ROHDE&SCHWARZ

Test and Measurement
Division

Operating Manual

Test Receiver

FSET22

1080.3508.30
1080.3508.31
1080.3508.32

FSET7

1080.3508.07
1080.3508.17
1080.3508.27

Volume 2
This Operating Manual consists of 2 volumes

Printed in the Federal
Republic of Germany

Tabbed Divider Overview

Volume 1

Data Sheet

Safety Instructions
Certificate of quality
EC Certificate of Conformity
Support Center
List of R & S Representatives
List of Manuals

Tabbed Divider

1	**Chapter 1:**	**Preparing for Operation**
2	**Chapter 2:**	**Getting Started**
3	**Chapter 3:**	**Manual Operation**
4	**Chapter 4:**	**Instrument Functions**
10	**Chapter 10:**	**Index**

Volume 2

Safety Instructions
List of Manuals

Tabbed Divider

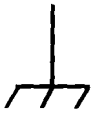
5	**Chapter 5:**	**Remote Control - Basics**
6	**Chapter 6:**	**Remote Control - Description of Commands**
7	**Chapter 7:**	**Remote Control - Programming Examples**
8	**Chapter 8:**	**Maintenance and Instrument Interfaces**
9	**Chapter 9:**	**Error Messages**
10	**Chapter 10:**	**Index**

Safety Instructions

This unit has been designed and tested in accordance with the EC Certificate of Conformity and has left the manufacturer's plant in a condition fully complying with safety standards.

To maintain this condition and to ensure safe operation, the user must observe all instructions and warnings given in this operating manual.

Safety-related symbols used on equipment and documentation from R&S:

							
Observe operating instructions	Weight indication for units >18 kg	PE terminal	Ground terminal	Danger! Shock hazard	Warning! Hot surfaces	Ground	Attention! Electrostatic sensitive devices require special care

1. The unit may be used only in the operating conditions and positions specified by the manufacturer. Unless otherwise agreed, the following applies to R&S products:

   IP degree of protection 2X, pollution severity 2 overvoltage category 2, only for indoor use, altitude max. 2000 m.

   The unit may be operated only from supply networks fused with max. 16 A.

   Unless specified otherwise in the data sheet, a tolerance of ±10% shall apply to the nominal voltage and of ±5% to the nominal frequency.

2. For measurements in circuits with voltages V_{rms} > 30 V, suitable measures should be taken to avoid any hazards.

   (using, for example, appropriate measuring equipment, fusing, current limiting, electrical separation, insulation).

3. If the unit is to be permanently wired, the PE terminal of the unit must first be connected to the PE conductor on site before any other connections are made. Installation and cabling of the unit to be performed only by qualified technical personnel.

4. For permanently installed units without built-in fuses, circuit breakers or similar protective devices, the supply circuit must be fused such as to provide suitable protection for the users and equipment.

5. Prior to switching on the unit, it must be ensured that the nominal voltage set on the unit matches the nominal voltage of the AC supply network.

   If a different voltage is to be set, the power fuse of the unit may have to be changed accordingly.

6. Units of protection class I with disconnectible AC supply cable and appliance connector may be operated only from a power socket with earthing contact and with the PE conductor connected.

7. It is not permissible to interrupt the PE conductor intentionally, neither in the incoming cable nor on the unit itself as this may cause the unit to become electrically hazardous.

   Any extension lines or multiple socket outlets used must be checked for compliance with relevant safety standards at regular intervals.

8. If the unit has no power switch for disconnection from the AC supply, the plug of the connecting cable is regarded as the disconnecting device. In such cases it must be ensured that the power plug is easily reachable and accessible at all times (length of connecting cable approx. 2 m). Functional or electronic switches are not suitable for providing disconnection from the AC supply.

   If units without power switches are integrated in racks or systems, a disconnecting device must be provided at system level.

9. Applicable local or national safety regulations and rules for the prevention of accidents must be observed in all work performed.

   Prior to performing any work on the unit or opening the unit, the latter must be disconnected from the supply network.

   Any adjustments, replacements of parts, maintenance or repair may be carried out only by authorized R&S technical personnel.

   Only original parts may be used for replacing parts relevant to safety (eg power switches, power transformers, fuses). A safety test must be performed after each replacement of parts relevant to safety.

   (visual inspection, PE conductor test, insulation-resistance, leakage-current measurement, functional test).

continued overleaf

10. Ensure that the connections with information technology equipment comply with IEC950 / EN60950.
11. Lithium batteries must not be exposed to high temperatures or fire.

    Keep batteries away from children.

    If the battery is replaced improperly, there is danger of explosion. Only replace the battery by R&S type (see spare part list).

    Lithium batteries are suitable for environmentally-friendly disposal or specialized recycling. Dispose them into appropriate containers, only.

    Do not short-circuit the battery.
12. Equipment returned or sent in for repair must be packed in the original packing or in packing with electrostatic and mechanical protection.
13. Electrostatics via the connectors may damage the equipment. For the safe handling and operation of the equipment, appropriate measures against electrostatics should be implemented.
14. The outside of the instrument is suitably cleaned using a soft, lint-free dustcloth. Never use solvents such as thinners, acetone and similar things, as they may damage the front panel labeling or plastic parts.
15. Any additional safety instructions given in this manual are also to be observed.

Patent Information

This product contains technology licensed by Marconi Instruments LTD. under US patent 4609881 and under the corresponding patent in Germany and elsewhere.

Contents of Manuals for Test Receiver FSET

Operating Manual FSET

The operating manual describes the following models and options:

- FSET7 100 Hz to 7 GHz
- FSET-Z2 100 Hz to 7 GHz
- FSET22 100 Hz to 22 GHz
- FSET-Z22 100 Hz to 22 GHz

- Option FSE-B15 Ethernet Adapter
- Option FSE-B17 Second IEC/IEEE Bus Interface

The present operating manual contains comprehensive information about the technical data of the instrument, the setup and putting into operation of the instrument, the operating concept and controls as well as the operation of the FSET via the menus and via remote control. Typical measurement tasks for the FSET are explained using the functions offered by the menus and a selection of program examples.

In addition the operating manual gives information about maintenance of the instrument and about error detection listing the error messages which may be output by the instrument. It is subdivided intothe data sheet plus 9 chapters:

Volume 1

The data sheet **i**nforms about guaranteed specifications and characteristics of the instrument.

Chapter 1 describes the control elements and connectors on the front and rear panel as well as all procedures required for putting the FSET into operation and integration into a test system.

Chapter 2 gives an introduction to typical measurement tasks of the FSET which are explained step by step.

Chapter 3 describes the operating principles, the structure of the graphical interface and offers a menu overview.

Chapter 4 forms a reference for manual control of the FSET and contains a detailed description of all instrument functions and their application.

Volume 2

Chapter 5 describes the basics for programming the FSET, command processing and the status reporting system.

Chapter 6 lists all the remote-control commands defined for the instrument. At the end of the chapter a alphabetical list of commands and a table of softkeys with command assignment is given.

Chapter 7 contains program examples for a number of typical applications of the FSET.

Chapter 8 describes preventive maintenance and the characteristics of the instrument's interfaces.

Chapter 8 gives a list of error messages that the FSET may generate.

Chapter 9 contains a list of error messages.

Chapter 10 contains an index for the operating manual.

Service Manual - Instrument

The service manual - instrument informs on how to check compliance with rated specifications (performance test) and on the self tests.

Service Manual

The service manual is not delivered with the instrument but may be obtained from your R&S service department using the order number 1080.3608.24.

The service manualinforms on instrument function, repair, troubleshooting and fault elimination. It contains all information required for the maintenance of FSET by exchanging modules.It contains information about the individual modules of FSET. This comprises the test and adjustment of the modules, fault detection within the modules and the interface description.

Contents - Chapter 5 "Remote Control - "Basics"

5 Remote Control - Basics 5.1

Introduction 5.1

Brief Instructions 5.2

Switchover to Remote Control 5.3

Indications during Remote Control 5.3

Remote Control via IEC Bus 5.4

Setting the Device Address 5.4

Return to Manual Operation 5.4

Remote Control via RS-232-Interface 5.5

Setting the Transmission Parameters 5.5

Return to Manual Operation 5.5

Limitations 5.5

Remote Control via RSIB Interface 5.6

Return to Manual Operation 5.6

Messages 5.7

IEE/IEEE-Bus Interface Messages 5.7

RSIB Interface Messages 5.7

Device Messages (Commands and Device Responses) 5.8

Structure and Syntax of the Device Messages 5.9

SCPI Introduction 5.9

Structure of a Command 5.9

Structure of a Command Line 5.12

Responses to Queries 5.12

Parameters 5.13

Overview of Syntax Elements 5.14

Instrument Model and Command Processing 5.15

Input Unit 5.15

Command Recognition 5.16

Data Set and Instrument Hardware 5.16

Status Reporting System 5.16

Output Unit 5.17

Command Sequence and Command Synchronization 5.17

Status Reporting System 5.18

Structure of an SCPI Status Register 5.18

Overview of the Status Registers 5.20

Description of the Status Registers 5.21

Status Byte (STB) and Service Request Enable Register (SRE) 5.21

IST Flag and Parallel Poll Enable Register (PPE) 5.22

Event-Status Register (ESR) and Event-Status-Enable Register (ESE) 5.22

STATus:OPERation Register 5.23

STATus:QUEStionable Register 5.24

STATus QUEStionable:FREQuency Register 5.25

STATus QUEStionable:LIMit Register 5.26

STATus QUEStionable:LMARgin Register 5.27

STATus QUEStionable:POWer Register 5.28

STATus QUEStionable:SYNC Register 5.29
STATus QUEStionable:TRANsducer Register 5.30
Application of the Status Reporting Systems..... 5.31
Service Request, Making Use of the Hierarchy Structure 5.31
Serial Poll 5.31
Parallel Poll..... 5.32
Query by Means of Commands..... 5.32
Error-Queue Query..... 5.32
Resetting Values of the Status Reporting System..... 5.33

5 Remote Control - Basics

In this chapter you find:

- instructions how to put the FSET into operation via remote control,
- a general introduction to remote control of programmable instruments. This includes the description of the command structure and syntax according to the SCPI standard, the description of command execution and of the status registers,
- diagrams and tables describing the status registers used in the FSET.

In chapter 6, all remote control functions are described in detail. The subsystems are listed by alphabetical order according to SCPI. All commands and their parameters are listed by alphabetical order in the command list at the end of chapter 6.

Program examples for the FSET can be found in chapter 7.

The remote control interfaces and their interface functions are described in chapter 8.

Introduction

The instrument is equipped with an IEC-bus interface according to standard IEC 625.1/IEEE 488.2 and two RS-232 interfaces. The connector is located at the rear of the instrument and permits to connect a controller for remote control.

The internal controller function may also be used as a controller (see chapter 1, section "Controller function).
In addition, the instrument is equipped with an RSIB interface that allows instrument control by Visual C++ and Visual Basic programs

The instrument supports the SCPI version 1994.0 (Standard Commands for Programmable Instruments). The SCPI standard is based on standard IEEE 488.2 and aims at the standardization of device-specific commands, error handling and the status registers (see section "SCPI Introduction").

This section assumes basic knowledge of IEC-bus programming and operation of the controller. A description of the interface commands is to be obtained from the relevant manuals. The RSIB interface functions are matched to the function interface for IEC/IEEE-bus programming from National Instruments. The functions supported by the DLLs are listed in chapter 8.

The requirements of the SCPI standard placed on command syntax, error handling and configuration of the status registers are explained in detail in the respective sections. Tables provide a fast overview of the commands implemented in the instrument and the bit assignment in the status registers. The tables are supplemented by a comprehensive description of every command and the status registers. Detailed program examples of the main functions are to be found in chapter 7.

The program examples for IEC-bus programming are all written in Quick BASIC.

Brief Instructions

The short and simple operating sequence given below permits fast putting into operation of the instrument and setting of its basic functions. As a prerequisite, the IEC-bus address, which is factory-set to 20, must not have been changed.

1. Connect instrument and controller using IEC-bus cable.
2. Write and start the following program on the controller:

```
CALL IBFIND("DEV1", analyzer%)              'Open port to the instrument
CALL IBPAD(analyzer%, 20)                   'Inform controller about instrument address
CALL IBWRT(analyzer%, "*RST;*CLS")          'Reset instrument
CALL IBWRT(analyzer%, 'FREQ:CENT 100MHz')   'Set center frequency to 100 MHz
CALL IBWRT(analyzer%, 'FREQ:SPAN 10MHz')    'Set span to 10 MHz
CALL IBWRT(analyzer%, 'DISP:TRAC:Y:RLEV -10dBm')
                                            'Set reference level to -10 dBm
```

The instrument now performs a sweep in the frequency range of 95 MHz to 105 MHz.

3. To return to manual control, press the *LOCAL* key at the front panel

Switchover to Remote Control

On power-on, the instrument is always in the manual operating state ("LOCAL" state) and can be operated via the front panel.
It is switched to remote control ("REMOTE" state)

IEC-bus	as soon as it receives an addressed command from a controller.
RS-232	as soon as it receives the command '@REM' from a controller.
RSIB	as soon as it receives an addressed command from a controller.

During remote control, operation via the front panel is disabled. The instrument remains in the remote state until it is reset to the manual state via the front panel or via remote control interfaces. Switching from manual operation to remote control and vice versa does not affect the remaining instrument settings.

Indications during Remote Control

Remote control mode is indicated by the LED "REMOTE" on the instrument's front panel. In this mode the softkeys, the function fields and the diagram labelling on the display are not shown.

Note: *Command `SYSTem:DISPlay:UPDate ON` activates all indications during remote control to check the instrument settings.*

Remote Control via IEC Bus

Setting the Device Address

In order to operate the instrument via the IEC-bus, it must be addressed using the set IEC-bus address. The IEC-bus address of the instrument is factory-set to 20. It can be changed manually in the *SETUP - GENERAL SETUP* menu or via IEC bus. Addresses 0 to 31 are permissible.

Manually:

- Call *SETUP - GENERAL SETUP* menu
- Enter desired address in table *GPIB ADDRESS*
- Terminate input using one of the unit keys (=ENTER).

Via IEC bus:

```
CALL IBFIND("DEV1", analyzer%)                    'Open port to the instrument
CALL IBPAD(analyzer%, 20)                         'Inform controller about old address
CALL IBWRT(analyzer%, "SYST:COMM:GPIB:ADDR 18")'Set instrument to new address
CALL IBPAD(analyzer%, 18)                         'Inform controller about new address
```

Return to Manual Operation

Return to manual operation is possible via the front panel or the IEC bus.

Manually:

- Press the *LOCAL* key.

***Notes*:**

– *Before switchover, command processing must be completed as otherwise switchover to remote control is effected immediately.*

– *The LOCAL key can be disabled by the universal command LLO (see chapter 8) in order to prevent unintentional switchover. In this case, switchover to manual mode is only possible via the IEC bus.*

– *The LOCAL key can be enabled again by deactivating the REN line of the IEC bus (see chapter 8).*

Via IEC bus:

```
...
CALL IBLOC(analyzer%)          'Set instrument to manual operation.
...
```

Remote Control via RS-232-Interface

Setting the Transmission Parameters

To enable an error-free and correct data transmission, the parameters of the unit and the controller should have the same setting. Parameters can be manually changed in menu *SETUP-GENERAL SETUP* in table *COM PORT 1/2* or via remote control using the command `SYSTem:COMMunicate:SERial1|2:...`.

The transmission parameters of the interfaces COM1 and COM2 are factory-set to the following values: baudrate = 9600, data bits = 8, stop bits = 1, parity = NONE and owner = INSTRUMENT.

Manually: Setting interface COM1|2

- Call *SETUP-GENERAL SETUP* menu
- Select desired baudrate, bits, stopbit, parity in table *COM PORT 1/2*.
- Terminate input using one of the unit keys (=ENTER).

Return to Manual Operation

Return to manual operation is possible via the front panel or via RS-232 interface.

Manually:

- Press the *LOCAL* key.

Notes:

- *Before switchover, command processing must be completed as otherwise switchover to remote control is effected immediately.*
- *The LOCAL key can be disabled by the universal command LLO (see chapter 8) in order to prevent unintentional switchover. In this case, switchover to manual mode is only possible via remote control.*
- *The LOCAL key can be enabled again by sending the control codes "@LOC" via RS-232 (see chapter 8).*

Via RS-232:

```
...
V24puts(port, '@LOC');          Set instrument to manual operation.
...
```

Limitations

The following limitations apply if the unit is remote-controlled via the RS-232-C interface:

- No interface messages, some control codes are defined (see chapter 8)., some control codes are defined (see chapter 8).
- Only the Common Commands *OPC? can be used for command synchronization, *WAI and *OPC are not available.
- Block data cannot be transmitted.
- When Windows NT is booted, data are output via the COM interface because of automatic external device recognition. Therefore, it is recommended to clear the input buffer of the controller before remote operation of the instrument via the COM interface.

Remote Control via RSIB Interface

Windows Environment

To access the measuring instruments via the RSIB interface the DLLs should be installed in the corresponding directories:

- `RSIB.DLL` in Windows NT `system` directory or control application directory.
- `RSIB32.DLL` in Windows NT `system32` directory or control application directory.

On the measuring instrument the DLL is already installed in the corresponding directory.

Unix Enviroment

In order to access the measuring equipment via the RSIB interface, copy the `librsib.so.X.Y` file to a directory for which the control application has read rights. `X.Y` in the file name indicates the version number of the library, for example `1.0` (for details see Chapter 8).

Remote Control

The control is performed with Visual C++ or Visual Basic programs. The local link to the internal controller is established with the name '@local'.If a remote controller is used, the instrument IP address is to be indicated here.

Via VisualBasic:

internal controller:	ud = RSDLLibfind ('@local', ibsta, iberr, ibcntl)
remote controller:	ud = RSDLLibfind ('82.1.1.200', ibsta, iberr, ibcntl)

Return to Manual Operation

The return to manual operation can be performed via the front panel (*LOCAL* key) or the RSIB interface.

Manually:

- Press the *LOCAL* key.

Note: *Before switchover, command processing must be completed as otherwise switchover to remote control is effected immediately.*

Via RSIB:

```
...
ud = RSDLLibloc (ud, ibsta, iberr, ibcntl);
...
```

Messages

The messages transferred via the data lines of the IEC bus or the RSIB interface (see chapter 8) can be divided into two groups:

– **interface messages and**
– **device messages.**

Some control characters are defined for the control of the RS-232-interface (see chapter 8).

IEE/IEEE-Bus Interface Messages

Interface messages are transferred on the data lines of the IEC bus, the "ATN" control line being active. They are used for communication between controller and instrument and can only be sent by a controller which has the IEC-bus control. Interface commands can be subdivided into

– **universal commands and**
– **addressed commands.**

Universal commands act on all devices connected to the IEC bus without previous addressing, addressed commands only act on devices previously addressed as listeners. The interface messages relevant to the instrument are listed in chapter 8.

RSIB Interface Messages

The RSIB interface enables the instrument to be controlled by Visual C++ or Visual Basic programs. The interface functions are matched to the function interface for IEC/IEEE-bus programming from National Instruments.
The functions supported by interface are listed in chapter 8.

Device Messages (Commands and Device Responses)

Device messages are transferred on the data lines of the IEC bus, the "ATN" control line not being active. ASCII code is used. The device messages are more or less equal for the different interfaces.
A distinction is made according to the direction in which they are sent on the IEC bus:

– **Commands** are messages the controller sends to the instrument. They operate the device functions and request informations.
The commands are subdivided according to two criteria::

1. According to the effect they have on the instrument:

   Setting commands cause instrument settings such as reset of the instrument or setting the center frequency.

   Queries cause data to be provided for output on the IEC-bus, e.g. for identification of the device or polling the marker.

2. According to their definition in standard IEEE 488.2:

   Common Commands are exactly defined as to their function and notation in standard IEEE 488.2. They refer to functions such as management of the standar-dized status registers, reset and selftest.

   Device-specific commands refer to functions depending on the features of the instrument such as frequency setting. A majority of these commands has also been standardized by the SCPI committee (cf. Section 3.5.1).

– **Device responses** are messages the instrument sends to the controller after a query. They can contain measurement results, instrument settings and information on the instrument status (cf. Section 3.5.4).

Structure and syntax of the device messages are described in the following section. The commands are listed and explained in detail in chapter 6.

Structure and Syntax of the Device Messages

SCPI Introduction

SCPI (Standard Commands for Programmable Instruments) describes a standard command set for programming instruments, irrespective of the type of instrument or manufacturer. The goal of the SCPI consortium is to standardize the device-specific commands to a large extent. For this purpose, a model was developed which defines the same functions inside a device or for different devices. Command systems were generated which are assigned to these functions. Thus it is possible to address the same functions with identical commands. The command systems are of a hierarchical structure.
Fig. 5-1 illustrates this tree structure using a section of command system SENSe, which controls the sensor functions of the devices.
SCPI is based on standard IEEE 488.2, i.e. it uses the same syntactic basic elements as well as the common commands defined in this standard. Part of the syntax of the device responses is defined with greater restrictions than in standard IEEE 488.2 (see Section "Responses to Queries").

Structure of a Command

The commands consist of a so-called header and, in most cases, one or more parameters. Header and parameter are separated by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). The headers may consist of several key words. Queries are formed by directly appending a question mark to the header.

Note: *The commands used in the following examples are not in every case implemented in the instrument.*

Common commands

Common commands consist of a header preceded by an asterisk "*" and one or several parameters, if any.

Examples:

`*RST`	RESET, resets the device
`*ESE 253`	EVENT STATUS ENABLE, sets the bits of the event status enable register
`*ESR?`	EVENT STATUS QUERY, queries the contents of the event status register.

Device-specific commands

Hierarchy**:** Device-specific commands are of hierarchical structure (see Fig. 5-1). The different levels are represented by combined headers. Headers of the highest level (root level) have only one key word. This key word denotes a complete command system.

Example: `SENSe` This key word denotes the command system `SENSe`.

For commands of lower levels, the complete path has to be specified, starting on the left with the highest level, the individual key words being separated by a colon ":".

Example: `SENSe:FREQuency:SPAN:LINK STARt`

This command lies in the fourth level of the SENSe system. It determines which parameter remains unchanged when the span is changed. If LINK is set to STARt, the values of CENTer and STOP are adjusted when the span is changed.

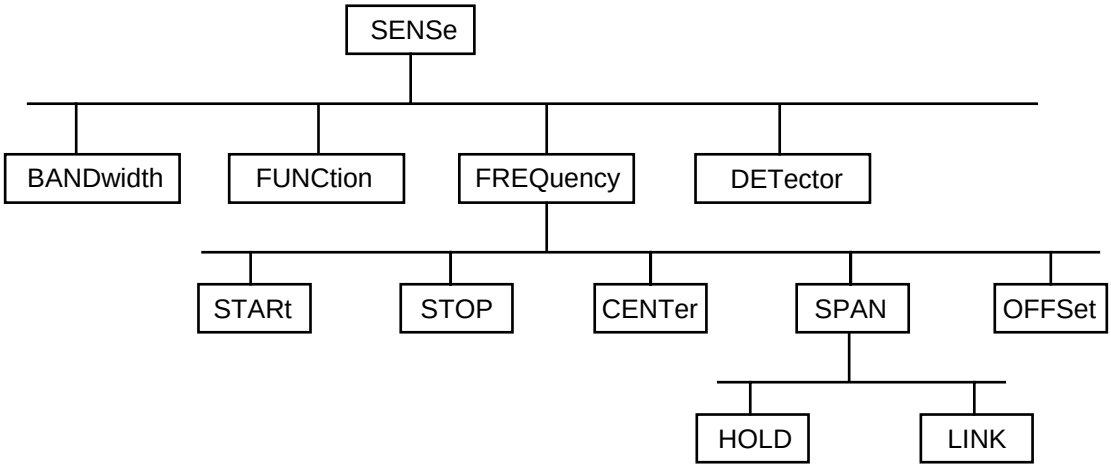


Fig. 5-1 Tree structure the SCPI command systems using the SENSe system by way of example

Some key words occur in several levels within one command system. Their effect depends on the structure of the command, that is to say, at which position in the header of a command they are inserted.

Example: `SOURce:FM:POLarity NORMal`
This command contains key word POLarity in the third command level. It defines the polarity between modulator and modulation signal.

`SOURce:FM:EXTernal:POLarity NORMal`
This command contains key word POLarity in the fourth command level. It defines the polarity between modulation voltage and the resulting direction of the modulation only for the external signal source indicated.

Optional key words: Some command systems permit certain key words to be optionally inserted into the header or omitted. These key words are marked by square brackets in the description. The full command length must be recognized by the instrument for reasons of compatibility with the SCPI standard. Some commands are considerably shortened by these optional key words.

Example: `[SENSe]:BANDwidth[:RESolution]:AUTO`
This command couples the resolution bandwidth of the instrument to other parameters. The following command has the same effect:
`BANDwidth:AUTO`

Note: *An optional key word must not be omitted if its effect is specified in detail by a numeric suffix.*

Long and short form: The key words feature a long form and a short form. Either the short form or the long form can be entered, other abbreviations are not permissible.

Beispiel: `STATus:QUEStionable:ENABle 1= STAT:QUES:ENAB 1`

Note: *The short form is marked by upper-case letters, the long form corresponds to the complete word. Upper-case and lower-case notation only serve the above purpose, the instrument itself does not make any difference between upper-case and lower-case letters.*

Parameter: The parameter must be separated from the header by a "white space". If several parameters are specified in a command, they are separated by a comma ",". A few queries permit the parameters MINimum, MAXimum and DEFault to be entered. For a description of the types of parameter, refer to Section 3.5.5.

Example: `SENSe:FREQuency:STOP? MAXimum` Response: 3.5E9
This query requests the maximal value for the stop frequency.

Numeric suffix: If a device features several functions or features of the same kind, e.g. inputs, the desired function can be selected by a suffix added to the command. Entries without suffix are interpreted like entries with the suffix 1.

Example:. `SYSTem:COMMunicate:SERial2:BAUD 9600`
This command sets the baudrate of the second serial interface.

Structure of a Command Line

A command line may consist of one or several commands. It is terminated by a <New Line>, a <New Line> with EOI or an EOI together with the last data byte. Quick BASIC automatically produces an EOI together with the last data byte.

Several commands in a command line are separated by a semicolon ";". If the next command belongs to a different command system, the semicolon is followed by a colon.

Example:
```
CALL IBWRT(analyzer, "SENSe:FREQuency:CENTer 100MHz;:INPut:ATTenuation 10")
```

This command line contains two commands. The first command is part of the SENSe system and is used to specify the center frequency of the analyzer. The second command is part of the INPut system and sets the attenuation of the input signal.

If the successive commands belong to the same system, having one or several levels in common, the command line can be abbreviated. To this end, the second command after the semicolon starts with the level that lies below the common levels (see also Fig. 5-1). The colon following the semicolon must be omitted in this case.

Example:
```
CALL IBWRT(analyzer, "SENSe:FREQuency:STARt 1E6;:SENSe:FREQuency:STOP 1E9")
```

This command line is represented in its full length and contains two commands separated from each other by the semicolon. Both commands are part of the SENSe command system, subsystem FREQuency, i.e. they have two common levels.
When abbreviating the command line, the second command begins with the level below SENSe:FREQuency. The colon after the semicolon is omitted.

The abbreviated form of the command line reads as follows:

```
CALL IBWRT(analyzer,  "SENSe:FREQuency:STARt 1E6;STOP 1E9")
```

However, a new command line always begins with the complete path.

Example:
```
CALL IBWRT(analyzer,  "SENSe:FREQuency:STARt 1E6")
CALL IBWRT(analyzer,  "SENSe:FREQuency:STOP 1E9")
```

Responses to Queries

A query is defined for each setting command unless explicitly specified otherwise. It is formed by adding a question mark to the associated setting command. According to SCPI, the responses to queries are partly subject to stricter rules than in standard IEEE 488.2.

1 The requested parameter is transmitted without header.
Example: `INPut:COUPling?` Response: `DC`

2. Maximum values, minimum values and all further quantities, which are requested via a special text parameter are returned as numerical values.
Example: `SENSe:FREQuency:STOP? MAX` Response: `3.5E9`

3. Numerical values are output without a unit. Physical quantities are referred to the basic units or to the units set using the Unit command.
Example: `SENSe:FREQuency:CENTer?` Response: `1E6` for 1 MHz

4. Truth values <Boolean values> are returned as 0 (for OFF) and 1 (for ON).
Example: `SENSe:BANDwidth:AUTO?` Response: `1` for ON

5. Text (character data) is returned in a short form (see also Section 3.5.5).
Example: `SYSTem:COMMunicate:SERial:CONTrol:RTS?` Response(for standard): `STAN`

Parameters

Most commands require a parameter to be specified. The parameters must be separated from the header by a "white space". Permissible parameters are numerical values, Boolean parameters, text, character strings and block data. The type of parameter required for the respective command and the permissible range of values are specified in the command description (see Section 3.6).

Numerical values Numerical values can be entered in any form, i.e. with sign, decimal point and exponent. Values exceeding the resolution of the instrument are rounded up or down. The value range is -9.9E37 to 9.9E37. The exponent is introduced by an "E" or "e". Entry of the exponent alone is not permissible. In the case of physical quantities, the unit can be entered. Permissible unit prefixes are G (giga), MA (mega), MOHM and MHZ are also permissible), K (kilo), M (milli), U (micro) and N (nano). It the unit is missing, the basic unit is used.

Example:
`SENSe:FREQuency:STOP 1.5GHz = SENSe:FREQuency:STOP 1.5E9`

Special numerical The texts MINimum, MAXimum, DEFault, UP and DOWN are interpreted as valuesspecial numerical values.

In the case of a query, the numerical value is provided.

Example: Setting command: `SENSe:FREQuency:STOP MAXimum`
Query: `SENSe:FREQuency:STOP?` Response: `3.5E9`

MIN/MAX MINimum and MAXimum denote the minimum and maximum value.

DEF DEFault denotes a preset value which has been stored in the EPROM. This value conforms to the default setting, as it is called by the *RST command

UP/DOWN UP, DOWN increases or reduces the numerical value by one step. The step width can be specified via an allocated step command for each parameter which can be set via UP, DOWN.

INF/NINF INFinity, Negative INFinity (NINF) Negative INFinity (NINF) represent the numerical values -9.9E37 or 9.9E37, respectively. INF and NINF are only sent as device reponses.

NAN Not A Number (NAN) represents the value 9.91E37. NAN is only sent as device response. This value is not defined. Possible causes are the division by zero, the subtraction/addition of infinite and the representation of undefined values.

Boolean Parameters Boolean parameters represent two states. The ON state (logically true) is represented by ON or a numerical value unequal to 0. The OFF state (logically untrue) is represented by OFF or the numerical value 0. 0 or 1 is provided in a query.

Example: Setting command: `DISPlay:WINDow:STATe ON`
Query: `DISPlay:WINDow:STATe?` Response: `1`

Text

Text parameters observe the syntactic rules for key words, i.e. they can be entered using a short or long form. Like any parameter, they have to be separated from the header by a white space. In the case of a query, the short form of the text is provided.

Example: Setting command: `INPut:COUPling GROund`
Query: `INPut:COUPling?` Response `GRO`

Strings

Strings must always be entered in quotation marks (' or ").

Example: `SYSTem:LANGuage "SCPI"` or
`SYSTem:LANGuage 'SCPI'`

Block data

Block data are a transmission format which is suitable for the transmission of large amounts of data. A command using a block data parameter has the following structure:

Example: `HEADer:HEADer #45168xxxxxxxx`

ASCII character # introduces the data block. The next number indicates how many of the following digits describe the length of the data block. In the example the 4 following digits indicate the length to be 5168 bytes. The data bytes follow. During the transmission of these data bytes all End or other control signs are ignored until all bytes are transmitted..

Overview of Syntax Elements

The following survey offers an overview of the syntax elements.

: The colon separates the key words of a command.
In a command line the separating semicolon marks the uppermost command level.

; The semicolon separates two commands of a command line.
It does not alter the path.

, The comma separates several parameters of a command.

? The question mark forms a query.

***** The asterisk marks a common command.

" **'** Double or single quotation marks introduce a string and terminate it.

The double dagger # introduces block data.

␣ A "white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates header and parameter.

Instrument Model and Command Processing

The instrument model shown in Fig. 5-2 has been made viewed from the standpoint of the servicing of IEC-bus commands. The individual components work independently of each other and simultaneously. They communicate by means of so-called "messages".

