSease
- DECT:PORTable:TRAFFic:RELSease

**Command Syntax**

DECT:TRAFFic:RELSease
**DECT:VOICe:DESTination**
This command sets the destination of the received speech/B-Field from the EUT.

**Command Syntax**

DECT:VOICe:DESTination <string>

Where <string> = 'None' | 'RearPanel' | 'Echo'

**Example Statements**

OUTPUT 714;"DECT:VOICe:DESTination 'RearPanel'"

**Query Syntax**

DECT:VOICe:DESTination?

**Return Format**

String
DISPlay Subsystem

This chapter describes all the commands related to the screen display of the HP 8923B.

Figure 12-1. DISPlay Syntax Diagram
DISPlay[:SCReen]
This command switches the display to the desired screen.

Command Syntax

DISPlay
or
DISPlay:SCReen <screen_identifier>
Where <screen_identifier> = AUDio | BETest | CALL | CONFIGure
DBField | EXTSource | FREQ | HELP | IOConfig | LOGging |
MCNTL | MESSAGE | NTPower | OSCilloscope | PCONfigure |
PRPrietary | PTFall | PTMid | PTRise | PUP | RFParameter |
SERVice | TCONfigure | TESTs | TReq | TIBasic | TMAKe | TPARm |
TSEQn | TSPEc

Example Statements

0OUTPUT 714;"DISPlay CALL"
or
0OUTPUT 714;"DISPlay:SCReen CALL"

Query Syntax

DISPlay?
or
DISPlay:SCReen?

Return Format

String
**DISPlay:BETest**

This command selects the display of the BER measurement to be either a count or ratio. Count shows the accumulated number of bits/frames in error, ratio shows the ratio of bits/frames in error to bits/frames tested.

**Command Syntax**

```
DISPlay:BETest <string>
   Where <string> = 'CNT' | 'RATIO'
```

**Example Statements**

```
OUTPUT 714;"DISPlay:BETest 'CNT'"
OUTPUT 714;"DISPlay:BETest 'RATIO'"
```

**Query Syntax**

```
DISPlay:BETest?
```

**Return Format**

String
**DISPlay:BE*Test:VIEW**

This command selects the measurement viewed to be either bit error ratio or word error ratio.

**Command Syntax**

```
DISPlay:BE*Test:VIEW <string>
Where <string> = 'BER' | 'WER'
```

**Example Statements**

```
OUTPUT 714;"DISPlay:BE*Test:VIEW 'BER'"
OUTPUT 714;"DISPlay:BE*Test:VIEW 'WER'"
```

**Query Syntax**

```
DISPlay:BE*Test:VIEW?
```

**Return Format**

String
DISPlay:FREQuency
This command selects which frequency screen is to be displayed.

Command Syntax
DISPlay:FREQuency <string>
  Where <string> = 'MAXMIN0' | 'MAXMIN1' | 'AVERAGE01'

Example Statements
OUTPUT 714;"DISPlay:FREQuency 'MAXMIN1'"

Query Syntax
DISPlay:FREQuency?

Return Format
String
**DISPlay:OSCilloscope**

This command selects which oscilloscope screen is to be displayed.

**Command Syntax**

\[ \text{DISPlay:OSCilloscope } <\text{string}> \]

Where \(<\text{string}> = '\text{MAIN}' | '\text{VOLT/TIME}' | '\text{MARKER}' | '\text{TRIGGER1}' | '\text{TRIGGER2}'\]

**Example Statements**

\[ \text{OUTPUT 714;}"\text{DISPlay:OSCilloscope 'VOLT/TIME'"} \]

**Query Syntax**

\[ \text{DISPlay:OSCilloscope?} \]

**Return Format**

String
ESOurre Subsystem

This subsystem provides access to external controls and parameters.

ESOurre Syntax Diagram
**ESOource:POWer:ADVance**

Positions the leading edge of the trigger signal that goes to the external source.

**Command Syntax**

ESOource:POWer:ADVance <power_advance_time_value>
Where <power_advance_time_value> is integer values of bit-periods in the range 0-31, or its equivalent real number in micro-seconds or milli-seconds.

**Note**
The ESOource:POWer:ADVance command is dependant on the units set by the ESOource:POWer:ADVance:UNits command.

**Example Statements**

OUTPUT 714; "ESOource:POWer:ADVance 13"

**Query Syntax**

ESOource:POWer:ADVance?

**Return Format**

Real
ESOure:POWer:OPOSiton

Sets the position of the trailing edge of the trigger signal that is supplied to the external source.

Command Syntax

ESOure:POWer:OPOSiton <off_position_time_value>
Where <off_position_time_value> is integer values of bit-periods in the range 416-447, or its equivalent real number in micro-seconds or milli-seconds.

Note
The ESOure:POWer:OPOSiton command is dependant on the units set by the ESOure:POWer:OPOSiton:UNits command.

Example Statements

OUTPUT 714:"ESOure:POWer:OPOSiton 421"

Query Syntax

ESOure:POWer:OPOSiton?

Return Format
Real
ESOrcue: SLOT
Sets the slot on which the data will appear.

**Command Syntax**

ESOrcue: SLOT <integer>
Where <integer> is in the range 0 through 23.

**Example Statements**

OUTPUT 714; "ESOrcue: SLOT 19"

**Query Syntax**

ESOrcue: SLOT?

**Return Format**

Integer
ESOunce:SLOT:INCReoment
Sets the increment value or instructs the increment/decrement to be made.

Command Syntax
ESOunce:SLOT:INCReoment <increment parameter>
Where <increment parameter> is UP | DOWN | <increment value>,
and <increment value> is an integer no greater than the difference between
the minimum and maximum values of the entry field.

Example Statements
OUTPUT 714;"ESOunce:SLOT:INCReoment 3"

Query Syntax
ESOunce:SLOT:INCReoment?

Return Format
Integer
**ESSource:STATE**
Enables/disables the trigger output.

**Command Syntax**

```
ESSource:STATE <boolean>
Where <boolean> is 'On' | 'Off' | '1' | '0'
```

**Example Statements**

```
OUTPUT 714;"ESSource:STATE 'On'"
```

**Query Syntax**

```
ESSource:STATE?
```

**Return Format**

String
**ESource:Pattern**

Sets the data pattern output to the external source.

**Command Syntax**

```
ESource:Pattern <string>
```

Where `<string>` is 'DM0' | 'DM1' | 'DM2' | 'FACC' | 'FDEV2_FS'.

**Example Statements**

```
OUTPUT 714;"ESource:Pattern 'DM2'"
```

**Query Syntax**

```
ESource:Pattern? 
```

**Return Format**

String
MEASure Subsystem

The MEASure subsystem is used to set up the measurement parameters and queries the results.
MEASure Syntax Diagram
MEASure Syntax Diagram
MEASure Syntax Diagram
MEASure:AUDio:ACVolts?
This command returns the measured AC voltage.

Query Syntax
MEASure:AUDio:ACVolts?

Note  The AFANalyzer:VOLTage control must be set to 'AC' for this query to return a value.

Return Formats
Real

Note  See Appendix B for additional related commands.
**MEASure:AU Dio:DCVolts?**

This command queries the DC voltmeter on the AU dio screen; it measures the DC voltage on the front panel AUDIO IN connector.

**Query Syntax**

`MEASure:AU dio:DCVolts?`

**Note**

The AFANalyzer:VOLTage control must be set to ‘AC’ for this query to return a value.

**Return Formats**

Real

See Appendix B for additional related commands.
MEASure:AU Dio:FR EQuency?

This command queries the frequency counter on the AU Dio screen; it measures the audio frequency presented to the front panel AU Dio IN connector, or the received audio from the internal CODEC.

Query Syntax

MEASure:AU Dio:FR EQuency?

Return Format

Real

See Appendix B for additional related commands.
MEASure:AUDio:OSCilloscope:MARKer:LEVel:VOLTs?
This command queries the level in volts of the marker on the oscilloscope screen.

Query Syntax
  MEASure:AUDio:OSCilloscope:MARKer:LEVel:VOLTs?

Return Format
  Real
See Appendix B for additional related commands.
MEASure:AUDio:OSCilloscope:MARKer:TIME?

This command queries the time measurement associated with the marker on the oscilloscope screen.

Query Syntax

MEASure:AUDio:OSCilloscope:MARKer:TIME?

Return Format

Real

See Appendix B for additional related commands.
MEASure:AUDio:OSCilloscope:TRACE?

This command reads the discrete data points which make up the oscilloscope trace.

Query Syntax
   MEASure:AUDio:OSCilloscope:TRACE?

Return Format
   417 real numbers corresponding to the values of the trace displayed on the screen.
MEASURE:BETest:BERror:COUNT?

This command queries the number of bit errors on the completed BER measurement.

Query Syntax

MEASURE:BETest:BERror:COUNT?

Return Format

Integer
MEASure:BETest:BERRor:ICOunt?
This command queries the number of bit errors on the intermediate BER measurement.

Query Syntax

MEASure:BETest:BERRor:ICOunt?

Return Format

Integer
MEASure:BETest:BERRor:IRATio?

This command queries the bit error ratio of the intermediate BER measurement.

Query Syntax

MEASure:BETest:BERRor:IRATio?

Return Format

Real

See Appendix B for additional related commands.
MEASure:BETest:BERRor:RATio?

This command queries the bit error ratio of the completed BER measurement.

Query Syntax

MEASure:BETest:BERRor:RATio?

Return Format

Real

See Appendix B for additional related commands.
MEASure:BETest:BTESsted?

This command queries the number of bits that have been tested on the completed DER measurement.

Query Syntax

MEASure:BETest:BTESsted?

Return Format

Integer
MEASure:BETest:IBTested?

This command queries the number of bits that have been tested on the intermediate BER measurement.

Query Syntax

MEASure:BETest:IBTested?

Return Format

Integer
MEASure:BETest:IWTested?

This command queries the number of words tested on the intermediate BER or WER measurement.

Query Syntax

   MEASure:BETest:IWTested?

Return Format

   Integer
MEASure:BETest:WERRor:COUNt?

This command queries the number of word/frame errors on the completed WER measurement.

Query Syntax

   MEASure:BETest:WERRor:COUNt?

Return Format

   Integer
MEASure:BETest:WERRor:ICount?

This command queries the number of word/frame errors on the intermediate WER measurement.

Query Syntax

MEASure:BETest:WERRor:ICount?

Return Format

Integer
**MEASure:BETest:WERRor:IRATio?**

This command queries the word/frame error ratio of the intermediate WER measurement.

**Query Syntax**

`MEASure:BETest:WERRor:IRATio?`

**Return Format**

Real

See Appendix B for additional related commands.
**MEASure:BETest:WERRor:RATio?**

This command queries the word/frame error ratio of the completed WER measurement.

**Query Syntax**

`MEASure:BETest:WERRor:RATio?`

**Return Format**

Real

See Appendix B for additional related commands.
MEASure:BETest:WTESted?

This command queries the number of words tested on the completed BER or WER measurement.

Query Syntax

MEASure:BETest:WTESted?

Return Format

Integer
MEASure:MODE

This command sets the IIP 8923 for normal (burst) or CW measurements.

Command Syntax

MEASure:MODE <string>

Where <string> = 'Normal' | 'CW'

Example Statements

OUTPUT 714;"MEASure:MODE 'Normal'"

Query Syntax

MEASure:MODE?

Return Format

String
MEASure:PACKet

This command sets the size of the DECT packet for measurement.

**Command Syntax**

```
MEASure:PACKet <string>
Where <string> = 'P00' | 'P32'
```

**Example Statements**

```
OUTPUT 714;"MEASure:PACKet 'P32'"
```

**Query Syntax**

```
MEASure:PACKet?
```

**Return Format**

String
MEASure:PATTern
This command sets the measurement pattern for the HP 8923.

Command Syntax

    MEASure:PATTern <string>
    Where <string> = 'DM0' | 'DM1' | 'DM2' | 'FACC' | 'FDEV1_FS' |
    'FDEV2_FS' | 'USER_DEF'

Example Statements

    OUTPUT 714;"MEASure:PATTern 'FACC'"

Query Syntax

    MEASure:PATTern?