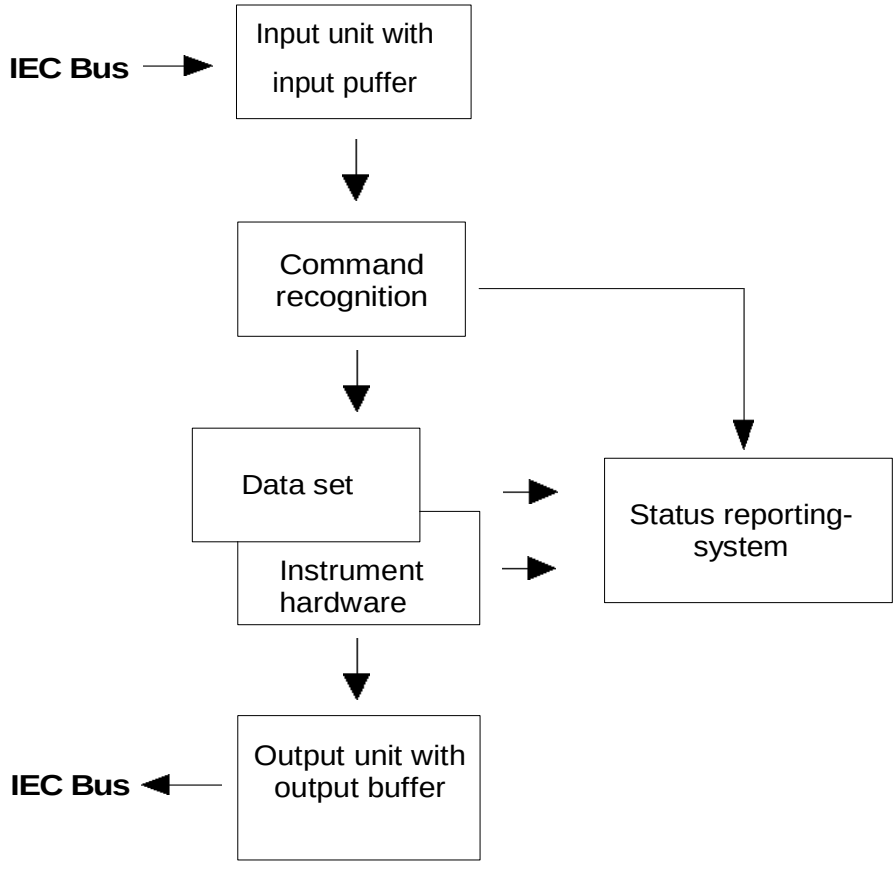


Fig. 5-2 Instrument model in the case of remote control by means of the IEC bus

Input Unit

The input unit receives commands character by character from the IEC bus and collects them in the input buffer. The input buffer has a size of 256 characters. The input unit sends a message to the command recognition as soon as the input buffer is full or as soon as it receives a delimiter, <PROGRAM MESSAGE TERMINATOR>, as defined in IEEE 488.2, or the interface message DCL.
If the input buffer is full, the IEC-bus traffic is stopped and the data received up to then are processed. Subsequently the IEC-bus traffic is continued. If, however, the buffer is not yet full when receiving the delimiter, the input unit can already receive the next command during command recognition and execution. The receipt of a DCL clears the input buffer and immediately initiates a message to the command recognition.

Command Recognition

The command recognition analyses the data received from the input unit. It proceeds in the order in which it receives the data. Only a DCL is serviced with priority, a GET (Group Execute Trigger), e.g., is only executed after the commands received before as well. Each recognized command is immediately transferred to the data set but without being executed there at once.
Syntactical errors in the command are recognized here and supplied to the status reporting system. The rest of a command line after a syntax error is analysed further if possible and serviced.
If the command recognition recognizes a delimiter or a DCL, it requests the data set to set the commands in the instrument hardware as well now. Subsequently it is immediately prepared to process commands again. This means for the command servicing that further commands can already be serviced while the hardware is still being set ("overlapping execution").

Data Set and Instrument Hardware

Here the expression "instrument hardware" denotes the part of the instrument fulfilling the actual instrument function - signal generation, measurement etc. The controller is not included.

The instrument data base is a detailed reproduction of the instrument hardware in the software.

IEC-bus setting commands lead to an alteration in the data set. The data base management enters the new values (e.g. frequency) into the data base, however, only passes them on to the hardware when requested by the command recognition.

The data are only checked for their compatibility among each other and with the instrument hardware immediately before they are transmitted to the instrument hardware. If the detection is made that an execution is not possible, an "execution error" is signalled to the status reporting system. The alteration of the data base are cancelled, the instrument hardware is not reset.

IEC-bus queries induce the data set management to send the desired data to the output unit.

Status Reporting System

The status reporting system collects information on the instrument state and makes it available to the output unit on request. The exact structure and function are described in the following section.

Output Unit

The output unit collects the information requested by the controller, which it receives from the data set management. It processes it according to the SCPI rules and makes it available in the output buffer. The output buffer has a size of 4096 characters. If the information requested is longer, it is made available "in portions" without this being recognized by the controller.
If the instrument is addressed as a talker without the output buffer containing data or awaiting data from the data set management, the output unit sends error message "Query UNTERMINATED" to the status reporting system. No data are sent on the IEC bus, the controller waits until it has reached its time limit. This behaviour is specified by SCPI.

Command Sequence and Command Synchronization

What has been said above makes clear that all commands can potentially be carried out overlapping. Equally, setting commands within one command line are not absolutely serviced in the order in which they have been received.

In order to make sure that commands are actually carried out in a certain order, each command must be sent in a separate command line, that is to say, with a separate IBWRT()-call.
In order to prevent an overlapping execution of commands, one of commands *OPC, *OPC? or *WAI must be used. All three commands cause a certain action only to be carried out after the hardware has been set and has settled. By a suitable programming, the contoller can be forced to wait for the respective action to occur (cf. Table 5-1).

Table 5-1 Synchronisation using *OPC, *OPC? and *WAI

Commnd	**Action after the hardware has settled**	**Programming the controller**
*OPC	Setting the opteration-complete bit in the ESR	- Setting bit 0 in the ESE - Setting bit 5 in the SRE - Waiting for service request (SRQ)
*OPC?	Writing a "1" into the output buffer	Addressing the instrument as a talker
*WAI	Continuing the IEC-bus handshake	Sending the next command

An example as to command synchronization can be found in chapter 7 "Program Examples".

Status Reporting System

The status reporting system (cf. Fig. 5-3) stores all information on the present operating state of the instrument, e.g. that the instrument presently carries out an AUTORANGE and on errors which have occurred. This information is stored in the status registers and in the error queue. The status registers and the error queue can be queried via IEC bus.

The information is of a hierarchical structure. The register status byte (STB) defined in IEEE 488.2 and its associated mask register service request enable (SRE) form the uppermost level. The STB receives its information from the standard event status register (ESR) which is also defined in IEEE 488.2 with the associated mask register standard event status enable (ESE) and registers STATus:OPERation and STATus:QUEStionable which are defined by SCPI and contain detailed information on the instrument.

The IST flag ("Individual STatus") and the parallel poll enable register (PPE) allocated to it are also part of the status reporting system. The IST flag, like the SRQ, combines the entire instrument status in a single bit. The PPE fulfills an analog function for the IST flag as the SRE for the service request.

The output buffer contains the messages the instrument returns to the controller. It is not part of the status reporting system but determines the value of the MAV bit in the STB and thus is represented in Fig. 5-3.

Table 5-12 at the end of this chapter comprises the different commands and events causing the status reporting system to be reset.

Structure of an SCPI Status Register

Each SCPI register consists of 5 parts which each have a width of 16 bits and have different functions (cf. Fig. 5-2). The individual bits are independent of each other, i.e. each hardware status is assigned a bit number which is valid for all five parts. For example, bit 3 of the STATus:OPERation register is assigned to the hardware status "wait for trigger" in all five parts. Bit 15 (the most significant bit) is set to zero for all parts. Thus the contents of the register parts can be processed by the controller as positive integer.

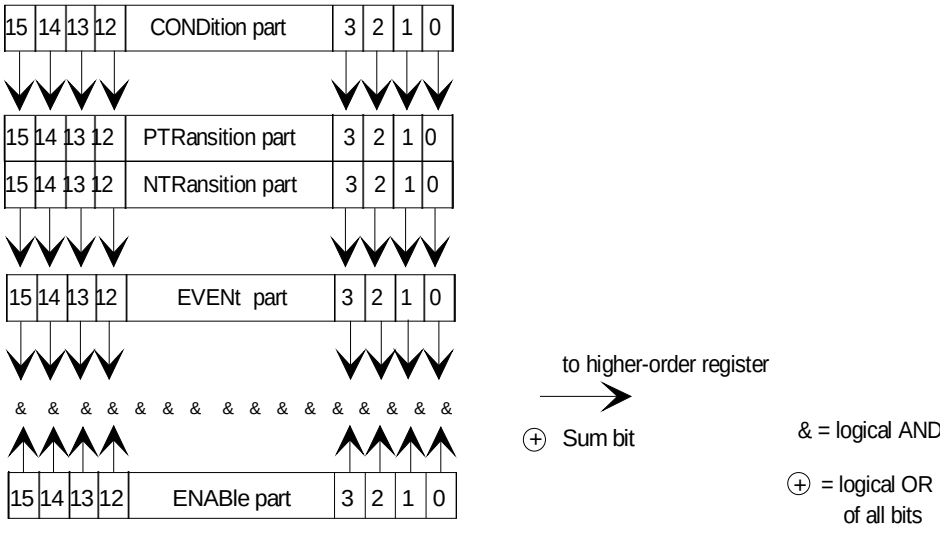


Fig. 5-2 The status-register model

CONDition part

The CONDition part is directly written into by the hardware or the sum bit of the next lower register. Its contents reflects the current instrument status. This register part can only be read, but not written into or cleared. Its contents is not affected by reading.

PTRansition part

The Positive-TRansition part acts as an edge detector. When a bit of the CONDition part is changed from 0 to 1, the associated PTR bit decides whether the EVENt bit is set to 1.
PTR bit =1: the EVENt bit is set.
PTR bit =0: the EVENt bit is not set.
This part can be written into and read at will. Its contents is not affected by reading.

NTRansition part

The Negative-TRansition part also acts as an edge detector. When a bit of the CONDition part is changed from 1 to 0, the associated NTR bit decides whether the EVENt bit is set to 1.
NTR-Bit = 1: the EVENt bit is set.
NTR-Bit = 0: the EVENt bit is not set.
This part can be written into and read at will. Its contents is not affected by reading.

With these two edge register parts the user can define which state transition of the condition part (none, 0 to 1, 1 to 0 or both) is stored in the EVENt part.

EVENt part

The EVENt part indicates whether an event has occurred since the last reading, it is the "memory" of the condition part. It only indicates events passed on by the edge filters. It is permanently updated by the instrument. This part can only be read by the user. During reading, its contents is set to zero. In linguistic usage this part is often equated with the entire register.

ENABle part

The ENABle part determines whether the associated EVENt bit contributes to the sum bit (cf. below). Each bit of the EVENt part is ANDed with the associated ENABle bit (symbol '&'). The results of all logical operations of this part are passed on to the sum bit via an OR function (symbol '+').
ENABle-Bit = 0: the associated EVENt bit does not contribute to the sum bit
ENABle-Bit = 1: if the associated EVENT bit is "1", the sum bit is set to "1" as well.
This part can be written into and read by the user at will. Its contents is not affected by reading.

Sum bit

As indicated above, the sum bit is obtained from the EVENt and ENABle part for each register. The result is then entered into a bit of the CONDition part of the higher-order register.
The instrument automatically generates the sum bit for each register. Thus an event, e.g. a PLL that has not locked, can lead to a service request throughout all levels of the hierarchy.

Note: *The service request enable register SRE defined in IEEE 488.2 can be taken as ENABle part of the STB if the STB is structured according to SCPI. By analogy, the ESE can be taken as the ENABle part of the ESR.*

Overview of the Status Registers

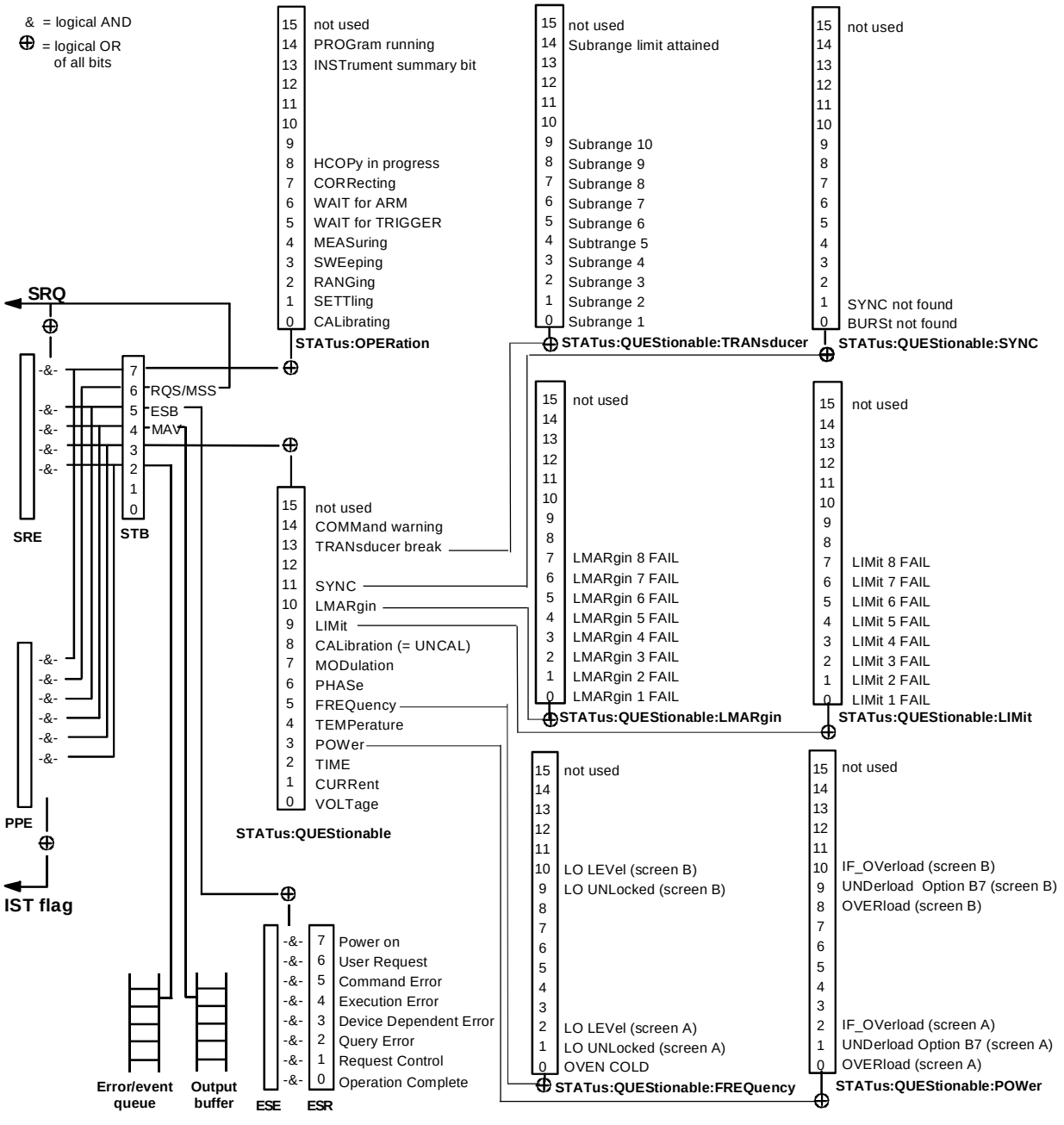


Fig. 5-3 Overview of the status registers

Description of the Status Registers

Status Byte (STB) and Service Request Enable Register (SRE)

The STB is already defined in IEEE 488.2. It provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. It can thus be compared with the CONDition part of an SCPI register and assumes the highest level within the SCPI hierarchy. A special feature is that bit 6 acts as the sum bit of the remaining bits of the status byte.
The STATUS BYTE is read out using the command "*STB?" or a serial poll.

The STB implies the SRE. It corresponds to the ENABle part of the SCPI registers as to its function. Each bit of the STB is assigned a bit in the SRE. Bit 6 of the SRE is ignored. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a Service Request (SRQ) is generated on the IEC bus, which triggers an interrupt in the controller if this is appropriately configured and can be further processed there.
The SRE can be set using command "*SRE" and read using "*SRE?".

Table 5-2 Meaning of the bits in the status byte

Bit No.	Meaning
2	**Error Queue not empty** The bit is set when an entry is made in the error queue. If this bit is enabled by the SRE, each entry of the error queue generates a Service Request. Thus an error can be recognized and specified in greater detail by polling the error queue. The poll provides an informative error message. This procedure is to be recommended since it considerably reduces the problems involved with IEC-bus control.
3	**QUEStionable status sum bit** The bit is set if an EVENt bit is set in the QUEStionable-Status register and the associated ENABle bit is set to 1. A set bit indicates a questionable instrument status, which can be specified in greater detail by polling the QUEStionable-Status register.
4	**MAV bit** (message available) The bit is set if a message is available in the output buffer which can be read. This bit can be used to enable data to be automatically read from the instrument to the controller (cf. chapter 7, program examples).
5	**ESB bit** Sum bit of the event status register. It is set if one of the bits in the event status register is set and enabled in the event status enable register. Setting of this bit implies an error or an event which can be specified in greater detail by polling the event status register.
6	**MSS bit** (master status smmary bit) The bit is set if the instrument triggers a service request. This is the case if one of the other bits of this registers is set together with its mask bit in the service request enable register SRE.
7	**OPERation status register sum bit** The bit is set if an EVENt bit is set in the OPERation-Status register and the associated ENABle bit is set to 1. A set bit indicates that the instrument is just performing an action. The type of action can be determined by polling the OPERation-status register.

IST Flag and Parallel Poll Enable Register (PPE)

By analogy with the SRQ, the IST flag combines the entire status information in a single bit. It can be queried by means of a parallel poll or using command "*IST?".

The parallel poll enable register (PPE) determines which bits of the STB contribute to the IST flag. The bits of the STB are ANDed with the corresponding bits of the PPE, with bit 6 being used as well in contrast to the SRE. The Ist flag results from the ORing of all results. The PPE can be set using commands "*PRE" and read using command "*PRE?".

Event-Status Register (ESR) and Event-Status-Enable Register (ESE)

The ESR is already defined in IEEE 488.2. It can be compared with the EVENt part of an SCPI register. The event status register can be read out using command "*ESR?".
The ESE is the associated ENABle part. It can be set using command "*ESE" and read using command "*ESE?".

Table 5-3 Meaning of the bits in the event status register

Bit No.	Meaning
0	**Operation Complete** This bit is set on receipt of the command *OPC exactly when all previous commands have been executed.
1	**Request Control** This bit is set if the instrument requests the controller function. This is the case when hardcopy is outputted to a printer or a plotter via the IEC-bus.
2	**Query Error** This bit is set if either the controller wants to read data from the instrument without having send a query, or if it does not fetch requested data and sends new instructions to the instrument instead. The cause is often a query which is faulty and hence cannot be executed.
3	**Device-dependent Error** This bit is set if a device-dependent error occurs. An error message with a number between -300 and -399 or a positive error number, which denotes the error in greater detail, is entered into the error queue (cf. chapter 9, Error Messages).
4	**Execution Error** This bit is set if a received command is syntactically correct, however, cannot be performed for other reasons. An error message with a number between -200 and -300, which denotes the error in greater detail, is entered into the error queue (cf. chapter 9, Error Messages).
5	**Command Error** This bit is set if a command which is undefined or syntactically incorrect is received. An error message with a number between -100 and -200, which denotes the error in greater detail, is entered into the rror queue (cf. chapter 9, -Error Messages).
6	**User Request** This bit is set on pressing the *LOCAL* key.
7	**Power On** (supply voltage on) This bit is set on switching on the instrument.

STATus:OPERation Register

In the CONDition part, this register contains information on which actions the instrument is being executing or, in the EVENt part, information on which actions the instrument has executed since the last reading. It can be read using commands "STATus:OPERation:CONDition?" or "STATus :OPERation[:EVENt]?".

Table 5-4 Meaning of the bits in the STATus.OPERation register

Bit No.	Meaning
0	**CALibrating** This bit is set as long as the instrument is performing a calibration.
1	**SETTling** This bit is set as long as the new status is settling after a setting command. It is only set if the settling time is longer than the command processing time.
2	**RANGing** This bit is set as long as the instrument is changing a range (e.g. Autorange).
3	**SWEeping** This bit is set while the instrument is performing a sweep.
4	**MEASuring** This bit is set while the instrument is performing a measurement.
5	**WAIT for TRIGGER** This bit is set as long as the instrument is waiting for a trigger event.
6	**WAIT for ARM** This bit is set as long as the instrument is waiting for an arming event.
7	**CORRecting** This bit is set while the instrument is performing a correction.
8	**HardCOPy in progress** This bit is set while the instrument is printing a hardcopy.
9-12	Device dependent
13	**INSTrument Summary Bit** This bit is set when one or more logical instruments is reporting a status message.
14	**PROGram running** This bit is set while the instrument is performing a program.
15	This bit is always 0

The FSET supports bits 0 and 8.

STATus:QUEStionable Register

This register comprises information about indefinite states which may occur if the unit is operated without meeting the specifications. It can be queried by commands `STATus:QUEStionable:CONDition?` and `STATus:QUEStionable[:EVENt]?`.

Table 5-5 Meaning of bits in STATus:QUEStionable register

Bit No.	Meaning
0	**VOLTage** This bit is set if a questionable voltage occurs.
1	**CURRent** This bit is set if a questionable current occurs.
2	**TIME** This bit is set if a questionable time occurs.
3	**POWer** This bit is set if a questionable power occurs (cf. also section "STATus:QUEStionable:POWerRegister")
4	**TEMPerature** This bit is set if a questionable temperature occurs.
5	**FREQuency** The bit is set if a frequency is questionable (cf. section "STATus:QUEStionable:FREQuency Register")
6	**PHASe** The bit is set if a phase value is questionable.
7	**MODulation** The bit is set if a modulation is performed questionably.
8	**CALibration** The bit is set if a measurement is performed uncalibrated (≙ label "UNCAL")
9	**LIMit** (unit-dependent) This bit is set if a limit value is violated (see also section STATus:QUEStionable:LIMit Register)
10	**LMARgin** (unit-dependent) This bit is set if a margin is violated (see also section STATus:QUEStionable:LMARgin Register)
11	**SYNC** (unit-dependent) This bit is set if, during measurements with Option B7 (Signal Vector Analysis), the synchronization with midamble or a successful search for bursts cannot be performed (see also STATus:QUEStionable:SYNC Register)
12	**ACPLimit** (unit-dependent) This bit is set if a limit for the adjacent channel power measurement is violated (see also section STATus:QUEStionable:ACPLimit Register)
13	**TRANsducer break** This bit is set when the limit of the transducer set subrange is attained.
14	**COMMand Warning** This bit is set if the instrument ignores parameters when executing a command.
15	This bit is always 0.

The FSET supports bits 3, 5, 7, 8, 9, 10, 11 and 13.

STATus QUEStionable:FREQuency Register

This register comprises information about the reference and local oscillator.
It can be queried with commands `STATus:QUEStionable:FREQuency:CONDition?` and "`STATus:QUEStionable:FREQuency[:EVENt]?`.

Table 5-6 Meaning of bits in STATus:QUEStionable:FREQuency register

Bit No.	Meaning
0	**OVEN COLD** This bit is set if the reference oscillator has not yet attained its operating temperature. 'OCXO' will then be displayed.
1	**LO UNLocked** (Screen A) This bit is set if the local oscillator no longer locks. 'LO unl' will then be displayed.
2	**LO LEVel** (Screen A**)** This bit is set if the level of the local oscillator is smaller than the nominal value. 'LO LVL' will then be displayed.
3	not used
4	not used
5	not used
6	not used
7	not used
8	not used
9	**LO UNLocked** (Screen B) This bit is set if the local oscillator no longer locks.' LO unl' will then be displayed.
10	**LO LEVel** (Screen B**)** This bit is set if the level of the local oscillator is smaller than the nominal value. 'LO LVL' will then be displayed.
11	not used
12	not used
13	not used
14	not used
15	This bit is always 0.

STATus QUEStionable:LIMit Register

This register comprises information about the observance of limit lines. It can be queried with commands `STATus:QUEStionable:LIMit:CONDition?` and `STATus:QUEStionable:LIMit[:EVENt]?`.

Table 5-7 Meaning of bits in STATus:QUEStionable:LIMit register

Bit No.	Meaning
0	**LIMit 1 FAIL** This bit is set if limit line 1 is violated.
1	**LIMit 2 FAIL** This bit is set if limit line 2 is violated.
2	**LIMit 3 FAIL** This bit is set if limit line 3 is violated.
3	**LIMit 4 FAIL** This bit is set if limit line 4 is violated.
4	**LIMit 5 FAIL** This bit is set if limit line 5 is violated.
5	**LIMit 6 FAIL** This bit is set if limit line 6 is violated.
6	**LIMit 7 FAIL** This bit is set if limit line 7 is violated.
7	**LIMit 8 FAIL** This bit is set if limit line 8 is violated.
8	not used
9	not used
10	not used
11	not used
12	not used
13	not used
14	not used
15	This bit is always 0.

STATus QUEStionable:LMARgin Register

This register comprises information about the observance of limit margins. It can be queried with commands `STATus:QUEStionable:LMARgin:CONDition?` and "`STATus:QUEStionable :LMARgin[:EVENt]?`.

Table 5-8 Meaning of bits in STATus:QUEStionable:LMARgin register

Bit No.	Meaning
0	**LMARgin 1 FAIL** This bit is set if limit margin 1 is violated.
1	**LMARgin 2 FAIL** This bit is set if limit margin 2 is violated.
2	**LMARgin 3 FAIL** This bit is set if limit margin 3 is violated.
3	**LMARgin 4 FAIL** This bit is set if limit margin 4 is violated.
4	**LMARgin 5 FAIL** This bit is set if limit margin 5 is violated.
5	**LMARgin 6 FAIL** This bit is set if limit margin 1 is violated.
6	**LMARgin 7 FAIL** This bit is set if limit margin 7 is violated.
7	**LMARgin 8 FAIL** This bit is set if limit margin 8 is violated.
8	not used
9	not used
10	not used
11	not used
12	not used
13	not used
14	not used
15	This bit is always 0.

STATus QUEStionable:POWer Register

This register comprises all information about possible overloads of the unit.
It can be queried with commands `STATus:QUEStionable :POWer:CONDition?` and "`STATus :QUEStionable:POWer [:EVENt]?`.

Table 5-9 Meaning of bits in STATus:QUEStionable:POWer register

Bit No.	Meaning
0	**OVERload** (Screen A) This bit is set if the RF input is overloaded. 'OVLD' will then be displayed.
1	**UNDerload** (Screen A) This bit is set if, during measurements in vector analyzer mode without capture buffer used, the lower level limit in the IF path is violated.
2	**IF_OVerload** (Screen A) This bit is set if the IF path is overloaded. 'IFOVLD' will then be displayed.
3	not used
4	not used
5	not used
6	not used
7	not used
8	**OVERload** (Screen B) This bit is set if the RF input is overloaded. 'OVLD' will then be displayed.
9	**UNDerload** (Screen B) This bit is set if, during measurements without capture buffer used, the lower level limit in the IF path is violated.
10	**IF_OVerload** (Screen B) This bit is set if the IF path is overloaded. 'IFOVLD' will then be displayed.
11	not used
12	not used
13	not used
14	not used
15	This bit is always 0.

STATus QUEStionable:SYNC Register

This register comprises information about sync and burst events.
It can be queried with commands `STATus:QUEStionable:SYNC:CONDition?` and `"STATus :QUEStionable:SYNC[:EVENt]?`.

Table 5-10 Meaning of bits in STATus:QUEStionable:SYNC register

Bit No.	Meaning
0	**BURSt not found** This bit is set if a burst was not found.
1	**SYNC not found** This bit is set if the sync sequence of midamble was not found.
2 to 14	not used
15	This bit is always 0.

STATus QUEStionable:TRANsducer Register

This register indicates that a transducer hold point is attained (bit 15) and what range is to be swept next (bit 0 to 10). The sweep can be continued with command `INITiate2:CONMeasure`.
It can be queried with commands `STATus:QUEStionable:TRANsducer:CONDition?` and "`STATus:QUEStionable:TRANsducer[:EVENt]?`.

Table 5-11 Meaning of bits in STATus:QUEStionable:TRANsducer register

Bit No.	Meaning
0	**Range 1** This bit is set when subrange 1 is attained.
1	**Range 2** This bit is set when subrange 2 is attained.
2	**Range 3** This bit is set when subrange 3 is attained.
3	**Range 4** This bit is set when subrange 4 is attained.
4	**Range 5** This bit is set when subrange 1 is attained.
5	**Range 6** This bit is set when subrange 6 is attained.
6	**Range 7** This bit is set when subrange 7 is attained.
7	**Range 8** This bit is set when subrange 8 is attained.
8	**Range 9** This bit is set when subrange 9 is attained.
9	**Range 10** This bit is set when subrange 10 is attained.
10	**not used**
11	**not used**
12	**not used**
13	**not used**
14	**Subrange limit** This bit is set when the transducer is at the point of changeover from one range to another.
15	This bit is always 0.

Application of the Status Reporting Systems

In order to be able to effectively use the status reporting system, the information contained there must be transmitted to the controller and further processed there. There are several methods which are represented in the following. Detailed program examples are to be found in chapter 7, Program Examples.

Service Request, Making Use of the Hierarchy Structure

Under certain circumstances, the instrument can send a service request (SRQ) to the controller. Usually this service request initiates an interrupt at the controller, to which the control program can react with corresponding actions. As evident from Fig. 5-3, an SRQ is always initiated if one or several of bits 2, 3, 4, 5 or 7 of the status byte are set and enabled in the SRE. Each of these bits combines the information of a further register, the error queue or the output buffer. The corresponding setting of the ENABle parts of the status registers can achieve that arbitrary bits in an arbitrary status register initiate an SRQ. In order to make use of the possibilities of the service request, all bits should be set to "1" in enable registers SRE and ESE.

Examples (cf. Fig. 5-3 and chapter 7, Program Examples, as well):

Use of command "*OPC" to generate an SRQ at the end of a sweep.

- Set bit 0 in the ESE (Operation Complete)
- Set bit 5 in the SRE (ESB)?

  After its settings have been completed, the instrument generates an SRQ.

The SRQ is the only possibility for the instrument to become active on its own. Each controller program should set the instrument such that a service request is initiated in the case of malfunction. The program should react appropriately to the service request. A detailed example for a service request routine is to be found in chapter 7, Program Examples.

Serial Poll

In a serial poll, just as with command "*STB", the status byte of an instrument is queried. However, the query is realized via interface messages and is thus clearly faster. The serial-poll method has already been defined in IEEE 488.1 and used to be the only standard possibility for different instruments to poll the status byte. The method also works with instruments which do not adhere to SCPI or IEEE 488.2.

The quick-BASIC command for executing a serial poll is "IBRSP()". Serial poll is mainly used to obtain a fast overview of the state of several instruments connected to the IEC bus.

Parallel Poll

In a parallel poll, up to eight instruments are simultaneously requested by the controller by means of a single command to transmit 1 bit of information each on the data lines, i.e., to set the data line allocated to each instrument to logically "0" or "1". By analogy to the SRE register which determines under which conditions an SRQ is generated, there is a parallel poll enable register (PPE) which is ANDed with the STB bit by bit as well considering bit 6. The results are ORed, the result is then sent (possibly inverted) as a response in the parallel poll of the controller. The result can also be queried without parallel poll by means of command "*IST".

The instrument first has to be set for the parallel poll using quick-BASIC command "IBPPC()". This command allocates a data line to the instrument and determines whether the response is to be inverted. The parallel poll itself is executed using "IBRPP()".

The parallel-poll method is mainly used in order to quickly find out after an SRQ which instrument has sent the service request if there are many instruments connected to the IEC bus. To this effect, SRE and PPE must be set to the same value. A detailed example as to the parallel poll is to be found in chapter 7, Program Examples.

Query by Means of Commands

Each part of every status register can be read by means of queries. The individual commands are indicated in the detailed description of the registers. What is returned is always a number which represents the bit pattern of the register queried. Evaluating this number is effected by the controller program.

Queries are usually used after an SRQ in order to obtain more detailed information on the cause of the SRQ.

Error-Queue Query

Each error state in the instrument leads to an entry in the error queue. The entries of the error queue are detailed plain-text error messages which can be looked at in the ERROR menu via manual control or queried via the IEC bus using command "SYSTem:ERRor?". Each call of "SYSTem:ERRor?" provides an entry from the error queue. If no error messages are stored there any more, the instrument responds with 0, "No error".

The error queue should be queried after every SRQ in the controller program as the entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program the error queue should be queried regularly since faulty commands from the controller to the instrument are recorded there as well.

Resetting Values of the Status Reporting System

Table 5-12 comprises the different commands and events causing the status reporting system to be reset. None of the commands, except for *RST and SYSTem:PRESet influences the functional instrument settings. In particular, DCL does not change the instrument settings.

Table 5-12 Resetting instrument functions

Event / Effect	Switching on supply voltage: Power-On-Status-Clear 0	Switching on supply voltage: Power-On-Status-Clear 1	DCL,SDC (Device Clear, Selected Device Clear)	*RST or SYSTem:PRESet	STATus:PRESet	*CLS
Clear STB,ESR	—	yes	—	—	—	yes
Clear SRE,ESE	—	yes	—	—	—	—
Clear PPE	—	yes	—	—	—	—
Clear EVENTt parts of the registers	—	yes	—	—	—	yes
Clear Enable parts of all OPERation and QUEStionable registers, Fill Enable parts of all other registers with "1".	—	yes	—	—	yes	—
Fill PTRansition parts with "1" , Clear NTRansition parts	—	yes	—	—	yes	—
Clear error queue	yes	yes	—	—	—	yes
Clear output buffer	yes	yes	yes	1)	1)	1)
Clear command processing and input buffer	yes	yes	yes	—	—	—

1) Every command being the first in a command line, i.e., immediately following a <PROGRAM MESSAGE TERMINATOR> clears the output buffer.

Contents - Chapter 6 "Remote Control - Description of Commands"

6 Description of Commands

Notation **6.1**

Common Commands **6.4**

ABORt Subsystem **6.7**

CALCulate Subsystem **6.7**

CALCulate:DELTamarker Subsystem 6.8

CALCulate:DLINe Subsystem 6.14

CALCulate:FEED Subsystem 6.17

CALCulate:FORMat and CALCulate:FSK Subsystems 6.18

CALCulate:LIMit Subsystem 6.20

CALCulate:MARKer Subsystem 6.28

CALCulate:MATH Subsystem 6.48

CALCulate:PEAKsearch - Subsystem 6.49

CALCulate:X and CALCulate:UNIT Subsystem 6.51

CALibration Subsystem **6.52**

DIAGnostic Subsystem **6.54**

DISPlay Subsystem **6.56**

FORMat Subsystem **6.67**

HCOPy Subsystem **6.69**

INITiate Subsystem **6.74**

INPut Subsystem **6.75**

INSTrument Subsystem **6.84**

MMEMory Subsystem **6.86**

OUTPut Subsystem **6.95**

SENSe Subsystem **6.96**

SENSe:AVERage Subsystem 6.96

SENSe:BANDwidth Subsystem 6.98

SENSe:CORRection-Subsystem 6.101

SENSe:DEMod, SENSe:AM, SENSe:FM and SENSe:PM Subsystems 6.106

SENSe:DETector Subsystem 6.108

SENSe:DDEMod Subsystem 6.110

SENSe:FILTer Subsystem 6.118

SENSe:FREQuency Subsystem 6.119

SENSe:POWer Subsystem 6.124

SENSe:ROSCillator Subsystem 6.125

SENSe:SCAN - Subsystem 6.126

SENSe:SWEep Subsystem 6.129

STATus Subsystem **6.133**

SYSTem Subsystem 6.143

TRACe Subsystem 6.149

TRIGger Subsystem 6.152

UNIT Subsystem 6.154

Alphabetical List of Commands 6.155

Table of Softkeys with IEC/IEEE-Bus Command Assignment 6.167

Basic Instrument - Signal Analysis Mode 6.167
FREQUENCY Key Group 6.167
LEVEL Key Group 6.169
INPUT Key 6.171
MARKER Key Group 6.172
LINES Key Group 6.176
TRACE Key Group 6.178
SWEEP Key Group 6.180

Basic Instrument - General Device Settings 6.183
DATA VARIATION Key Group 6.183
SYSTEM Key Group 6.184
CONFIGURATION Key Group 6.187
STATUS Key Group 6.190
HARDCOPY Key Group 6.191
MEMORY Key Group 6.192
USER Key 6.194

Receiver Mode 6.195
CONFIGURATION Key Group 6.195
FREQUENCY Key Group 6.199
LEVEL Key Group 6.199
INPUT Key 6.201
MARKER Key Group 6.202
LINES Key Group 6.205
TRACE Key Group 6.206
SWEEP Key Group 6.207

AF Demodulation Mode 6.209
CONFIGURATION Key Group 6.209

Vector Signal Analysis Mode 6.210
CONFIGURATION Key Group - Digital Demodulation 6.210
FREQUENCY Key Group 6.213
LEVEL Key Group 6.213
INPUT Key 6.214
MARKER Key Group 6.215
LINES Key Group 6.217
TRACE Key Group 6.218
SWEEP Key Group 6.219

6 Description of Commands

Notation

In the following sections, all commands implemented in the instrument are first listed in tables and then described in detail, separated according to the command system. The notation corresponds to the one of the SCPI standards to a large extent. The SCPI conformity information can be taken from the individual description of the commands.

Table of Commands

Command: In the command column, the table provides an overview of the commands and their hierarchical arrangement (see indentations).

Parameter: The parameter column indicates the requested parameters together with their specified range.

Unit: The unit column indicates the basic unit of the physical parameters.

Remark: In the remark column an indication is made on:
- whether the command does not have a query form,
- whether the command has only one query form
- whether this command is implemented only with a certain option of the instrument

Indentations

The different levels of the SCPI command hierarchy are represented in the table by means of indentations to the right. The lower the level is, the farther the indentation to the right is. Please observe that the complete notation of the command always includes the higher levels as well.

Example: `:SENSe:FREQuency:CENTer` is represented in the table as follows:

`:SENSe`	first level
`:FREQuency`	second level
`:CENTer`	third level

Individual description

In the individual description, the complete notation of the command is given. An example for each command, the *RST value and the SCPI information is written out at the end of the individual description.
The modes for which a command can be used are indicated by the following abbreviations:

A	Spectrum analysis
A-F	Spectrum analysis - frequency domain only
A-Z	Spectrum analysis - time domain only (zero span)
VA	Vector signal analysis - digital demodulation
AF-D	AF demodulation
R	Receiver

Upper/lower case notation

Upper/lower case letters serve to mark the long or short form of the key words of a command in the description (see Chapter 5). The instrument itself does not distinguish between upper and lower case letters.

Special characters

| A selection of key words with an identical effect exists for several commands. These key words are indicated in the same line, they are separated by a vertical stroke. Only one of these key words has to be indicated in the header of the command. The effect of the command is independent of which of the key words is indicated.

Example: `SENSe:FREQuency:CW|:FIXed`

The two following commands of identical meaning can be formed. They set the frequency of the constantly frequent signal to 1 kHz:

`SENSe:FREQuency:CW 1E3` = `SENSe:FREQuency:FIXed 1E3`

A vertical stroke in indicating the parameters marks alternative possibilities in the sense of "or". The effect of the command is different, depending on which parameter is entered.

Example: Selection of the parameters for the command

`INPut:COUPling AC | DC`

If parameter AC is selected, only the AC content is fed through, in the case of DC, the DC as well as the AC content.

[ ] Key words in square brackets can be omitted when composing the header (cf. Chapter 5, Section "Optional Keywords"). The full command length must be accepted by the instrument for reasons of compatibility with the SCPI standards.
Parameters in square brackets can optionally be incorporated in the command or omitted as well.

{ } Parameters in braces can optionally be incorporated in the command either not at all, once or several times.

Description of parameters Due to the standardization, the parameter section of SCPI commands consists always of the same syntactical elements. SCPI has specified a series of definitions therefore, which are used in the tables of commands. In the tables, these established definitions are indicated in angled brackets (<...>) and will be briefly explained in the following (see also Chapter 5, Section "Parameters").

<Boolean> This indication refers to parameters which can adopt two states, "on" and "off". The "off" state may either be indicated by the keyword **OFF** or by the numeric value 0, the "on" state is indicated by **ON** or any numeric value other than zero. Parameter queries are always returned the numeric value 0 or 1.

<numeric_value>

<num> These indications mark parameters which may be entered as numeric values or be set using specific keywords (character data).

The keywords given below are permitted:

MINimum This keyword sets the parameter to the smallest possible value.

MAXimum This keyword sets the parameter to the largest possible value.

DEFault This keyword is used to reset the parameter to its default value.

UP This keyword increments the parameter value.

DOWN This keyword decrements the parameter.

The numeric values associated to MAXimum/MINimum/DEFault can be queried by adding the corresponding keywords to the command. They must be entered following the quotation mark.

Example: `SENSe:FREQuency:CENTer? MAXimum`

returns the maximum possible numeric value of the center frequency as result.

<arbitrary block program data>

This keyword is provided for commands the parameters of which consist of a binary data block.

Common Commands

The common commands are taken from the IEEE 488.2 (IEC 625-2) standard. Same commands have the same effect on different devices. The headers of these commands consist of an asterisk "*" followed by three letters. Many common commands refer to the status reporting system which is described in detail in Chapter 5.

Command	Designation	Parameter	Remark
*CAL?	Calibration Query		query only
*CLS	Clear Status		no query
*ESE	Event Status Enable	0 to 255	
*ESR?	Standard Event Status Query	0 to 255	query only
*IDN?	Identification Query	<string>	query only
*IST?	Individual Status Query	0 to 255	query only
*OPC	Operation Complete		
*OPT?	Option Identification Query		query only
*PCB	Pass Control Back	0 to 30	no query
*PRE	Parallel Poll Register Enable	0 to 255	
*PSC	Power On Status Clear	0 | 1	
*RST	Reset		no query
*SRE	Service Request Enable	0 to 255	
*STB?	Status Byte Query		query only
*TRG	Trigger		no query
*TST?	Self Test Query		query only
*WAI	Wait to continue		no query

***CAL?**

CALIBRATION QUERY triggers a calibration of the instrument and subsequently query the calibration status. Any responses > 0 indicate errors.

***CLS**

CLEAR STATUS sets the status byte (STB), the standard event register (ESR) and the EVENt-part of the QUEStionable and the OPERation register to zero. The command does not alter the mask and transition parts of the registers. It clears the output buffer.

***ESE 0 to 255**

EVENT STATUS ENABLE sets the event status enable register to the value indicated. Query *ESE? returns the contents of the event status enable register in decimal form.

***ESR?**

STANDARD EVENT STATUS QUERY returns the contents of the event status register in decimal form (0 to 255) and subsequently sets the register to zero.

***IDN?**

IDENTIFICATION QUERY queries the instrument identification.

The instrument identification consists of the following elements which are separated by commas:

Manufacturer
Device (analyzer model)
Serial number of the instrument
Firmware version number
Example: "Rohde&Schwarz, FSET7, 825082/007, 2.09"

***IST?**

INDIVIDUAL STATUS QUERY returns the contents of the IST flag in decimal form (0 | 1). The IST flag is the status bit which is sent during a parallel poll (cf. Chapter 5).

***OPC**

OPERATION COMPLETE sets bit 0 in the event status register when all preceding commands have been executed. This bit can be used to initiate a service request (cf. Chapter 5).

***OPC?**

OPERATION COMPLETE QUERY writes message "1" into the output buffer as soon as all preceding commands have been executed (cf. Chapter 5).

***OPT?**

OPTION IDENTIFICATION QUERY queries the options included in the instrument and returns a list of the options installed. The options are separated from each other by means of commas.

Position	Option	
1 to 4		reserved
2	FSE-B4	Low Phase Noise & OCXO
5	FSE-B7	Vector Signal Analysis
6 to 12		reserved
13	FSE-B15	DOS option

Example: 0, FSE-B4, 0, 0, FSE-B7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,..

Note: *The standard FSET is equipped with options FSE-B4, FSE-B5 and FSE-B7 The options are indicated for reasons of compatibility to instrument family FSE.*

***PCB 0 to 30**

PASS CONTROL BACK indicates the controller address which the IEC-bus control is to be returned to after termination of the triggered action.

***PRE 0 to 255**

PARALLEL POLL REGISTER ENABLE sets parallel poll enable register to the value indicated. Query *PRE? returns the contents of the parallel poll enable register in decimal form.

*__PSC 0 | 1__

POWER ON STATUS CLEAR determines whether the contents of the ENABle registers is maintained or reset in switching on.

*PSC = 0 causes the contents of the status registers to be maintained. Thus a service request can be triggered in switching on in the case of a corresponding configuration of status registers ESE and SRE.

*PSC ≠ 0 resets the registers.

Query *PSC? reads out the contents of the power-on-status-clear flag. The response can be 0 or 1.

***RST**

RESET sets the instrument to a defined default status. The command essentially corresponds to pressing the [PRESET] key. The default setting is indicated in the description of the commands.

***SRE 0 to 255**

SERVICE REQUEST ENABLE sets the service request enable register to the value indicated. Bit 6 (MSS mask bit) remains 0. This command determines under which conditions a service request is triggered. Query *SRE? reads the contents of the service request enable register in decimal form. Bit 6 is always 0.

***STB?**

READ STATUS BYTE QUERY reads out the contents of the status byte in decimal form.

***TRG**

TRIGGER triggers a measurement. This command corresponds to `INITiate:IMMediate` (cf. Section "TRIGger subsystem", as well).

***TST?**

SELF TEST QUERY triggers all selftests of the instrument and outputs an error code in decimal form.

***WAI**

WAIT-to-CONTINUE only permits the servicing of the subsequent commands after all preceding commands have been executed and all signals have settled (cf. Chapter 5 and "*OPC" as well).

ABORt Subsystem

The ABORt subsystem contains the commands for aborting triggered actions. An action can be triggered again immediately after being aborted. All commands trigger events which is why they are not assigned any *RST value.

COMMAND	PARAMETERS	UNIT	COMMENT
ABORt	--	--	no query
HOLD	--	--	no query

:ABORt

This command aborts a current measurement and resets the trigger system.

Example: "ABOR;INIT:IMM"

Features: ***RST value:** 0
SCPI: conforming

Modes: R, A, VA

:HOLD

This command interrupts a current scan measurement.

Example: "HOLD"

Features: ***RST value:** -
SCPI: conforming

Modes: R

CALCulate Subsystem

The CALCulate subsystem contains commands for converting instrument data, transforming and carrying out corrections. These functions are carried out subsequent to data acquistion, i.e., following the SENSe subsystem.

In the split-screen representation, a distinction is made between CALCulate1 and CALCulate2:

CALCulate1 ≙ screen A;

CALCulate2 ≙ screen B

For setting REAL/IMAG PART in Vector Analyzer mode a distinction is also made between CALCulate3 and CALCulate4 in the split-screen representation:

CALCulate3 ≙ screen C;

CALCulate4 ≙ screen D

CALCulate:DELTamarker Subsystem

The CALCulate:DELTamarker subsystem checks the delta-marker functions in the instrument.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:DELTamarker<1 to 4>			
[:STATe]	<Boolean>	--	
:MODE	ABSolute|RELative		
:AOFF			no query
:TRACe	<numeric_value>	--	
:X	<numeric_value>	HZ | S | SYM	
:RELative?	--	--	query only
:Y?	--	--	query only
:MAXimum			
[:PEAK]	--	--	no query
:APEak	--	--	no query (vector signal analysis)
:NEXT	--	--	no query
:RIGHt	--	--	no query
:LEFT	--	--	no query
:MINimum			
[:PEAK]	--	--	no query
:NEXT	--	--	no query
:RIGHt	--	--	no query
:LEFT	--	--	no query
:FUNCtion			
:FIXed			
[:STATe]	<Boolean>		
:RPOint			
:Y	<numeric_value>	DBM	
:OFFSet	<numeric_value>	DB	
:X	<numeric_value>	HZ |S | SYM	
:PNOise			
[:STATe]	<Boolean>		
:RESult?	--	--	query only
:STEP			
[:INCRement]	<numeric_value>	HZ |S | SYM	
:AUTO	<Boolean>	--	

:CALCulate<1|2>:DELTamarker<1 to 4>[:STATe] ON | OFF

This command switches on or off the selected delta marker. If no indication is made, delta marker 1 is selected automatically.

Example: ":CALC:DELT3 ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:DELTamarker<1 to 4>:MODE ABSolute | RELative

This command switches over between relative and absolute input of frequency of the delta marker.

Example: ":CALC:DELT:MODE ABS"

Features: *RST value: REL
SCPI: device-specific

Modes: R, A, VA

In the RELative mode, the frequency of the delta marker is programmed relative to the reference marker. In the ABSolute mode, the frequency is defined by the absolute values.

:CALCulate<1|2>:DELTamarker<1 to 4>:AOFF

This command switches off all active delta markers.

Example: ":CALC:DELT:AOFF"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:DELTamarker<1 to 4>:TRACe 1 to 4

This command assigns the selected delta marker to the indicated measuring curve.

Example: ":CALC:DELT3:TRAC 2"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:DELTamarker<1 to 4>:X 0 to MAX (frequency | sweep time | symbols)

This command positions the selected delta marker to the indicated frequency (span > 0) or time (span = 0). The query always returns the absolute value of frequency or time.

Example: ":CALC:DELT:X 10.7MHz"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

The SYM unit is only valid in Vector Signal Analysis mode.

:CALCulate<1|2>:DELTamarker<1 to 4>:X:RELative?

This command queries the frequency (span > 0) or time (span = 0) of the selected delta marker relative to the reference marker.

Example: ":CALC:DELT:X:REL?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:DELTamarker<1 to 4>:Y?

This command queries the value of the selected marker.

Example: ":CALC:DELT:Y?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

In complex presentations (vector signal analysis - polar diagrams), the real and the imaginary component as well as magnitude and phase are output separated by a comma.

:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum[:PEAK]

This command positions the delta marker to the current maximum value in the trace memory.

Example: ":CALC:DELT:MAX"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:APEak

This command positions the delta marker to the maximum absolute value of the trace.

Example: ":CALC:DELT:MAX:APE"

Features: *RST value: -
SCPI: device-specific

Modes: VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:NEXT

This command positions the delta marker to the next smaller maximum value in the trace memory.

Example: ":CALC:DELT:MAX:NEXT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:RIGHt

This command positions the delta marker to the next smaller maximum value to the right of the current value (i.e., in ascending X direction) in the trace memory.

Example: ":CALC:DELT:MAX:RIGH"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:LEFT

This command positions the delta marker to the next smaller maximum value to the left of the current value (i.e., in descending X direction) in the trace memory.

Example: ":CALC:DELT:MAX:LEFT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum[:PEAK]

This command positions the delta marker to the current minimum value in the trace memory.

Example: ":CALC:DELT:MIN"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum:NEXT

This command positions the delta marker to the next higher minimum value in the trace memory.

Example: ":CALC:DELT:MIN:NEXT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum:RIGHt

This command positions the delta marker to the next higher minimum value to the right of the current value (ie in ascending X direction).

Example: ":CALC:DELT:MIN:RIGH"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum:LEFT

This command positions the delta marker to the next higher minimum value to the left of the current value (ie in descending X direction).

Example: ":CALC:DELT:MIN:LEFT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed[:STATe] ON | OFF

This command switches the relative measurement to a fixed reference value on or off.

Example: ":CALC:DELT:FUNC:FIX ON"

Features: *RST value: OFF
SCPI: device-specific.

Modes: R, A, VA

The reference value is independent of the current trace.

:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed:RPOint:Y <numeric_value>

This command defines a new fixed reference value for the relative measurement.

Example: ":CALC:DELT:FUNC:FIX:RPO:Y -10dBm"

Features: *RST value: - (FUNction:FIXed[:STATe] is set to OFF)
SCPI: device-specific

Modes: A, VA

The reference value is independent of the current trace.

:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed:RPOint:Y:OFFSet <numeric_value>

This command defines an additional level offset for the relative measurement.

Example: ":CALC:DELT:FUNC:FIX:RPO:Y:OFFS 10dB"

Features: *RST value: 0 dB
SCPI: device-specific

Modes: A, VA

The level offset is included in the output of the level value.

:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed:RPOint:X <numeric_value>

This command defines the new fixed reference frequency, time or symbols for the relative measurement.

Example: ":CALC:DELT:FUNC:FIX:RPO:X 10.7MHz"

Features: *RST value: - (FUNction:FIXed[:STATe] is set to OFF)
SCPI: device-specific

Mode: A

The reference value is independent of the current trace. With span = 0, the reference time, otherwise the reference frequency is defined.

:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:PNOise[:STATe] ON | OFF

This command switches the measurement of the phase noise on or off.

Example: ":CALC:DELT:FUNC:PNO ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A

When the phase noise is measured, the correction values for the bandwidth and the log amplifier are automatically considered. The measurement uses the reference values defined by `FUNCtion:FIXed:RPOint:X` or `:Y`.

:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:PNOise:RESult?

This command queries the result of the phase noise measurement.

Example: ":CALC:DELT:FUNC:PNO:RES?"

Features: *RST value: -
SCPI: device-specific

Mode: A

This command is only a query which is why it is not assigned an *RST value.

:CALCulate<1|2>:DELTamarker<1 to 4>:STEP[:INCRement] <numeric_value>

This command defines the delta marker step width.

Example: ":CALC:DELT:STEP 10kHz" (frequency domain)
":CALC:DELT:STEP 5ms" (time domain)

Features: *RST value: - (STEP is set to AUTO)
SCPI: device-specific

Mode: A

:CALCulate<1|2>:DELTamarker<1 to 4>:STEP:AUTO ON | OFF

This command switches the automatic adaptation of the marker step width on or off.

Example: ":CALC:DELT:STEP:AUTO OFF"

Features: *RST value: ON
SCPI: device-specific

Mode: A

With AUTO ON, the step width is 10% of the span.

CALCulate:DLINe Subsystem

The CALCulate:DLINe subsystem checks the display lines in the instrument, i.e., the level, frequency and time lines (depending on the X-axis) as well as threshold and reference lines.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:DLINe<1|2>	<numeric_value>	DBM | DB | DEG | RAD | S | HZ | PCT	
:STATe	<Boolean>		
:THReshold	<numeric_value>	DBM | DB | DEG | RAD | S | HZ | PCT	
:STATe	<Boolean>		
:CTHReshold	<numeric_value>	DBM | DB | DEG | RAD | S | HZ | PCT	
:STATe	<Boolean>		
:RLINe	<numeric_value>	DBM | DB | DEG | RAD | S | HZ | PCT	
:STATe	<Boolean>		
:FLINe<1|2>	<numeric_value>	HZ	
:STATe	<Boolean>		
:TLINe<1|2>	<numeric_value>	S | SYM	
:STATe	<Boolean>		

:CALCulate<1|2>:DLINe<1|2> MINimum to MAXimum (depending on current unit)

This command defines the position of the display line.

Example: `":CALC:DLIN -20dBm"`

Features: *RST value: - (STATe to OFF)
SCPI: device-specific

Modes: R, A, VA

The display lines mark the given level in the display.

The units DEG, RAD, S, and HZ are only valid in operating mode Vector Signal Analysis.

:CALCulate<1|2>:DLINe<1|2>:STATe ON | OFF

This command switches the display line on or off.

Example: `":CALC:DLIN2:STAT OFF"`

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:THReshold MINimum to MAXimum (depending on current unit)

This command defines the position of the thresholds.

Example: `":CALC:THR -82dBm"`

Features: *RST value: - (STATe to OFF)
SCPI: device-specific

Modes: R, A, VA

For marker scan functions MAX PEAK, NEXT PEAK etc., the threshold serves as the lowest limit for maximum or minimum search.

The units DEG, RAD, S, and HZ are only valid in operating mode Vector Signal Analysis.

:CALCulate<1|2>:THReshold:STATe ON | OFF

This command switches the threshold on or off.

Example: ":CALC:THR:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:CTHReshold MINimum to MAXimum (depending on the current unit)

This command defines the position of a threshold line (base line), below which all measured values are cleared.

Example: ":CALC:CTHR -82dBm"

Features: *RST value: - (STATe to OFF)
SCPI: device-specific

Mode: R, A, VA

The units DEG, RAD, S, and HZ are only valid in operating mode Vector Signal Analysis.

:CALCulate<1|2>:CTHReshold:STATe ON | OFF

This command is for switching on or off the threshold line (base line), below which all measured values are cleared.

Example: ":CALC:CTHR:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: R, A, VA

:CALCulate<1|2>:RLINe MINimum to MAXimum (depending on the current unit)

This command defines the position of the reference line.

Example: ":CALC:RLIN -10dBm"

Features: *RST value: - (STATe to OFF)
SCPI: device-specific

Modes: R, A, VA

The reference line serves as a reference for the arithmetic operation of traces.

The units DEG, RAD, S, and HZ are only valid in operating mode Vector Signal Analysis.

:CALCulate<1|2>:RLINe:STATe ON | OFF

This command switches the reference line on or off.

Example: ":CALC:RLIN:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:FLINe<1|2> 0 Hz to f_{max}

This command defines the position of the frequency lines.

Example: ":CALC:FLIN2 120MHz"

Features: *RST value: - (STATe to OFF)
SCPI: device-specific

Modes: R, A-F, VA

The frequency lines mark the given frequencies in the display. Frequency lines are only valid for a SPAN >0.

:CALCulate<1|2>:FLINe<1|2>:STATe ON | OFF

This command switches the frequency line on or off.

Example: ":CALC:FLIN2:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A-F, VA

:CALCulate<1|2>:TLINe<1|2> 0 to 1000s

This command defines the position of the time lines.

Example: ":CALC:TLIN 10ms"

Features: *RST value: - (STATe to OFF)
SCPI: device-specific

Modes: A-Z, VA

The time lines mark the given times in the display. Time lines are only valid for a SPAN = 0.

:CALCulate<1|2>:TLINe<1|2>:STATe ON | OFF

This command switches the time line on or off.

Example: ":CALC:TLIN2:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: A-Z, VA

CALCulate:FEED Subsystem

The CALCulate:FEED subsystem selects the measured data in operating mode vector signal analysis.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2> :FEED	<string>		Vector Signal Analysis no query

:CALCulate<1|2>:FEED <string>

This command selects the measured data that are to be displayed.

Parameter: <string>::= ‘XTIM:DDEM:MEAS’ |
‘XTIM:DDEM:REF’ |
‘XTIM:DDEM:ERR:MPH’ |
‘XTIM:DDEM:ERR:VECT’ |
‘XTIM:DDEM:SYMB’ |
‘TCAP’

Example: `":CALC:FEED ‘XTIM:DDEM:SYMB’"`

Features: *RST value: ‘XTIM:DDEM:MEAS’
SCPI: conforming

Mode: VA

The string parameters have the following meaning:

‘XTIM:DDEM:MEAS’	Test signal (filtered, synchronized to symbol clock)
‘XTIM:DDEM:REF’	Reference signal (internally generated from demodulated test signal)
‘XTIM:DDEM:ERR:MPH’	Error signal (magnitude and phase error)
‘XTIM:DDEM:ERR:VECT’	Vector error signal
‘XTIM:DDEM:SYMB’	Symbol table (demodulated bits and table with modulation errors)
‘TCAP’	Test signal in capture buffer

CALCulate:FORMat and CALCulate:FSK Subsystems

The CALCulate:FORMat and CALCulatr:FSK subsystems determine further processing and conversion of measured data in operating mode vector signal analysis.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2> :FORMat	MAGNitude | PHASe | UPHase | RIMag | FREQuency | IEYE | QEYE | TEYE | FEYE | COMP | CONS		Vector Signal Analysis
:FSK :DEViation :REFerence	<numeric_value>	HZ	Vector Signal Analysis

:**CALCulate<1|2>:FORMat** MAGNitude | PHASe | UPHase | RIMag | FREQuency | IEYE | QEYE | TEYE | FEYE | COMP | CONS

This command defines the display of the traces.

Example: ":CALC:FORM CONS"

Features: *RST value: MAGNitude
SCPI: conforming

Mode: VA

The availability of the parameters depends on the selected data (see command :CALCulate:FEED).

Available for selection measurement signal, reference signal and modulation error (CALCulate:FEED 'XTIM:DDEM:MEAS','XTIM:DDEM:REF) ,'XTIM:DDEM:ERR:MPH'):

MAGNitude	Display of the magnitude in the time domain (only available for settings :CALCulate:FEED 'XTIM:DDEM:ERR:MPH' (error signal) or 'XTIM:DDEM:MEAS' (measurement signal) or 'XTIM:DDEM:REF' (reference signal)
PHASe | UPHase	Display of the phase in the time domain with or without ("unwrapped") limitation to ±180°
RIMag	Display of the time characteristic of inphase and quadrature component
FREQuency	Display of the frequency response in the time domain
COMP	Display of the polar vector diagram (complex)
CONS	Display of the polar vector diagaram (constellation)

Available for selection measurement signal and reference signal (CALCulate:FEED 'XTIM:DDEM:MEAS','XTIM:DDEM:REF):

IEYE | QEYE	Eye diagram of the inphase or quadrature component
TEYE	Display of the trellis diagram
FEYE	Eye diagram of FSK modulation

:CALCulate<1|2>:FSK:DEViation:REFerence <numeric_value>

This command defines the reference value of the frequency deviation for FSK modulation.

Example: ":CALC:FSK:DEV:REF 20kHz"

Features: *RST value: -
SCPI: device-specific

Mode: VA

CALCulate:LIMit Subsystem

The CALCulate:LIMit subsystem comprises the limit lines and the corresponding limit checks. Limit lines can be defined as upper and lower limit lines. The individual values of the limit lines correspond to the values of the X-axis (CONTrol) which have to have the same number.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:LIMit<1 to 8>			
:ACTive?			Query only
:TRACe	<numeric_value>		
:STATe	<Boolean>	--	
:UNIT	DBM | DBPW | DBPT | WATT | DBUV | DBMV | VOLT | DBUA | AMPere | DB | DBUV_MHZ | DBMV_MHZ | DBUA_MHZ | DBUV_M | DBUA_M | DBUV_MMHZ | DBUA_MMHZ | DEG | RAD | S | HZ | PCT | UNITLESS		
CATalog?			Query only
:CONTrol	<numeric_value>,<numeric_value>..	HZ | S | SYM	
[:DATA]	FREQuency | TIME		
:DOMain	<numeric_value>	HZ | S | SYM	
:OFFSet	RELative | ABSolute		
:MODE			
:UNIT	S | SYM		
[:TIME]	<numeric_value>	HZ | S | SYM	Vector Signal Analysis
:SHIFt	LINear | LOGarithmic		
:SPACing			
:UPPer	<numeric_value>,<numeric_value>..	DBM | DB | DEG | RAD | S | HZ | PCT	
[:DATA]	<Boolean>	--	
:STATe	<numeric_value>	DB | DEG | RAD | S | HZ | PCT	
:OFFSet	<numeric_value>	DB| DEG | RAD | S | HZ | PCT	
:MARGin	RELative | ABSolute	--	
:MODE	<numeric_value>	DB | DEG | RAD| S | HZ | PCT	
:SHIFt	LINear | LOGarithmic		
:SPACing			
:LOWer	<numeric_value>,<numeric_value>..	DBM | DB | DEG | RAD | S | HZ | PCT	
[:DATA]	<Boolean>	--	
:STATe	<numeric_value>	DB| DEG | RAD | S | HZ | PCT	
:OFFSet	<numeric_value>	DB| DEG | RAD | S | HZ | PCT	
:MARGin	RELative | ABSolute	--	
:MODE	<numeric_value>	DB | DEG | RAD | S | HZ | PCT	
:SHIFt	LINear | LOGarithmic		
:SPACing			
:FAIL?		--	query only
:CLEar	--	--	
[:IMMediate]			no query
:COMMent	<string>		
:COPY	1 to 8 | < name>		
:NAME	<string>		
:DELete			

:CALCulate<1|2>:LIMit<1 to 8>:ACTive?

This command queries the names of all activated limit lines. The names are output in alphabetical order. If no limit line is activated, an empty string will be output. The numeric suffixes in `CALCulate<1|2>` and `LIMit<1 to 8>` are not significant.

Example: `":CALC:LIM:ACT?"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:LIMit<1 to 8>:TRACe 1 to 4

This command assigns a trace to a limit line.

Example: `":CALC:LIM2:TRAC 2"`

Features: *RST value: 1
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:LIMit<1 to 8>:STATe ON | OFF

This command switches the limit check for the selected limit line on or off. With limit check switched off, the limit line is disabled.

Example: `":CALC:LIM:STAT ON"`

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

In analyzer and vector analyzer mode, the result of the limit check can be queried with `CALCulate:LIMit:FAIL?`.

:CALCulate<1|2>:LIMit<1 to 8>:UNIT DBM | DBPW | DBPT | WATT | DBUV | DBMV | VOLT |DBUA | AMPere | DB | DBUV_MHZ | DBMV_MHZ | DBUA_MHZ | DBUV_M | DBUA_M | DBUV_MMHZ | DBUA_MMHZ | UNITLESS|

This command defines the unit of the selected limit line.