Return Format

    String

For a description of the patterns used in making measurements, refer to the HP 8923B DECT Test Set User's Manual.
MEASure:PATTern:DBField

This command sets the B-field of the HP 8923's user definable packet.

Command Syntax

    MEASure:PATTern:DBField <string>

Where <string> is a list of eighty hexadecimal characters which correspond to the 320 bits of the B-field.

Example Statements

    OUTPUT 714;"MEASure:PATTern 'USER_DEF'"

Query Syntax

    MEASure:PATTern:DBField?

Return Format

    String

For a description of valid test patterns for use with the defined B-Field, refer to the HP 8923B DECT Test Set User's Manual.
MEASure:RF:FREQuency:ACCuracy?
This query returns the value of the frequency accuracy measurement.

Query Syntax
  MEASure:RF:FREQuency:ACCuracy?

Return Format
  Real
See Appendix B for additional related commands.
MEASure:RF:FREQuency:COMPosite?

This query returns the frequency accuracy and all of the frequency deviation measurements. They are returned in the order:

- Frequency accuracy,
- Frequency deviation maximum for a zero,
- Frequency deviation minimum for a zero,
- Frequency deviation average for a zero,
- Frequency deviation maximum for a one,
- Frequency deviation minimum for a one,
- Frequency deviation average for a one.

Query Syntax

MEASure:RF:FREQuency:COMPosite?

Return Format

Real,real,real,real,real,real,real.
MEASure:RF:FREQuency:DEViation:ONE:BAverage?

This query returns the average frequency deviation measurement of a modulated one(1) across a single burst.

Query Syntax

MEASure:RF:FREQuency:DEViation:ONE:BAverage?

Return Format

Real

See Appendix B for additional related commands.
MEASURE:RF:FREQUENCY:DEViation:ONE:BMAXimum?

This query returns the maximum frequency deviation measurement of a modulated one(1) across a single burst.

Query Syntax

MEASURE:RF:FREQUENCY:DEViation:ONE:BMAXimum?

Return Format

Real

See Appendix B for additional related commands.
**MEASure:RF:FREQuency:DEViation:ONE:BMINimum?**

This query returns the minimum frequency deviation measurement of a modulated one(1) across a single burst.

**Query Syntax**

```
MEASure:RF:FREQuency:DEViation:ONE:BMINimum?
```

**Return Format**

Real

See Appendix B for additional related commands.
MEASure:RF:FREQuency:DEViation:ZERO:BAVerage?

This query returns the average frequency deviation measurement of a modulated zero(0) across a single burst.

Query Syntax

MEASure:RF:FREQuency:DEViation:ZERO:BAVerage?

Return Format

Real

See Appendix B for additional related commands.
MEASure:RF:FREQuency:DEViation:ZERO:BMAXimum?

This query returns the maximum frequency deviation measurement of a modulated zero(0) across a single burst.

Query Syntax

MEASure:RF:FREQuency:DEViation:ZERO:BMAXimum?

Return Format

Real

See Appendix B for additional related commands.
**MEASure:RF:FREQuency:DEViation:ZERO:BMINimum?**

This query returns the minimum frequency deviation measurement of a modulated zero(0) across a single burst.

**Query Syntax**

`MEASure:RF:FREQuency:DEViation:ZERO:BMINimum?`

**Return Format**

Real

See Appendix B for additional related commands.
MEASure:RF:FREQuency:DRIFt?

This query returns the frequency drift across a single burst.

Query Syntax

MEASure:RF:FREQuency:DRIFt?

Return Format

Real

See Appendix B for additional related commands.
MEASure:RF:NTPower?
This query returns the value of the Normal Transmitted Power measurement (NTP).

Query Syntax

MEASure:RF:NTPower?

Return Format

Real

See Appendix B for additional related commands.
MEASure:RF:PTIME:MARKer:LEVEL:FALL?

This query returns the level of the marker on the power time falling edge screen.

Query Syntax

MEASure:RF:PTIME:MARKer:LEVEL:FALL?

Return Format

Real

See Appendix B for additional related commands.
**MEASure:RF:PTIMe:MARKer:LEVEL:MID?**

This query returns the level of the marker on the power time middle section screen.

**Query Syntax**

MEASure:RF:PTIMe:MARKer:LEVEL:MID?

**Return Format**

Real

See Appendix B for additional related commands.
MEASure:RF:PTIMe:MARKer:LEVEL:RISE?
This query returns the level of the marker on the power time rising edge screen.

**Query Syntax**

MEASure:RF:PTIMe:MARKer:LEVEL:RISE?

**Return Format**

Real

See Appendix B for additional related commands.
MEASURE:RF:PTIME:MARKer:TIME:FALL?

This query returns the timing of the marker on the power time falling edge screen.

Query Syntax

MEASURE:RF:PTIME:MARKer:TIME:FALL?

Return Format

Real

See Appendix B for additional related commands.
MEASure:RF:PTIME:MARKer:TIME:MID?
This query returns the timing of the marker on the power time middle section screen.

Query Syntax
MEASure:RF:PTIME:MARKer:TIME:MID?

Return Format
Real
See Appendix B for additional related commands.
**MEASure:RF:PTIMe:MARKer:TIME:RISE?**

This query returns the timing of the marker on the power time rising edge screen.

**Query Syntax**

`MEASure:RF:PTIMe:MARKer:TIME:RISE?`

**Return Format**

Real

See Appendix B for additional related commands.
MEASure:RF:PTIMe:MASK:FALL?

This query returns whether the DECT signal falls within the power/time template for the fall section of the burst.

**Query Syntax**

```
MEASure:RF:PTIMe:MASK:FALL?
```

**Return Format**

String - "PASS" or "FAIL"
MEASure:RF:PTIMe:MASK:MID?

This query returns whether the DECT signal falls within the power/time template for the middle section of the burst.

Query Syntax

MEASure:RF:PTIMe:MASK:MID?

Return Format

String - “PASS” or “FAIL”
MEASure:RF:PTIMe:MASK:RISE?

This query returns whether the DECT signal falls within the power mask for the rise section of the burst.

Query Syntax

MEASure:RF:PTIMe:MASK:RISE?

Return Format

String - "PASS" or "FAIL"
MEASure:RF:PTIME:TRACe:FALL?
This query returns the trace data points for the falling edge measurements.

Query Syntax

MEASure:RF:PTIME:TRACe:FALL?

Return Format

417 data points in real numbers
MEASURE:RF:PTIME:TRACE:MID?

This query returns the trace data points for the mid section measurements.

Query Syntax

MEASURE:RF:PTIME:TRACE:MID?

Return Format

417 data points in real numbers
MEASure:RF:PTIME:TRACe:RISE?

This query returns the trace data points for the rising edge measurements.

Query Syntax

MEASure:RF:PTIME:TRACe:RISE?

Return Format

417 data points in real numbers
MEASure:RF:TIMing:JITTer?
This query returns the timing jitter value in fundamental units of seconds.

Query Syntax
MEASure:RF:TIMing:JITTer? <real number>

Return Format
Real
See Appendix B for additional related commands.
MEASURE:SYNC?
This query returns the measurement synchronisation display.

Query Syntax

OUTPUT 714;"MEASURE:SYNC?"

Return Format

String
OSClloscope Subsystem

This subsystem controls the operation of the oscilloscope.
OSCilloscope Syntax Diagram
**OSCilloscope:MARKer:NPEak**

This command moves the marker on the oscilloscope to the negative peak of the on screen trace.

**Command Syntax**

OSCilloscope:MARKer:NPEak

**Example Statements**

OUTPUT 714:"OSCilloscope:MARKer:NPEak"
**OSCilloscope:MARKeR:PPETak**

This command moves the marker on the oscilloscope to the positive peak of the on screen trace.

**Command Syntax**

`OSCilloscope:MARKeR:PPETak`

**Example Statements**

`OUTPUT 714;"OSCilloscope:MARKeR:PPETak"`
**OSCilloscope:MARKer:POSition**

This command sets the position of the marker on the oscilloscope trace.

**Command Syntax**

OSCilloscope:MARKer:POSition <increment_parameter>

or

OSCilloscope:MARKer:POSition <value> [<units_parameters> - (optional)]

Where <increment_parameter> = UP or DOWN

Where <value> = real number

Where <measurement_parameter> = DIV

**Example Statements**

OUTPUT 714;"OSCilloscope:MARKer:POSition UP"

OUTPUT 714;"OSCilloscope:MARKer:POSition 3.2 DIV"

**Query Syntax**

OSCilloscope:MARKer:POSition?

**Return Format**

Real

See Appendix C for additional related commands.
**OSCilloscope:SCALE:TIME**

This command sets the oscilloscope timebase.

**Command Syntax**

OSCilloscope:SCALE:TIME <string>

Where <string> = '1 US' | '2 US' | '5 US' | '10 US' | '20 US' | '50 US' | '100 US' | '200 US' | '500 US' | '1 MS' | '2 MS' | '5 MS' | '10 MS' | '20 MS' | '50 MS' | '100 MS' | '200 MS'

**Example Statements**

OUTPUT 714;"OSCilloscope:SCALE:TIME '500 US'"

**Query Syntax**

OSCilloscope:SCALE:TIME?

**Return Format**

String
**OSCilloscope:SCALE:VERTical:OFFSet**

This command sets the amplitude offset for the oscilloscope trace.

**Command Syntax**

OSCilloscope:SCALE:VERTical:OFFSet <value> [<measurement_parameters>] - optional
Where <value> = real number
Where <measurement_parameter> = DIV

**Example Statements**

OUTPUT 714; "OSCilloscope:SCALE:VERTical:OFFSet 3.2 DIV"

**Query Syntax**

OSCilloscope:SCALE:VERTical:OFFSet?

**Return Format**

Real
See Appendix C for additional related commands.
**OSCilloscope:SCALe:VERTical:VOLTs**

This command sets the amplitude scale of the oscilloscope.

**Command Syntax**

```
OSCilloscope:SCALe:VERTical:VOLTs <string>
```

Where `<string>` = '1 MV' | '2 MV' | '5 MV' | '10 MV' | '20 MV' | '50 MV' | '100 MV' | '200 MV' | '500 MV' | '1 V' | '2 V' | '5 V' | '20 V'.

**Example Statements**

```
OUTPUT 714;"OSCilloscope:SCALe:VERTical:VOLTs '500 MV'"
```

**Query Syntax**

```
OSCilloscope:SCALe:VERTical:VOLTs?
```

**Return Format**

String
**OSCilloscope:**TRIGger:LEVel

This command sets the trigger level of the oscilloscope.

**Command Syntax**

OSCilloscope:TRIGger:LEVel <value> [ <measurement(parameters)> ] - optional
Where <value> = real number
Where <measurement(parameter)> = DIV

**Example Statements**

OUTPUT 714;"OSCilloscope:TRIGger:LEVel 1.2 DIV"

**Query Syntax**

OSCilloscope:TRIGger:LEVel?

**Return Format**

Real
OSCilloscope:TRIGger:LEVel:INCRement

This command controls the increment of the oscilloscope's trigger level input. This can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

Command Syntax

OSCilloscope:TRIGger:LEVel:INCRement <increment_parameters> | <value>
  Where <increment_parameters> = UP | DOWN
  Where <value> = real number

Example Statements

OUTPUT 714;"OSCilloscope:TRIGger:LEVel:INCRement UP"

Query Syntax

OSCilloscope:TRIGger:LEVel:INCRement?

Return Format

Real

See Appendix D for additional related commands.
**OSCilloscope:TRIGger:MODE**

This command sets the oscilloscope to continuous or single shot triggering.

**Command Syntax**

```
OSCilloscope:TRIGger:MODE <string>
Where <string> = 'Cont' | 'Sngl'
```

**Example Statements**

```
OUTPUT 714;"OSCilloscope:TRIGger:MODE 'Cont'"
```

**Query Syntax**

```
OSCilloscope:TRIGger:MODE?
```

**Return Format**

String
**OSCilloscope:TRIGger:PRETrigger**

This command sets the oscilloscope pretrigger value.

**Command Syntax**

OSCilloscope:TRIGger:PRETrigger \(<value>\\)
Where \(<value> = DIV\)

**Example Statements**

```
OUTPUT 714;"OSCilloscope:TRIGger:PRETrigger 3.2 DIV"
```

**Query Syntax**

OSCilloscope:TRIGger:PRETrigger?

**Return Format**

Real
**Oscilloscope:TRIGger:PRETrigger:INCREMENT**

This command controls the increment of the oscilloscope's pretrigger level input. This can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

**Command Syntax**

```
Oscilloscope:TRIGger:PRETrigger:INCREMENT <increment_parameter> | <value>
Where <increment_parameter> = UP | DOWN
Where <value> = real number
```

**Example Statements**

```
OUTPUT 714:"Oscilloscope:TRIGger:PRETrigger:INCREMENT 3"
```

**Query Syntax**

```
Oscilloscope:TRIGger:PRETrigger:INCREMENT?
```

**Return Format**

Real

See Appendix D for additional related commands.
**OSCilloscope:TRIGger:RESet**

This command retriggers an oscilloscope measurement when on single trigger mode.

**Command Syntax**

```
OSCilloscope:TRIGger:RESet
```

**Example Statements**

```
OUTPUT 714;"OSCilloscope:TRIGger:RESet"
```
**OSCilloscope:TRIGger:SENSe**

This command sets the triggering edge for making oscilloscope measurements.

**Command Syntax**

```
OSCilloscope:TRIGger:SENSe <string>
   Where <string> = 'Pos' | 'Neg'
```

**Example Statements**

```
OUTPUT 714;'OSCilloscope:TRIGger:SENSe 'Pos'"
```

**Query Syntax**

```
OSCilloscope:TRIGger:SENSe?
```

**Return Format**

String
OSCilloscope:TRIGger:SOURce

This command sets the oscilloscope trigger source to internal or external.

Command Syntax

OSCilloscope:TRIGger:SOURce <string>
Where <string> = 'Ext' | 'Int'

Example Statements

OUTPUT 714;"OSCilloscope:TRIGger:SOURce 'Int'"

Query Syntax

OSCilloscope:TRIGger:SOURce?

Return Format

String
**OSCilloscope:TRIgger:TYPE**

This command sets the oscilloscope to automatic level triggering or manual level triggering using the trigger level control.

**Command Syntax**

OSCilloscope:TRIgger:TYPE <string>

Where <string> = 'Norm' | 'Auto'

**Example Statements**

OUTPUT 714;"OSCilloscope:TRIgger:TYPE 'Norm'"

**Query Syntax**

OSCilloscope:TRIgger:TYPE?

**Return Format**

String
PTIME Subsystem

This subsystem controls the power-time configuration and measurement on the HP 8923B.
PTIMe:MARKer:POSition:FALL

This command sets the position of the marker on the power time fall screen in screen divisions.

Command Syntax

PTIMe:MARKer:POSition:FALL <real number>
Where <real number> = 0.75 through 7.25

Example Statements

OUTPUT 714;"PTIMe:MARKer:POSition:FALL 4.25"

Query Syntax

PTIMe:MARKer:POSition:FALL?

Return Format

Real

See Appendix C for additional related commands.
**PTIMe:MARKer:POSition:MID**

This command sets the position of the marker on the power time middle screen in screen divisions.

**Command Syntax**

```
PTIMe:MARKer:POSition:MID <value>
Where <value> = 0.4 through 16
```

**Example Statements**

```
OUTPUT 714:"PTIMe:MARKer:POSition:MID 6.5"
```

**Query Syntax**

```
PTIMe:MARKer:POSition:MID?
```

**Return Format**

Real

See Appendix C for additional related commands.
PTIMe:MARKer:POSitioNE:RISE
This command sets the position of the marker on the power time middle screen in screen divisions.

Command Syntax
PTIMe:MARKer:POSitioNE:RISE <value>
Where <value> = 0.75 through 7.25

Example Statements
OUTPUT 714:"PTIMe:MARKer:POSitioNE:RISE 1.5"

Query Syntax
PTIMe:MARKer:POSitioNE:RISE?

Return Format
Real
See Appendix C for additional related commands.
PTIMe:MASK
This command activates/deactivates the power time on all of the power versus time measurement screens.

Command Syntax
PTIMe:MASK <command_parameter>
Where <command_parameter> = 'On' | 'Off'

Example Statements
OUTPUT 714:"PTIMe:MASK 'On'"

Query Syntax
PTIMe:MASK?

Return Format
String
PTIME:ZField
This command allows access to control the Z Field.

Command Syntax

PTIME:ZField <string>
Where <string> = 'Yes' | 'No'

Example Statements

OUTPUT 714;'PTIME:ZField 'Yes'"

Query Syntax

PTIME:ZField?

Return Format

String
REGister Subsystem

The Register subsystem controls the instrument state memory.

Figure 17-1. REGister Syntax Diagram
[REGister:]CLEar

This command clears an instrument state register.

Command Syntax

CLeaR <integer> or <string>
or
REGister:CLeaR <integer> or <string>
Where <integer> and <string> are customer definable identifiers.

Example Statements

OUTPUT 714; "REGister:CLeaR 53"
or
OUTPUT 714; "CLeaR 53"
[REGister:]CLEar:ALL
This command clears all the instrument state registers.

Command Syntax

CLEar:ALL
or
REGister:CLEar:ALL

Example Statements

OUTPUT 714;"REGister:CLEar:ALL"
or
OUTPUT 714;"CLEar:ALL"
[REGister:]RECall

This command recalls an instrument state from the memory. An instrument state stored using the common command *SAV may be recalled using *RCL or [REGister:]RECall. The *SAV uses an integer which can be recalled using the same integer as the identifier. Using [REGister:]SAVE allows you to use alphanumeric strings instead of integers. [REGister:]RECall can use alphanumeric strings or integers.

Command Syntax

RECall <integer> or <string>
or
REGister:RECall <integer> or <string>
Where <integer> and <string> are customer definable identifiers.

Example Statements

OUTPUT 714;"REGister:RECall 13"
or
OUTPUT 714;"RECall 13"
[REGister:]SAVE

This command save an instrument state to the memory.

The "RCL command may be used to return the instrument to the state at which the instrument was saved using [REGister:]SAVE <integer>. The "RCL must use the same integer to return to this state. If [REGister:]SAVE used a string to store an instrument state, only [REGister:]RECall can be used to restore the instrument to its initial state. The total number of registers which can be saved is limited by the number of settings which differ from their preset condition and the memory available.

Command Syntax

SAVE <integer> or <string>

or

REGister:SAVE <integer> or <string>

Where <integer> and <string> are customer definable identifiers.

Example Statements

OUTPUT 714;"REGister:SAVE 29"

or

OUTPUT 714;"SAVE 29"
RFANalyzer Subsystem

The RFANalyzer subsystem controls the input frequency and amplitude settings of the HP 8923B.

Figure 18-1. RFANalyzer Syntax Diagram
**RFA
dalyzer:AMPLitude**

This command sets the input amplitude expected from the EUT at the RF In/Out connector of the HP 8923B.

**Command Syntax**

RFA
dalyzer:AMPLitude <value> [ <amplitude_parameters> (optional) ]
Where <value> = real number
Where <amplitude_parameters> = 'DBM' | 'DBMW' | 'DBUV' | 'V' | 'W'.

**Example Statements**

OUTPUT 714;"RFA
dalyzer:AMPLitude 3.2"
OUTPUT 714;"RFA
dalyzer:AMPLitude 24 DBM"

**Query Syntax**

RFA
dalyzer:AMPLitude?

**Return Format**

Real

See Appendix C for additional related commands.
**RFANalyzer:AMPLitude:CORRection:LOSS**

This command allows you to set the known power out of the UUT and the known loss in the cabling/coupling, rather than having to add/subtract the two and then alter the instrument settings.

**Command Syntax**

RFANalyzer:AMPLitude:CORRection:LOSS <real number>
Where <real number> is in the range 0.0 to 40.0.

**Example Statements**

OUTPUT 714;"RFANalyzer:AMPLitude:CORRection:LOSS 3.142"

**Query Syntax**

RFANalyzer:AMPLitude:CORRection:LOSS?

**Return Format**

Real

See Appendix C for additional related commands.
**RFANalyzer:CARRier**

This command sets the DECT RF carrier number to which the HP 8923B is tuned for making measurements.

**Note**

This command only has effect when the RFANalyzer:COUPling is set to 'Manual' and the RFANalyzer:COUPling:INPut is set to 'Carrier No'.

**Command Syntax**

```
RFANalyzer:CARRier <value>
Where <value> = 0 through 9
```

**Example Statements**

```
OUTPUT 714; "RFANalyzer:CARRier 3"
```

**Query Syntax**

```
RFANalyzer:CARRier?
```

**Return Format**

Integer
RFANalyzer:CARRier:INCRement

This command controls the increment of the RF analyzer's carrier frequency. This can be used to:

1. Set the increment magnitude, using a numeric parameter.
2. Apply the increment, using either the UP or DOWN parameter.

Note: This command only has effect when the RFANalyzer:COUPling is set to 'Manual' and the RFANalyzer:COUPling:INPut is set to 'Carrier No'.

Command Syntax

RFANalyzer:CARRier:INCRement <value> | <increment_parameter>
Where <value> = integer
Where <increment_parameter> = UP | DOWN

Example Statements

OUTPUT 714;"RFANalyzer:CARRier:INCRement 3"
OUTPUT 714;"RFANalyzer:CARRier:INCRement DOWN"

Query Syntax

RFANalyzer:CARRier:INCRement?

Return Format

Integer
RFA\text{alyzer}:COUP\text{ling}

This command sets the input frequency of the HP 8923B to be coupled to the traffic bearer, the dummy bearer or set independently.

Command Syntax

RFANalyzer:COUPling <string>
Where <string> = 'Manual' | 'Traffic' | 'Dummy'

Example Statements

OUTPUT 714;"RFANalyzer:COUPling 'Traffic'"

Query Syntax

RFANalyzer:COUPling?

Return Format

String
RFANalyzer:COUPling:INPut

This command sets the form of frequency input of the HP 8923B. This can either be in terms of a frequency value or a DECT RF carrier number.

Note
This command only has effect when RFANalyzer:COUPling is set to 'Manual'.

Command Syntax

RFANalyzer:COUPling:INPut <string>
Where <string> = 'Freq' | 'Carrier No'

Example Statements

OUTPUT 714:"RFANalyzer:COUPling:INPut 'Freq'"

Query Syntax

RFANalyzer:COUPling:INPut?

Return Format

String
RFANalyzer:FREQuency
This command sets the RF frequency to which the HP 8923B is tuned.

**Note**
This command only has effect when RFANalyzer:COUPling is set to 'Manual' and RFANalyzer:COUPling:INPut is set to 'Freq'

**Command Syntax**
RFANalyzer:FREQuency <value> [ units _ parameter > (optional)]
Where <value> = real number (final value should read between 1880 MHz and 1990 MHz).
Where <units _ parameter > = 'HZ' | 'KHZ' | 'MHZ' | 'GHZ'.

**Example Statements**
OUTPUT 714;"RFANalyzer:FREQuency 1897.344MHZ"

**Query Syntax**
RFANalyzer:FREQuency?

**Return Format**
Real

See Appendix C for additional related commands.
RFANalyzer:PMETer:ZERO

This command zero's the power meter in the presence of no input signal. All power should be removed from the input connector prior to sending this command.

This is the order of the actions:
- Remove the input power.
- Set the RFAN:AMPL to its lowest value.
- Execute PMET:ZERO
- Replace the input power.
- Reset the RFAN:AMPL to its expected value.

Command Syntax

RFANalyzer:PMETer:ZERO

Example Statements

OUTPUT 714;"RFANalyzer:PMETer:ZERO"
RFGenerator Subsystem

The RFGenerator subsystem controls the output amplitude of the HP 8923B.

Figure 19-1. RFGenerator Syntax Diagram
**RFCGenerator:AMPLitude**

This command sets the output amplitude of the IIP 8923B.