Example: `":CALC:LIM:UNIT DBUV"`

Features: *RST value: DBM
SCPI: device-specific

Modes: R, A, VA

In receiver mode the units DBM, DBUV, DBUA, DBPW, DBPT, DBUV_M, and DBUA_M are available

In analyzer mode all units are available except for DEG, RAD, S and HZ.

DBUV_MHZ and DBUA_MHZ denote the units DBUV/MHZ or DBUA/MHZ.
Upon selection of the unit DB the limit line is automatically switched to the relative mode. For units different from DB the limit line is automatically switched to the absolute mode.

The units DEG, RAD, S, HZ are available in the vector analysis mode only.

:CALCulate<1|2>:LIMit:CATalog?

This command reads out the names of all limit lines stored on the harddisk.

Syntax of output format:

<Sum of file lengths of all subsequent files>,<free memory on hard disk>,
<1st file name>,,<1st file length>,<2nd file name>,,<2nd file length>,....,<nth file name>,
<nth file length>

Example: `":CALC:LIM:CAT?"`

Feature: *RST value: -
SCPI: device-specific

Mode: R, A, VA

:CALCulate<1|2>:LIMit<1 to 8>:CONTrol[:DATA] <numeric_value>,<numeric_value>..

This command defines the X-axis values (frequencies or times) of the upper or lower limit lines.

Example: `":CALC:LIM:CONT 1MHz,30MHz,300MHz,1GHz"`

Features: *RST value: - (LIMit:STATe is set to OFF)
SCPI: conforming

Modes: R, A, VA

The number of values for the CONTrol axis and the corresponding UPPer- and/or LOWer limit lines have to be identical. The following units are available:

receiver	HZ
spectrum analysis	HZ | S
vector analysis	HZ | S | SYM.

:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:DOMain FREQuency | TIME

This command defines the X-axis in the frequency or time domain.

Example: `":CALC:LIM:CONT:DOM TIME"`

Features: *RST value: FREQuency
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:OFFSet <numeric_value>

This command defines an offset for the X-axis value of the selected relative limit line in the frequency or time domain.

Example: `":CALC:LIM:CONT:OFFS 100us"`

Features: *RST value: 0
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:MODE RELative | ABSolute

This command selects the relative or absolute scaling for the X-axis of the selected limit line.

Example: ":CALC:LIM:CONT:MODE REL"

Features: *RST value: ABSolute
SCPI: device-specific

Modes: A, VA

Upon selection of RELative, the unit is switched to DB.

:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:UNIT[:TIME] S | SYM

This command defines the unit of the x-axis scaling of limit lines.

Example: ":CALC:LIM:CONT:UNIT SYM"

Features: *RST value: S
SCPI: device-specific

Mode: VA

:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:SHIFt <numeric_value>

This command shifts a limit line which has been specified for relative frequencies or times (X-axis).

Example: ":CALC:LIM:CONT:SHIF 50kHz"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:SPACing LINear | LOGarithmic

This command makes a selection between linear and logarithmic interpolation for determining the limit line from the frequency points.

Example: ":CALC:LIM:CONT:SPAC LIN"

Features: *RST value: LIN
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:LIMit<1 to 8>:UPPer[:DATA] <numeric_value>,<numeric_value>..

This command defines the values for the upper limit lines.

Example: ":CALC:LIM:UPP -10,0,0,-10"

Features: *RST value: - (LIMit:STATe is set to OFF)
SCPI: conforming

Modes: R, A, VA

The number of values for the CONTrol axis and the corresponding UPPer limit line have to be identical. The unit must be identical with the unit selected by command CALC:LIM:UNIT. If the measured values exceed the UPPer limit line in analyzer or vector analyzer mode, the limit check signals errors. The unit must be identical with the unit selected by CALC:LIM:UNIT. The units DEG, RAD, S, and HZ are available in the vector signal analysis mode only.

:CALCulate<1|2>:LIMit<1 to 8>:UPPer:STATe ON | OFF

This command defines the selected limit line as upper limit line.

Example: ":CALC:LIM:UPPer:STAT ON"

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

The limit check is switched on with command CALCulate:LIMit:STATe ON. In analyzer and vector analyzer mode, the result of the limit check can be queried with CALCulate:LIMit<1 to 8>:FAIL?.

:CALCulate<1|2>:LIMit<1 to 8>:UPPer:OFFSet <numeric_value>

This command defines an offset for the Y-axis of the selected relative upper limit line.

Example: ":CALC:LIM:UPP:OFFS 3dB"

Features: *RST value: 0
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:UPPer:MARGin <numeric_value>

This command defines the margin of the selected upper limit line.

Example: ":CALC:LIM:UPP:MARG 10dB"

Features: *RST value: 0
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:UPPer:MODE RELative | ABSolute

This command selects the relative or absolute scaling for the Y-axis of the selected upper limit line.

Example: ":CALC:LIM:UPP:MODE REL"

Features: *RST value: ABSolute
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:UPPer:SHIFt <numeric_value>

This command shifts a limit line, which has relative values for the Y-axis (levels or linear units such as volt).

Example: ":CALC:LIM:UPP:SHIF 20dB"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:LIMit<1 to 8>:UPPer:SPACing LINear | LOGarithmic

This command makes a selection between linear and logarithmic interpolation for the upper limit line.

Example: ":CALC:LIM:UPP:SPAC LIN"

Features: *RST value: LIN
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:LIMit<1 to 8>:LOWer[:DATA] <numeric_value>,<numeric_value>..

This command defines the values for the selected lower limit line.

Example: ":CALC:LIM:LOW -30,-40,-40,-30"

Features: *RST value: - (LIMit:STATe is set to OFF)
SCPI: conforming

Modes: R, A, VA

The number of values for the CONTrol axis and the corresponding LOWer limit line have to be identical.If the measured values violate the LOWer limit line in analyzer or vector analyzer mode, the limit check signals errors. The unit must be identical with the unit selected by CALC:LIM:UNIT.

The units DEG, RAD, S, and HZ are available in the vector signal analysis mode only.

:CALCulate<1|2>:LIMit<1 to 8>:LOWer:STATe ON | OFF

This command defines the selected limit line as lower limit line.

Example: ":CALC:LIM:LOWer:STAT ON"

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

The limit check is switched on with command CALCulate:LIMit:STATe ON. In analyzer and vector analyzer mode, the result of the limit check can be queried with CALCulate:LIMit:FAIL?.

:CALCulate<1|2>:LIMit<1 to 8>:LOWer:OFFSet <numeric_value>

This command defines an offset for the Y-axis of the selected relative lower limit line.

Example: ":CALC:LIM:LOW:OFFS 3dB"

Features: *RST value: 0
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:LOWer:MARGin <numeric_value>

This command defines the margin of the selected lower limit line.

Example: ":CALC:LIM:LOW:MARG 10dB"

Features: *RST value: 0
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:LOWer:MODE RELative | ABSolute

This command selects the relative or absolute scaling for the Y-axis of the selected lower limit line.

Example: ":CALC:LIM:LOW:MODE REL"

Features: *RST value: ABSolute
SCPI: device-specific

Modes: A, VA

:CALCulate<1|2>:LIMit<1 to 8>:LOWer:SHIFt <numeric_value>

This command shifts a limit line, which has relative values for the Y-axis (levels or linear units such as volt).

Example: ":CALC:LIM:LOW:SHIF 20dB"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:LIMit<1 to 8>:LOWer:SPACing LINear | LOGarithmic

This command makes a selection between linear and logarithmic interpolation for the lower limit line.

Example: ":CALC:LIM:LOW:SPAC LIN"

Features: *RST value: LIN
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:LIMit<1 to 8>:FAIL?

This command queries the result of the limit check.

Example: ":CALC:LIM:FAIL?"

Features: *RST value: -
SCPI: conforming

Modes: A, VA

The result of the limit check responds with 0 in case of PASS and with 1 in case of FAIL.

:CALCulate<1|2>:LIMit<1 to 8>:CLEar[:IMMediate]

This command deletes the result of the current limit check.

Example: ":CALC:LIM:CLE"

Features: *RST value: -
SCPI: conforming

Modes: A, VA

This command is an event which is why it is not assigned an *RST value.

:CALCulate<1|2>:LIMit<1 to 8>:COMMent <string>

This command defines a comment for the limit line selected.

Example: ":CALC:LIM:COMM 'Upper limit for spectrum'"

Features: *RST value: blank comment
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:LIMit<1 to 8>:COPY 1 to 8 | <name>

This command copies one limit line onto another one.

Parameter: 1 to 8 ::= number of the new limit line or, alternatively:
<name> ::= name of the new limit line given as a string

Example: ":CALC:LIM1:COPY 2"
":CALC:LIM1:COPY 'GSM2'"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

The name of the limit line may contain a maximum of 8 characters. This command is an "event" which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:LIMit<1 to 8>:NAME <name of limit line>

This command assigns a name to a limit line numbered 1 to 8. If it doesn't exist previously, a limit line with this name is created.

Example: ":CALC:LIM1:NAME 'GSM1'"

Features: *RST value: 'REM1' to 'REM8' for lines 1 to 8
SCPI: device-specific

Modes: R, A, VA

The name of the limit line may contain a maximum of 8 characters.

:CALCulate<1|2>:LIMit<1 to 8>:DELete

This command deletes the limit line selected.

Examples: ":CALC:LIM1:DEL"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is an "event" which is why it is not assigned an *RST value and has no query.

CALCulate:MARKer Subsystem

The CALCulate:MARKer subsystem checks the marker functions in the instrument.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:MARKer<1 to 4>			
[:STATe]	<Boolean>	--	
:AOFF			no query
:TRACe	<numeric_value>	--	
:X	<numeric_value>	HZ | S | SYM	
:SLIMits			
[:STATe]	<Boolean>		
:COUNt	<Boolean>	--	
:RESolution	<numeric_value>	HZ	
:FREQuency?	--	--	query only
:COUPled			
[:STATe]	<Boolean>		
:SCOupled			
[:STATe]	<Boolean>		
:LOEXclude	<Boolean>		
:Y?	--	--	query only
:MAXimum			
[:PEAK]	--	--	no query
:APEak	--	--	no query, Vector Signal Analysis
:NEXT	--	--	no query
:RIGHt	--	--	no query
:LEFT	--	--	no query
:MINimum			
[:PEAK]	--	--	no query
:NEXT	--	--	no query
:RIGHt	--	--	no query
:LEFT	--	--	no query
:STEP			
[:INCRement]	<numeric_value>	HZ | S | SYM	
:AUTO	<Boolean>	--	
:PEXCursion	<numeric_value>	DB	
:READout	MPHase | RIMaginary		Vector Signal Analysis
:FUNCtion			
:NDBDown	<numeric_value>	DB	
:STATe	<Boolean>		
:RESult?	--	--	query only
:FREQuency?	--	--	query only
:ZOOM	<numeric_value>	HZ	no query
:NOISe			
[:STATe]	<Boolean>		
:RESult?	--	--	query only
:SFACtor	<expr>		
:STATe	<Boolean>		
:RESult?	--	--	query only
:FREQuency?	--	--	query only

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:MARKer			
:FUNCtion			
:STRack	<Boolean>		
[:STATe]			
:DDEMod			Vector Signal Analysis
:RESult?	MERM | MEPK | MEPS | PERM | PEPK | PEPS |EVRM | EVPK | EVPS | IQOF | IQIM | ADR | FERR | DEV | FSRM | FSPK | FSPS | RHO |FEPK | DTTS		query only
:POWer			
:SELect	OBANdwidth | OBWidth		
:RESult?	OBANdwidth | OBWidth		query only
[:STATe]	OFF		
:SUMMary			
:STATe	<Boolean>		
:MAXimum			Vector Signal Analysis
[:STATe]	<Boolean>		
:RESult?			query only
:AVERage			
:RESult?			query only
:PHOLd			
RESult?			query only
:PPEak			Vector Signal Analysis
[:STATe]	<Boolean>		
:RESult?			query only
:AVERage			
:RESult?			query only
:PHOLd			
RESult?			query only
:MPEak			Vector Signal Analysis
[:STATe]	<Boolean>		
:RESult?			query only
:AVERage			
:RESult?			query only
:PHOLd			
RESult?			query only
:MIDDle			Vector Signal Analysis
[:STATe]	<Boolean>		
:RESult?			query only
:AVERage			
:RESult?			query only
:PHOLd			
RESult?			query only

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:MARKer			
:FUNCtion			
:SUMMary			
:RMS			
[:STATe]	<Boolean>		
:RESult?			query only
:AVERage			
:RESult?			query only
:PHOLd			
RESult?			query only
:MEAN			
[:STATe]	<Boolean>		
:RESult?			query only
:AVERage			
:RESult?			query only
:PHOLd			
RESult?			query only
:AOFF			no query
:PHOLd	<Boolean>		
:AVERage	<Boolean>		
:CENTer			no query
:CSTep			no query
:STARt			no query
:STOP			no query
:MSTep			no query
:REFerence			no query

:CALCulate<1|2>:MARKer<1 to 4>[:STATe] ON | OFF

This command switches on or off the currently selected marker. If no indication is made, marker 1 is selected automatically.

Example: ":CALC:MARK3 ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:MARKer<1 to 4>:AOFF

This command switches off all active markers.

Example: ":CALC:MARK:AOFF"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:TRACe 1 to 4

This command assigns the selected marker (1 to 4) to the indicated test curve.

Example: ":CALC:MARK3:TRAC 2"

Features: *RST value -
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:MARKer<1 to 4>:X 0 to MAX (frequency | sweep time | symbols)

This command positions the selected marker to the indicated frequency (span > 0) or time (span = 0).

Example: ":CALC:MARK:X 10.7MHz"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

Available units: receiver: Hz; analyzer: HZ | S; vector analyzer: HZ | S | SYM

:CALCulate<1|2>:MARKer<1 to 4>:X:SLIMits[:STATe] ON | OFF

This command switches between a limited (ON) and unlimited (OFF) search range.

Example: ":CALC:MARK:X:SLIM ON"

features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:MARKer<1 to 4>:COUNt ON | OFF

This command switches on or off the frequency counter at the marker position.

Example: ":CALC:MARK:COUN ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A

:CALCulate<1|2>:MARKer<1 to 4>:COUNt:RESolution 0.1 | 1 | 10 | 100 | 1000 | 10000 Hz

This command specifies the resolution of the frequency counter.

Example: ":CALC:MARK:COUN:RES 1kHz"

Features: *RST value: 1kHz
SCPI: device-specific

Mode: A

The numeric suffix in MARKer<1 to 4> is not significant.

:CALCulate<1|2>:MARKer<1 to 4>:COUNt:FREQuency?

This command queries the result of the frequency counter.

Example: ":CALC:MARK:COUN:FREQ?"

Features: *RST value: -
SCPI: device-specific

Mode: A

This command is only a query and thus has no *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:COUPled[:STATe] ON | OFF

This command switches the coupling of markers on or off.

Example: ":CALC:MARK:COUP ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: R, VA

The numeric suffix in MARKer<1 to 4> is not significant.

:CALCulate<1|2>:MARKer<1 to 4>:SCOupled[:STATe] ON | OFF

This command switches the coupling of the receiver frequency at the marker frequency on or off.

Example: ":CALC:MARK:SCO ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: R

The numeric suffix in MARKer<1 to 4> is not significant.

:CALCulate<1|2>:MARKer<1 to 4>:LOEXclude ON | OFF

This command switches the local oscillator suppression on or off.

Example: ":CALC:MARK:LOEX ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A-F

The numeric suffixes 1|2 and 1 to 4 are not significant.

:CALCulate<1|2>:MARKer<1 to 4>:Y?

This command queries the selected marker value.

Example: ":CALC:MARK:Y?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALCulate<1|2>:MARKer<1 to 4>:MAXimum[:PEAK]

This command positions the marker to the current maximum value in the trace memory.

Example: ":CALC:MARK:MAX"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:APEak

This command positions the marker to the maximum absolute value of the trace.

Example: ":CALC:MARK:MAX:APE"

Features: *RST value: -
SCPI: device-specific

Mode: VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:NEXT

This command positions the marker to the next lower maximum value in the trace memory.

Example: ":CALC:MARK:MAX:NEXT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:RIGHt

This command positions the marker to the next smaller maximum value to the right of the current value (i.e., in ascending X direction) in the trace memory.

Example: ":CALC:MARK:MAX:RIGH"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:LEFT

This command positions the marker to the next smaller maximum value to the left of the current value (i.e., in descending X direction) in the trace memory.

Example: ":CALC:MARK:MAX:LEFT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MINimum[:PEAK]

This command positions the marker to the current minimum value in the trace memory.

Example: ":CALC:MARK:MIN"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MINimum:NEXT

This command positions the marker to the next higher minimum value in the trace memory.

Example: ":CALC:MARK:MIN:NEXT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MINimum:RIGHt

This command positions the marker to the next higher minimum value to the right of the current value (ie in ascending X direction).

Example: ":CALC:MARK:MIN:RIGH"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:MINimum:LEFT

This command positions the marker to the next higher minimum value to the left of the current value (ie in descending X direction).

Example: ":CALC:MARK:MIN:LEFT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

is command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:STEP[:INCRement] <numeric_value>

This command defines the marker step width.

Example: ":CALC:MARK:STEP 10kHz" (frequency domain)
:CALC:MARK:STEP 5ms" (time domain)

Features: *RST value: - (STEP is set to AUTO)
SCPI: device-specific

Mode: A

This command sets STEP:AUTO to OFF. The numeric suffix in MARKer<1 to 4> is not significant.

:CALCulate<1|2>:MARKer<1 to 4>:STEP:AUTO ON | OFF

This command switches the automatic adaptation of the marker step width on or off.

Example: ":CALC:MARK:STEP:AUTO OFF"

Features: *RST value: ON
SCPI: device-specific

Mode: A

With AUTO ON, the step width is 10% of the span. The numeric suffix in MARKer<1 to 4> is not significant.

:CALCulate<1|2>:MARKer<1 to 4>:PEXCursion <numeric_value>

This command defines the peak excursion.

Example: ":CALC:MARK:PEXC 10dB"

Features: *RST value: 6dB
SCPI: device-specific

Modes: R, A, VA

The numeric suffix in MARKer<1 to 4> is not significant.

:CALCulate<1|2>:MARKer<1 to 4>:READout MPHase | RIMaginary

This command determines the type of the marker display.

Example: ":CALC:MARK:READ RIM"

Features: *RST value: -
SCPI: device-specific

Mode: VA

The numeric suffix in MARKer<1 to 4> is not significant.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown <numeric_value>

This command defines the "N dB Down" value.

Example: ":CALC:MARK:FUNC:NDBD 3dB"

Features: *RST value: 6dB
SCPI: device-specific

Mode: A

The temporary markers T1 and T2 are positioned by n dB below the active reference marker. The frequency spacing of these markers can be queried with CALCulate:MARKer:FUNCtion:NDBDown:RESult?.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown:STATe ON | OFF

This command switches the "N dB Down" function on or off.

Example: ":CALC:MARK:FUNC:NDBD:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown:RESult?

This command queries the frequency spacing (bandwidth) of the "N dB Down" markers.

Example: ":CALC:MARK:FUNC:NDBD:RES?"

Features: *RST value: -
SCPI: device-specific

Mode: A

This command is only a query which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown:FREQuency?

This command queries the frequencies of the "N dB Down" marker.

Example: ":CALC:MARK:FUNC:NDBD:FREQ?"

Features: *RST value: -
SCPI: device-specific

Mode: A

The two frequency values are separated by comma and indicated in ascending order. This command is only a query which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:ZOOM <numeric_value>

This command defines the range to be enlarged around the active marker.

Example: ":CALC:MARK:FUNC:ZOOM 1kHz"

Features: *RST value: -
SCPI: device-specific

Mode: A-F

The subsequent frequency sweep is stopped at the marker position and the frequency of the signal is counted. This frequency becomes the new center frequency, the zoomed span is then set. This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NOISe[:STATe] ON | OFF

This command switches the noise measurement on or off.

Example: ":CALC:MARK:FUNC:NOIS ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A

The noise power density is measured at the position of the markers. The result can be queried with CALCulate:MARKer:FUNCtion:NOISe:RESult?.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NOISe:RESult?

This command queries the result of the noise measurement.

Example: ":CALC:MARK:FUNC:NOIS:RES?"

Features: *RST value: -
SCPI: device-specific

Mode: A

This command is an event which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor (60dB/3dB) | (60dB/6dB)

This command defines the shape factor measurement 60dB/6dB or 60dB/3dB.

Example: ":CALC:MARK:FUNC:SFAC (60dB/3dB)"

Features: *RST value: (60dB/6dB)
SCPI: device-specific

Mode: A

The temporary markers T1 to T4 are positioned in pairs by 60dB and by 3dB or 6dB below the active reference marker. The frequency spacing ratio of these markers - the shape factor - can be queried with CALCulate:MARKer:FUNCtion:SFACtor:RESult?.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor:STATe ON | OFF

This command switches the shape factor measurement on or off.

Example: ":CALC:MARK:FUNC:SFAC:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor:RESult?

This command queries the result of the shape factor measurement.

Example: ":CALC:MARK:FUNC:SFAC:RES?"

Features: *RST value: -
SCPI: device-specific

Mode: A

This command is only a query which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor:FREQuency?

This command queries the frequencies of the shape factor measurement.

Example: ":CALC:MARK:FUNC:SFAC:FREQ?"

Features: *RST value: -
SCPI: device-specific

Mode: A

Four frequency values (at -60 dB, -6 or. -3 dB, -6 or -3 dB, -60dB) are indicated in ascending order. They are separated by a comma. This command is only a query which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:STRack[:STATe] ON | OFF

This command switches the signal-track function on or off.

Example: ":CALC:MARK:FUNC:STR ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A-F

With SIGNAL TRACK function activated, the maximum signal is determined after each frequency sweep and the center frequency of this signal is set. With drifting signals the center frequency follows the signal.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:DDEMod:RESult?
MERM | MEPK | MEPS | PERM | PEPK | PEPS | EVRM | EVPK | EVPS | IQOF | IQIM | ADR | FERR | FEPK | RHO| DEV | FSRM | FSPK | FSPS | DTTS

This command queries the error measurement results of digital demodulation.The results correspond to the values obtained when the symbol table (SYMBOL TABLE/ ERRORS softkey) is selected in manual operation. Marker values can be queried queried with command `CALCulate<1|2>:MARKer<1...4>:Y?` and trace data with command `TRACe[:DATA]`.

Example: `":CALC:MARK:FUNC:DDEM:RES? EVRM"`

Features: *RST value: -
SCPI: device-specific

Mode: VA

MERM	magnitude error in %rms	FERR	frequency error in Hz
MEPK	maximum of magnitude error in %pk	FEPK	maximum of frequency error in Hz
MEPS	symbol number by which the maximum of the magnitude error occurred	ADR	amplitude drop in dB/symbol
PERM	phase error in deg	RHO	Rho-Factor
PEPK	maximum of phase error in deg	DEV	FSK deviation in Hz
PEPS	symbol number by which the maximum of the phase error occurred		
EVRM	vector error in %rms	FSPK	maximum of FSK deviation error in Hz
EVPK	maximum of vector error in %pk	FSRM	FSK deviation error in Hz
EVPS	symbol number by which the maximum of the vector error occurred	FSPS	symbol number by which the maximum of error occurred
IQOF	I/Q-offset error in %	DTTS	trigger delay of synchronization
IQIM	I/Q Imbalance in %		

This command is only a query which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:POWer:SELect OBANdwidth | OBWidth

This command switches on the power measurement.

Example: `":CALC:MARK:FUNC:POW:SEL OBAN"`

Features: *RST value: -
SCPI: device-specific

Mode: A-F

This command is an event which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:POWer:RESult? OBANdwidth | OBWidth

This command queries the results of the occupied bandwidth power measurement.

Example: `":CALC:MARK:FUNC:POW:RES? OBW"`

Features: *RST value: -
SCPI: device-specific

Modes: A, VA

This command is only a query which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:POWer[:STATe] OFF

This command switches the power measurement off.

Example: ":CALC:MARK:FUNC:POW OFF"

Features: *RST value: -
SCPI: device-specific

Modes: A-F, VA

This command is an event which is why it is not assigned an *RST value.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:STATe ON | OFF

This command switches the messages selected by the summary marker (eg RMS and MEAN) on and off. One or several measurements can be selected with the commands listed in the following and then switched jointly on and off with SUMMary:STATe.

Example: ":CALC:MARK:FUNC:SUMM:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A-Z, VA

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAXimum[:STATe] ON | OFF

This command switches on or off the measurement of the maximum of the absolute value.

Example: ":CALC:MARK:FUNC:SUMM:MAX ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: VA

When the measurement is switched on, the summary marker is automatically activated (command SUMMary:STATe set to ON). When it is switched off, the summary marker remains switched on provided further measurements are selected. Otherwise the marker is switched off automatically.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAXimum:RESult?

This command queries the results of the measurement of the maximum of the absolute value. Results of average calculation and peak hold are queried with commands ...:MAXimum:AVERage:RESult? and ...:MAXimum:PHOLd:RESult?.

Example: ":CALC:MARK:FUNC:SUMM:MAX:RES?"

Features: *RST value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAXimum:AVERage:RESult?

This command is used to query the results of the measurement of the maximum of the absolute value if the average is calculated using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage`.

Example: `":CALC:MARK:FUNC:SUMM:MAX:AVER:RES?"`

Features: *RST value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAXimum:PHOLd:RESult?

This command is used to query the results of the measurement of the maximum of the absolute value when the peak hold function is switched on with command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PHOLd`.

Example: `":CALC:MARK:FUNC:SUMM:MAX:PHOL:RES?"`

Features: *RST value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak[:STATe] ON | OFF

This command switches on or off the measurement of the positive peak value if the calculation and.

Example: `":CALC:MARK:FUNC:SUMM:PPE ON"`

Features: *RST value: OFF
SCPI: device-specific

Mode: VA

When the measurement is switched on, the summary marker is automatically activated (command `SUMMary:STATe` set to `ON`). When it is switched off, the summary marker remains switched on provided further measurements are selected. Otherwise the marker is switched off automatically.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak:RESult?

This command is used to query the result of the measurement of the positive peak value. Results of average calculation and peak hold are queried with commands `...:PPEak:AVERage:RESult?` and `...:PPEak:PHOLd:RESult?`.

Example: `":CALC:MARK:FUNC:SUMM:PPE:RES?"`

Features: *RST value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak:AVERage:RESult?

This command is used to query the result of the measurement of the positive peak value if the average is calculated using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage`.

Example: `":CALC:MARK:FUNC:SUMM:PPE:AVER:RES?"`

Features: *RST value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak:PHOLd:RESult?

This command is used to query the result of the measurement of the positive peak value if the peak hold function is switched on with command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PHOLd`.

Example: `":CALC:MARK:FUNC:SUMM:PPE:PHOL:RES?"`

Features: *RST value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak[:STATe] ON | OFF

This command switches on or off the measurement of the negative peak value.

Example: `":CALC:MARK:FUNC:SUMM:MPE ON"`

Features: *RST value: OFF
SCPI: device-specific

Mode: VA

When the measurement is switched on, the summary marker is automatically activated (command `SUMMary:STATe` set to `ON`). When it is switched off, the summary marker remains switched on provided further measurements are selected. Otherwise the marker is switched off automatically.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak:RESult?

This command queries the result of the measurement of the negative peak value. Results of average calculation and peak hold are queried with commands `...:MPEak:AVERage:RESult?` and `...:MPEak:PHOLd:RESult?`.

Example: `":CALC:MARK:FUNC:SUMM:MPE:RES?"`

Features: *RST- value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak:AVERage:RESult?

This command queries the result of the measurement of the negative peak value if the average is calculated using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage`.

Example: `":CALC:MARK:FUNC:SUMM:MPE:AVER:RES?"`

Features: *RST- value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak:PHOLd:RESult?

This command queries the result of the measurement of the negative peak value if the peak hold function is switched on with command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PHOLd`.

Example: `":CALC:MARK:FUNC:SUMM:MPE:RES?"`

Features: *RST- value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle[:STATe] ON | OFF

This command switches on or off the measurement of the arithmetical mean between positive and negative peak value.

Example: `":CALC:MARK:FUNC:SUMM:MIDD ON"`

Features: *RST value: OFF
SCPI: device-specific

Mode: VA

When the measurement is switched on, the summary marker is automatically activated (command `SUMMary:STATe` set to `ON`). When it is switched off, the summary marker remains switched on provided further measurements are selected. Otherwise the marker is switched off automatically.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle:RESult?

This command queries the result of the measurement of the arithmetical mean between positive and negative peak value. Results of average calculation and peak hold are queried with commands `...:MIDDle:AVERage:RESult?` and `...:MIDDle:PHOLd:RESult?`.

Example: `":CALC:MARK:FUNC:SUMM:MIDD:RES? "`

Features: *RST- value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle:AVERage:RESult?

This command queries the result of the measurement of the arithmetical mean between positive and negative peak value if the average is calculated using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage`.

Example: `":CALC:MARK:FUNC:SUMM:MIDD:AVER:RES? "`

Features: *RST- value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle:PHOLd:RESult?

This command queries the result of the measurement of the arithmetical mean between positive and negative peak value if the peak hold function is switched on using the command `:CALCulate<1|2>:MARKer<1 to4>:FUNCtion:SUMMary:PHOLd`.

Example: `":CALC:MARK:FUNC:SUMM:MIDD:PHOL:RES? "`

Features: *RST- value: -
SCPI: device-specific

Mode: VA

This command is only a query and thus has no *RST value assigned

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS[:STATe] ON | OFF

This command switches on or off the measurement of the effective (rms) value of the total trace.

Example: `":CALC:MARK:FUNC:SUM:RMS ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: A-Z, VA

When the measurement is switched on, the summary marker is automatically activated (command `SUMMary:STATe` set to `ON`). When it is switched off, the summary marker remains switched on provided further measurements are selected. Otherwise the marker is switched off automatically.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS:RESult?

This command queries the result of the measurement of the mean value of the total trace. Results of average calculation and peak hold are queried with commands `...:RMS:AVERage:RESult?` and `...:RMS:PHOLd:RESult?`..

Example: `":CALC:MARK:FUNC:SUMM:RMS:RES?"`

Features: *RST- value: -
SCPI: device-specific

Modes: A-Z, VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS:AVERage:RESult?

This command queries the result of the measurement of the mean value of the total trace if the average is calculated using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage`.

Example: `":CALC:MARK:FUNC:SUMM:RMS:AVER:RES?"`

Features: *RST- value: -
SCPI: device-specific

Modes: A-Z, VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS:PHOLd:RESult?

This command queries the result of the measurement of the mean value of the total trace if the peak hold function is switched on using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PHOLd`.

Example: `":CALC:MARK:FUNC:SUMM:RMS:PHOL:RES?"`

Features: *RST- value: -
SCPI: device-specific

Modes: A-Z, VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN[:STATe] ON | OFF

This command switches on or off the measurement of the mean value of the total trace.

Example: `":CALC:MARK:FUNC:SUMM:MEAN ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: A-Z, VA

When the measurement is switched on, the summary marker is automatically activated (command `SUMMary:STATe` set to `ON`). When it is switched off, the summary marker remains switched on provided further measurements are selected. Otherwise the marker is switched off automatically.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN:RESult?

This command queries the result of the measurement of the mean value of the total trace. Results of average calculation and peak hold are queried with commands `...:MEAN:AVERage:RESult?` and `...:MEAN:PHOLd:RESult?`..

Example: `":CALC:MARK:FUNC:SUMM:MEAN:RES?"`

Features: *RST- value: -
SCPI: device-specific

Modes: A-Z, VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN:AVERage:RESult?

This command queries the result of the measurement of the mean value of the total trace if the average is calculated using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage`.

Example: `":CALC:MARK:FUNC:SUMM:MEAN:AVER:RES?"`

Features: *RST- value: -
SCPI: device-specific

Modes: A-Z, VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN:PHOLd:RESult?

This command queries the result of the measurement of the mean value of the total trace if the peak hold function is switched on using the command `:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PHOLd`.

Example: `":CALC:MARK:FUNC:SUMM:MEAN:PHOL:RES?"`

Features: *RST- value: -
SCPI: device-specific

Modes: A-Z, VA

This command is only a query and thus has no *RST value assigned.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PHOLd ON | OFF

This command switches on or off the peak-hold function.

Example: `":CALC:MARK:FUNC:SUMM:PHOL ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: A-Z, VA

The peak-hold function is reset by switching off and on, again.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage ON | OFF

This command switches the calculation of the average value on or off.

Example: `":CALC:MARK:FUNC:SUMM:AVER ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: A-Z, VA

The calculation of the average is reset by switching off and on, again.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AOFF

This command switches off all measuring functions.