**Command Syntax**

RFCGenerator:AMPLitude <value> [ <units_parameter> (optional) ]
Where <value> = real number
Where <units_parameter> = 'DBM' | 'DBMW' | 'DBUV' | 'MV' | 'MW' | 'UV' | 'V' | 'W'.

**Example Statements**

OUTPUT 714:"RFCGenerator:AMPLitude -83 DBM"

**Query Syntax**

RFCGenerator:AMPLitude?

**Return Format**

Real

See Appendix C for additional related commands.
RFGenerator:ATTenuator
This command sets the output attenuator of the HP 8923B.

**Note**  
This command will force the RFGenerator:ATTenuator:AUTO to 'OFF' and will restrict the range of RFGenerator:AMPLitude.

---

**Command Syntax**

```
RFGenerator:ATTenuator <string>
Where <string> = '100 dB' | '90 dB' | '80 dB' | '70 dB' |
'60 dB' | '50 dB' | '40 dB' | '30 dB' | '20 dB' | '10 dB' | '0 dB'.
```

**Example Statements**

```
OUTPUT 714;"RFGenerator:ATTenuator '80 dB''
```

**Query Syntax**

```
RFGenerator:ATTenuator?
```

**Return Format**

String
RFGenerator:ATTenuator:AUTO

This command sets the output attenuator of the HP 8923B to automatic or manual selection. When set to "Off" the output level range is restricted and attenuator is selected manually. When set to "On" the instrument will set the appropriate attenuator for the output level.

Command Syntax

RFGenerator:ATTenuator:AUTO <string>
Where <string> = 'On' | 'Off'

Example Statements

OUTPUT 714; "RFGenerator:ATTenuator:AUTO 'Off'"

Query Syntax

RFGenerator:ATTenuator:AUTO?

Return Format

String
RFGenerator: CW:CARRIER
This command sets the output frequency in terms of DECT RF carrier number.

**Note**
This command only has effect when RFGenerator:MODE is set to 'CW'

---

**Command Syntax**

RFGenerator: CW:CARRIER <value>
Where <value> = 0 through 9

**Example Statements**

OUTPUT 714:"RFGenerator: CW:CARRIER 3"

**Query Syntax**

RFGenerator: CW:CARRIER?

**Return Format**

Integer
RFGenerator: CW: PATTERN
This command sets the pattern which is modulated onto the CW carrier.

Note
This command only has effect when RFGenerator: MODE is set to 'CW'

Command Syntax
RFGenerator: CW: PATTERN <string>
Where <string> = '0000 ... 0000' | '1111 ... 1111' | '0101 ... 0101' |
'00001111 ... '.

Example Statements
OUTPUT 714; "RFGenerator: CW: PATTERN '1111 ... 1111'"

Query Syntax
RFGenerator: CW: PATTERN?

Return Format
String
RFGenerator:MODE

This command sets the HP 8923B output to be a normal DECT pulse modulated signal or a modulated CW signal.

Command Syntax

RFGenerator:MODE <string>
    Where <string> = 'Normal' | 'CW'

Example Statements

OUTPUT 714;"RFGenerator:MODE 'CW'"

Query Syntax

RFGenerator:MODE?

Return Format

String
STATus Subsystem

The STATus subsystem provides access to information about the instrument’s status, transient events, and errors.

Figure 20-1. STATus Syntax Diagram
STATUs:CALibration

The STATUs:CALibration series of commands allows you to poll information on some of the HP 8923B's internal functions. The Sampler, Counter and Voltmeter can all be tested and queried using this command.

STATUs:CALibration has several commands associated with it. These are:

- STATus:CALibration:CONDition?
- STATus:CALibration:ENABLE[?]
- STATus:CALibration[:EVENT]?
- STATus:CALibration:NTRansition[?]
- STATus:CALibration:PTRansition[?]

Refer to “Additional Commands in the STATus Subsystem” at the end of this chapter for additional related commands.
STATus:COMMunicate

The STATus:COMMunicate series of commands allows you to query the status of the HP 8923B. This includes any communication failures as well as the current transmit/receive state.

STATus:COMMunicate has several commands associated with it. These are:

- STATus:COMMunicate:CONDition?
- STATus:COMMunicate:ENABLE[?]
- STATus:COMMunicate[:EVENt]?
- STATus:COMMunicate:NTransition[?]
- STATus:COMMunicate:PTRansition[?]

Refer to “Additional Commands in the STATus Subsystem” at the end of this chapter for additional related commands.
STATus: HARDware1

The STATus: HARDware1 series of commands allows you read the state of the power-up self test.

STATus: HARDware1 has several commands associated with it. These are:

- STATus: HARDware1:CONDition?
- STATus: HARDware1:ENABle[?]
- STATus: HARDware1[:EVENT]?
- STATus: HARDware1:NTRansition[?]
- STATus: HARDware1:PTRansition[?]

Refer to “Additional Commands in the STATus Subsystem” at the end of this chapter for additional related commands.
STATus:HARDware2

The STATus:HARDware2 series of commands allows you read the state of the pattern and trigger selection and the analyzer coupling.

STATus:HARDware2 has several commands associated with it. These are:

- STATus:HARDware2:CONDition?
- STATus:HARDware2:ENABle[?]
- STATus:HARDware2:[EVENT]?
- STATus:HARDware2:NTRansition[?]
- STATus:HARDware2:PTRansition[?]

Refer to "Additional Commands in the STATus Subsystem" at the end of this chapter for additional related commands.
STATus:OPERation

The STATus:OPERation series of commands allows you to describe the operation of the HP 8923B.

STATus:OPERation has several commands associated with it. These are:

- STATus:OPERation:CONDition?
- STATus:OPERation:ENABLE[?]
- STATus:OPERation:[EVENT]?
- STATus:OPERation:NTRansition[?]
- STATus:OPERation:PTransition[?]

Refer to “Additional Commands in the STATus Subsystem” at the end of this chapter for additional related commands.
STATus:PRESet
The STATus:PRESet command sets the status of the IIP 8923B to its default condition.

Command Syntax

STATus:PRESet
STATus:QUEStionable

The STATus:QUEStionable series of commands allows you to read the status of the instrument calibration.

Status:QUEStionable has several commands associated with it. These are:

- STATus:QUEStionable:CONDition?
- STATus:QUEStionable:ENABLE[?]
- STATus:QUEStionable[:EVENT]?
- STATus:QUEStionable:NTRansition[?]
- STATus:QUEStionable:PTRansition[?]

Refer to “Additional Commands in the STATus Subsystem” at the end of this chapter for additional related commands.
Additional Commands in the STATus Subsystem

These subcommands refer to the following commands in the STATus Subsystem:

- STATus:CALibration
- STATus:COMMunicate
- STATus:HARDware1
- STATus:HARDware2
- STATus:OPERation
- STATus:QUESTionable

Refer to Figure 20-2 for diagrammatic information on the way the subcommands are implemented in the subsystem.

:CONDition?

The condition register indicates the current/instantaneous condition of the register.

Query Syntax - <command>:CONDition?
Return Format - Integer

:ENABLE

The enable register selects which event bits in the corresponding event register will cause a TRUE summary register when set.

Command Syntax - <command>:ENABLE <integer>
Query Syntax - <command>:ENABLE?
Return Format - Integer

[:EVENT]?

The event register captures changes in conditions. Each event bit in an event register corresponds to the condition bit in the condition register.

Query Syntax - <command>[:EVENT]?

:NTRansition

The negative transition filter indicates that the change captured by the event register corresponding bit will be TRUE(1) to FALSE(0) condition.

Command Syntax - <command>:NTRansition <integer>
Query Syntax - <command>:NTRansition?
Return Format - Integer

:PTRansition

The negative transition filter indicates that the change captured by the event register corresponding bit will be FALSE(0) to TRUE(1) condition.

Command Syntax - <command>:PTRansition <integer>
Query Syntax - <command>:PTRansition?
Return Format - Integer
Figure 20-2. Additional Commands Syntax Diagram
The SYSTem subsystem provides access to error information.

Figure 21-1. SYSTem Syntax Diagram
SYSTem[:ERRor]?

This command queries the HP 8923B error queue. The error queue is a FIFO, when there are no errors or all errors have been read it will return 0, "No Error"

Query Syntax

SYSTem[:ERRor]?

Return Format

Integer
TRIGger Subsystem

This subsystem controls the HP 8923B's triggering.
TRIGger[:IMMediate]
This command instructs the instrument to trigger.

Command Syntax
   TRIGger
or
   TRIGger:IMMediate

Example Statements
   OUTPUT 714;"TRIGger"
or
   OUTPUT 714;"TRIGger:IMMediate"
TRIGger:ABORt
This command instructs the instrument to abort its triggering.

Command Syntax
TRIGger:ABORt

Example Statements
OUTPUT 714;"TRIGger:ABORt"
TRIGger:BETest
This command sets the BER measurement to run or stop.

Command Syntax
   TRIGger:BETest <string>
   Where <string> = 'Run' | 'Stop'

Example Statements
   OUTPUT 714;"TRIGger:BETest 'Run'"

Query Syntax
   TRIGger:BETest?

Return Format
   String
TRIGger:BETest:MODE
This command sets the BER measurement to run continuously or singly.

Command Syntax
TRIGger:BETest:MODE <string>
Where <string> = 'Sngl' | 'Cont'

Example Statements
OUTPUT 714;"TRIGger:BETest:MODE 'Cont'"

Query Syntax
TRIGger:BETest:MODE?

Return Format
String
TRIGger:DE Lay

This command sets the trigger delay.

Note  This command only has effect when the trigger source is external.

Command Syntax

TRIGger:DEL ay <value> [ <units_parameter> (optional) ]
Where <value> = a real number
Where <units_parameter> = 'US' | 'MS' | 'S' | 'T'

Example Statements

OUTPUT 714; "TRIGger:DELay 4 MS"

Query Syntax

TRIGger:DE Lay?

Return Format

Real

See Appendix B for additional related commands.
TRIGger:MODE:RETRigger

This command sets the triggering for measurements to be continuous/repetitive or single. Single triggering of measurements is only available in REMOTE mode. When the HP 8923B is returned to manual control (LOCAL) the triggering of measurements is continuous.

Command Syntax

TRIGger:MODE:RETRigger <trigger_parameters>
Where <trigger_parameters> = REPetitive | SINGle

Example Statements

OUTPUT 714;"TRIGger:MODE:RETRigger REPetitive"

Query Syntax

TRIGger:MODE:RETRigger?

Return Format

String
TRIGger:SOURce
This command sets the trigger source for measurements.

Command Syntax
TRIGger:SOURce <string>
Where <string> = 'Ext' | 'RF Rise' | 'Traffic' | 'Dummy'

Example Statements
OUTPUT 714; "TRIGger:SOURce 'RF Rise'

Query Syntax
TRIGger:SOURce?

Return Format
String
Example Programs

This chapter contains four example HP-IB programs. All of the example programs are written in HP BASIC.