Example: `":CALC:MARK:FUNC:SUMM:AOFF"`

Features: *RST value:
SCPI: device-specific

Modes: A-Z, VA

This command is an "event" and therefore has no *RST value assigned and no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:CENTer

This command sets the center or receiver to that of the current marker.

Example: ":CALC:MARK:FUNC:CENT"

Features: *RST value: -
SCPI: device-specific

Modes: R, A-F

This command is an "event" and therefore has no *RST value assigned and no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:CSTep

This command sets the step width of the center frequency to the x-value of the current marker.

Example: ":CALC:MARK:FUNC:CST"

Features: *RST value: -
SCPI: device-specific

Modes: R, A-F

This command is an "event" and therefore has no *RST value assigned and no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:STARt

This command sets the start frequency to the frequency of the current marker.

Example: ":CALC:MARK:FUNC:STAR"

Features: *RST value: -
SCPI: device-specific

Modes: R, A-F

This command is an "event" which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:STOP

This command sets the stop frequency to the frequency of the current marker.

Example: ":CALC:MARK:FUNC:STOP"

Features: *RST value: -
SCPI: device-specific

Modes: R, A-F

This command is an "event" which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:MSTep

This command sets the marker step width to the x-value of the current marker.

Example: ":CALC:MARK:FUNC:MST"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an "event" which is why it is not assigned an *RST value and has no query.

:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:REFerence

This command sets the reference level to that of the current marker.

Example: ":CALC:MARK:FUNC:REF"

Features: *RST value: -
SCPI: device-specific

Modes: A, VA

This command is an "event" and therefore has no *RST value assigned and no query.

CALCulate:MATH Subsystem

The CALCulate:MATH - subsystem allows to process data from the SENSe-subsystem in numeric expressions.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:MATH<1 to 4>			
[:EXPRession]			
[:DEFine]	<expr>	--	
:STATe	<Boolean>	--	
:MODE	LINear | LOGarithmic	--	

:CALCulate<1|2>:MATH<1 to 4>[:EXPression][:DEFine] <expr>

This command defines the mathematical expression for relating traces and reference line. Command `CALCulate:MATH:STATe` switches the mathematical relation of traces on or off .

Parameter: <expr>::= 'OP1 - OP2 [+ RLINE]'
OP1 ::= TRACE1 | TRACE2 | TRACE3 | TRACE4
OP2 ::= TRACE1 | TRACE2 | TRACE3 | TRACE4 | RLINE

Examples:
```
":CALC:MATH1 (TRACE1 - TRACE3 + RLINE)"
":CALC:MATH4 (TRACE4 - RLINE)"
```

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The operand `[+ RLINE]` may be used only if `OP2` is different from `RLINE`. The numeric suffix in `CALCULATE<1|2>` is not significant. The numeric suffix in `MATH<1 to 4>` denotes the trace where the result of the mathematical operation is stored. The number must be identical to the number of the operand `OP1`.

:CALCulate<1|2>:MATH<1 to 4>:STATe ON | OFF

This command switches the mathematical relation of traces on or off.

Example:
```
":CALC:MATH1:STAT ON"
```

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

The numeric suffix in `CALCULATE<1|2>` is not significant. The numeric suffix in `MATH<1 to 4>` denotes the trace which the command refers to.

:CALCulate<1|2>:MATH<1 to 4>:MODE LINear | LOGarithmic

This command selects linear or logarithmic calculation of the mathematical functions relating the traces.

Example:
```
":CALC:MATH1:MODE LIN"
```

Features: *RST value: LOG
SCPI: device-specific

Modes: A, VA

The numeric suffix in `CALCULATE<1|2>` is not significant. The numeric suffix in `MATH<1 to 4>` denotes the trace which the command refers to.

CALCulate:PEAKsearch - Subsystem

The CALCulate:PEAKsearch subsystem allows data to be processed for final measurement.

Command	PARAMETER	Unit	Comment
CALCulate<1|2>			
:PEAKsearch			
[:IMMediate]			
:MARGin	<numeric_value>	dB	
:SUBRanges	<numeric_value>	--	
:METHod	SUBRange | PEAK		
:PSEarch			
[:IMMediate]			
:MARGin	<numeric_value>	dB	
:SUBRanges	<numeric_value>	--	
:METHod	SUBRange | PEAK		

:CALCulate<1|2>:PEAKsearch|PSEarch[:IMMediate]

This command activates the generation of final measurement data.

Example: `"CALC:PEAK"`

Features: *RST value: -
SCPI: device-specific

Mode: **R**

The numeric suffix in `CALCULATE<1|2>` is not significant.

:CALCulate<1|2>:PEAKsearch|PSEarch:MARGin MINimun to MAXimum

This command defines the margin of the final measurement.

Example: `"CALC:PEAK:MARG 5 dB"`

Features: *RST value: 6 dB
SCPI: device-specific

Mode: R

The numeric suffix in `CALCULATE<1|2>` is not significant.

:CALCulate<1|2>:PEAKsearch|PSEarch:SUBRanges 1 to 500

This command defines the number of subranges for the final measurement.

Example: `"CALC:PEAK:SUBR 10"`

Features: *RST value: 25
SCPI: device-specific

Mode: R

The numeric suffix in `CALCULATE<1|2>` is not significant.

:CALCulate<1|2>:PEAKsearch|PSEarch:METHod SUBRange | PEAK

This command defines the method that is used to determine the maxima of a scan.

Example: `"CALC:PEAK:METH SUBR"`

Features: *RST value: PEAK
SCPI: device-specific

Mode: R

The numeric suffix in CALCULATE<1|2> is not significant.

CALCulate:X and CALCulate:UNIT Subsystem

The CALCulate:X and CALCulate:Unit subsystems define the units for vector signal analyzer mode and power measurements.

COMMAND	PARAMETERS	UNIT	COMMENT
CALCulate<1|2>			
:X			Vector Signal Analysis
:UNIT			
:TIME	S | SYM		
:UNIT			
:ANGLe	DEG | RAD		Vector Signal Analysis
:POWer	DBM | V | W | DB | PCT | UNITLESS | DBPW | WATT | DBUV | DBMV | VOLT | DBPT | DBUA | AMPere DBUV_MHZ | DBMV_MHZ | DBUA_MHZ | DBUV_M | DBUA_M | DBUV_MMHZ | DBUA_MMHZ		

:CALCulate<1|2>:X:UNIT:TIME S | SYM

This command selects seconds or symbols as an x-axis unit.

Example: `":CALC:X:UNIT:TIME S"`

Features: *RST value: S
SCPI: device-specific

Mode: VA

:CALCulate<1|2>: UNIT:ANGLe DEG | RAD

This command selects the unit for angular measurement.

Example: `":CALC:UNIT:ANGL DEG"`

Features: *RST- value: RAD
SCPI: device-specific

Mode: VA

:CALCulate<1|2>: UNIT:POWer DBM | V | W | DB | PCT | DBPT | UNITLESS | DBPW | WATT | DBUV |DBMV | VOLT | DBUA | AMPere | DBUV_MHZ |DBMV_MHZ| DBUA_MHZ | DBUV_M | DBUA_M |DBUV_MMHZ | DBUA_MMHZ

This command selects the unit for power.

Example: `":CALC:UNIT:POW DBM"`

Features: *RST value: _
SCPI: device-specific

Modes: R, A, VA

In the receiver mode, the units DBM, DBUV, DBPT, DBUA, DBPW, DBT, DBUV_M and DBUA_M are available. In the analyzer mode, all unit are available except for DBPT, DEG, RAD, S, and HZ. The units DEG, RAD, S, and HZ are available only in the vector analyzer mode.

DBUV_MHZ and DBUA_MHZ denote the units DBUV/MHZ or DBUA/MHZ.

CALibration Subsystem

The commands of the CALibration subsystem perform instrument calibrations.

COMMAND	PARAMETERS	UNIT	COMMENT
CALibration			
[:ALL]?	--	--	query only
:BANDwidth			
[:RESolution]?	--	--	query only
:BWIDth			
[:RESolution]?	--	--	query only
:IQ?	--	--	query only / Vector Signal Analysis
:LDETector?	--	--	query only
:LOSuppression?	--	--	query only
:PPEak?	--	--	query only
:SHORt?	--	--	
:STATe	<Boolean>	--	query only

::CALibration[:ALL]?

This command performs a complete calibration of the instrument. A "0" is returned if the calibration was successful.

Example: "CAL?"

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

:CALibration:BANDwidth|BWIDth[:RESolution]?

This command performs a calibration of the filter bandwidths. A "0" is returned if the calibration was successful.

Example: ":CAL:BAND?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALibration:IQ?

This command performs a calibration of the vector signal analyzer. A "0" is returned if the calibration was successful.

Example: ":CAL:IQ?"

features: *RST value: -
SCPI: device-specific

Mode: VA

:CALibration:LDETector?

This command performs a calibration of the log module's characteristic and of the detectors. A "0" is returned if the calibration was successful.

Example: ":CAL:LDET?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALibration:LOSuppression?

This command performs a calibration of the local oscillator suppression. . A "0" is returned if the calibration was successful.

Example: ":CAL:LOS?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALibration:PPEak?

This command performs a calibration of the tracking YIG filter (preselector peaking). A "0" is returned if the calibration was successful.

Example: ":CAL:PPE?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

:CALibration:SHORt?

This command performs a short calibration. A "0" is returned if the calibration was successful.

Example: ":CAL:SHOR?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:CALibration:STATe ON | OFF

This command determines whether (ON) or not (OFF) the current calibration data are taken into consideration.

Example: ":CAL:STAT OFF"

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

DIAGnostic Subsystem

The DIAGnostic subsystem contains the commands which support instrument diagnostics for maintenance, service and repair. In accordance with the SCPI standard, all of these commands are device-specific.

COMMAND	PARAMETERS	UNIT	COMMENT
DIAGnostic			
:SERVice			
:INPut			
[:SELect]	CALibration | RF	--	
:FUNCtion	<numeric_value>,>numeric_value>	--	no query
:NSOurce	<Boolean>	--	
:HGENerator	OFF | 10KHZ | 100 KHZ | BALanced	HZ	
:INFO			
:CCOunt			
:ATTenuation<1>?			query only
:PRESelector<1 to 10>?			query only

:DIAGnostic:SERVice:INPut[:SELect] CALibration | RF

This command toggles between the RF input on the front panel and the internal 120-MHz reference signal.

Example: `":DIAG:SERV:INP CAL"`

Features: *RST value: RF
SCPI: device-specific

Modes: R, A, VA

:DIAGnostic:SERVice:FUNCtion <numeric_value>,<numeric_value>...

This command activates a service function.

Example: `":DIAG:SERV:FUNC 2,0,2,12,1"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

The service function is selected via five parameters: functional group number, board number, function number, parameter 1 and parameter 2 (see service manual).

:DIAGnostic:SERVice:NSOurce ON | OFF

This command switches the 28-V supply at the rear connector of the noise source on and off.

Example: `":DIAG:SERV:NSO ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:DIAGnostic:SERVice:HGENerator OFF | 10KHZ | 100KHZ | BALanced

This command controls the harmonic generator.

Example: ":DIAG:SERV:HGEN 10KHZ"

Features: *RST value: OFF
SCPI: device specific

Modes: R, A, VA

:DIAGnostic:INFO:CCOunt:ATTenuation?

This command queries the cycle counters of the attenuator of the basic instrument

The result is output as a list of values separated by a ','. The list starts with the date.

Example: ":DIAG:INFO:CCO:ATT?"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is a query and has therefore no *RST value assigned.

:DIAGnostic:INFO:CCOunt:PRESelector<1..6>?

This command queries the cycle counters of the attenuators of preselector FSET-Zxx. The suffix selects the attenuator:

1: Input 1; 2: Input 2; 3: Low pass
4: High pass 5: Band pass; 6: Micro wave

The result is output as a list of values separated by a ','. The list starts with the date.

Example: ":DIAG:INFO:CCO:PRES2?"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is a query and has therefore no *RST value assigned.

DISPlay Subsystem

The DISPlay subsystem controls the selection and presentation of textual and graphic information as well as of trace data on the display.
The displays in the split-screen mode are assigned to WINDow 1 (screen A) or 2 (screen B) .

COMMAND	PARAMETERS	UNIT	COMMENT
DISPlay			
:FORMat	SINGle | SPLit		
:PROgram			
[:MODE]	<Boolean>		
:ANNotation			
:FREQuency	<Boolean>		
:LOGO	<Boolean>		
:CMAP<1 to 13>			
:DEFault			
:HSL	0 to 100,0 to 100,0 to 100		
:PDEFined	BLACk | BLUE | BROWn | GREen | CYAN | RED | MAGenta | YELLow | WHITe | GRay | LGRay | LBLue | LGReen | LCYan | LRED | LMAGenta		
[:WINDow<1|2>]			
:TEXT			
[:DATA]	<string>		
:STATe	<Boolean>		
:TIME	<Boolean>		
:FEED	<string>		
:MINFo	<Boolean>		
:TRACe<1 to 4>			
:X			
[:SCALe]			Vector Analysis
:RVALue	<numeric_value>		
:ZOOM	<Boolean>		
[:FREQuency]			
:STARt	<numeric_value>	HZ	
:STOP	<numeric_value>	HZ	
:CENTer	<numeric_value>	HZ	
:SPACing	LINear | LOGarithmic	--	
:Y			
[:SCALe]	<numeric_value>	DB	
:MODE	ABSolute | RELative		
:RLEVel	<numeric_value>	DBM	
:OFFSet	<numeric_value>	DB	
:RVALue	<numeric_value>	DBM|DB|HZ|DEG|RAD	
:AUTO	<Boolean>		Analyzer mode only
:RPOSition	<numeric_value>	PCT	Vector Analysis
:PDIVision	<numeric_value>	DBM|DB|HZ|DEG|RAD	Vector Analysis
:SPACing	LINear|LOGarithmic|PERCent	--	
:MODE	WRITe | VIEW | AVERage | MAXHold | MINHold	--	
:CWRite	<Boolean>	--	Vector Analysis
:ANALog	<Boolean>	--	
:HCONtinuous	<Boolean>	--	
[:STATe]	<Boolean>	--	
:SYMBol	DOTS | BARS | OFF		Vector Analysis
:EYE			
:COUNt	<numeric_value>	SYM	Vector Analysis

COMMAND	PARAMETERS	UNIT	COMMENT
:BARGraph			
:LEVel			
:LOWer?	--		
:UPPer?	--		
:PSAVe			
[:STATe]	<Boolean>		
HOLDoff	<numeric_value>		

:DISPlay:FORMat SINGle | SPLit

This command switches the test result display between FULL SCREEN and SPLIT SCREEN.

Example: ":DISP:FORM SPL"

Features: *RST value: SINGle
SCPI: device-specific

Modes: R, A, VA

:DISPlay:PROGram[:MODE] ON | OFF

This command switches the display between the measuring instrument and the computer function.

Example: ":DISP:PROG ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:DISPlay:ANNotation:FREQuency ON | OFF

This command switches the x-axis annotation on or off.

Example: ":DISP:ANN:FREQ OFF"

Features: *RST value: ON
SCPI: conforming

Modes: R, A, VA

:DISPlay:LOGO ON | OFF

This command switches the factory logo on the screen on or off.

Example: ":DISP:LOGO OFF"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

:DISPlay:CMAP<1 to 13>:DEFault

This command resets the screen colors of the instrument to their default settings.

Example: ":DISP:CMAP:DEF"

Features: *RST value: --
SCPI: conforming

Modes: R, A, VA

This command is an event and has thus no query and no *RST value assigned. The numeric suffix in CMAP<1 to 13> is not significant.

:DISPlay:CMAP<1 to 13>:HSL <hue>,<sat>,<lum>

This command defines the color table of the instrument.

Parameter: hue = TINT
sat = SATURATION
lum = BRIGHTNESS

The value range is 0.0 to 100.0 for all parameters.

Example: `":DISP:CMAP2:HSL 0.3,0.8,1.0"`

Features: *RST value: --
SCPI: conforming

Modes: R, A, VA

To each numeric suffix of CMAP is assigned one or several picture elements which can be modified by varying the corresponding color setting. The following assignment applies:

CMAP1	Trace 1	CMAP8	Soft key State Data Entry
CMAP2	Trace 2	CMAP9	Soft key State OFF
CMAP3	Trace 3	CMAP10	Soft key Shade
CMAP4	Trace 4	CMAP11	Text
CMAP5	Marker	CMAP12	Title
CMAP6	Grid	CMAP13	Background
CMAP7	softkey State On		

The values set are not changed by *RST.

:DISPlay:CMAP<1 to 13>:PDEFined BLACk | BLUE | BROWn | GREen | CYAN | RED | MAGenta | YELLow |WHITe | DGRay | LGRay | LBLue | LGReen | LCYan |LRED | LMAGenta

This command defines the color table of the instrument using predefined color values. To each numeric suffix of CMAP is assigned one or several picture elements which can be modified by varying the corresponding color setting. The same assignment as for `:DISPlay:CMAP<1 to 13>:HSL` applies.

Example: `":DISP:CMAP2:PDEF GRE"`

Features: *RST value: --
SCPI: conforming

Modes: R, A, VA

The values set are not changed by *RST.

:DISPlay[:WINDow<1|2>]:FEED 'AF' | 'VIDeo'

This command switches the AF-indication on or off.

Example: `":DISP:WIND1:FEED 'AF'"`

Features: *RST value: 'VIDeo'
SCPI: device-specific

Mode: AF-D

'AF' AF-Display on
'VIDeo' AF-Display off

:DISPlay[:WINDow<1|2>]:MINFo ON | OFF

This command switches the marker info list on the screen on or off.

Example: `":DISP:MINF ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:DISPlay[:WINDow<1|2>]:TEXT[:DATA] <string>

This command defines a comment (max. 50 characters) which can be displayed on the screen.

Example: `":DISP:TEXT "signal/noise power measurement"`

Features: *RST value: "" (empty)
SCPI: conforming

Modes: R, A, VA

The numeric suffix in WINDow<1|2> is not significant.

:DISPlay[:WINDow<1|2>]:TEXT:STATe ON | OFF

This command switches on or off the screen display of the comment.

Example: `":DISP:TEXT:STAT ON"`

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

The numeric suffix in WINDow<1|2> is not significant.

:DISPlay[:WINDow<1|2>]:TIME ON | OFF

This command switches on or off the screen display of date and time.

Example: `":DISP:TIME ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

The numeric suffix in WINDow<1|2> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:RVALue <numeric_value>

This command defines the reference value for the X-axis of the measuring diagram.

Example: `":DISP:TRAC:X:RVAL 20SYM"`

Features: *RST value: -
SCPI: device-specific

Mode: VA

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM ON | OFF

This command displays the zoomed current frequency range in the opposite window of the split screen representation.

Example: ":DISP:TRAC:X:ZOOM ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A-F

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM[:FREQuency]:STARt <numeric_value>

This command defines the start frequency of the zoomed display range. The value must lie between start and stop frequency of the original range.

Example: ":DISP:TRAC:X:ZOOM:STAR 100MHZ"

Features: *RST value: -- (depending on the current frequency setting)
SCPI: device-specific

Mode: A-F

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM[:FREQuency]:STOP <numeric_value>

This command defines the stop frequency of the zoomed display range. The value must lie between start and stop frequency of the original range.

Example: ":DISP:TRAC:X:ZOOM:STOP 200MHZ"

Features: *RST value: -- (depending on the current frequency setting)
SCPI: device-specific

Mode: A-F

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM[:FREQuency]:CENTer <num_value>

This command shifts the zoomed frequency range to the specified center frequency.

Example: ":DISP:TRAC:X:ZOOM:CENT 1GHZ"

Features: *RST value: -- (depending on the current frequency setting)
SCPI: device-specific

Mode: A-F

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X:SPACing LINear | LOGarithmic

This command toggles between linear and logarithmic display.

Example: ":DISP:TRAC:X:SPAC LIN"

Features: *RST value: LOGarithmic
SCPI: conforming

Modes: R, A

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe] 10dB to 200dB

This command defines the display range of the Y-axis (level axis) with logarithmic scaling (DISP:TRAC:Y:SPAC LOG).

Example: ":DISP:TRAC:Y 110dB"

Features: *RST value: 100dB
SCPI: device-specific

Mode: A

For linear scaling, (DISP:TRAC:Y:SPAC LIN | PERC) the display range is fixed and cannot be set. The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:MODE ABSolute | RELative

This command defines the scale of the y-axis (absolute or relative).

Example: ":DISP:TRAC:Y:MODE REL"

Features: *RST value: ABS
SCPI: device-specific

Mode: A

As long as SYSTem:DISPlay is set to OFF, this command does not directly influence the screen. The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RLEVel -200dBm to 200dBm

This command defines the reference level.

Example: ":DISP:TRAC:Y:RLEV -60dBm"

Features: *RST value: -20dBm
SCPI: conforming

Modes: A, VA

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RLEVel:OFFSet -200dB to 200dB

This command defines the offset of the reference level.

Example: ":DISP:TRAC:Y:RLEV:OFFS -10dB"

Features: *RST value: 0dB
SCPI: conforming

Modes: A, VA

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RVALue <numeric_value>

This command defines the reference value for the y-axis of the measurement diagram.
This defines the upper limit of the display range, the corresponding parameters of the manual control vary according to the measurement mode.

In the spectrum analysis mode the setting corresponds to the parameter MAX LEVEL.

In the vector signal analysis mode the setting corresponds to the parameter REFERENCE VALUE Y AXIS.

Example: ":DISP:TRAC:Y:RVAL -20dBm" (spectrum analysis)
":DISP:TRAC:Y:RVAL +1.20" (vector signal analysis)

Features: *RST value: - coupled to reference level
0 (vector signal analysis)
SCPI: device specific

Modes: A, VA

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RVALue:AUTO ON | OFF

This command defines whether the reference value for the y-axis of the measured diagram is coupled to the reference level (default) or can be set independently.

Example: ":DISP:TRAC:Y:RVAL:AUTO ON"

Features: *RST value: ON
SCPI: device-specific

Mode: A

This command is available in the analyzer mode only. The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RPOSition 0 to 100PCT

This command defines the position of the reference value.

Example: ":DISP:TRAC:Y:RPOS 50PCT"

Features: *RST value: 50PCT
SCPI: conforming

Mode: VA

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:PDIVision <numeric_value>

This command defines the scaling of the Y-axis in the current unit.

Example: ":DISP:TRAC:Y:PDIV +1.20"

Features: *RST value: -
SCPI: conforming

Mode: VA

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:BOTTom <numeric_value>

This command defines the minimum grid level for the display in the receiver mode.

Example: ":DISP:TRAC:Y:BOTT -20"

Features: *RST value: 0
SCPI: conforming

Mode: R

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:TOP <numeric_value>

This command defines the maximum grid level for the display in the receiver mode.

Example: ":DISP:TRAC:Y:TOP 120"

Features: *RST value: 100
SCPI: conforming

Mode: R

The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y:SPACing LINear | LOGarithmic | PERCent

This command toggles between linear and logarithmic display.

Example: ":DISP:TRAC:Y:SPAC LIN"

Features: *RST value: LOGarithmic
SCPI: conforming

Mode: A

The linear display can be LIN/% (PERCent) or LIN/dB (LINear). The numeric suffix in TRACe<1 to 4> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE WRITe | VIEW | AVERage | MAXHold | MINHold | FRESult

This command defines the type of display and the evaluation of the traces.

For selection FRESult (display of final measurement) only suffix 3 or 4 is valid for TRACe<1 to 4>.

Example: ":DISP:TRAC3:MODE MAXH"

Features: *RST value: WRITe for TRACe1, STATe OFF for TRACe2 to 4
SCPI: device-specific

Modes: R, A, VA

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE:CWRite ON | OFF

This command selects continuous display of the measured values (continuous write).

Example: ":DISP:TRAC3:MODE:CWR ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: VA

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE:ANALog ON | OFF

This command selects continuous display of the measured values in the analyzer mode (analog trace).

Example: ":DISP:TRAC3:MODE:ANAL ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE:HCONtinuous ON | OFF

This command defines whether the traces in the min hold/max hold mode are reset after some definite parameter changes.

Example: ":DISP:TRAC3:MODE:HCON ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A

OFF The traces are reset after some definite parameter changes.

ON This mechanism is switched off.

In general, parameter changes require a restart of the measurement before results are evaluated (e.g. with markers). For those changes that are known to require a new measurement (e.g. modification of the span), the trace is automatically reset so that erroneous evaluations of previous results are avoided. This mechanism can be switched off for those exceptional cases where the described behavior is unwelcome.

The numeric suffix in WINDow<1|2> is not significant.

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>[:STATe] ON | OFF

This command switches the display of the corresponding trace on or off.

Example: `":DISP:TRAC3 ON"`

Features: *RST value: ON for TRACe1, OFF for TRACe2 to 4
SCPI: conforming

Modes: R, A, VA

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:SYMBol DOTS | BARS | OFF

This command determines the display of the points of decision on the trace.

Example: `":DISP:TRAC:SYMB BARS"`

Features: *RST value: OFF
SCPI: device-specific

Mode: VA

:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:EYE:COUNt 1 to Result Length

This command determines the display range of the eye diagram in symbols.

Example: `":DISP:TRAC:EYE:COUNt 5"`

Features: *RST value: 2
SCPI: device-specific

Mode: VA

:DISPlay:BARGraph:LEVel:LOWer?

This command queries the minimum level of the bar graph.

Example: `":DISP:BARG:LEV:LOW?"`

Features: *RST value: --
SCPI: device-specific

Mode: R

:DISPlay:BARGraph:LEVel:UPPer?

This command queries the maximum level of the bar graph.

Example: `":DISP:BARG:LEV:UPP?"`

Features: *RST value: --
SCPI: device-specific

Mode: R

:DISPlay:PSAVe[:STATe] ON | OFF

This command switches the screen saver mode on or off.

Example `":DISP:PSAV ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: A, VA

:DISPlay: PSAVe:HOLDoff <numeric_value>

This command determines the time after which the screen saver is switched on. Values are input in minutes, the valid range is 1 to 100 minutes.

Example: `":DISP:PSAV:HOLD 10"`

Features: *RST value: 1
SCPI: device-specific

Modes: A, VA

FORMat Subsystem

The FORMat subsystem specifies the data format of the data transmitted from and to the instrument.

COMMAND	PARAMETERS	UNIT	COMMENT
FORMat			
[:DATA]	ASCii|REAL|UINT[,<numeric_value>]	-	
:DEXPort			
:DSEParator	POINt|COMMa		
:HEADer			
[:STATe]	<Boolean>		
:APPend			
[:STATe]	<Boolean>		
:COMMent	<string>		

:FORMat[:DATA] ASCii | REAL | UINT [, 32]

This command specifies the data format of the data transmitted from and to the instrument.

Example: `":FORM REAL,32"`
`":FORM ASC"`

Features: *RST value: ASCii
SCPI: conforming

Modes: R, A, VA

The data format is either ASCii or one of the formats REAL or UINT (Unsigned Integer). ASCii data are transmitted in plain text, separated by commas. REAL data are transmitted as 32-bit IEEE 754 floating-point numbers in the "definite length block". The format UINT is only used in operating mode Vector Signal Analysis, for the symbol table.

Format setting for the binary transmission of trace data (see also `TRACE:DATA?`):

Analyzer mode: REAL, 32

Vector analyzer: UINT, 8 with digital demodulation, symbol table
REAL, 32otherwise

Note: *Incorrect format setting will result in numerical conversion, which may lead to incorrect results.*

If the format ”FORM REAL, 32” is selected for the above example, the data stream from the analyzer to the controller is as follows:

```
#432085334.....
 |||  |
 |||  From here follow the data bytes of the data block, here 5334
 ||Number of bytes of data block stated in ASCII plain text, here 3208
 |ASCII byte stating the length of the subsequent length counter, here 4
Head marker of binary data stream
```

For this example, the number of transmitted data bytes is as follows:

Number = number of test points * bytes/value* 2

3208 = 401 * 4* 2

The FORMat command is valid for the transmission of trace data. The data format of trace data received by the instrument is automatically recognized, regardless of the format which is programmed.

:FORMat:DEXPort:DSEParator POINt|COMMA

This command determines the decimal separator (decimal point or comma)for the output of the measurement data in ASCII format. Thus, evaluation programms (i.e. MS-Exel) of different language versions are supported.

Example: ":FORM:DEXP:DSEP POIN

Features: *RST value: POINt
SCPI: device specific

Modes: R, A, VA

:FORMat:DEXPort:HEADer[:STATe] ON | OFF

This commad determines if the output file starts with a header (start frequency, sweep time, detector, .. = ON) or not. OFF = only measurement data is output.

Example: ":FORM:DEXP:HEAD OFF

Features: *RST value: ON
SCPI: device specific

Modes: R, A, VA

:FORMat:DEXPort:APPend[:STATe] ON | OFF

This commad determines if the output file is overwritten or the data is added to the end of the file.

Example: ":FORM:DEXP:APP OFF

Features: *RSTvalue: ON
SCPI: device-specific

Modes: R, A, VA

:FORMat:DEXPort:COMMent <string>

This command defines a comment for the output file.

Example: ":FORM:DEXP:COMM 'ASCII EXPORT TRACE 1'"

Features: *RST value: „“
SCPI: conforming

Modes: R, A, VA

HCOPy Subsystem

The HCOPy subsystem controls the output of display information for documentation purposes on output devices or files.

COMMAND	PARAMETERS	UNIT	COMMENT
HCOPy			
:ABORt	--		no query
:DESTination<1|2>	SYST:COMM:PRIN'| 'SYST:COMM:CLIP' | 'MMEM'		no query
:DEVice			
:COLor	<Boolean>		
:LANGuage<1|2>	WMF | GDI | EWMF | BMP		
[:IMMediate<1|2>]	--		
:ITEM			no query
:ALL			
:FFEed<1|2>			no query
:STATe	<Boolean>		
:LABel			
:TEXT	<string>		
:PFEed<1|2>			
:STATe	<Boolean>		
:WINDow<1|2>			
:TABLe			
:STATe	<Boolean>		
:TEXT	<string>		
:TRACe			
:STATe	<Boolean>		
:CAINcrement	<Boolean>		
:PAGE			
:DIMensions			
:QUADrant<1 to 4>			no query
:FULL			no query
:ORIentation<1|2>	LANDscape|PORTrait		

:HCOPy:ABORt

This command aborts a running hardcopy output.

Example: ":HCOP:ABOR"

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and no query.

:HCOPy:DESTination<1|2> <string>

This command selects the device for outputting the hardcopy. The availability of the parameters depends on the selected data format (see command `:HCOPy:DEVice:LANGuage`).

Parameter: <string>::= 'MMEM' |
'SYST:COMM:PRIN' |
'SYST:COMM:CLIP'

Example: `":HCOP:DEST2 'MMEM'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

`'MMEM'` creates a file for the hardcopy output. Command `MMEM:NAME <file_name>` defines the file name. All formats can be selected for `:HCOPy:DEVice:LANGuage`.

`'SYST:COMM:PRIN'` directs the hardcopy to the printer. The printer is selected with command `SYSTEM:COMMunicate:PRINter:SELect`. `GDI` should be selected for `:HCOPy:DEVice:LANGuage`.

`'SYST:COMM:CLIP'` directs the hardcopy to the clipboard. `EWMF` should be selected for `:HCOPy:DEVice:LANGuage`.

This command is an event which is why it is not assigned an *RST value and no query.

:HCOPy:DEVice:COLor ON | OFF

This command selects between color and monochrome hardcopy of the screen.

Example: `":HCOP:DEV:COL ON"`

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

:HCOPy:DEVice:LANGuage<1|2> WMF| EWMF | GDI | BMP

This command determines the data format of the printout.

Example: `":HCOP:DEV:LANG WMF"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

WMF and (WINDOWS Metafile and Enhanced Metafile Format)

EWMF Data formats for output files which can at a later time be integrated in corresponding programs for documentation purposes.

BMP (Bitmap) Data format for output files.

GDI (Graphics Device Interface) Default format for the output to a printer configured under Windows. For the output file the printer driver configured under Windows is used and thus a printer-specific format is generated.

:HCOPy[:IMMediate<1|2>]

This command starts a hardcopy output.

Example: "HCOP"

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

HCOP[1] starts the hardcopy output to device 1 (default), HCOP2 starts the output to device 2. This command is an event which is why it is not assigned an *RST value and has no query.

:HCOPy:ITEM:ALL

This command selects the complete screen to be output.

Example: ":HCOP:ITEM:ALL"

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

The hardcopy output is always provided with comments, title, time and date. As an alternative to the whole screen, only traces (commands ':HCOPy:DEVice:WINDow:TRACe: STATe ON') or tables (command ':HCOPy:DEVice:WINDow:TABLe:STATe ON') can be output.

:HCOPy:ITEM:FFEed<1|2>:STATe ON | OFF

The command adds a form feed command to the hardcopy output of the screen.