Example Program 1

This example program demonstrates the following:

- Call setup for a portable part (or fixed part with a simple modification).
- Querying Frequency, Power, and BER measurements.
- Using Serial Polling to determine the existence of a call.

```
10  COM /Addr/ @Hp8923,INTEGER Status_byte,Comm_event,Errno,Err_mess$[200]
20  CLEAR SCREEN
30  DIM Rise$[7]
40  DIM Mid$[7]
50  DIM Fall$[7]
60  DIM Mask$[25]
70  DIM Part_to_test$[11]
80  Part_to_test$=""'portable'"" ! set Part_to_test$="'fixed'" to test fixed part
90  !
100 ASSIGN @Hp8923 TO 714
110 !
120 ! Setup RF output and input levels
130 OUTPUT @Hp8923:"RFG:AMPL -10" ! RFGenerators default units are DBM
140 OUTPUT @Hp8923:"RFA:AMPL 24" ! RFAnalyzer defaults units are DBM
150 !
160 ! Call setup parameters
170 !
180 OUTPUT @Hp8923:"disp call"
190 !
200 ! Part to be tested
210 OUTPUT @Hp8923:"DECT:EUT "&Part_to_test$
220 !
230 SELECT Part_to_test$
240 CASE "'portable'"
250 !
260 ! Dummy bearer physical channel
270 OUTPUT @Hp8923:"DECT:PP:DUMMY:CARRIER 0" ! RANGE 0 to 9
280 OUTPUT @Hp8923:"DECT:PP:DUMMY:SLOT 0" ! RANGE 0 to 11
290 !
300 ! Traffic bearer physical channel
310 OUTPUT @Hp8923:"DECT:PP:TRAFFIC:CARRIER 0"! RANGE 0 to 9
320 OUTPUT @Hp8923:"DECT:PP:TRAFFIC:SLOT 2" ! RANGE 0 to 11
330 !
```
340 ! PARI
350 OUTPUT @Hp8923;"DECT:PARI '000049D3A'" ! string of 9 hex characters
360 !
370 ! START TRANSMITTING DUMMY BEARER
380 OUTPUT @Hp8923;"DECT:PP:DUMMY:STATE ON"
390 !
400 BEEP
410 DISP "Press continue when PP is locked to dummy bearer."
420 PAUSE ! Wait until PP is in Test Standby mode and locked to dummy
430 DISP
440 GOSUB Set_for_call! prepare status for traffic bit
450 !
460 ! Start setting up a traffic bearer
470 Call_start=TIMEDATE
480 OUTPUT @Hp8923;"DECT:PP:TRAFFIC:CONNECT"
490 !
500 CASE "."'fixed'"
510 !
520 ! Traffic bearer physical channel
530 ! Dummy bearer fields are read only when testing fixed parts
540 ! since the dummy bearer is defined by the fixed part itself.
550 OUTPUT @Hp8923;"DECT:FP:TRAFFIC:CARRIER 0" ! Range 0 to 9
560 OUTPUT @Hp8923;"DECT:FP:TRAFFIC:SLOT 2" ! Range 0 to 11
570 !
580 ! Set PMID only if FP requires it for access
590 OUTPUT @Hp8923;"DECT:PMID '00195'" ! string of 5 hex characters
600 !
610 GOSUB Set_for_lock ! prepare status for dummy bit
620 !
630 ! Start synchronising to the dummy bearer
640 OUTPUT @Hp8923;"DECT:PP:DUMMY:SYNC"
650 !
660 GOSUB Test_for_locked
670 !
680 GOSUB Set_for_call! prepare status for traffic bit
690 !
700 ! Start setting up the traffic bearer
710 OUTPUT @Hp8923;"DECT:FP:TRAFFIC:CONNECT"
720 !
730 END SELECT
740 !
750 GOSUB Test_for_call
760 !
770 Tstart=TIMEDATE
780 !
790 ! Setup triggering from the traffic bearer.
800 OUTPUT @Hp8923;"TRIG:SOURCE 'traffic'"
810 !
820 ! Setup single triggering of measurements
830 OUTPUT @Hp8923;"TRIG:MODE:RETR SING"
840 !
850 ! Measure frequency accuracy
860 OUTPUT @Hp8923;"DISP FREQ::MEAS:MODE 'facc';:TRIG:IMM"
870 OUTPUT @Hp8923;"MEAS:FREQ:ACC?"
880 ENTER @Hp8923;Freq

23-2 Example Programs
PRINT "Freq Acc:";TAB(25);Freq/1000;"kHz"

! Measure frequency deviation
OUTPUT @Hp8923;"MEAS:OUTPUT OFF;TRIG:IMM"
OUTPUT @Hp8923;"MEAS:RF:FREQ:FREQ:ZERO:BMAX?;BMIN?;BAV?"
ENTER @Hp8923;Max0;Min0;Avg0
OUTPUT @Hp8923;"MEAS:RF:FREQ:FREQ:ONE:BMAX?;BMIN?;BAV?"
ENTER @Hp8923;Max1;Min1;Avg1
!
! Measure frequency drift; must have pattern 'fdev2_fs' set
OUTPUT @Hp8923;"MEAS:RF:FREQ:DRIFT?"
ENTER @Hp8923;Drift
PRINT "Freq Drift:";TAB(25);Drift/1000;"kHz"
PRINT "Freq Dev(0) Max:";TAB(25);Max0/1000;"kHz"
PRINT "Freq Dev(0) Min:";TAB(25);Min0/1000;"kHz"
PRINT "Freq Dev(0) Avg:";TAB(25);Avg0/1000;"kHz"
PRINT "Freq Dev(1) Max:";TAB(25);Max1/1000;"kHz"
PRINT "Freq Dev(1) Min:";TAB(25);Min1/1000;"kHz"
PRINT "Freq Dev(1) Avg:";TAB(25);Avg1/1000;"kHz"
!
! Measure NTP and the power time rise, mid, and fall
! pass/fall flags
OUTPUT @Hp8923;"DISP NTP"
OUTPUT @Hp8923;"MEAS:RF:NTP?;PTIM:MASK:RISE?;MID?;FALL?"
ENTER @Hp8923;Ntp;Rise$;Mid$;Fall$
PRINT "NTP:";TAB(25);Ntp;"dBm"
Mask$="FAILED"
IF Rise$[2;4]="FAIL" THEN Mask$=Mask$&" Rise"
IF Mid$[2;4]="FAIL" THEN Mask$=Mask$&" Middle"
IF Fall$[2;4]="FAIL" THEN Mask$=Mask$&" Fall"
IF Mask$="FAILED" THEN Mask$="PASSED"
PRINT "Power Time Template:";TAB(26);Mask$
!
! Measure Bit Error Ratio and Word Error Ratio.
OUTPUT @Hp8923;"DISP BET;:RF:AMPL -20;:TRIG:BET 'run'"
OUTPUT @Hp8923;"MEAS:BIT:BERR:RATIO?;:MEAS:BIT:WERR:RATIO?"
ENTER @Hp8923;Ber;War
OUTPUT @Hp8923;"RF:AMPL -10" ! reset to high level
PRINT "Bit Error Ratio:";TAB(25);Ber/10000;"%"
PRINT "Frame Error Ratio:";TAB(25);War/10000;"%"
PRINT "Elapsed Time:";TAB(25);TIME;"s"
!
Release the call on the traffic bearer.
OUTPUT @Hp8923;"DECT:TRAFFIC:RELEASE"
!
STOP
!
1440  
1450  Set_for_lock:  !
1460  OUTPUT @Hp8923:"*CLS" !Clear all status bits
1470  OUTPUT @Hp8923:"*SRE 4" ! set status byte interrupt mask for
1480      ! the communicate register.
1490  !
1500  ! setup communicate register enable and bit transitions
1510  OUTPUT @Hp8923;"STATUS:COMM:ENABLE 32"
1520  OUTPUT @Hp8923;"STATUS:COMM:PTR 32"
1530  RETURN
1540  !
1550  !
1560  Test_for_locked:  !
1570  Sync_start=TIMEDATE
1580  LOOP
1590  Stat_byte=SPOLL(@Hp8923)
1600  EXIT IF BIT(Stat_byte,2)
1610  END LOOP
1620  PRINT "Sync time =";TIMEDATE-Sync_start
1630  RETURN
1640  !
1650  !
1660  Set_for_call:  !
1670  OUTPUT @Hp8923:"*CLS" !Clear all status bits
1680  OUTPUT @Hp8923:"*SRE 4" ! set status byte interrupt mask for
1690      ! the communicate register.
1700  !
1710  ! setup communicate register enable and bit transitions
1720  OUTPUT @Hp8923;"STATUS:COMM:ENABLE 64"
1730  OUTPUT @Hp8923;"STATUS:COMM:PTR 64"
1740  RETURN
1750  !
1760  !
1770  Test_for_call:  !
1780  LOOP
1790  WAIT .01
1800  Stat_byte=SPOLL(@Hp8923)
1810  EXIT IF BIT(Stat_byte,2)
1820  END LOOP
1830  PRINT "Call time =";TIMEDATE-Call_start
1840  RETURN
1850  !
1860  !
1870  END
Example Program 2

This example program demonstrates call setup and release for a portable part using the instruments ability to generate a service request(SRQ) based upon bits in the COMMunicate register.

BIT 6 of the COMMunicate register will be 1 when the traffic bearer is active/connected, it will be 0 when the traffic bearer is inactive/released.

10   Comm_eve=0 ! clear the variable that holds
20    ! the communicate register value
30    
40   Status=0 ! clear the variable that holds
50    ! the status byte value
60    
70   Bus=7 ! Select code of HP-IB interface
80    
90   No_print=0 ! Set to 0 will print out status
100    ! byte and communicate register
110    ! values.
120    ! Set to 1 will not print out.
130    
140  CLEAR SCREEN
150  PRINT "Connecting..."
160    
170  ASSIGN @Hp8923 TO 714 ! Instrument HP-IB address
180    
190  OUTPUT @Hp8923;"*RST"
200    
210    
220  OUTPUT @Hp8923;"Disp call"
230  OUTPUT @Hp8923;"DECT:PAR1 '000049D3A'"
240    
250  !Use instrument defaults for traffic and dummy bearer
260    
270  OUTPUT @Hp8923;"DECT:Dummy:STATE ON"
280    
290  ! clear the status registers then set the instrument to
300  ! generate a Service ReQuest(SRQ) when the traffic bearer
310  ! is active/connected.
320    
330  OUTPUT @Hp8923;"*CLS" ! clear status registers.
340    
350  OUTPUT @Hp8923;"*SRE 4" ! generate an SRQ when communicate
360    ! register summary bit in the
370    ! status byte changes to 1
380    
390  OUTPUT @Hp8923;"status:comm:enable 64" ! enable the traffic_active bit
400    ! of the communicate register
410  OUTPUT @Hp8923;"status:comm:ptr 64" ! for positive transitions,
420  OUTPUT @Hp8923;"status:comm:intr 0" ! 0 to 1. is traffic active
430    
440    
450  ON INTR Bus,15 GOSUB Connection_isr ! setup computer to call interrupt
460    ! routine when an interrupt occurs
470 ! on the HP-IB
480 !
490 ENABLE INTR Bus;2
500 ! enable the computer to accept
510 ! interrupts.
520 !
530 BEEP 400,.2
540 DISP "Press continue FP is locked and in test standby"
550 PAUSE
560 DISP "waiting on interrupt..."
570 !
580 Tstart=TIMEDATE
590 OUTPUT @Hp8923:"DECT:TRAFFIC:CONNECT"
600 !
610 LOOP ! Wait in this loop until the
620 EXIT IF BIT(Comm_eve,6) ! interrupt occurs
630 END LOOP
640 !
650 !
660 BEEP 4000,.3
670 PRINT " Connection time: ";DROUND(Tstop-Tstart,3);"S"
680 !
690 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
700 !* Measurement code could go here *
710 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
720 !
730 DISP "Press Continue to release the call"
740 PAUSE
750 DISP "Releasing call . ."
760 PRINT
770 PRINT "Releasing"
780 !
790 Comm_eve=0
800 Status=0
810 !
820 ! clear the status registers then set the instrument to
830 ! generate a Service ReQuest(SRQ) when the traffic bearer
840 ! is released.
850 !
860 OUTPUT @Hp8923;"*CLS"
870 ! clear status registers.
880 !
890 OUTPUT @Hp8923;"*SRE 4"
900 ! generate an SRQ when communicate
910 ! register summary bit in the
920 ! status byte changes to 1
930 !
940 OUTPUT @Hp8923;"status:comm:enable 64" ! enable the traffic_active bit
950 ! of the communicate register
960 OUTPUT @Hp8923;"status:comm:ntr 64" ! for only negative transitions,
970 OUTPUT @Hp8923;"status:comm:ptr 0" ! 1 to 0. ie traffic inactive
980 !
990 ON INTR Bus,15 GOSUB Connection_isr ! setup computer to call interrupt
1000 ! routine when an interrupt occurs
1010 ! on the HP-IB
1020 ENABLE INTR Bus:2 ! enable the computer to accept
1030 ! interrupts.
1040 !
1050 !
1060 DISP "waiting on interrupt..."
1070 !
1080 Tstart=TIMEDATE
1090 OUTPUT @Hp8923:"DECT:TRAFFIC:RELEASE"
1100 !
1110 LOOP ! Wait in this loop until the
1120 EXIT IF BIT(Comm_eve,6) ! interrupt occurs
1130 END LOOP
1140 !
1150 BEEP 4000,.3
1160 PRINT " Release time: ";DROUND(Tstop-Tstart,3);"S"
1170 !
1180 DISP "Released and program stopped. "
1190 STOP
1200 !
1210 Connection_isr: !
1220 OFF INTR ! Disable interrupts
1230 IF No_print THEN GOSUB Read_comms
1240 IF No_print THEN RETURN
1250 PRINT " Subroutine Connection_isr"
1260 PRINT
1270 !
1280 GOSUB Read_status ! Read status bytes
1290 GOSUB Read_comms ! read the communicate register
1300 RETURN
1310 !
1320 Read_comms:!
1330 !
1340 OUTPUT @Hp8923;"status:comm:EVENT?" ! query communicate event register
1350 ENTER @Hp8923;Comm_eve
1360 !
1370 IF BIT(Comm_eve,6) THEN Tstop=TIMEDATE
1380 IF No_print THEN RETURN
1390 !
1400 DIM Comm_eve$[32]
1410 Comm_eve$=DVAL$(Comm_eve,2)
1420 PRINT " Communicate event:"; ! an easy to read register bit
1430 FOR X=1 TO 29 STEP 4 ! representation
1440 PRINT " &Comm_eve$[X:4];
1450 NEXT X
1460 PRINT "||"&VAL$(Comm_eve) ! append decimal value of register
1470 !
1480 OUTPUT @Hp8923;"status:comm:COND?" ! query the communicate condition
1490 ENTER @Hp8923;Comm_cond ! register.
1500 DIM Comm_cond$[32]
1510 Comm_cond$=DVAL$(Comm_cond,2)
1520 PRINT " Communicate cond: "; ! an easy to read register bit
1530 FOR X=1 TO 29 STEP 4 ! representation
1540 PRINT " &Comm_cond$[X:4];
1550 NEXT X
1560 PRINT "||"&VAL$(Comm_cond) ! append decimal value of register
1570    RETURN
1580  !
1590 Read_status:  !
1600    DIM Status$[32]
1610    OUTPUT @Hp8923;"*STB?"    ! read status byte
1620    ENTER @Hp8923;Status
1630  !
1640    Status$=DVAL$(Status,2)
1650    PRINT " Status byte:    ";Status$[25;4];" ";
1660    PRINT Status$[29;4]&""&VAL$(Status).
1670    RETURN
1680  !
1690    END
Example Program 3