Example: ":HCOP:ITEM:FFE2:STAT ON"

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

:HCOPy:ITEM:LABel:TEXT <string>

This command defines the title of the screen with a maximum of 60 characters.

Example: ":HCOP:ITEM:LAB:TEXT 'My Title'"

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

:HCOPy:ITEM:PFEed<1|2>:STATe ON | OFF

This command adds a paper feed command to the hardcopy output of the screen (ON).

Example: ":HCOP:ITEM:PFE2:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:HCOPy:ITEM:WINDow<1|2>:TABLe:STATe ON | OFF

This command selects the output of the currently displayed tables (ON).

Example: ":HCOP:ITEM:WIND:TABL:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

The command :HCOPy:ITEM:WINDow<1|2>:TABLe:STATe OFF same as command :HCOPy:ITEM:ALL enables the output of the whole screen.

:HCOPy:ITEM:WINDow<1|2>:TEXT <string>

This command defines the comment text for printout to trace 1 or 2 with a maximum of 100 characters.

Example: ":HCOP:ITEM:WIND2:TEXT 'comment'"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:HCOPy:ITEM:WINDow<1|2>:TRACe:STATe ON | OFF

This command selects the output of the currently displayed trace (ON).

Example: ":HCOP:ITEM:WIND:TRACe:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

The command :HCOPy:ITEM:WINDow<1|2>:TRACe:STATe OFF same as command :HCOPy:ITEM:ALL enables the output of the whole screen.

:HCOPy:ITEM:WINDow<1|2>:TRACe:CAINcrement ON | OFF

The command automatically changes the colour of the currently displayed trace after printout (ON).

Example: ":HCOP:ITEM:WIND:TRACe:CAIN ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

The automatic change of colour of the trace allows outputting to a plotter of several traces of the same diagram. For a better distinction, the colour of the trace is changed (”Color Auto Increment”).

:HCOPy:PAGE:DIMensions:QUADrant<1 to 4>

The command defines the quadrant which is allocated to the screen output.

Example: `":HCOP:PAGE:DIM:QUAD1"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The quadrants are defined as QUAD1 at the top right, QUAD2 at the top left, QUAD3 at the bottom left and QUAD4 at the bottom right. This command is an event which is why it is not assigned an *RST value and has no query.

:HCOPy:PAGE:DIMensions:FULL

This command defines that the full screen is to be printed out.

Example: `":HCOP:PAGE:DIM:FULL"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an event which is why it is not assigned and *RST value and has no query.

:HCOPy:PAGE:ORIentation<1|2> LANDscape | PORTrait

The command selects the format of the output (portrait and landscape) (hardcopy unit 1 or 2).

Example: `":HCOP:PAGE:ORI LAND"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

INITiate Subsystem

The INITiate subsystem checks the initialization of the trigger subsystem.
In receiver mode, a distinction is made between single measurement (INITiate1) and scan (INITiate2). In signal analyzer mode, a distinction is made between INITiate1 (screen A) and INITiate2 (screen B) in split screen representation.

COMMAND	PARAMETERS	UNIT	COMMENT
INITiate<1|2>			
:CONTinuous	<boolean>	--	
:CONMeas	--	--	no query
[:IMMediate]	--	--	no query
:DISPlay	<boolean>	--	

:INITiate<1|2>:CONTinuous ON | OFF

This command determines if the trigger system is continuously initiated ("Free Run").

Example: `":INIT:CONT OFF"`

Features: *RST value: ON
SCPI: conforming

Modes: R, A, VA

Setting "`INITiate:CONTinuous ON`" corresponds to function SCAN/SWEEP CONTinuous, ie the scan/sweep of the receiver/analyzer is cyclically repeated. The setting "`INITiate:CONTinuous OFF`" corresponds to function SCAN/SWEEP SINGLE.

:INITiate<1|2>:CONMeas

This command continues the sweep from the current sweep position.

Example: `":INIT:CONM"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an event and therefore has no *RST value and no query.
Sweeps are stopped, for example, in the case of transducer sets between the different transducer factors.

:INITiate<1|2>[:IMMediate]

The command initiates a new sweep/scan or starts a single sweep/scan.

Example: `"INIT"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:INITiate<1|2>:DISPlay ON | OFF

This command switches the display on or off during a single sweep.

Example: `":INIT:DISP OFF"`

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

INPut Subsystem

The INPut subsystem checks the input features of the instrument. In receiver mode, the suffix is not significant. In analyzer mode, a distinction is made between INPut1 (screen A) and INPut2 (screen B) in the split-screen representation.

COMMAND	PARAMETERS	UNIT	COMMENT
INPut<1|2>			
:ATTenuation	<numeric_value>	DB	
:AUTO	<Boolean>	--	
:MODE	NORMal | LNOise|LDIStortion	--	
:PROTection	<Boolean>	--	
:STEPsize	1 | 10	dB	Option 1-dB attenuator
:UPORt<1|2>			
[:VALue]?	--	--	query only
:STATe	<Boolean>	--	
:MIXer	<numeric_value>	DBM	
:COUPling	AC | DC		
:GAIN	<numeric_value>	dB	
:STATe	<Boolean>		
:AUTO	<Boolean>		
:BIMPedance	150 | 600 | 10000		
:PRESelection		OHM	
[:STATe]	<Boolean>		
:CATalog?			query only
:COUPling	<Boolean>		
:LOW			
:FREQuency	<numeric value>		
:SET	<string>	Hz	
:MID			
:SET	<string>		
:HIGH			
:FREQuency	<numeric value>		
:SET	<string>	Hz	
:SET	NARRow | NORMal | WIDE		
:USET			
[:SELect]	<string>		
:NAME	<string>		
:COMMent	<string>		
:LRANge			
[:DATA]	<numeric value>,<numeric value>..		
:MRANge		Hz	
[:DATA]	<numeric value>,<numeric value>..		
:CLEar		Hz	
:FILTer	<Boolean>		
[:STATe]			
:LPASs	<numeric value>		
[:FREQuency]		Hz	
:HPASs	<numeric value>		
[:FREQuency]		Hz	
:BPASs	<numeric value>,<numeric value>		
[:FREQuency]	RF | BALanced	Hz	
:TYPE			

:INPut<1|2>:ATTenuation 0 to 70 dB | 0 to 80 dB

This command programs the input attenuator.

Parameter: <numeric_value>::= 0 to 70 dB (without preselector /with preselection, frequency range 2 GHz to 22 GHz)
0 to 80 dB (with preselector, frequency range 100 Hz to 2 GHz)

Example: `":INP:ATT 40dB"`

Features: *RST value: - (AUTO is set to ON)
SCPI: conforming

Modes: R, A, VA

The attenuation of the input calibration line can be programmed in steps of 10 dB (frequency limits 2 GHz to 22 GHz, with/without preselection) or 1 dB (frequency limits 100 Hz to 2 GHz with preselection). If the attenuation is programmed directly, the coupling to the reference level is switched off.

:INPut<1|2>:ATTenuation:AUTO ON | OFF

This command automatically couples the input attenuation to the reference level. (analyzer) or activates the auto range function (receiver).

Example: `":INP:ATT:AUTO ON"`

Features: *RST value: ON
SCPI: conforming

Modes: R, A, VA

:INPut<1|2>:ATTenuation:AUTO:MODE NORMal | LNOise | LDIStortion

This command optimizes the coupling of the input attenuation to the reference level to high sensitivity or to high intermodulation immunity.

Example: `":INP:ATT:AUTO:MODE LDIS"`

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

For `LNOise`, the input attenuator value is set 10 dB lower than for `INPut:ATTenuation:AUTO:MODE NORMal`, for `LDIStortion` it is set 10 dB higher.

:INPut<1|2>:ATTenuation:PROTection ON | OFF

This command defines whether the 0 dB position of the attenuator is to be used in manual or automatic adjustment.

Example: `":INP:ATT:PROT ON"`

Features: *RST value: OFF
SCPI: device-specific

Mode: R

:INPut<1|2>:ATTenuation:STEPsize 1dB | 10dB

This command defines the attenuator stepsize.

Example: ":INP:ATT:STEP 1dB"

Features: *RST value: 10dB
SCPI: device-specific

Mode: A, VA, BTS, MS

Mode: R, A, VA

This command is only available in conjunction with option FSE-B13, 1-dB attenuator.

:INPut<1|2>:UPORt<1|2>[:VALue]?

This command queries the control lines of the user ports.

Example: ":INP:UPOR2?"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is a query command which is why it is not assigned an *RST value.

:INPut<1|2>:UPORt<1|2>:STATe ON | OFF

This command toggles the control lines of the user ports between INPut and OUTPut.

Example: ":INP:UPOR2:STAT ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

With ON, the user port is switched to INPut, with OFF to OUTPut.

:INPut<1|2>:MIXer -10 to -100 dBm

This command defines the nominal mixer level of the instrument.

Example: ":INP:MIX -30"

Features: *RST value: -
SCPI: device-specific

Modes: A, VA

:INPut<1|2>:COUPling AC | DC

This command switches the input coupling of the RF input between A.C. and D.C.

When using the unbalanced input or if the FSET is operated without the preselector, only DC coupling is possible.

Example: ":INP:COUP DC"

Features: *RST value: AC
SCPI: conforming

Modes: R, A, VA

:INPut<1|2>:GAIN 0 to 30dB

This command defines the input gain. If a value < 0 dB is set, the preamplifier will be automatically switched on, and if 0 dB is set, it will be automatically switched off.

Example: `":INP:GAIN 20dB"`

Features: *RST value: 30dB
SCPI: conforming

Modes: R, A, VA

The gain can be programmed in steps of 10 dB.

:INPut<1|2>:GAIN:STATe ON | OFF

This command switches on the preamplifier. The value defined with the command `INPut:GAIN` will be set as gain. The preselector has to be activated for the preamplifier to become effective. Therefore, the preselector is switched on automatically when the preamplifier is activated.

Example: `":INP:GAIN 20dB"`

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

:INPut<1|2>:GAIN:AUTO ON | OFF

This command includes the preamplifier into the autoranging function of the receiver.

Example: `":INP:GAIN:AUTO ON"`

Features: *RST value: OFF
SCPI: conforming

Mode: R

:INPut<1|2>:BIMPedance 150 | 600 | 10000

This command sets the impedance of the balanced input.

Example: `":INP:BIMP 600"`

Features: *RST value: 600
SCPI: conforming

Modes: A, VA, R

:INPut<1|2>:PRESelection[:STATe] ON | OFF

This command switches the preselection and preamplification on or off.

Example: `":INP:PRES:STAT ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: A, VA, R

:INPut<1|2>:PRESelection:CATalog?

This command reads out the names of all preselector data sets stored on the harddisk.

Syntax of output format:
<Sum of file lengths of all subsequent files>,<free memory on hard disk>,
<1st file name>,,<1st file length>,<2nd file name>,,<2nd file length>,....,<nth file name>,
<nth file length>

Example: `":INP:PRES:CAT?"`

Feature: *RST value: -
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:COUPling ON | OFF

This command switches the coupling of the preselection on or off.

Example: `":INP:PRES:COUP ON"`

Features: *RST value: ON
SCPI: device-specific

Modes: A, VA, R

:INPut<1|2>:PRESelection:COUPling:LOW:FREQuency 10Hz to 5MHz

This command sets the low resolution bandwidths while the preselector sets are coupled to the resolution bandwidth.

Example: `":INP:PRES:COUP:LOW:FREQ 5KHZ"`

Feature: *RST value: 2MHz
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:COUPling:LOW:SET <string>

This command selects a preselector set for the low resolution bandwidth, with the preselector set being coupled to the resolution bandwidth.

Parameter: <string> = 'Name of preselector set for low resolution bandwidth'

Example: `":INP:PRES:COUP:LOW:SET 'NARROW'"`

Feature: *RST value: NARROW
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:COUPling:MID:SET <string>

This command selects a preselector set for medium resolution bandwidth, with the preselector sets being coupled to the resolution bandwidth.

Parameter: <string> = 'Name of preselector set for medium resolution bandwidth'

Example: `":INP:PRES:COUP:MID:SET 'NORMAL'"`

Feature: *RST value: NORMAL
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:COUPling:HIGH:FREQuency 5MHz to 500MHz

Setting of high resolution bandwidth with preselector sets being coupled to the resolution bandwidth

Example: `":INP:PRES:COUP:HIGH:FREQ 50MHZ"`

Feature: *RST value: 5MHz
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:COUPling:HIGH:SET <string>

This command selects a preselector set for high resolution bandwidth, with the preselector sets being coupled to the resolution bandwidth.

Parameter: <string> = 'Name of preselector set for high resolution bandwidth'

Example: `":INP:PRES:COUP:LOW:SET 'WIDE'"`

Feature: *RST value: WIDE
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:SET NARRow | NORMal | WIDE

This command selects the preselector set.

Example: `":INP:PRES:SET WIDE"`

Features: *RST value: NORMal
SCPI: device-specific

Modes: A, VA, R

:INPut<1|2>:PRESelection:USET[:SELect] 'name of preselector user set'

This command selects the user defined preselector set.

Example: `":INP:PRES:USET 'set1'"`

Features: *RST value: -
SCPI: device-specific

Modes: A, VA, R

:INPut<1|2>:PRESelection:USET:NAME <Name of user-defined preselector set>

This command selects a user-defined preselector set to be entered anew or modified.

Example: `":INP:PRES:USET:NAME 'NEW_SET'"`

Feature: *RST value: -
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:USET:COMMent <Comment on preselector set>

This command determines the comment on the user-defined preselector set.
At first, the name must be selected by entering the command
"INPut:PRESelector:USET:NAME„.

Example: ":INP:PRES:USET:COMM 'Set for dipole antenna'"

Feature: *RST value: -
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:USET:LRANGe[:DATA] <num_value>,<num_value>,<num_value>...

This command defines the values of the lower preselector range (low range) for a user-defined preselector set.

Each entry comprises three parameters: minimum frequency (start frequency), highpass filter, lowpass filter. The maximum frequency for the last entry is set automatically to 40 MHz.
At first, the name must be selected by entering the command
"INPut:PRESelector:USET:NAME„.

Example: ":INP:PRES:USET:LRAN 0,100,20KHZ,2KHZ,1KHZ,50KHZ"
This selects two entries in the lower range:
1. 0Hz - 2KHz, HP = 100Hz, LP = 20KHz
2. 2KHz - 40MHz, HP = 1KHz, LP = 50KHz

Feature: *RST value: -
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:USET:MRANGge[:DATA] <num_value>,<num_value>,<num_value>..

This command defines the values of the medium preselector range (mid range) for a user-defined preselector set.

Each entry comprises three parameters: minimum frequency (start frequency), start frequency of the wideband filter, stop frequency of the wideband filter. The maximum frequency for the last entry is set automatically to 2 GHz.
At first, the name must be selected by entering the command
"INPut:PRESelector:USET:NAME„.

Example: ":INP:PRES:USET:MRAN 40MHZ,40MHZ,65MHZ,100MHZ,160MHZ,250MHZ"
This selects two entries in the medium range:
1. 40MHz - 100MHz, BP = 40MHz - 65MHz
2. 100MHz - 2GHz, BP = 160MHz - 250MHz

Feature: *RST value: -
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:USET:CLEar

This command deletes the selected preselector set. The set to be deleted must first be selected by the command `INPut:PRESelector:USET:NAME`.

Example: "`PRES:USET:CLE`"

Feature: *RST value: -
SCPI: device-specific

Mode: R, A, VA

This command is an event and therefore has no *RST value.

:INPut<1|2>:PRESelection:FILTer[:STATe] ON | OFF

This command defines if the filters are entered by the following commands (ON) or selected via a preselector set (OFF).

Example: "`:INP:PRES:FILT ON`"

Feature: *RST value: OFF
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:FILTer:LPASs[:FREQuency] 20KHz to 40MHz

This command selects a lowpass for the preselection. The following values can be set:
20KHz, 50KHz, 100KHz, 500KHz, 1MHz, 5MHz, 40MHz

Example: "`:INP:PRES:FILT:LPAS:FREQ 500KHZ`"

Feature: *RST value: 50KHz
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:FILTer:HPASs[:FREQuency] 100Hz to 5MHz

This command selects a highpass for the preselection. The following values can be set:
100Hz, 1KHz, 5KHz, 20KHz, 50KHz, 100KHz, 200KHz, 1MHz, 2MHz, 5MHz

Example: "`:INP:PRES:FILT:HPAS:FREQ 20KHZ`"

Feature: *RST value: 100Hz
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:PRESelection:FILTer:BPASs[:FREQuency] 40MHz to 1Ghz, 65MHz to 2GHz

This command selects a wideband filter for the preselection. The following values can be set:
(40MHz, 65MHz), (65MHz, 100MHz), (100MHz, 160MHz), (160MHz, 250MHz), (40MHz, 100MHz), (100MHz, 200MHz), (200MHz, 500MHz), (500MHz, 1GHz), (1GHz, 2GHz), (40MHz, 2GHz)

Example: "`:INP:PRES:FILT:BPAS:FREQ 100MHZ,160MHZ`"

Feature: *RST value: 40MHz,2GHZ
SCPI: device-specific

Mode: R, A, VA

:INPut<1|2>:TYPE RF | BALanced

This command selects the type of input.

Example: `":INP:TYPE BAL"`

Features: *RST value: RF
SCPI: conforming

Modes: A, VA, R

INSTrument Subsystem

The INSTrument subsystem selects the operating mode of the unit either via text parameters or fixed numbers. In the receiver mode, the suffix is not significant. In the analyzer and vector analyzer mode, a distinction is made between INSTrument1 (screen A) and INSTrument2 (screen B) in the split-screen representation.

COMMAND	PARAMETERS	UNIT	COMMENT
INSTrument<1|2>			
[:SELect]	SANalyzer | DDEMod | RECeiver		Vector Signal Analysis
:NSELect	<numeric_value>		
:COUPled	NONE | MODE | X | Y | CONTrol | XY | XCONtrol | YCONtrol | ALL		

:INSTrument<1|2>[:SELect] SANalyzer | DDEMod | RECeiver

This command switches between the operating modes by means of text parameters.

Parameter: SANalyzer: spectrum analysis
DDEMod: vector signal analysis, digital demodulation
RECeiver: receiver mode

Example: `":INST DDEM"`

Features: *RST value: SANalyzer
SCPI: conforming

Modes: R, A, VA

:INSTrument<1|2>:NSELect 1 | 2 | 6

This command switches between the two modes by means of numbers.

Example: `":INST:NSEL 2"`

Features: *RST value: 1
SCPI: conforming

Modes: R, A, VA
1: spectrum analysis
2: vector signal analysis, digital demodulation
6: receiver mode

:INSTrument<1|2>:COUPle NONE | MODE | X | Y | CONTrol | XY | XCONtrol | YCONtrol | ALL

This command defines the coupling between the two measurement windows screen A and B.

Example: `":INST:COUP NONE"`

Features: *RST value: ALL
SCPI: device specific

Modes: A, VA

NONE	no coupling.
MODE	the operating mode of the two screens is coupled.
X or Y	the scaling of the x- or y axis of the two screens is coupled.
CONTrol	the trigger and gate parameter ,and the sweep parameters SINGle/ CONTinous and COUNt of the two screens are coupled.
XY	the scaling of the x- and y-axis of the two screens are coupled.
XCONTrol bzw. YCONTrol	the trigger and gate parameter, and the sweep parameters SINGle CONTinous/ COUNt of the two screens are coupled.
ALL	the scaling of the x- or y axis the trigger and gate parameter and the sweep parameters SINGle/ CONTinous/ COUNt of the two screens are coupled.

The numeric suffix in INSTrument<1|2> is not significant.

MMEMory Subsystem

The MMEMory (mass memory) subsystem provides commands which allow for access to the storage media of the instrument and for storing and loading various instrument settings.
The NAME command stores the HCOPy outputs in a file.

The various drives can be addressed via the mass storage unit specifier <msus> using the conventional DOS syntax. The internal hard disk is addressed by "C:", the floppy-disk drive installed by "A:".
The file names <file_name> are indicated as string parameters with the commands being enclosed in quotation marks. They correspond to the DOS conventions.

DOS file names consist of max. 8 ASCII characters and an extension of up to three characters separated from the file name by a colon "." Both, the colon and the extension are optional. The colon is not part of the file name. DOS file names do not differ between uppercase and lowercase notation. All letters and digits are permitted as well as the special characters "_", "^", "$", "~", "!", "#", "%", "&", "-", "{", "}", "(", ")", "@" and "`". Reserved file names are CLOCK$, CON, AUX, COM1 to COM4, LPT1 to LPT3, NUL and PRN.

The two characters "*" and "?" have the function of so-called "wildcards", i.e., they are variables for selection of several files. The question mark "?" replaces exactly one character which may be any, the asterisk means any of the remaining characters in the file name. "*.*" thus means all files in a directory.

COMMAND	PARAMETERS	UNIT	COMMENT
MMEMory			
:CATalog	<string>		
:CDIRectory	<directory_name>	--	
:COPY	<file_name>,<file_name>	--	no query
:DATA	<file name>[,<block>]	--	
:DELete	<file_name>	--	no query
:INITialize	<msus>	--	no query
:LOAD			
:STATe	1,<file_name>	--	no query
:AUTO	1,<file_name>	--	no query
:MDIRectory	<directory_name>	--	no query
:MOVE	<file_name>,<file_name>	--	no query
:MSIS	<msus>	--	
:NAME	<file_name>	--	
:RDIRectory	<directory_name>	--	no query
:STORe			
:STATe	1,<file_name>	--	no query
:TRACe	<numeric_value>, <file_name>		no query
:FINal	<final_name>		no query
:CLEar			
:STATe	1,<file_name>	--	no query
:ALL			no query

COMMAND	PARAMETERS	UNIT	COMMENT
MMEMory			
:SELect			
[:ITEM]			
:GSETup	<Boolean>		
:HWSettings	<Boolean>		
:TRACE<1 to 4>	<Boolean>		
:LINes			
[:ACTive]	<Boolean>		
:ALL	<Boolean>		
:CSETup	<Boolean>		
:HCOPy	<Boolean>		
:MACRos	<Boolean>		
:TRANsducer			
[:ACTive]	<Boolean>		
:ALL	<Boolean>		
:ALL	--		no query
:NONE	--		no query
:DEFault	<string>		no query
:COMMent			

:MMEMory:CATalog? <string>

This command reads out the files and subdirectories of the current directory. A mask, eg "*.bat", can be defined so that only files with "bat" as extension are selected.

Syntax of output format:

<Sum of file lengths of all subsequent files>,<free memory on hard disk>,
<1st file name or name of subdirectory>,<file or subdirectory>,<1st file length>,<2nd file name or name of subdirectory>,<file or subdirectory>,<2nd file length>,....,<nth file name>,<file or subdirectory>, <nth file length>.

<File or subdirectory>: in case of a file the field remains empty,
in case of a subdirectory the field contains "DIR".

Parameter: <string>::= DOS file name

Example: `":MMEM:CAT 'rem?.lin'"`

Characteristics: *RST value: -
SCPI: conformal

Modes: R, A, VA

:MMEMory:CDIRectory <directory_name>

This command changes the current directory.

Parameter: <directory_name>::= DOS path name

Example: `":MMEM:CDIR 'C:\USER\DATA'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

In addition to the path name, the indication of the directory may contain the drive name. The path name complies with the DOS conventions.

:MMEMory:COPY <file_source>,<file_destination>

This command copies the files indicated.

Parameter: <file_source>,<file_destination> ::= <file_name>
<file_name> ::= DOS file name

Example: `":MMEM:COPY 'C:\USER\DATA\SETUP.CFG','A:'"`

Features: *RST value: -
SCPI conforming

Modes: R, A, VA

The indication of the file name may include the path and the drive. The file names and path information must be in accordance with the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:DATA <file_name>[,<block data>]

This command writes block data to the specified file.

Syntax: :MMEMory:DATA <file_name>,<block data>
:MMEMory:DATA? <file_name>

Example: `":MMEM:DATA? 'TEST01.HCP'"`
`":MMEM:DATA 'TEST01.HCP', #217This is the file"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

Each <block> starts with the character '#', followed by the value stating the length of the length information. This is followed by one or several characters for the length information; then come the data themselves. The end character must be set to EOI to ensure correct data transmission.

:MMEMory:DELete <file_name>

This command deletes the files indicated.

Parameter: <file_name> ::= DOS file name

Example: `":MMEM:DEL 'TEST01.HCP'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The indication of the file name contains the path and, optionally, the drive. Indication of the path corresponds to the DOS conventions. The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:INITialize 'A:'

This command formats the disk in drive A.

Example: `":MMEM:INIT 'A:'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

Formatting deletes all data stored on the floppy disk. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:LOAD:STATe 1,<file_name>

This command loads instrument settings from files.

Parameter: <file_name> ::= DOS file name without extension

Example: `":MMEM:LOAD:STAT 1,'A:TEST'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The contents of the file is loaded and set as new instrument state. The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:LOAD:AUTO 1,<file_name>

This command defines which device setting is automatically loaded after the instrument is switched on.

Parameter: <file_name> ::= DOS file name without extension;
FACTORY denotes the data set previously in the instrument

Example: `":MMEM:LOAD:AUTO 1,'C:\USER\DATA\TEST'"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

The contents of the file are read after switching on the instrument and used to define the new device state. The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:MDIRectory <directory_name>

This command creates a new directory.

Parameter: <directory_name>::= DOS path name

Example: `":MMEM:MDIR 'C:\USER\DATA'"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:MOVE <file_source>,<file_destination>

This command renames existing files.

Parameter: <file_source>,<file_destination> ::= <file_name>
<file_name> ::= DOS file name

Example: `":MMEM:MOVE 'TEST01.CFG','SETUP.CFG'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:MSIS 'A:' | 'C:'

This command changes to the drive indicated.

Example: `":MMEM:MSIS 'A:'"`

Features: *RST value: "C:'
SCPI: conforming

Modes: R, A, VA

The drive may be the internal hard disk C: or the floppy-disk drive A:. The drive is indicated according to the DOS conventions.

:MMEMory:NAME <file_name>

This command specifies a file which is printed or plotted to.

Parameter: <file_name> ::= DOS filename

Example: `":MMEM:NAME 'PLOT1.HPG'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The file name includes indication of the path and may also include the drive. The file name and path information correspond to the DOS conventions. The output to the printer is routed into a file using the command `":HCOP:DEST 'MMEM"`'.

:MMEMory:RDIRectory <directory_name>

This command deletes the directory indicated.

Parameter: <directory_name>::= DOS path name

Example: `":MMEM:RDIR 'C:\TEST'"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

The directory name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:STORe:STATe 1,<file_name>

This command stores the current instrument setting in a file.

Parameter: <file_name> ::= DOS file name without extension

Example: `":MMEM:STOR:STAT 1,'TEST'"`

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The current instrument state is stored as a file. The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:STORe:TRACe 1 to 4,<file_name>

This command stores the selected trace (1 to 4) in ASCII format in a file.

Parameter: 1 to 4 := selected trace 1 to 4
<file_name> := DOS file name

Example: `":MMEM:STOR:TRAC 3,'A:\TEST.ASC'"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

The file name includes indication of the path and may also include the drive. The file name and path information correspond to the DOS conventions.This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:CLEar:STATe 1,<file_name>

This command deletes the instrument setting denoted by <file_name>.

Parameter: <file_name> ::= DOS file name without extension

Example: `":MMEM:CLE:STAT 1,'TEST'"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

The device data set specified is deleted. The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

MMEMory:STORe:FINal <file_name>

This command stores the final measurement data in an ASCII file.

Parameter: <file_name> := DOS file name

Example: `"MMEM:STOR:FIN 'A:\TEST.ASC'"`

Features: *RST value: -
SCPI: device-specific

Mode: R

The file name includes indication of the path and may also include the drive. The path name corresponds to the DOS conventions. This command is an event which is why it is not assigned an *RST value and has no query.

:MMEMory:CLEar:ALL

This command deletes all instrument settings in the current directory.

Example: ":MMEM:CLE:ALL"

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

This command is an "event" which is why it is not assigned an *RST value and has no query.

:MMEMory:SELect[:ITEM]:GSETup ON | OFF

This command includes the data of the general setup in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:GSET ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:MMEMory:SELect[:ITEM]:HWSettings ON | OFF

This command includes the hardware settings in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:HWS ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

Level and frequency lines are stored with this command as well.

:MMEMory:SELect[:ITEM]:TRACe<1 to 4> ON | OFF

This command includes the data of the selected trace in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:TRACE3 ON"

Features: *RST value: OFF for all Traces
SCPI: device-specific

Modes: R, A, VA

:MMEMory:SELect[:ITEM]:LINes[:ACTive] ON | OFF

This command includes the active limit lines in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:LIN ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

Upon MMEM:LOAD the limit lines which are not currently active but contained in the data set are restored as well.

:MMEMory:SELect[:ITEM]:LINes:ALL ON | OFF

This command includes all limit lines in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:LIN:ALL ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

This command includes selection of the active limit lines.

:MMEMory:SELect[:ITEM]:CSETup ON | OFF

This command includes the current color setting in the list of partial datasets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:CSET ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

:MMEMory:SELect[:ITEM]:HCOPy ON | OFF

This command includes the hardcopy settings in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:HCOPy ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

:MMEMory:SELect[:ITEM]:MACRos ON | OFF

This command includes the keyboard macros in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:MACRos ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

:MMEMory:SELect[:ITEM]:TRANsducer[:ACTive] ON | OFF

This command includes the active transducer factors and set in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:TRAN ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

Upon MMEM:LOAD the transducer factors and sets which are not currently active but contained in the data set are restored as well.

:MMEMory:SELect[:ITEM]:TRANsducer:ALL ON | OFF

This command includes all transducer factors and sets in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:TRAN:ALL ON"

Features: *RST value: ON
SCPI: device-specific

Modes: R, A, VA

This command is an event and therefore has no *RST value assigned.

:MMEMory:SELect[:ITEM]:ALL

This command includes all data subsets in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:ALL"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is an event and therefore has no *RST value assigned.

:MMEMory:SELect[:ITEM]:NONE

This command deletes all data subsets in the list of data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:NONE"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is an event and therefore has no *RST value assigned.

:MMEMory:SELect[:ITEM]:DEFault

This command sets the default list of the data subsets of a device setting to be stored/loaded.

Example: ":MMEM:SEL:DEFault"

Features: *RST value: --
SCPI: device-specific

Modes: R, A, VA

This command is an event and therefore has no *RST value assigned.

:MMEMory:COMMent <string>

This command defines a comment for a device setting to be stored.

Example: ":MMEM:COMM 'Setup for GSM measurement'"

Features: *RST value: blank comment
SCPI: device-specific

Modes: R, A, VA

OUTPut Subsystem

The OUTPut subsystem checks the output features of the instrument.

COMMAND	PARAMETERS	UNIT	COMMENT
OUTPut<1|2>			
:UPORt<1|2>			
[:VALue]	<binary>	--	
:STATe	<boolean>	--	

:OUTPut<1|2>UPORt<1|2>[:VALue] #B00000000 to #B11111111

This command sets the control lines of the user ports.

Example: `":OUTP:UPOR2 #B10100101"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

User port 1 or 2 is written with the given binary pattern. If the user port is programmed to INPut instead of OUTPut, the output value is temporarily stored.

:OUTPut<1|2>UPORt<1|2>:STATe ON | OFF

This command switches the control line of the user ports between INPut and OUTPut.

Example: `":OUTP:UPOR:STAT ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A, VA

With ON, the user port is switched to OUTPut, with OFF to INPut.

SENSe Subsystem

The SENSe subsystem is itself divided up into several subsystems. The commands of these subsystems directly control device-specific settings, they do not refer to the signal characteristics of the measurement signal.
The SENSe subsystem controls the essential parameters of the analyzer, vector analyzer and receiver. In accordance with the SCPI standard, it is for this reason optional, which means that it is not necessary to include the SENSe node in command sequences.

In the split-screen representation, a distinction is made between SENSe1 and SENSe2:

SENSe1 ≙ screen A;

SENSe2 ≙ screen B

SENSe:AVERage Subsystem

The SENSe:AVERage subsystem calculates the average of the data acquired. A new test result is obtained from various successive measurements. The amount of test points and the axis reference of the new result correspond to those of the original measurements.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:AVERage			
:COUNt	<numeric_value>	--	
:AUTO	<Boolean>	--	
[:STATe]	<Boolean>	--	
:TYPE	MAXimum | SCALar	--	

:[SENSe<1|2>:]AVERage:COUNt 0 to 32767

The command specifies the number of measurements which are combined.

Example: ":AVER:COUN 16"

Features: ***RST value:** 0
SCPI: conforming

Modes: R, A, VA

:[SENSe<1|2>:]AVERage:COUNt:AUTO ON | OFF

AUTO ON selects a suitable number of :COUNt for the respective type of measurement.

Example: ":AVER:COUN:AUTO ON"

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

:[SENSe<1|2>:]AVERage[:STATe] ON | OFF

The command switches on or off the average function.

Example: "AVER OFF"

Features: ***RST value:** OFF
SCPI: conforming

Modes: R, A, VA

:[SENSe<1|2>:]AVERage:TYPE MAXimum | MINimum | SCALar

The command selects the type of average function.

Example: ":AVER:TYPE SCAL"

Features: ***RST value: SCALar
SCPI: conforming

Modes: R, A, VA

The following average functions have been defined:

MAXimum (MAX HOLD): $AVG(n) = MAX(X_1 \text{ to } .X_n)$

MINimum (MIN HOLD): $AVG(n) = Min(X_1 \text{ to } .X_n)$

SCALar (AVERAGE): $AVG(n) = \frac{1}{n} \times \sum_{=}^{n} xi$

SENSe:BANDwidth Subsystem

This subsystem controls the setting of the instrument's filter bandwidths. Both groups of commands (BANDwidth and BWIDth) perform the same functions.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:BANDwidth | :BWIDth			
[:RESolution]	<numeric_value>	HZ	
:AUTO	<Boolean>	--	
:RATio	<numeric_value>	--	
:VIDeo	<numeric_value>	HZ	
:AUTO	<Boolean>	--	
:EXTernal			
[:STATe]	<Boolean>		
:RATio	<numeric_value> | SINe | PULSe | NOISe		
:DEMod	<numeric_value>		
:COUPling	<Boolean>		
:PLL	AUTO | HIGH | MEDium | LOW		

:[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution] 10 Hz to 500 MHz

This command defines the IF bandwidth (6-dB bandwidth) of the receiver or resolution bandwidth of the analyzer.

Example: `":BAND 1MHz"`

Features: *RST value: - (AUTO is set to ON)
SCPI: conforming

Modes: R, A, VA

In the receiver mode, FSET offers the IF bandwidths (6 dB bandwidths) 10 Hz, 100 Hz, 200 Hz, 1 kHz, 9 kHz, 10 kHz, 100 kHz, 120 kHz, 1 MHz and 10 MHz. The bandwidth is limited by the set receive frequency: RES BW $\leq f_E/2$.

In the analyzer and vector analyzer mode, the values for the resolution bandwidth are rounded in 1 | 2 | 5 steps.

:[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:AUTO ON | OFF

This command either automatically couples the resolution bandwidth of the instrument to the span or cancels the coupling.

Example: `":BAND:AUTO OFF"`

Features: *RST value: ON
SCPI: conforming

Modes: R, A, VA

The automatic coupling matches the resolution bandwidth to the currently set span according to the relationship between span and resolution bandwidth.

:[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:RATio 0.0001 to 1

This command defines the ratio resolution bandwidth (Hz) / span (Hz).

Example: ":BAND:RAT 0.1"

Features: *RST value: -- (AUTO is set to ON)
SCPI: conforming

Modes: A, VA, R

The ratio to be entered is reciprocal to the ratio Span/RBW used in manual control.

:[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo 1Hz to 500MHz

This command defines the instrument's video bandwidth.

Example: ":BAND:VID 10kHz"

Features: *RST value: - (AUTO is set to ON)
SCPI: conforming

Mode: A

The values for the video bandwidth are rounded in 1 | 2 | 5 steps.

:[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:AUTO ON | OFF

This command either automatically couples the instrument's video bandwidth to the resolution bandwidth or cancels the coupling.

Example: ":BAND:VID:AUTO OFF"

Features: *RST value: ON
SCPI: conforming

Mode: A

:[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:RATIO 0.001to 1000 | SINe | PULSe | NOISe

This command defines the ratio video bandwidth (Hz) / resolution bandwidth (Hz).

Parameter: The parameters SINe, PULSe and NOISe may be used as synonyms for the following values:
SINe: 1
PULSe: 10
NOISe: 0.1

Example: ":BAND:VID:RAT 10"

Features: *RST value: - (AUTO is set to ON)
SCPI: conforming

Modes: A, VA

The ratio to be entered is reciprocal to the ratio RBW/VBW used in manual control.

:[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:EXTernal[:STATe] ON | OFF

This command switches on the external video signal.

Example: ":BAND:VID:EXT:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: A, VA

:[SENSe<1|2>:]BANDwidth|BWIDth:DEMod <num_value>

This command defines the audio bandwidth for AF demodulation. Entries for audio bandwidth can always be made, however they influence the RBW only if `SENSe:BWIDth:DEMod:COUPling` is ON.

Example: `"BAND:DEM 100KHz"`

Features: *RST value: 10 kHz
SCPI: device-specific

Mode: A, R

The values for the demodulation bandwidth are rounded in steps of 1 | 2 | 5.

[SENSe<1|2>:]BANDwidth|BWIDth:DEMod:COUPling ON | OFF

This command deactivates (OFF) the coupling between receiver bandwidth and IF bandwidth for the demodulation signal path.

Example: `"BAND:DEM:COUP ON"`

Features: *RST value: OFF
SCPI: device-specific

Mode: A, R

:[SENSe<1|2>:]BANDwidth|BWIDth:PLL AUTO | HIGH | MEDium | LOW

This command defines the bandwidth of the main PLL of the instrument synthesizer.

Example: `":BAND:PLL HIGH"`

Features: *RST value: AUTO
SCPI: device-specific

Mode: A

SENSe:CORRection-Subsystem

The SENSe:CORRection-subsystem controls the correction of measured results by means of frequency-dependent correction factors (e. g. for antenna or cable attenuation).

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:TRANsducer			
:CATalog?			query only
:ACTive?			query only
:SELect	<name>		
:UNIT	<string>		
:SCALing	LINear | LOGarithmic		
:COMMent	<string>		
:DATA	<freq> , <level> ..	HZ , --	
[:STATe]	<Boolean>		
:DELete	--	--	no query
:TSET			
:CATalog?			query only
:ACTive?			query only
:SELect	<name>		
:UNIT	<string>		
:BREak	<Boolean>		
:COMMent	<string>		
:RANGe<1 to 10>	<freq> , <freq>, <name> ..	HZ, HZ, --	
[:STATe]	<Boolean>		
:DELete	--	--	no query

:[SENSe<1|2>:]CORRection:TRANsducer:CATalog?

This command reads out the names of all transducer factors stored on the harddisk.

Syntax of output format:

<Sum of file lengths of all subsequent files>,<free memory on hard disk>,
<1st file name>,,<1st file length>,<2nd file name>,,<2nd file length>,....,<nth file name>,
<nth file length>

Example: `":CORR:TRAN:CAT?"`

Feature: *RST value: -
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]CORRection:TRANsducer:ACTive?

This command reveals the active transducer factor. If no transducer factor is switched on, a void string will be output.

Example: `":CORR:TRAN:ACT?"`

Feature: *RST value: -
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]CORRection:TRANsducer:SELect <name>

This command selects the transducer factor designated by <name>. If <name> does not exist yet, a new transducer factor is created.

Parameter: <name>::= Name of the transducer factor in string data form with a maximum of 8 characters.

Example: `":CORR:TRAN:SEL 'FACTOR1'"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command must be sent prior to the subsequent commands for modifying/activating transducer factors.

:[SENSe<1|2>:]CORRection:TRANsducer:UNIT <string>

This command defines the unit of the transducer factor selected.

Parameter: <string>::= 'DB' | 'DBM' | 'DBMV' | 'DBUV' | 'DBUV/M' | 'DBUA' 'DBUA/M' | 'DBPW' | 'DBPT'

Example: `":CORR:TRAN:UNIT 'DBUV'"`

Features: *RST value: 'DB'
SCPI: device-specific

Modes: R, A

Prior to this command, the command `SENS:CORR:TRAN:SEL` must be sent.

:[SENSe<1|2>:]CORRection:TRANsducer:SCALing LINear | LOGarithmic

This command defines whether the frequency scaling of the transducer factor is linear or logarithmic.

Example: `":CORR:TRAN:SCAL LOG"`

Features: *RST value: LINear
SCPI: device-specific

Modes: R, A

Prior to this command, the command `SENS:CORR:TRAN:SEL` must be sent.

:[SENSe<1|2>:]CORRection:TRANsducer:COMMent <string>

This command defines the comment for the selected transducer factor.

Example: `":CORR:TRAN:COMM 'FACTOR FOR ANTENNA'"`

Features: *RST value '' (empty comment)
SCPI: device specific

Modes: R, A

Prior to this command, the command `SENS:CORR:TRAN:SEL` must be sent.

:[SENSe<1|2>:]CORRection:TRANsducer:DATA <freq>,<level>..

This command defines the test points for the selected transducer factor. The values are entered as a series of frequency/level pairs. The frequencies must be in ascending order.

Example: `":CORR:TRAN:TRANsducer:DATA 1MHZ,-30,2MHZ,-40"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A

Prior to this command, the command `SENS:CORR:TRAN:SEL` must be sent. The level values are sent as dimensionless numbers; the unit is specified by means of the command `SENS:CORR:TRAN:UNIT`.

:[SENSe<1|2>:]CORRection:TRANsducer[:STATe] ON | OFF

This command switches the selected transducer factor on or off.

Example: `":CORR:TRAN ON"`

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A

Prior to this command, the command `SENS:CORR:TRAN:SEL` must be sent.

:[SENSe<1|2>:]CORRection:TRANsducer:DELete

This command deletes the selected transducer factor.

Example: `":CORR:TRAN:DEL"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event and therefore has no *RST value.
Prior to this command, the command `SENS:CORR:TRAN:SEL` must be sent.

:[SENSe<1|2>:]CORRection:TSET:CATalog?

This command polls the names of all transducer factors stored on the harddisk.

Syntax of output format:

<Sum of file lengths of all subsequent files>,<free memory on hard disk>,
<1st file name>,,<1st file length>,<2nd file name>,,<2nd file length>,....,<nth file name>,
<nth file length>

Example: `":CORR:TSET:CAT?"`

Modes: R, A

Feature: *RST value: -
SCPI: device-specific

:[SENSe<1|2>:]CORRection:TSET:ACTive?

This command reveals the active transducer set. If no transducer set is switched on, a void string will be output.

Example: ":CORR:TSET:ACT?"

Feature: *RST value: -
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]CORRection:TSET:SELect <name>

This command selected the transducer set designated by <name>. If <name> does not exist yet, a new set is created.

Parameter: <name>::= name of the transducer set in string data form with a maximum of 8 characters.

Example: ":CORR:TSET:SEL 'SET1'"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command must be sent prior to the subsequent commands for changing/activating the transducer sets.

:[SENSe<1|2>:]CORRection:TSET:UNIT 'DB' | 'DBM' | 'DBUV' | 'DBUV/M' | 'DBUA' | 'DBUA/M'' | DBPW' | 'DBPT <string>

This command defines the unit of the selected transducer sets. When assigning transducer factors to the set, only factors which are compatible to the selected unit, i. e. factors with the same unit or the unit dB, are allowed.

Example: ":CORR:TSET:UNIT 'DBUV'"

Features: *RST value: 'DB'
SCPI: device-specific

Modes: R, A

Prior to this command, the command SENS:CORR:TSET:SEL must be sent.

:[SENSe<1|2>:]CORRection:TSET:BREak ON | OFF

This command defines if the sweep is to be stopped on changeover from range to another.

Example: ":CORR:TSET:BRE ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A

Prior to the above command, the command SENS:CORR:TSET:SEL must be sent.

:[SENSe<1|2>:]CORRection:TSET:COMMent <string>

This command defines the comment for the selected transducer set.

Example: ":CORR:TSET:COMM 'SET FOR ANTENNA'"

Features: *RST value '' (empty comment)
SCPI: device specific

Modes: R, A

Prior to this command, the command SENS:CORR:TSET:SEL must be sent.

:[SENSe<1|2>:]CORRection:TSET:RANGe<1 to 10> <freq>,<freq>,<name>..

This command defines a partial range of the selected transducer set. The partial range is determined by its start and stop frequencies plus a list of names of the assigned transducer factors. The ranges 1 to 10 must be sent in ascending order.

Parameter: <freq>,<freq>::= start frequency, stop frequency of the range
<name>...::= list of names for the assigned transducer factors. The individual names must be characterized by single quotation marks (') and separated by commas.

Example: ":CORR:TRAN:TSET:RANG 1MHZ,2MHZ,'FACTOR1,'FACTOR2'"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

Prior to this command, the command SENS:CORR:TSET:SEL must be sent.

:[SENSe<1|2>:]CORRection:TSET[:STATe] ON | OFF

This command switches the selected transducer set on or off.

Example: ":CORR:TSET ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: R, A

Prior to this command, the command SENS:CORR:TSET:SEL must be sent.

:[SENSe<1|2>:]CORRection:TSET:DELete

This command deletes the selected transducer set.

Example: ":CORR:TSET:DEL"

Features: *RST value: -
SCPI: device-specific

Modes: R, A

This command is an event and thus has no *RST value assigned.
Prior to this command, the command SENS:CORR:TSET:SEL must be sent.

SENSe:DEMod, SENSe:AM, SENSe:FM and SENSe:PM Subsystems

The SENSe:DEMod, SENSe:AM, SENSe:FM and SENSe:PM subsystems control the analog demodulation of the video signal.

COMMAND	PARAMETER	UNIT	COMMENT
[SENSe<1|2>]			
:DEMod	OFF | AM | AMVideo | FM | PM		
:FILTer			
[:LPASs]			
:FREQuency	<numeric_value>	HZ	
:AUTO	<Boolean>	--	
:HPASs			
:FREQuency	0 | 10 | 100 | 1000	HZ	
:AM			
:RANGe			
[:UPPer]	3 | 10 | 100	PCT	
:FM			
[:DEViation]			
:RANGe			
[:UPPer]	<numeric_value>	HZ	
:AUTO	<Boolean>	--	
:PM			
[:DEViation]			
:RANGe			
[:UPPer]	<numeric_value>	DEG	

:[SENSe<1|2>:]DEMod OFF | AM | AMVideo | FM | PM

This command selects the type of analog demodulation.

Example: `":DEM FM"`

Features: *RST value: OFF
SCPI: device-specific

Mode: AF-D

:[SENSe<1|2>:]DEMod:FILTer[:LPASs]:FREQuency <numeric_value>

This command selects the audio low-pass filter for demodulation.

Example: `":DEM:FILT:FREQ 300KHz"`

Features: *RST value: -
SCPI: device-specific

Mode: AF-D

:[SENSe<1|2>:]DEMod:FILTer[:LPASs]:AUTO ON | OFF

This command either automatically couples the audio low-pass filters to the selected resolution bandwidth or cancels the coupling.

Example: ":DEM:FILT:AUTO OFF"

Features: *RST value: ON
SCPI: device-specific

Mode: AF-D

:[SENSe<1|2>:]DEMod:FILTer:HPASs:FREQuency 0HZ| 10HZ | 100HZ | 1KHZ

This command selects the audio high-pass filters for cutting off the D.C.-current.

Example: ":DEM:FILT:HPAS:FREQ 100"

Features: *RST value: -
SCPI: device-specific

Mode: AF-D

:[SENSe<1|2>:]AM:RANGe[:UPPer] 3PCT | 10PCT | 100PCT

This command selects the range for the AM demodulation.

Example: ":AM:RANG 10"

Features: *RST value: -
SCPI: conforming

Mode: AF-D

:[SENSe<1|2>:]FM[:DEViation]:RANGe[:UPPer] <numeric_value>

This command selects the display range for the FM demodulation.

Example: ":FM:RANG 300kHZ"

Features: *RST value: -
SCPI: conforming

Mode: AF-D

:[SENSe<1|2>:]FM[:DEViation]:RANGe:AUTO ON | OFF

This command couples the display range for the FM demodulation to the selected resolution bandwidth.

Example: ":FM:RANG:AUTO OFF"

Features: *RST value: ON
SCPI: conforming

Mode: AF-D

:[SENSe<1|2>:]PM[:DEViation]:RANGe[:UPPer] <numeric_value>

This command selects the display range for the PM demodulation.

Example: ":PM:RANG 180DEG"

Features: *RST value: -
SCPI: conforming

Mode: AF-D

SENSe:DETector Subsystem

The SENSe:DETector subsystem controls the recording of measurement values via the type of detector selected for each trace.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:DETector<1to4>			
:CMEM			
[:STATe]	<Boolean>		
[:FUNCtion]	APEak | NEGative| POSitive | SAMPle| RMS | AVERage | ACVideo		
:AUTO	<Boolean>	--	
:RECeiver			
[:FUNCtion]	POSitive | NEGative | RMS | AVERage | ACVideo, POSitive | NEGative | RMS | AVERage | ACVideo..		
:PSTRetch			
[:STATe]	<Boolean>		
:AUTO	<Boolean>		
:FMEasurement	POSitive | NEGative | RMS | AVERage | ACVideo		

:[SENSe<1|2>:]DETector:CMEM[:STATe] ON | OFF

This command switches the capture memory on or off

Example: `":DET:CMEM ON"`

Features: *RST value: OFF
SCPI: device specific

Mode: A

:[SENSe<1|2>:]DETector<1 to 4>[:FUNCtion] APEak | NEGative | POSitive | SAMPle | RMS | AVERage | ACVideo

This command switches the detector for recording of the measured value.

Example: `":DET POS"`

Features: *RST value: POSitive
SCPI: conforming

Modes: R, A

In scan mode of the receiver, the detectors POSitive, RMS and AVERage and ACVideo are available.

In the analyzer mode, the detectors APEak, POSitive, NEGative, RMS, SAMPLe and AVERage are available. The value "APEak" (AutoPeak) displays both the positive peak value and the negative peak value when noise is present. The positive peak value is displayed when one signal is present. The trace is selected by means of the numeric suffix after DETector.

:[SENSe<1|2>:]DETector<1 to 4>[:FUNCtion]:AUTO ON | OFF

This command either couples the detector to the current trace setting or turns coupling off.

Example: ":DET:AUTO OFF"

Features: *RST value: ON
SCPI: conforming

Mode: A

The trace is defined by the numeric suffix at DETector.

:[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] POSitive | NEGative | RMS | AVERage | ACVideo

This command switches on the detectors for single measurements.

Example: ":DET:REC POS,AVER,QPE"

Features: *RST value: POS
SCPI: device-specific

Mode: R

The trace is not selectable; up to four detectors may be switched on simultaneously. The RMS, NEGative and ACVideo detector cannot be switched on simultaneously.

:[SENSe<1|2>:]DETector<1 to 4>:PSTRetch[:STATe] ON | OFF

This command switches the pulse stretcher on or off.

Example: ":DET:PSTR ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: A, VA

The trace is defined by the numeric suffix at DETector.

:[SENSe<1|2>:]DETector<1 to 4>:PSTRetch:AUTO ON | OFF

This command switches the automatic mode of the pulse stretcher on or off.

Example: ":DET:PSTR:AUTO ON"

Features: *RST value: OFF
SCPI: device-specific

Modes: A, VA

The trace is defined by the numeric suffix at DETector.

:[SENSe<1|2>:]DETector<1 to 4>:FMEasurement NEGative | POSitive | RMS | AVERage | ACVideo

This command selects the detector for the final measurement (the detector used for the subsequent final measurement).

Example: "DET:FME POS"

Features: *RST value: Trace 1, 3 POSitive
Trace 2, 4 AVERage
SCPI: device specific

Mode: R

SENSe:DDEMod Subsystem

This subsystem controls the parameters for digital demodulation. It is only active in conjunction with operating mode Vector Signal Analysis.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:DDEMod			Vector Signal Analysis
:FORMat	QPSK | PSK | MSK | QAM | FSK		
:SBANd	NORMal | INVerse		
:QPSK			
:FORMat	NORMal | DIFFerential | OFFSet | DPI4		
:PSK			
:NSTate	2 | 8		
:FORMat	NORMal | DIFFerential | N3Pi8		
:MSK			
:FORMat	TYPE1 | TYPE2 | NORMal | DIFFerential		
:QAM			
:NSTate	16		
:FSK			
:NSTate	2 | 4		
:SRATe	<numeric_value>	HZ	
:TIMe	<numeric_value>	SYM	
:PRATe	1 | 2 | 4 | 8 | 16		
:FILTer			
:MEASurement	OFF | RCOSine | RRCosine | GAUSsian | B22 | B25 | B44 | QFM | FM95 | QFR | FR95 | QRM |RM95 | QRR | RR95 | A25Fm | EMES | EREF		
:REFerence	RCOSine | RRCosine | GAUSsian | B22 | B25 | B44 | QFM | FM95 | QFR | FR95 | QRM | RM95 | QRR | RR95 | A25Fm | EMES | EREF		
:ALPHa	<numeric_value>		
:NORMalize	<Boolean>		
:PRESet	GSM | EDGe | TETRa | DCS1800 | PCS1900 |PHS | PDCup | PDCDown | APCO25CQPSK | APCO25C4FM | CDPD | DECT | CT2 |ERMes | MODacom | PWT | TFTS | F16 | F322 | F324 | F64| FQCDma | F95Cdma | RQCDma | R95Cdma | FNADc | RNADc		
:SEARch			
:PULSe			
:STATe	<Boolean>		
:SYNC			
:CATalog?			query only
:SELect	<string>		
:OFFSet	<numeric_value>	SYM	
:PATTern	<string>		
:STATe	<Boolean>		
:NAME	<string>		
:COMMent	<string>		
:DATA	<string>		
:DELete			
:MONLy	<Boolean>		
:TIME	<numeric value>	SYM	
:TCAPture			Vector Signal Analysis
:LENGth	<numeric_value>		

:[SENSe<1|2>:]DDEMod:FORMat QPSK | PSK | MSK | QAM | FSK

This command selects the digital demodulation type.

Example: ":DDEM:FORM QPSK"

Features: *RST value: MSK
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SBANd NORMal | INVerse

This command selects the sideband for the demodulation.

Example: ":DDEM:SBAN INV"

Features: *RST value: NORMal
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:QPSK:FORMat NORMal | DIFFerential | OFFSet | DPI4

This command determines the specific demodulation type for QPSK.

Example: ":DDEM:QPSK:FORM DPI4"

Features: *RST value: -
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:PSK:NSTate 2 | 8

This command determines the specific demodulation type for PSK.

Example: ":DDEM:PSK:NST 2"

Features: *RST value: -
SCPI: device-specific

Mode: VA

Value 2 (i.e. PSK2) corresponds to BPSK demodulation, value 8 the 8PSK demodulation.

:[SENSe<1|2>:]DDEMod:PSK:FORMat NORMal | DIFFerential | N3Pi8

This command determines the specific demodulation type for PSK.

Example: ":DDEM:PSK:FORM DIFF"

Features: *RST value: -
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:MSK:FORMat TYPE1 | TYPE2 | NORMal | DIFFerential

This command determines the specific demodulation type for MSK.

Example: ":DDEM:MSK:FORM TYPE2"

Features: *RST value: TYPE2 | DIFFerential
SCPI: device-specific

Mode: VA

TYPE1 | NORMal corresponds to MSK demodulation, TYPE2 | DIFFerential corresponds to DMSK demodulation.

:[SENSe<1|2>:]DDEMod:QAM:NSTate 16

This command determines the specific demodulation type for QAM.

Example: ":DDEM:QAM:NST 16"

Features: *RST value: 16
SCPI: device-specific.

Mode: VA

:[SENSe<1|2>:]DDEMod:FSK:NSTate 2 | 4

This command determines the specific demodulation type for FSK.

Example: ":DDEM:FSK:NST 2"

Features: *RST- value: -
SCPI: device-specific

Mode: VA

The parameter 2 corresponds to the demodulation type 2FSK, the parameter 4 to the demodulation type 4FSK.

:[SENSe<1|2>:]DDEMod:SRATe 160 Hz to 1.6 MHz

This command defines the symbol rate.

Example: ":DDEM:SRAT 18kHz"

Features: *RST value: 270.833333kHz
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:TIME 1 to Frame Length

The command determines the number of displayed symbols (result length).

Example: ":DDEM:TIME 80"

Features: *RST value: 147
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:PRATe 1 | 2 | 4 | 8 | 16

This command determines the number of points per symbol.

Example: ":DDEM:PRAT 8"

Features: *RST value: 4
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:FILTer:MEASurement OFF | RCOSine | RRCosine | GAUSsian| B22 | B25 | B44 | QFM | FM95 | QFR | FR95 | QRM | RM95 | QRR | RR95 | A25Fm | EMES | EREF

This command selects the input filter for the test signal.

B22	Bessel 22
B25	Bessel 25
B44	Bessel 44
QFM or FM95	IS95-CDMA fm
QFR or FR95	IS95-CDMA fr
QRM or RM95	IS95-CDMA rm
QRR or RR95	IS95-CDMA rr
A25Fm	APCO 25 fm
EMES	EDGE mes
EREF	EDGE ref

Example: ":DDEM:FILT:MEAS RCOS"

Features: *RST value: OFF
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:FILTer:REFerence RCOSine | RRCosine | GAUSsian| B22 | B25 | B44 | QFM | FM95 | QFR | FR95 | QRM | RM95 | QRR | RR95 | A25Fm | EMES | EREF

This command selects the input filter for the reference signal.

Example: ":DDEM:FILT:REF RCOS"

Features: *RST value: GAUSsian
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:FILTer:ALPHa 0.2 to 1

This command determines the filter characteristic (ALPHA/BT). Step width is 0.05.

Example: ":DDEM:FILT:ALPH 0.5"

Features: *RST value: 0.3
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:NORMalize ON | OFF

This command switches normalizing of the unit circle on or off using IQ offset.

Example: ":DDEM:NORM OFF"

Features: *RST value: ON
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:PULSe:STATe ON | OFF

This command switches the signal burst search on or off.

Example: ":DDEM:SEAR:PULS:STAT OFF"

Features: *RST value: ON
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:CATalog?

This command polls the names of all sync-file data sets stored on the harddisk. Syntax of output format:

<Sum of file lengths of all subsequent files>,<free memory on hard disk>,
<1st file name>,,<1st file length>,<2nd file name>,,<2nd file length>,....,<nth file name>,

Example: ":DDEM:SEAR:SYNC:CAT?"

Feature: *RST value: -
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:SELect <string>

This command selects a predefined sync file. A file that has been set previously by the command DDEM:SEARch:SYNC:PATTern becomes invalid.

Example: ":DDEM:SEAR:SYNC:SEL "PATT_1"

Feature: *RST value: ""
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:OFFSet <numeric_value>

This command defines the offset of the display with reference to the synchronization sequence.

Example: ":DDEM:SEAR:SYNC:OFFS 10SYM"

Features: *RST value: 0 SYM
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:PATTern <string>

This command defines the synchronization sequence.A file that has been set previously by the command `DDEM:SEARch:SYNC:SELect` becomes invalid.

Example: `":DDEM:SEAR:SYNC:PATT "1101001"`

Features: *RST value: ""
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:STATe ON | OFF

This command switches the search for a synchronization sequence on or off.

Example: `":DDEM:SEARch:SYNC:STAT ON"`

Features: *RST value: OFF
SCPI: device-specific

Mode: VA-D

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:NAME <string>

This command selects a sync file to be edited or entered anew.

Example: `":DDEM:SEAR:SYNC:NAME "PATT_NEW"`

Feature: *RST value: ""
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:COMMent <string>

This command defines a comment on a sync file. This sync file must be previously selected by the command `DDEM:SEARch:SYNC:NAME`.

Example: `":DDEM:SEAR:SYNC:COMM "PATTERN FOR PPSK"`

Feature: *RST value: ""
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:DATA <string>

This command defines a synchronization pattern for the sync file. Inputs other than "1" or "0" are interpreted as "Don't Care Bits". This sync file must be previously selected by the command `DDEM:SEARch:SYNC:NAME`.

Example: `":DDEM:SEAR:SYNC:DATA "1101001"`

Feature: *RST value: ""
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:DELete <string>

This command deletes a synchronization pattern on the hard disk. This sync file must be previously selected by the command `DDEM:SEARch:SYNC:NAME`.

Example: `":DDEM:SEAR:SYNC:DEL`

Features: *RST-value: ""
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:SYNC:MONLy ON | OFF

For setting ON, this command sets the analyzer so that the measured values are displayed and taken into account in the error analysis only if the set sync pattern is found . Bursts with a wrong sync pattern (sync not found) are ignored. If an invalid sync pattern is found or if there is none, the measurement is halted and continued in the presence of a valid sync pattern.

This command is available only if the search for a synchronization sequence is activated (`DDEM:SEARch:SYNC:STATe = ON`).

Example: `":DDEM:SEAR:SYNC:MONL ON"`

Features: *RST value: OFF
SCPI: device specific

Mode: VA

:[SENSe<1|2>:]DDEMod:SEARch:TIME 100 to 1600

This command determines the number of symbols required for demodulation (frame length). The values > 800 are only permissible if the number of points per symbol is <16, the values > 1600 only if the number of points per symbol is <8. Step width is 100.

Example: `":DDEM:SEAR:TIME 800"`

Features: *RST value: 400
SCPI: device-specific

Mode: VA

:[SENSe<1|2>:]DDEMod:PRESet GSM | EDGe | TETRa | DCS1800 | PCS1900 |PHS | PDCup | PDCDown | APCO25CQPSK | APCO25C4FM | CDPD | DECT | CT2 | ERMes | MODacom | PWT | TFTS | F16 | F322 | F324 | F64| FQCDma | F95Cdma | RQCDma | R95Cdma | FNADc | RNADc |

This command selects an automatic setting of all modulation parameters according to a standard transmission method.

Example: `":DDEM:PRES TETR"`

Features: *RST value: GSM
SCPI: device-specific

Mode: VA

APCO25CQPSK	APCO25 Continous Phase QPSK
APCO25C4FM	APCO25Continous Phase 4FM
F16	FLEX 1600 - 2FSK
F322	FLEX 3200 - 2FSK,
F324	FLEX 3200 - 4FSK
F64	FLEX 6400 - 4FSK
FNADc	Forward NADC
RNADc	Reverse NADC
FQCDma or F95Cdma	Forward CDMA acc. to IS95 Standard
RQCDma or R95Cdma	Reverse CDMA acc. to IS95 Standard

:[SENSe<1|2>:]TCAPture:LENGth 1024 | 2048 | 4096 | 8192 | 16384

This command determines the number of sampling points that are written into the memory for each measurement (memory size).

Example: `"TCAP:LENG 1024"`

Features: *RST value: 16384
SCPI: device-specific

Mode: VA

SENSe:FILTer Subsystem

The SENSe:FILTer subsystem activates the notch filter.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:FILTer			Vector Signal Analyzer
:HPASs			
[:STATe]	<Boolean>	--	
:FREQuency	<numeric_value>	HZ	
:NOTCh			
[:STATe]	<Boolean>	--	

:[SENSe<1|2>:]FILTer:HPASs[:STATe] ON | OFF

This command switches the AC-video coupling in the analyzer mode on or off

Example: ":FILT:HPAS ON"

Features: *RST value: OFF
SCPI: conforming

Mode: A

:[SENSe<1|2>:]FILTer:HPASs:FREQuency 10 kHz | 1 kHz | 100 Hz

This command defines the cut-off frequency of the A.C. video coupling

Example: ":FILT:HPAS:FREQ 1KHZ"

Features: *RST value: - (STATe = OFF)
SCPI: conforming

Mode: A

:[SENSe<1|2>:]FILTer:NOTCh[:STATe] ON | OFF

This command switches the notch filter on or off.

Example: ":FILT:NOTCH ON"

Features: *RST value: OFF
SCPI: device specific

Modes: A, VA

SENSe:FREQuency Subsystem

The SENSe:FREQuency subsystem defines the frequency axis of the active display. The frequency axis can either be defined via the start/stop frequency or via the center frequency and span.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:FREQuency			
:CENTer	<numeric_value>	HZ	
:LINK	STARt | STOP | SPAN	--	
:STEP	<numeric_value>	HZ	
:LINK	SPAN | RBW | OFF	--	
:FACTor	<numeric_value>	PCT	
:RANGe	2 GHZ | 22 GHZ	HZ	
:SPAN	<numeric value>	HZ	
:FULL	--	--	
:LINK	CENTer | STARt | STOP	--	
:STARt	<numeric_value>	HZ	
:FLINe			
[:STATe]	<Boolean>		
:LINK	CENTer | STOP | SPAN	--	
:STOP	<numeric_value>	HZ	
:FLINe			
[:STATe]	<Boolean>		
:LINK	CENTer | STARt |SPAN	--	
:MODE	CW | FIXed | SWEep	--	
:OFFSet	<numeric_value>	HZ	
:CW:	<numeric_value>	HZ	
:STEP	<numeric_value>	HZ	
:FIXed	<numeric_value>	HZ	
:STEP	<numeric_value>	HZ	

:[SENSe<1|2>:]FREQuency:CENTer 0 GHz to f_{max}

This command defines the receiver frequency ot the center frequency of the analyzer.