This example programme demonstrates dummy bearer synchronisation, call setup and call release when the EUT is a fixed part. The programme uses the HP8923B's ability to generate a service request(SRQ) based upon bits in the COMMunicate register.

The programme assumes that the fixed part being tested is in test standby mode.

BIT 5 of the COMMunicate register will be 1 when the HP8923B has synchronised/locked to a dummy bearer, it will be 0 when the instrument cannot find a dummy bearer.

BIT 6 of the COMMunicate register will be 1 when the traffic bearer is active/connected, it will be 0 when the traffic bearer is inactive/released.

10 Comm_eve=0
20 ! clear the variable that holds ! the communicate register value
30 !
40 Status=0
50 ! clear the variable that holds ! the status byte value
60 !
70 Bus=7
80 ! Select code of HP-IB interface
90 No_print=0
100 ! Set to 0 will print out status ! byte and communicate register ! values.
110 ! Set to 1 will not print out.
120 !
130 CLEAR SCREEN
140 PRINT "Synchronising..."
150 !
160 ASSIGN @Hp8923 TO 714 ! Instrument HP-IB address
170 !
180 OUTPUT @Hp8923;"*RST"
190 !
200 WAIT 2
210 !
220 OUTPUT @Hp8923;"disp call" ! Show the call screen so that ! the call status can be seen
230 !
240 !
250 OUTPUT @Hp8923;"DECT:EUT 'Fixed'"
260 OUTPUT @Hp8923;"DECT:PMID '00195'"
270 !
280 !***********************************************************************
290 ! Synchronise to Fixed Part's Dummy Bearer !
300 !***********************************************************************
310 !
320 ! clear the status registers then set the instrument to ! generate a Service ReQuest(SRQ) when the dummy bearer ! is active/locked.
330 !
340 OUTPUT @Hp8923;"*CLS" ! clear status registers.
350 !
360 OUTPUT @Hp8923;"*SRE 4" ! generate an SRQ when communicate ! register summary bit in the ! status byte changes to 1
370 !
380 OUTPUT @Hp8923;"status:comm:enable 32" ! enable the dummy_active bit
390 !
430 OUTPUT @Hp8923;"status:comm:ptr 32" ! of the communicate register
440 OUTPUT @Hp8923;"status:comm:ntr 0" ! for positive transitions,
450 ! 0 to 1. i.e dummy active
460 
470 !
480 ON INTR Bus,15 GOSUB Connection_isr ! setup computer to call interrupt
490 ! routine when an interrupt occurs
500 ! on the HP-IB
510 !
520 Test_bit=5 ! bit number in communicate
530 ! register for dummy_bearer
540 
550 ENABLE INTR Bus;2 ! enable the computer to accept
560 ! interrupts.
570 
580 !
590 Tstart=TIME|DATE
600 OUTPUT @Hp8923;"DECT:Dummy:SYNC" ! instruct 8923 to start searching
610 ! for a dummy bearer.
620 !
630 LOOP ! Wait in this loop until the
640 EXIT IF BIT(Comm_eve,Test_bit) ! interrupt occurs
650 END LOOP
660 !
670 !
680 BEEP 4000,.3
690 PRINT " Synchronisation time: ";DROUND(Tstop-Tstart,3);"S"
700 !
710 !
720 PRINT
730 PRINT "Connecting..."
740 !***********************
750 !* Setup a call to the Fixed Part *
760 !***********************
770 !
780 ! clear the status registers then set the instrument to
790 ! generate a Service ReQuest(SRQ) when the traffic bearer
800 ! is active/connected.
810 !
820 OUTPUT @Hp8923;"*CLS" ! clear status registers.
830 !
840 OUTPUT @Hp8923;"*SRE 4" ! generate an SRQ when communicate
850 ! register summary bit in the
860 ! status byte changes to 1
870 !
880 OUTPUT @Hp8923;"status:comm:enable 64" ! enable the traffic_active bit
890 ! of the communicate register
900 OUTPUT @Hp8923;"status:comm:ptr 64" ! for positive transitions,
910 OUTPUT @Hp8923;"status:comm:ntr 0" ! 0 to 1. i.e traffic active
920 
930 !
940 ON INTR Bus,15 GOSUB Connection_isr ! setup computer to call interrupt
950 ! routine when an interrupt occurs
960 ! on the HP-IB
970 !
980  Test_bit=6                ! bit number in communicate
990  ! register for traffic bearer
1000
1010
1020  ENABLE INTR Bus;2        ! enable the computer to accept
1030  ! interrupts.
1040
1050
1060  Tstart=TIME DATE
1070  OUTPUT @Hp8923;"DECT:TRAFFIC:CONNECT"
1080
1090  LOOP                      ! Wait in this loop until the
1100  EXIT IF BIT(Comm_eve,Test_bit)  ! interrupt occurs
1110  END LOOP
1120
1130
1140  BEEP 4000,.3
1150  PRINT " Connection time: ";DROUND(Tstop-Tstart,3);"S"
1160
1170 !********************************************************************************
1180  !* Measurement code could go here *
1190 !********************************************************************************
1200
1210  DISP "Press Continue to release the call"
1220  PAUSE
1230  DISP "Releasing call . . ."
1240  PRINT
1250  PRINT "Releasing...
"
1260
1270  Comm_eve=0
1280  Status=0
1290
1300  ! clear the status registers then set the instrument to
1310  ! generate a Service ReQuest(SRQ) when the traffic bearer
1320  ! is released.
1330
1340  OUTPUT @Hp8923;"CLS"       ! clear status registers.
1350
1360  OUTPUT @Hp8923;"*SRE 4"    ! generate an SRQ when communicate
1370
1380
1390
1400  OUTPUT @Hp8923;"status:comm:enable 64" ! enable the traffic_active bit
1410  ! of the communicate register
1420  OUTPUT @Hp8923;"status:comm:ntr 64" ! for only negative transitions,
1430  OUTPUT @Hp8923;"status:comm:ptr 0" ! 1 to 0, ie traffic inactive
1440
1450
1460  NW INTR Bus,15 GOSUB Connection_isr ! setup computer to call interrupt
1470
1480
1490
1500  Test_bit=6                 ! bit number in communicate
1510  ! register for traffic bearer
1520
ENABLE INTR Bus;2 ! enable the computer to accept
! interrupts.

! DISP "waiting on interrupt...

! Tstart=TIMEDATE

! OUTPUT @Hp8923;"DECT:TRAFFIC:RELEASE"

! LOOP ! Wait in this loop until the
EXIT IF BIT(Comm_eve,Test_bit) ! interrupt occurs
END LOOP

! BEEP 4000,,3

! PRINT " Release time: ";DROUND(Tstop-Tstart,3);"S"

! DISP "Released and program stopped. "
STOP

! Connection_isr: !
OFF INTR ! Disable interrupts

IF No_print THEN GOSUB Read_comms

IF No_print THEN RETURN

PRINT " Subroutine Connection_isr"
PRINT

GOSUB Read_status ! Read status byte
GOSUB Read_comms ! read the communicate register
RETURN

! Read_comms:

! OUTPUT @Hp8923;"status:comm:event?" ! query communicate event register

ENTER @Hp8923;Comm_eve

! IF BIT(Comm_eve,Test_bit) THEN Tstop=TIMEDATE

IF No_print THEN RETURN

! DIM Comm_eve$[32]
Comm_eve$=DVAL$(Comm_eve,2)
PRINT " Communicate event:"; ! an easy to read register bit
FOR X=1 TO 29 STEP 4 ! representation
PRINT " "&Comm_eve$[X;4];
NEXT X

PRINT ";"&VAL$(Comm_eve) ! append decimal value of register

! OUTPUT @Hp8923;"status:comm:cond?" ! query the communicate condition

ENTER @Hp8923;Comm_cond ! register.

DIM Comm_cond$[32]
Comm_cond$=DVAL$(Comm_cond,2)
PRINT " Communicate cond: "; ! an easy to read register bit
FOR X=1 TO 29 STEP 4 ! representation
PRINT " "&Comm_cond$[X;4];
NEXT X

23-12 Example Programs
2080 PRINT ""&"VAL$(Comm_cond)" ! append decimal value of register
2090 RETURN
2100 !
2110 Read_status: !
2120 DIM Status$[32]
2130 OUTPUT @Hp8923;"+STB?"
2140 ENTER @Hp8923;Status
2150 !
2160 Status$=DVAL$(Status,2)
2170 PRINT " Status byte: ;Status$[25,4];" ";
2180 PRINT Status$[29;4]""&"VAL$(Status)
2190 RETURN
2200 !
2210 END
Example Program 4

This example programme demonstrates reading the trace display from the oscilloscope screen and displaying the measured data either in numeric or graphical form. The internal audio source is the measured signal.

```
10  COM /Osc/ Osc_trace(1:417),Scale_vols
20  !
30  ASSIGN @Hp8923 TO 714
40  OUTPUT @Hp8923;"*RST"
50  WAIT 2
60  !
70  CLEAR SCREEN
80  !
90  ! Switch to the audio screen, select an audio frequency
100 ! and turn on the audio source.
110 !
120 OUTPUT @Hp8923;"disp aud"
130 OUTPUT @Hp8923;"afg:stat 'on'"
140 OUTPUT @Hp8923;"afg:freq '400Hz'"
150 !
160 ! Prompt user to make connections
170 !
180 PRINT TABXY(1,10)
190 PRINT "Connect the 'AUDIO OUT' connector on the front panel"
200 PRINT "......to the 'AUDIO IN' connector on the front panel"
210 BEEP 400,.2
220 DISP "press Continue when connection is made"
230 PAUSE
240 DISP
250 CLEAR SCREEN
260 !
270 ! Switch to the oscilloscope screen and to its volt/time controls
280 ! Change the timebase to get a few cycles on the screen
290 ! Change the scaling to get a taller trace
300 OUTPUT @Hp8923;"disp osc"
310 OUTPUT @Hp8923;"disp:osc 'volt/time'"
320 OUTPUT @Hp8923;"osc:scale:time '1 ms'"
330!
340 OUTPUT @Hp8923;"osc:scale:vert:volts '500 mV'"
350 Scale_vols=.5  ! should always be set to the same value in volts
360 ! as the vertical scale
370 !
380 ! Read the 417 trace points into the array.
390 !
400 OUTPUT @Hp8923;"meas:aud:osc:trace?"
410 ENTER @Hp8923;Osc_trace(*)
420 !
430 ! Setup menu for user to view the trace results either as the direct
440 ! array contents or as a graphical trace.
450 !
460 Menu:LOOP
470  CLEAR SCREEN
480  BEEP 6000,.2
490  PRINT TABXY(1,10);"To view the trace press 't'"
```
500 PRINT TABXY(1,11);"To view the trace data press 'd'
510 PRINT TABXY(1,12);"To exit programme press 'q'
520 INPUT "Input Choice?", View$
530 EXIT IF LWC$(View$)="q"
540 IF LWC$(View$)="t" THEN GOSUB Show_trace
550 IF LWC$(View$)="d" THEN GOSUB Show_data
560 END LOOP
570 !
580 CLEAR SCREEN
590 DISP "Programme Stopped"
600 STOP
610 !
620 !
630 Show_data:!
640 ! Displays the contents of the trace array
650 ! a screenfull at a time
660 !
670 Lastx=1
680 CLEAR SCREEN
690 PRINT "Trace data"
700 FOR X=1 TO 417
710 PRINT ";&VAL$(X)&" of ";&VAL$(417)&" ", osc_trace(X)
720 IF X=Lastx+18 THEN
730 Lastx=X
740 DISP "Press continue to see more data"
750 PAUSE
760 DISP
770 END IF
780 NEXT X
790 DISP "Press Continue to return to Menu"
800 PAUSE
810 DISP
820 RETURN
830 !
840 !
850 Show_trace:!
860 ! Display the contents of the trace array
870 ! as a graphical trace.
880 !
890 CLEAR SCREEN ! clear any text on the screen
900 GINIT ! initialize the graphics
910 PLOTTER IS CRT,"INTERNAL" ! Use the internal screen
920 GRAPHICS ON ! Turn on the graphics screen
930 !
940 X_max=100*MAX(1,RATIO) ! Determine how wide the screen is
950 Y_max=100*MAX(1,1/RATIO) ! Determine how high the screen is
960 !
970 ! Define subset of screen area so that the graph and the prompt
980 ! line can both be seen
990 !
1000 VIEWPORT .01*X_max,.99*X_max,.26*Y_max,.99*Y_max
1010 !
1020 ! Scale the Window based on the known data range
1030 !
1040 WINDOW 1,420,-4*Scale_volts,4*Scale_volts
1050 !
1060 ! Draw axes and grid for the trace
1070 !
1080 PEN 1
1090 LINE TYPE 4
1100 AXES 42,5,0,0,1,1,2
1110 AXES 42,5,1,417,4,5,4
1120 GRID 21,Scale_vols,0,0,2,1
1130 !
1140 ! Plot the trace data
1150 !
1160 PEN 1
1170 LINE TYPE 1
1180 FOR X=1 TO 417
1190 PLOT X,0sc_trace(X)
1200 NEXT X
1210 DISP "Press Continue to return to Menu"
1220 PAUSE
1230 DISP
1240 RETURN
1250 !
1260 END
Appendix A

Definition of the Units Used in HP-IB Programming

Display Units (DUNits)

Display Units are those shown on the HP 8923B's screen; the full set of units used to manually control the instrument. For instance; Hz, kHz, MHz, and GHz can all be used to set and display the RF Generator frequency.

Changing Display Units

Use the DUNits syntax to change the Display Unit for any measurement unit, or setting. For example, to change the Display Unit for the output amplitude from dBm to Watts, you would enter the following BASIC command:

```
OUTPUT 714;"RFG:AMPL:DUN W"
```

<table>
<thead>
<tr>
<th>Display Units</th>
<th>HP-IB Syntax Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHz</td>
<td>MEAS:RF:FREQ:ACC:DUN GHZ</td>
</tr>
<tr>
<td>MHz</td>
<td>MEAS:RF:FREQ:ACC:DUN MHZ</td>
</tr>
<tr>
<td>kHz</td>
<td>MEAS:RF:FREQ:ACC:DUN KHZ</td>
</tr>
<tr>
<td>Hz</td>
<td>MEAS:RF:FREQ:ACC:DUN HZ</td>
</tr>
<tr>
<td>ppm</td>
<td>MEAS:BET:BERR:RAT:DUN PPM</td>
</tr>
<tr>
<td>%</td>
<td>MEAS:BET:BERR:RAT:DUN PCT</td>
</tr>
<tr>
<td>V</td>
<td>MEAS:AUD:DCV:DUN V</td>
</tr>
<tr>
<td>mV</td>
<td>MEAS:AUD:DCV:DUN MV</td>
</tr>
<tr>
<td>μV</td>
<td>RFG:AMPL:DUN UV</td>
</tr>
<tr>
<td>dBμV</td>
<td>RFG:AMPL:DUN DBUV</td>
</tr>
<tr>
<td>W</td>
<td>RFG:AMPL:DUN W</td>
</tr>
<tr>
<td>mW</td>
<td>RFG:AMPL:DUN MW</td>
</tr>
<tr>
<td>dBm</td>
<td>RFG:AMPL:DUN DBM</td>
</tr>
<tr>
<td>db</td>
<td>RFG:AMPL:DUN DB</td>
</tr>
</tbody>
</table>

Reading-Back Display Units

Use the DUNits? syntax for a measurement or setting to read-back its Display Unit. For example, to read the Display Unit for the Rise measurement, you would enter these BASIC commands:

```
OUTPUT 714;"MEAS:RF:NTP:DUNits?" ! Read Display Units for Normal Tx Power .
ENTER 714;A$ ! Enter the returned value into a string variable.
PRINT A$ ! Print the units.
```

The returned value will be the unit displayed on the HP 8923B's front-panel display for the measurement: dBm, V, mV, dBμV, or W. (Note: all returned characters are in upper case. Example; dBμV displayed returns DBUV.)
Guidelines for Display Units

- Querying measurements over HP-IB always returns numeric values in fundamental units, regardless of the current Display Unit.
- You can always use any appropriate Display Unit for a setting or measurement to display the value on the HP 8923B, regardless of its HP-IB Unit.
- The Display Unit for a numeric field is not affected when the field’s value is changed over HP-IB.
  
  For example, if the RFG:AMPL field is set to 0.0001 W, and you send the command RFG:AMPL 0.09 mW, the instrument displays '0.00000 W'; not 0.09 mW.

HP-IB UNITS

The :UNITs command is used to set the HP-IB units for a field. The HP-IB unit determines the units in which a query of a particular field is returned, and the default unit for setting. All :UNITs commands can only use fundamental units, that is, Hz for frequency, seconds for time, volts for voltage, and watts/dBm for power.

Changing HP-IB units has no effect on the front-panel display units. Use the UNITS syntax to change the HP-IB unit.

<table>
<thead>
<tr>
<th>Table A-1. Units for HP-IB Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Bit Error</td>
</tr>
<tr>
<td>Amplitude</td>
</tr>
<tr>
<td>Frequency</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td>Trace Displays</td>
</tr>
</tbody>
</table>

A field can be set in units other than the HP-IB units. Where normal multiples can be applied to fundamental units, the field can be set using these. A frequency field which always returns values in Hz can be set using KHZ, MHZ, or GHZ as a unit qualifies to the value setting, for example, RFAN:FREQ 1861 MHZ.

Reading-Back HP-IB Units

Use the UNITS? syntax for a measurement or setting to determine its HP-IB unit. For example, to determine the HP-IB Unit for the NTF measurement, you would enter these BASIC commands:

```
OUTPUT 714:"MEAS:RF:NTF:UNITS?"  ! Read HP-IB Units for Power-time.
ENTER 714:A$                ! Enter the returned value into a string variable.
PRINT A$                     ! Print the units.
```

In this example, ‘W’ would be printed (Watts).

Guidelines for HP-IB Units

- When changing a numeric setting, any non-HP-IB unit must be specified in the HP-IB command, or the HP-IB Unit is assumed by the HP 8923B to be the default HP-IB unit.
For example, if the Freq field is already set to 1850 MHz, and you send the command

\texttt{RFG:FREQ 1900}

the HP 8923B interprets that you are trying to set it to 1900 Hz, and results in an “Input value out of range” error. Sending the command

\texttt{RFG:FREQ 1900 MHz}

would set the value to 1900 MHz.

The returned numeric value for a measurement is \emph{always} in HP-IB units, regardless of the displayed units. The numeric value is expressed in scientific notation.

For example, if the Frequency measurement is shown as 1850.000000 MHz on the HP 8923B’s display, the returned value over HP-IB is 1.85000000E+09 (or $1.85 \times 10^9$). To convert the returned value to a non-HP-IB unit, you must enter the value into a conversion formula in your program.

\section*{Attribute Units (AUNits)}

Attribute units are used with the measurement Data Function, REF SET. AUNits provide a way to remotely query and change the units used to define a Data Function’s value.

Attribute units use the same set of units as HP-IB units, but are only applied to the measurement Data Functions.

\section*{Changing Attribute Units}

Some measurements can use more than one Attribute unit. For example, the bit error measurement allows you to set a Reference in units of ppm or %.

The Display Units shown on the HP 8923B’s front panel are not affected by changing Attribute Units. However, the Reference value and unit returned over HP-IB are in the new Attribute unit you specified using the above command.

\section*{Reading-back Attribute Units}

Use the AUNits? syntax to read-back the Attribute unit for a measurement.
Using STATE Commands

STATE commands correspond to using the ON/OFF key to turn measurements, Data Functions, and settings on and off.

Use a logic one or ON to turn a measurement on; logic zero or OFF to turn a measurement off. When queried, the returned value will be either '1' (on) or '0' (off).

For example, to turn off the voltage marker, and turn on measurement averaging for the RF rise measurement, enter the following BASIC commands:

```
OUTPUT 714;"MEAS:RF:MARK:LEV:STAT 0" !Turn off marker
OUTPUT 714;"MEAS:RF:FREQ:DEV:ZERO:BAV:STAT 1" !Turn on average measurement
```

State Command Guidelines

- All of the DATA FUNCTIONS can be turned on and off.
- Any function that generates a signal can be turned on and off.
Appendix B

Introduction

These commands appear predominantly in the MEASure Subsystem.

:STATe

The :STATe command activates or de-activates a measurement. All measurements default to active/on. Setting the boolean parameter to 0 of OFF turns the measurement off. Setting the boolean parameter to 1 of ON turns the measurement on.

<command>:STATe <boolean>
Where <boolean> = 0 | 1 | OFF | ON

:DUNits

The :DUNits command sets the units to be displayed on the screen. Table B-1 gives the valid :DUNits parameters for each command referred to Appendix B.

<command>:DUNits <display_units>

:UNItS

The :UNItS command sets the units of the <command> to be communicated between the instrument and the controller. Table B-1 gives the valid :UNItS parameters for each command referred to Appendix B.

<command>:UNIts <display_units>

:AUNits

The Attribute units are used with making measurements using the REF, INCR SET, and AVG functions. Table B-1 gives the valid :AUNits parameters for each command referred to Appendix B.

<command>:AUNits <display_units>

:AVERage[:VALue]

The :AVERage[:VALue] command sets the number of measurements used to calculate the averaged value.

<command>:AVERage[:VALue] <real>

:AVERage:RESet

The :AVERage:RESet command resets the measurement required to make the averaged result.

OUTPUT 714:"<command>:AVERage:RESet"
:AVerAge:STATe
The state of the average function can be switched on or off using this command. Enter a logic
0 to switch this off, or a logic 1 to switch this on.

  <command>:AVerAge:STATe <boolean>
  Where <boolean> = 0 | 1 | OFF | ON

:REFerence[:VALue]
Measurements can be made from the specified reference point, (the default setting is zero).
This reference point is defined by the user.
Refer to Table B-1 for information on the REFerence[:VALue] units displayed for each command
using Appendix B.

  <command>:REFerence[:VALue] <value> [[<display_units>]] optional
  Where <value> = a real number

:REFerence:DUNits
The reference value displayed on the screen is defined by this command. Refer to Table B-1 for
information on the REFerence:DUNits for each command using Appendix B.

  <command>:REFerence:DUNits <display_units>

:REFerence:STATe
The state of the reference is defined as either being on or off. The on state is programmed as a
logic one and the off state as logic zero.

  <command>:REFerence:STATe <boolean>
  Where <boolean> = 0 | 1 | OFF | ON.
Programming Command Guidelines

<table>
<thead>
<tr>
<th>Command</th>
<th>:DUN</th>
<th>:AUN</th>
<th>:UNIT</th>
<th>:REF::DUN</th>
<th>REF[:VAL]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAS:AUD:DCV</td>
<td>MV, V</td>
<td>V</td>
<td>V</td>
<td>MV, V</td>
<td>UV, MV, V</td>
</tr>
<tr>
<td>MEAS:AUD:FREQ</td>
<td>HZ, KHZ, MHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ</td>
<td>HZ, KHZ</td>
</tr>
<tr>
<td>MEAS:AUD:OSC:MARK:TIME</td>
<td>MS, S</td>
<td>S</td>
<td>S</td>
<td>MS, S</td>
<td>US, MS, S</td>
</tr>
<tr>
<td>MEAS:BET:BERR:RAT</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
</tr>
<tr>
<td>MEAS:BET:BERR:IRAT</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
</tr>
<tr>
<td>MEAS:BET:WERR:RAT</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
</tr>
<tr>
<td>MEAS:BET:WERR:IRAT</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
<td>PCT, PPM</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:ACC</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ZERO:BMAX</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ZERO:BMIN</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ZERO:BMAXH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMAXH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMINH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMAXH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMINH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMAXH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMINH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMAXH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DEV:ONE:BMINH</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
<tr>
<td>MEAS:RF:FREQ:DRIF</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
</tbody>
</table>
Table B-1. Programming Command Guidelines (continued)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAS:RF:PTIM:MARK:LEVEL:FALL</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>MEAS:RF:PTIM:MARK:LEVEL:MID</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>MEAS:RF:PTIM:MARK:LEVEL:RISE</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td></td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
</tr>
<tr>
<td></td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
</tr>
<tr>
<td></td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
<td>US, MS</td>
</tr>
<tr>
<td>MEAS:RF:TIMING:JITTER</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>NS</td>
<td>Real</td>
</tr>
</tbody>
</table>
Appendix C

Introduction
These commands appear predominantly in the OSCilloscope, PTIne and TRIGger subsytems.

:DUNits
The :DUNits command sets the units to be displayed on the screen. Table C-1 gives the valid :DUNits parameters for each command referred to Appendix C.

<command>:DUNits <display_units>

:UNITs
The :UNITs command sets the default units of the <command> to be communicated between the instrument and the controller. Table C-1 gives the valid :UNITs parameters for each command referred to Appendix C.

<command>:UNITs <units>

:INCReement
The :INCReement command sets the value by which the field value increases (using the UP arrow/command) or decreases (using the DOWN arrow/command). Table C-1 gives the valid :INCReement parameters for each command referred to Appendix C.

<command>:INCReement <increment_parameter> or <value>[<units_parameter>]
  Where <increment_parameter> = UP or DOWN
  Where <value> is the same type as <command>

:INCReement:DUNits
The :INCReement:DUNits command sets the description of the increment value. Table C-1 gives the valid :INCReement:DUNits parameters for each command referred to Appendix C.

<command>:INCReement:DUNits <display_units>

:INCReement:MODE

<command>:INCReement:MODE <scaling_mode>
  Where <scaling_mode> = 'LOGarithm' or 'LINear'
**:INCRement:MULTiply**

The multiply command increases the increment value by a factor of ten.

<command>:INCRement:MULTiply

**:INCRement:DIVide**

The divide command decreases the increment value by a factor of ten.

<command>:INCRement:DIVide

### Programming Command Guidelines

<table>
<thead>
<tr>
<th>Command</th>
<th>:DUN</th>
<th>:UNIT</th>
<th>:INCR:DUN</th>
<th>:INCR:[VALUE]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSC:MARK:POS</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>DB, DIV</td>
</tr>
<tr>
<td>OSC:SCALE:VERT:OFFS</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>DB, DIV</td>
</tr>
<tr>
<td>OSC:TRIG:LEV</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
</tr>
<tr>
<td>OSC:TRIG:PRET</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>EB, DIV</td>
</tr>
<tr>
<td>OSC:SCALE:VERT:OFFS</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>EB, DIV</td>
</tr>
<tr>
<td>PTIM:MARK:POS:FALL</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>DB, DIV</td>
</tr>
<tr>
<td>PTIM:MARK:POS:MID</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>DB, DIV</td>
</tr>
<tr>
<td>PTIM:MARK:POS:RISE</td>
<td>DIV</td>
<td>DIV</td>
<td>DIV</td>
<td>DB, DIV</td>
</tr>
<tr>
<td>RFAN:FREQ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
<td>HZ, KHZ, MHZ, GHZ</td>
</tr>
</tbody>
</table>
Appendix D

Introduction
These commands appear in the Oscilloscope Subsystem.

:INCRement:MULTiply
The multiply command increases the increment value by a factor of ten.
<command>:INCRement:MULTiply

:INCRement:DIVide
The multiply command decreases the increment value by a factor of ten.
<command>:INCRement:DIVide
Appendix E

To enter an integer in hexadecimal, octal, or binary, select the following base:

Using Different Numerical Bases

Hexadecimal

#H0123456
#h0123456

#H or #h is used to denote that a hexadecimal number is entered. The characters that follow are entered as the integer.

Octal

#Q0123456
#q0123456

#Q or #q is used to denote that a octal number is entered. The characters that follow are entered as the integer.

Binary

#B0101010
#b0101010

#B or #b is used to denote that a binary number is entered. The characters that follow are entered as the integer.
Index

A
ABORT, 11-5, 11-12, 11-45, 22-3
ACCuracy, 14-27, B-3
ACVolts, 14-5
ADDRess, 10-4
ADDRess, 10-14
AFANalyzer
INPut, 7-2
VOLTage, 7-3
AFANalyzer_Subsystem, 7-1
AFGenerator, 8-1
AFGenerator_TRAN, 8-5
AFGenerator_TX, 8-6
AField, 11-36, 11-37, 11-38, 11-39, 11-40, 11-41, 11-42
AMPL.CORR.LOSS, 18-3
AMPLitude, 18-2, 19-2
ANTenna, 11-38, 11-39, 11-40
ATTenuator, 19-3
AUTO, 19-4
AUDio, 14-5, 14-6, 14-7, 14-8, 14-9, 14-10, B-3
AUNits, A-3, B-1
AVERage, B-1, B-2

B
BADDRess, 10-4
BAUD, 10-21, 11-21
BAverage, 14-29, 14-32
BAVERage, B-3
BEEPer, 10-5
BERRor, 14-11, 14-12, 14-13, 14-14, B-3
BETest, 9-1, 12-3, 12-4, 14-11, 14-12, 14-13, 14-14, 14-15, 14-16, 14-17, 14-18, 14-19, 14-20, 14-21, 14-22, 22-4, 22-5, B-3
BITS, 9-2, 9-3
BMAXimum, 14-30, 14-33, B-3
BMNimum, 14-31, 14-34, B-3
BMODE, 10-6
BTESHed, 14-15

C
CALibration, 6-7, 20-2
CARRier, 11-10, 11-14, 11-15, 11-26, 11-27, 11-30, 11-31, 18-4, 18-5, 19-5
CLEar, 17-2

ALL, 17-3
*CLS, 5-2
COMMunicate, 5-13, 6-3, 6-10, 20-3
COMposite, 14-28
CONFigure, 10-1
CONNect, 11-16, 11-32, 11-46
COUNT, 14-11, 14-18
COUPLing, 18-6, 18-7
CW, 19-5, 19-6

D
DATA, 10-22
DATE, 10-7
DAY, 10-10
DCVolts, 14-6, B-3
DECT, 10-11, 11-1
DELay, 22-6
DESTination, 10-19, 11-48
DEVIation, 14-29, 14-30, 14-31, 14-32, 14-33, 14-34, B-3
DISPLAY, 12-1
DIVide, C-2, D-1
DRIFT, 14-35, B-3
DUMMY, 11-4, 11-5, 11-6, 11-10, 11-11, 11-12, 11-13, 11-25, 11-26, 11-27, 11-28, 11-29
DUNits, A-1, B-1, B-2, C-1

E
EDISK, 10-12
ERRror, 21-2
ESCape, 11-36, 11-37, 11-41, 11-42
*ESE, 5-3
ESOPATT, 13-7
ESOPW.OPOS, 13-3
ESOSLOT, 13-4
ESOSLOT.INCR, 13-5
ESOSTATE, 13-6
ESOURce, 13-1
POWer.ADVance, 13-2
*ESR, 5-5
EUT, 11-7
EUT.PARI, 11-8
EUT.PMID, 11-9

Index-1
RELease, 11-17, 11-33, 11-47
RFSet, 15-14, B-1
RETigger, 22-7
RF, 14-27, 14-28, 14-29, 14-30, 14-31, 14-32, 14-33, 14-34, 14-35, 14-36, 14-37, 14-38, 14-39, 14-40, 14-41, 14-42, 14-43, 14-44, 14-45, 14-46, 14-47, 14-48, B-3
RFANalyzer, 18-1
RFGenerator, 19-1
RISE, 14-39, 14-42, 14-45, 14-48, 16-4, B-3
RPACe, 10-26
RX, 11-36, 11-37, 11-38

S
*SAV, 5-12
SAVE, 17-5
SCALE, 15-6, 15-7, 15-8
SCReen, 12-2
SEND, 11-40, 11-42
SENSe, 15-15
SINPut, 10-27
SLOT, 11-11, 11-18, 11-19, 11-28, 11-29, 11-34, 11-35
SOURce, 15-16, 22-8
SPORT, 10-21, 10-22, 10-23, 10-24, 10-25, 10-26, 10-27, 10-28, 10-29, 11-21, 11-22
*SRE, 5-13
SRLocation, 10-30
STATE, 8-2, 11-4, 11-20, 11-25, A-4, B-1, B-2
STATUS, 11-37, 20-1
HARDware2, 20-5
STATUS?, 11-43
*STB, 5-14
STOP, 10-28
SYNC, 11-5, 11-6, 11-12, 11-13, 11-44, 11-45, 14-50

T
TEST, 11-36, 11-37, 11-38, 11-39, 11-40, 11-41, 11-42
TIME, 10-31, 14-9, 14-40, 14-41, 14-42, 15-6, B-3
TITLE, 10-20
TRACE, 14-47, 14-48
TRACE, 14-10
*TRG, 5-15
TRIGger, 15-9, 15-10, 15-11, 15-12, 15-13, 15-14, 15-15, 15-16, 15-17, 22-1
*TST, 5-16
TX, 11-39, 11-40, 11-41, 11-42
TYPE, 15-17

U
UNITS, A-2, B-1, C-1

V
VALue, B-1, B-2
VERTical, 15-7, 15-8
VIEW, 12-4
VOIce, 11-48
VOLT, B-3
VOLTs, 14-8, 15-8

W
*WAI, 5-17
warranty, iii
WERRor, 9-4, 14-18, 14-19, 14-20, 14-21, B-3
WTESted, 14-22

X
XPACe, 10-29

Y
YEAR, 10-8

Z
ZERO, 14-32, 14-33, 14-34, 18-9, B-3
ZField, 16-6