Example: `":FREQ:CENT 100MHz"`

Features: *RST value: f_{max} / 2
SCPI: conforming

Modes: R, A, VA

In the analyzer mode, the automatic coupling of the parameters is set to SPAN FIXED.

:[SENSe<1|2>:]FREQuency:CENTer:LINK STARt | STOP | SPAN

This command defines the coupling of the center frequency to the start, stop frequency or the frequency span.

Example: `":FREQ:CENT:LINK STAR"`

Features: *RST value: SPAN
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]FREQuency:CENTer:STEP 0 to f_{max}

This command defines the step width of the receiver or center frequency.

Example: ":FREQ:CENT:STEP 120MHz"

Features: *RST value: - (AUTO 0.1 × SPAN is switched on)
SCPI: conforming

Modes: A, VA

:[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK SPAN | RBW | OFF

This command couples the step width of the center frequency to span (span >0) or to the resolution bandwidth (span = 0) or cancels the couplings.

Example: ":FREQ:CENT:STEP:LINK SPAN"

Features: *RST value: SPAN
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor 1 to 100 PCT

This command couples the step width of the center frequency with a factor to the span (span >0) or to the resolution bandwidth (span = 0).

Example: ":FREQ:CENT:STEP:LINK:FACT 20PCT"

Features: *RST value: - (AUTO 0.1 × SPAN is switched on)
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]FREQuency:RANGe 2GHZ | 22GHZ

This command defines the frequency range.

Example: ":FREQ:RANG 2GHZ"

Features: *RST value: 22GHZ
SCPI: device-specific

Mode: A

The automatic coupling of the parameters is set to CENTER FIXED.

:[SENSe<1|2>:]FREQuency:SPAN 0 GHz to f_{max}

This command defines the frequency span of the analyzer.

Example: ":FREQ:SPAN 10MHz"

Features: *RST value: f_{max}
SCPI: conforming

Mode: A

The automatic coupling of the parameters is set to CENTER FIXED.

:[SENSe<1|2>:]FREQuency:SPAN:FULL

This command sets the maximum frequency span of the analyzer.

Example: ":FREQ:SPAN:FULL"

Features: *RST value: -
SCPI: conforming

Mode: A

This command is an event which is why it is not assigned an *RST value and has no query.

:[SENSe<1|2>:]FREQuency:SPAN:LINK CENTer | START | STOP

This command defines the coupling for frequency-span changes.

Example: ":FREQ:SPAN:LINK STOP"

Features: *RST value: CENTer
SCPI: conforming

Mode: A

:[SENSe<1|2>:]FREQuency:STARt 0 GHz to f_{max}

This command defines the start frequency of the scan in receiver mode or the start frequency of the analyzer.

Example: ":FREQ:STAR 20MHz"

Features: *RST value: 0
SCPI: conforming

Modes: R, A

In analyzer mode, the automatic coupling of the parameters is set to STOP FIXED.

:[SENSe<1|2>:]FREQuency:STARt:FLINe[:STATe] ON | OFF

If ON is selected for this command, the sweep will start at the frequency line (instead of the entered start frequency).

Example: ":FREQ:STAR ON"

Feature: *RST value: OFF
SCPI: device-specific

Mode: A-F

:[SENSe<1|2>:]FREQuency:STARt:LINK CENTer | STOP | SPAN

This command defines the coupling for start-frequency changes.

Example: ":FREQ:STAR:LINK SPAN"

Features: *RST value: STOP
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]FREQuency:STOP 0 GHz to f_{max}

This command defines the stop frequency of the scan in receiver mode or the stop frequency of the analyzer.

Example: ":FREQ:STOP 2000MHz"

Features: *RST value: f_{max}
SCPI: conforming

Modes: R, A

In analyzer mode, the automatic coupling of the parameters is set to STARt FIXED.

:[SENSe<1|2>:]FREQuency:STOP:FLINe[:STATe] ON | OFF

If ON is selected for this command, the sweep will stop at the frequency line 2 (instead of the entered stop frequency).

Example: ":FREQ:STOP ON"

Feature: *RST value: OFF
SCPI: device-specific

Mode: A-F

:[SENSe<1|2>:]FREQuency:STOP:LINK CENTer | STARt | SPAN

This command defines the coupling for stop-frequency changes.

Example: ":FREQ:STOP:LINK SPAN"

Features: *RST value: STARt
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]FREQuency:MODE CW | FIXed | SWEep

This command switches between frequency (SWEep) and time (CW | FIXed) domain in the analyzer mode.

Example: ":FREQ:MODE SWE"

Features: *RST value: CW
SCPI: conforming

Mode: A

For CW and FIXed, the frequency setting is via command FREQuency:CENTer. In the SWEep mode, the setting is via commands FREQuency:STARt, STOP, CENTer and SPAN.

:[SENSe<1|2>:]FREQuency:OFFSet <numeric_value>

This command defines the frequency offset of the instrument.

Example: ":FREQ:OFFS 1GHZ"

Features: *RST value: 0 Hz
SCPI: conforming

Modes: A, VA

:[SENSe<1|2>:]FREQuency[:CW|:FIXed] f_{min} to f_{max}

This command defines the receiver frequency

Example: `":FREQ:CW 50MHz"`

Features: *RST value: 100MHz
SCPI: conforming

Mode: R

:[SENSe<1|2>:]FREQuency[:CW|:FIXed]:STEP f_{min} .. f_{max}

This command defines the step width of the receiver frequency.

Example: `":FREQ:FIX:STEP 50kHz"`

Features: *RST value: 10kHz
SCPI: conforming

Mode: R

SENSe:POWer Subsystem

This subsystem controls the setting of the instrument's power measurements.

COMMAND	PARAMETER	UNIT	COMMENT
[SENSe<1|2>]			
:POWer			
:BANDwidth	<numeric_value>	PCT	
:BWIDth	<numeric_value>	PCT	

:[SENSe<1|2>:]POWer:BANDwidth|BWIDth 0 to 100PCT

This command defines the percentage of the power with respect to the total power.

Example: `":POW:BWID 95PCT"`

Features: *RST value: 99PCT
SCPI: device-specific

Mode: A-F

This value defines the occupied bandwidth (measurement `POW:ACH:PRES OBW`).

SENSe:ROSCillator Subsystem

This subsystem controls the reference oscillator. The suffix in SENSe<1|2> is not significant in this subsystem.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:ROSCillator			
:SOURce	INTernal | EXTernal	--	
:EXTernal			
:FREQuency	<numeric_value>	HZ	
[:INTernal]			
:TUNe	<numeric_value>		
:SAVe			no query

:[SENSe<1|2>:]ROSCillator:SOURce INTernal | EXTernal

This command controls selection of the reference oscillator.

Example: `":ROSC:SOUR EXT"`

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

If the external reference oscillator is selected, the reference signal must be connected to the rear panel of the instrument. *RST has no influence on this setting.

:[SENSe<1|2>:]ROSCillator:EXTernal:FREQuency 1MHz to 16MHz

This command defines the frequency of the external reference oscillator.

Example: `":ROSC:EXT:FREQ 5MHz"`

Features: *RST value: 10MHz
SCPI: conforming

Modes: R, A, VA

The value of the external reference frequency (1MHz to 16MHz) is rounded in steps of 1MHz.

:[SENSe<1|2>:]ROSCillator[:INTernal]:TUNe 0 to 4095

This command defines the value for the tuning of the internal reference oscillator.

Example: `":ROSC:TUN 128"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

The reference oscillator should be tuned only if an error has been detected in the frequency accuracy check. After PRESET or switching on the instrument, the saved value of the reference frequency is restored.

:[SENSe<1|2>:]ROSCillator[:INTernal]:TUNe:SAVe

This command saves the new value for the tuning of the internal reference oscillator. The factory-set value in the EEPROM is overwritten.

Example: `":ROSC:TUN:SAV"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

SENSe:SCAN - Subsystem

This subsystem controls the parameters for the receiver scan data. The numeric suffix in [SENSe<1|2>]SCAN is not significant.

COMMAND	PARAMETER	UNIT	COMMENT
[SENSe<1|2>]			
:SCAN<1 to 10>			Receiver Mode
:STARt	<numeric_value>	HZ	
:STOP	<numeric_value>	HZ	
:STEP	<numeric_value>	HZ	
:BANDwidth			
:RESolution	<numeric_value>	HZ	
:TIME	<numeric_value>	s	
:INPut			
:TYPe	RF | BALanced		
:BIMPedance	150 | 600 | 10 000	OHM	
:ATTenuation	<numeric_value>	dB	
:AUTO	<Boolean>	--	
:GAIN	0 | 10 | 20 | 30	dB	
:STATe	<Boolean>	--	
:AUTO	<Boolean>	--	
:RANGes			
[:COUNt]	<numeric_value>	--	

:[SENSe<1|2>:]SCAN<1 to 10>:STARt f_{min} to f_{max}

This command defines the start frequency of a receiver subscan.

Example: `":SCAN1:STAR 50kHz"`

Features: *RST value: 150 kHz (range1), 30 MHz (range2)
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:STOP f_{min} to f_{max}

This command defines the stop frequency of a receiver subscan.

Example: `":SCAN1:STOP 50kHz"`

Features: *RST value: 30 MHz (range1), 1 GHz (range2)
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:STEP f_{min} to f_{max}

This command defines the step size for the frequency of a receiver subscan.

Example: `":SCAN1:STEP 100Hz"`

Features: *RST value: 4 kHz (range 1), 40 kHz (range 2)
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:BANDwidth:RESolution f_{min} to f_{max}

This command defines the bandwidth for a receiver subscan.

Example: `":SCAN1:BAND:RES 1MHz"`

Features: *RST value: 9 kHz (range 1)
120 kHz (range 2)
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:TIME 100 µs to 100 s

This command defines the measurement time of the receiver subscan.

Example: `":SCAN1:TIME 1 ms"`

Features: *RST value: 1 ms (range 1)
100µs (range 2)
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:INPut:TYPE RF | BALanced

This command defines the input for a receiver subscan.

Example: `":SCAN1:INP:TYPE BAL"`

Features: *RST value: BAL
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:INPut:BIMPedance 150 | 600 | 10000

This command sets the impedance of the balanced input.

Example: `":SCAN1:INP:BIMP 600"`

Features: *RST value: 600
SCPI: conforming

Modes: R

:[SENSe<1|2>:]SCAN<1 to 10>:INPut:ATTenuation dB_{min} to dB_{max}

This command defines the RF attenuation for a receiver subscan.

Example: `":SCAN1:INP:ATT 30dB"`

Features: *RST value: 10 dB
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:INPut:ATTenuation:AUTO ON | OFF

This command switches on or off the autoranging function in a receiver subscan.

Example: ":SCAN1:INP:ATT:AUTO ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:INPut:GAIN 0 dB | 10 dB | 20 dB | 30 dB

This command sets the preamplification in a receiver subscan.

Example: ":SCAN1:INP:GAIN 30 dB"

Features: *RST value: 0 dB
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:INPut:GAIN:STATE ON | OFF

This command switches on or off the preamplifier in a receiver subscan.

Example: ":SCAN1:INP:GAIN:STAT ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN<1 to 10>:INPut:GAIN:AUTO ON | OFF

This command includes the preamplifier in the autoranging function of a receiver subscan.

Example: ":SCAN1:INP:GAIN:AUTO ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: R

:[SENSe<1|2>:]SCAN:RANGes[:COUNt] 1 to 10

This command determines the number of ranges.

Example: "SCAN:RANG:COUN 3"

Features: *RST value: 2
SCPI: device-specific

Mode: R

SENSe:SWEep Subsystem

This subsystem controls the sweep parameters.

COMMAND	PARAMETERS	UNIT	COMMENT
[SENSe<1|2>]			
:SWEep			
:TIME	<numeric_value>	S	
:AUTO	<Boolean>	--	
:FMEasurement	<numeric_value>	S	
:COUNt	<numeric_value>	--	
:EGATe	<Boolean>	--	
:LEVel	<numeric_value>	V	
:TYPE	LEVel | EDGE	--	
:POLarity	POSitive|NEGative	--	
:HOLDoff	<numeric_value>	S	
:LENGth	<numeric_value>	S	
:GAP	<Boolean>	--	
:PRETrigger	<numeric_value>	S	
:TRGTogap	<numeric_value>	S	
:LENGth	<numeric_value>		
:SPACing	LINear | LOGarithmic | AUTO		

:[SENSe<1|2>:]SWEep:TIME <numeric_value>

This command defines the duration of a single measurement or the duration of the sweep.

The allowed value range depends on the operating mode (see Chapter 4, Softkeys MEAS TIME (receiver mode) and SWEEP TIME MANUAL (analyzer mode)).

Example: `":SWE:TIME 10s"`

Features: *RST value - (AUTO is set to ON)
SCPI: conforming

Modes: R, A

In analyzer and vector analyzer mode, automatic coupling is switched off if `SWEep:TIME` is directly programmed.

:[SENSe<1|2>:]SWEep:TIME:AUTO ON | OFF

This command switches the automatic coupling of the sweep time for the frequency span or bandwidth settings.

Example: `":SWE:TIME:AUTO ON"`

Features: *RST value: ON
SCPI: conforming

Mode: A

If `SWEep:TIME` is directly programmed, automatic coupling is switched off.

:[SENSe<1|2>:]SWEep:TIME:FMEasurement <numeric_value>:

This command defines the measurement time in which the values indicated in the peak list (final measurement values) are re-checked.

Example: `"SWE:TIME:FME 0.01"`

Features: *RST value: 1s
SCPI: device specific

Mode: R

:[SENSe<1|2>:]SWEep:COUNt 0 to 32767

This command defines the number of sweeps started with single sweep.

Example: `":SWE:COUN 64"`

Features: *RST value: 0
SCPI: conforming

Modes: R, A, VA

This parameter defines the number of sweeps or the number of averaging procedures. In the average mode, the value 0 defines a running averaging of measurement data over 10 sweeps.

:[SENSe<1|2>:]SWEep:EGATe ON | OFF

This command controls the sweep with the external gate signal.

Example: `":SWE:EGAT ON"`

Features: *RST value: OFF
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]SWEep:EGATe:LEVel -5V to +5V

This command determines the threshold for the external gate signal.

Example: `":SWE:EGAT:LEV 3V"`

Features: *RST value: 2V
SCPI: device-specific

Mode: A

If `SWEep:TIME` is directly programmed, automatic coupling is switched off.

:[SENSe<1|2>:]SWEep:EGATe:TYPE LEVel | EDGE

This command sets the type of triggering (level or edge) by the external gate signal.

Example: `":SWE:EGAT:TYPE EDGE"`

Features: *RST value: EDGE
SCPI: device-specific

Mode: A

Parameter `EGATe:LENGth` is not used for level triggering.

:[SENSe<1|2>:]SWEep:EGATe:POLarity POSitive | NEGative

This command determines the polarity of the external gate signal.

Example: ":SWE:EGAT:POL POS"

Features: *RST value: POSitive
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]SWEep:EGATe:HOLDoff 0 to 100s

This command defines the delay time between the external gate signal and the continuation of the sweep.

Example: ":SWE:EGAT:HOLD 100us"

Features: *RST value: 0s
SCPI: device-specific

Mode: A

The values for the delay time can be set in steps 1, 2, 3 and 5.

:[SENSe<1|2>:]SWEep:EGATe:LENGth 0 to 100s

In case of edge triggering, this command determines the time interval in which the instrument sweeps.

Example: ":SWE:EGAT:LENG 10ms"

Features: *RST value: 0s
SCPI: device-specific

Modes: A

The values for the delay time can be set in steps 1, 2, 3 and 5.

:[SENSe<1|2>:]SWEep:GAP ON | OFF

This command controls the operating mode GAP SWEEP.

Example: ":SWE:GAP ON"

Features: *RST value: OFF
SCPI: device-specific

Mode: A-Z

Operating mode GAP SWEEP for measurements in the time domain controls the display of measured values. Measured values can be blanked for a defined time range (GAP).

:[SENSe<1|2>:]SWEep:GAP:PRETrigger 0 to 100s

This command defines the evaluation time for measured values before the pretrigger time (resolution: 50 ns).

Example: ":SWE:GAP:PRET 100us"

Features: *RST value: 0s
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]SWEep:GAP:TRGTogap 0 to 100s

This command defines the time between the pretrigger time and the beginning of the gap (trigger-to-gap time) (resolution: 50 ns).

Example: ":SWE:GAP:TRGT 50us"

Features: *RST value: 0s
SCPI: device-specific

Mode: A

:[SENSe<1|2>:]SWEep:GAP:LENGth 0 to 100s

This command defines the gap length.

Example: ":SWE:GAP:LENG 400us"

Features: *RST value: 0s
SCPI: device-specific

Mode: A

The gap length can be programmed from 0 to 100s in steps of 1, 2, 3 and 5.

:[SENSe<1|2>:]SWEep:SPACing LINear | LOGarithmic | AUTO

This command toggles between linear and logarithmic step modes of the receiver. In analyzer mode, it toggles between linear and logarithmic sweep.

Example: ":SWE:SPAC LOG"

Features: *RST value: LIN
SCPI: conforming

Modes: R, A

The frequency axis is set to linear or logarithmic scaling accordingly.

STATus Subsystem

The STATus subsystem contains the commands for the status reporting system (see Chapter 5, Section "Status Reporting System"). *RST does not influence the status registers. The commands and events causing the status reporting system to be reset are comprised in a table at the end of chapter 5.

COMMAND	PARAMETERS	UNIT	COMMENT
STATus			
:OPERation			
[:EVENt]?	--	--	
:CONDition?	--	--	
:ENABle	0 to 65535	--	
:PTRansition	0 to 65535	--	
:NTRansition	0 to 65535	--	
:PRESet	--	--	
:QUEStionable			
[:EVENt]?	--	--	
:CONDition?	--	--	
:ENABle	0 to 65535	--	
:PTRansition	0 to 65535	--	
:NTRansition	0 to 65535	--	
:POWer			
[:EVENt]?	--	--	
:CONDition?	--	--	
:ENABle	0 to 65535	--	
:PTRansition	0 to 65535	--	
:NTRansition	0 to 65535	--	
:LIMit			
[:EVENt]?	--	--	
:CONDition?	--	--	
:ENABle	0 to 65535	--	
:PTRansition	0 to 65535	--	
:NTRansition	0 to 65535	--	
:LMARgin			
[:EVENt]?	--	--	
:CONDition?	--	--	
:ENABle	0 to 65535	--	
:PTRansition	0 to 65535	--	
:NTRansition	0 to 65535	--	
:SYNC			
[:EVENt]?	--	--	
:CONDition?	--	--	
:ENABle	0 to 65535	--	
:PTRansition	0 to 65535	--	
:NTRansition	0 to 65535	--	

COMMAND	PARAMETERS	UNIT	COMMENT
STATus			
:QUEStionable	--	--	
:FREQuency	--	--	
[:EVENt]?	0 to 65535	--	
:CONDition?	0 to 65535	--	
:ENABle	0 to 65535	--	
:PTRansition	--	--	
:NTRansition	--	--	
:TRANsducer			
[:EVENt]?	--	--	
:CONDition?	--	--	
:ENABle	0 to 65535	--	
:PTRansition	0 to 65535	--	
:NTRansition	0 to 65535	--	
:QUEue?	--	--	
[:NEXT]?	--	--	

:STATus:OPERation[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:OPERation register.

Example: `":STAT:OPER?"`

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The contents of the EVENt section is deleted with readout.

:STATus:OPERation:CONDition?

This command queries the CONDition section of the :STATus:OPERation register.

Example: `":STAT:OPER:COND?"`

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

Readout does not delete the contents of the CONDition section. The value returned reflects the current hardware status.

:STATus:OPERation:ENABle 0 to 65535

This command sets the bits of the ENABle section of the :STATus:QUEStionable register.

Example: `":STAT:OPER:ENAB 65535"`

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:OPERation:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:OPERation register from 0 to 1 for the transitions of the CONDition bit.

Example: ":STAT:OPER:PTR 65535"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

:STATus:OPERation:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:OPERation register from 1 to 0 for the transitions of the CONDition bit.

Example: ":STAT:OPER:NTR 65535"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

:STATus:PRESet

This command resets the edge detectors and ENABle parts of all registers to a defined value. All PTRansition parts are set to FFFFh, i.e., all transitions from 0 to 1 are detected. All NTRansition parts are set to 0, i.e., a transition from 1 to 0 in a CONDition bit is not detected. The ENABle part of the :STATus:OPERation and :STATus:QUEStionable registers are set to 0, i.e., all events in these registers are not passed on.

Example: ":STAT:PRES"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

:STATus:QUEStionable[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:QUEStionable register.

Example: ":STAT:QUES?"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

Readout deletes the contents of the EVENt section.

:STATus:QUEStionable:CONDition?

This command queries the CONDition section of the :STATus:QUEStionable register.

Example: ":STAT:QUES:COND?"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

Readout does not delete the contents of the CONDition section.

:STATus:QUEStionable:ENABle 0 to 65535

This command sets the bits of the ENABle section of the STATus-QUEStionable register.

Example: ":STAT:QUES:ENAB 65535"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:QUEStionable:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:OPERation register from 0 to 1 for the transitions of the CONDition bit.

Example: ":STAT:QUES:PTR 65535"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

:STATus:QUEStionable:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:OPERation register from 1 to 0 for the transitions of the CONDition bit.

Example: ":STAT:QUES:NTR 65535"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

:STATus:QUEStionable:POWer[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:QUEStionable:POWer register.

Example: ":STAT:QUES?"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

Readout deletes the contents of the EVENt section.

:STATus:QUEStionable:POWer:CONDition?

This command queries the content of the CONDition section of the :STATus:QUEStionable:POWer register.

Example: ":STAT:QUES:COND?"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

Readout does not delete the contents of the CONDition section.

:STATus:QUEStionable:POWer:ENABle 0 to 65535

This command sets the bits of the ENABle section of the :STATus:QUEStionable:POWer register.

Example: `":STAT:QUES:ENAB 65535"`

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:QUEStionable:POWer:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:POWer register from 0 to 1 for the transitions of the CONDition bit.

Example: `":STAT:QUES:PTR 65535"`

Features: *RST-value: –
SCPI: conforming

Modes: R, A, VA

:STATus:QUEStionable:POWer:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:POWer register from 1 to 0 for the transitions of the CONDition bit.

Example: `":STAT:QUES:NTR 65535"`

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

:STATus:QUEStionable:LIMit[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:QUEStionable:LIMit register.

Example: `":STAT:QUES?"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout deletes the contents of the EVENt section.

:STATus:QUEStionable:LIMit:CONDition?

This command queries the contents of the CONDition section of the :STATus:QUEStionable:LIMit register.

Example: `":STAT:QUES:COND?"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout does not delete the contents of the CONDition section.

:STATus:QUEStionable:LIMit:ENABle 0 to 65535

This command sets the bits of the ENABle section of the :STATus:QUEStionable register.

Example: ":STAT:QUES:ENAB 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:QUEStionable:LIMit:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:LIMit register from 0 to 1 for the transitions of the CONDition bit.

Example: ":STAT:QUES:PTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:LIMit:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:LIMit register from 1 to 0 for the transitions of the CONDition bit.

Example: ":STAT:QUES:NTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:LMARgin[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:QUEStionable:LMARgin register.

Example: ":STAT:QUES?"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout deletes the contents of the EVENt section.

:STATus:QUEStionable:LMARgin:CONDition?

This command queries the contents of the CONDition section of the :STATus:QUEStionable:LMARgin register.

Example: ":STAT:QUES:COND?"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout does not delete the contents of the CONDition section.

:STATus:QUEStionable:LMARgin:ENABle 0 to 65535

This command sets the bits of the ENABle section of the :STATus:QUEStionable:LMARgin register.

Example: `":STAT:QUES:ENAB 65535"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:QUEStionable:LMARgin:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable: LMARgin register from 0 to 1 for the transitions of the CONDition bit.

Example: `":STAT:QUES:PTR 65535"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:LMARgin:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable: LMARgin register from 1 to 0 for the transitions of the CONDition bit.

Example: `":STAT:QUES:NTR 65535"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:SYNC[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:QUEStionable:SYNC register.

Example: `":STAT:QUES?"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout deletes the contents of the EVENt section.

:STATus:QUEStionable:SYNC:CONDition?

This command queries the contents of the CONDition section of the :STATus:QUEStionable:SYNC register.

Example: `":STAT:QUES:COND?"`

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout does not delete the contents of the CONDition section.

:STATus:QUEStionable:SYNC:ENABle 0 to 65535

This command sets the bits of the ENABle section of the :STATus:QUEStionable: SYNC register.

Example: ":STAT:QUES:ENAB 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:QUEStionable:SYNC:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable: SYNC register from 0 to 1 for the transitions of the CONDition bit.

Example: ":STAT:QUES:PTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:SYNC:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable: SYNC register from 1 to 0 for the transitions of the CONDition bit.

Example: ":STAT:QUES:NTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:FREQuency[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:QUEStionableFREQuency register.

Example: ":STAT:QUES:FREQ?"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout deletes the contents of the EVENt section.

:STATus:QUEStionable:FREQuency:CONDition?

This command queries the contents of the CONDition section of the :STATus:QUEStionable:FREQuency register.

Example: ":STAT:QUES:FREQ:COND?"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout does not delete the contents of the CONDition section.

:STATus:QUEStionable:FREQuency:ENABle 0 to 65535

This command sets the bits of the ENABle section of the :STATus:QUEStionable:FREQuency register.

Example: ":STAT:QUES:FREQ:ENAB 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:QUEStionable:FREQuency:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:FREQuency register from 0 to 1 for the transitions of the CONDition bit.

Example: ":STAT:QUES:FREQ:PTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:FREQuency:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:FREQuency register from 1 to 0 for the transitions of the CONDition bit.

Example: ":STAT:QUES:FREQ:NTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:TRANsducer[:EVENt]?

This command queries the contents of the EVENt section of the :STATus:QUEStionable:TRANsducer register.

Example: ":STAT:QUES:TRAN?"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout deletes the contents of the EVENt section.

:STATus:QUEStionable:TRANsducer:CONDition?

This command queries the contents of the CONDition section of the :STATus:QUEStionable:FREQuency register.

Example: ":STAT:QUES:TRAN:COND?"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

Readout does not delete the contents of the CONDition section.

:STATus:QUEStionable:TRANsducer:ENABle 0 to 65535

This command sets the bits of the ENABle section of the :STATus:QUEStionable:TRANsducer register.

Example: ":STAT:QUES:TRAN:ENAB 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

The ENABle register selectively enables the individual events of the associated EVENt section for the sum bit in the status byte.

:STATus:QUEStionable:TRANsducer:PTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:TRANsducer register from 0 to 1 for the transitions of the CONDition bit.

Example: ":STAT:QUES:TRAN:PTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEStionable:TRANsducer:NTRansition 0 to 65535

This command sets the edge detectors of all bits of the :STATus:QUEStionable:TRANsducer register from 1 to 0 for the transitions of the CONDition bit.

Example: ":STAT:QUES:TRAN:NTR 65535"

Features: *RST value: –
SCPI: device-specific

Modes: R, A, VA

:STATus:QUEue[:NEXT]?

This command queries the earliest entry to the error queue, thus deleting it.

Example: ":STAT:QUE?"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

Positive error numbers indicate device-specific errors, negative error numbers are error messages defined by SCPI (cf. Chapter 9). If the error queue is empty, the error number 0, "no error", is returned. This command is identical with the command SYSTem:ERRor.

SYSTem Subsystem

This subsystem comprises a series of commands for general functions.

COMMAND	PARAMETERS	UNIT	COMMENT
SYSTem			
:COMMunicate			
:GPIB			
[:SELF]			
:ADDRess	0 to 30	--	
:RTERminator	LFEoi | EOI		
:RDEVice<1|2>			
:ADDRess	0 to 30	--	
:SERial<1|2>			
:CONTrol			
:DTR	IBFull | OFF		
:RTS	IBFull | OFF		
[:RECeive]		--	
:BAUD	<numeric_value>	--	
:BITS	7 | 8		
:PARity			
[:TYPE]	EVEN | ODD | NONE	--	
:SBITs	1 | 2		
:PACE	XON | NONE	--	
:PRINter1|2>			
:ENUMerate			
[:NEXT]?			query only
:FIRSt?			query only
:SELect	<printer_name>		
:DATE	<num>, <num>, <num>	--	
:DISPlay			
:UPDate	ON | OFF		
:ERRor?	--		query only
:PASSword		--	
[:CENable]	<string>		no query
:PRESet	--	--	no query
:SET	<block>		
:TIME	0 to 23, 0 to 59, 0 to 59		
:VERSion?	--		query only
:BINFo?	--		query only

:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess 0 to 30

This command changes the IEC/IEEE-bus address of the unit.

Example: `":SYST:COMM:GPIB:ADDR 18"`

Features: *RST value: - (no influence on this parameter)
SCPI: conforming

Modes: R, A, VA

:SYSTem:COMMunicate:GPIB[:SELF]:RTERminator LFEOI | EOI

This command changes the delimiter.

Example: ":SYST:COMM:GPIB:RTER EOI"

Features: *RST value: LFEOI
SCPI: device-specific

Modes: R, A, VA

The instrument contains a DMA-channel for communication via IEC-bus. This ensures maximum speed for the transfer of commands and data. The parser for command decoding integrated in the instrument is, however, only activated by the delimiter when the command is completely transferred. In order to make this possible for the transfer of binary data, too (e. g. trace data which are retransferred into the instrument), the delimiter recognition must be switched to the EOI signal prior to the transfer. Output of binary data from the instrument does not require such a switching.

:SYSTem:COMMunicate:GPIB:RDEVice<1|2>:ADDRess 0 to 30

This command changes the IEC/IEEE-bus address of the unit which is selected as hardcopy device 1 or 2, provided that the IEC/IEEE-bus interface of this unit is set as an interface.

Example: ":SYST:COMM:GPIB:RDEV2:ADDR 5"

Features: *RST value: 4
SCPI: conforming

Modes: R, A, VA

:SYSTem:COMMunicate:SERial<1|2>:CONTrol:DTR IBFull | OFF
:SYSTem:COMMunicate:SERial<1|2>:CONTrol:RTS IBFull | OFF

These commands switch the hardware handshake procedure for the given serial interface off (OFF) or on (IBFull).

Examples: ":SYST:COMM:SER:CONT:DTR OFF"
":SYST:COMM:SER2:CONT:RTS IBF"

Features: *RST value: OFF
SCPI: conforming

Modes: R, A, VA

The two commands have the same meaning. SERial1 and SERial 2 correspond to device COM1and COM2, respectively.

:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:BAUD 75 | 150 | 300 | 600 | 1200 | 2400 | 9600

This command sets the transmission speed for the given serial interface.

Example: ":SYST:COMM:SER:BAUD 2400"

Features: *RST value: 9600
SCPI: conforming

Modes: R, A, VA

SERial1 and SERial 2 correspond to device interface COM1 and COM2, respectively. Permissible values are 75 Baud, 150 Baud, 300 Baud, 600 Baud, 1200 Baud, 2400 Baud, 4800 Baud, 9600 Baud.

:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:BITS 7 | 8

This command defines the number of data bits per data word for the given serial interface.

Example: `":SYST:COMM:SER2:BITS 7"`

Features: *RST value: 8
SCPI: conforming

Modes: R, A, VA

SERial1 and SERial 2 correspond to device interface COM1 and COM2, respectively.

:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:PARity[:TYPE] EVEN | ODD | NONE

This command defines the parity check for the given serial interface.

Example: `":SYST:COMM:SER:PAR EVEN"`

Features: *RST value: NONE
SCPI: conforming

Modes: R, A, VA

SERial1 and SERial 2 correspond to device interface COM1 and COM2, respectively. Permissible values are:
EVEN even parity
ODD odd parity
NONE no parity check.

:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:SBITs 1|2

This command defines the number of stopbits per data word for the given serial interface.

Example: `":SYST:COMM:SER:SBITs 2"`

Features: *RST value: 1
SCPI: conforming

Modes: R, A, VA

SERial1 and SERial 2 correspond to device interface COM1 and COM2, respectively.

:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:PACE XON | NONE

This command switches on or off the software handshake for the given serial interface.

Example: `":SYST:COMM:SER:PACE XON"`

features: *RST value: NONE
SCPI: conforming

Modes: R, A, VA

SERial1 and SERial 2 correspond to device interface COM1 and COM2, respectively.

:SYSTem:COMMunicate:PRINter<1|2>:ENUMerate:FIRSt?

This command queries the name of the first printer (in the list of printers) under Windows NT. The names of other installed printers can be queried with command `:SYSTem:COMMunicate:PRINter:ENUMerate:NEXT?`. The numeric suffix in PRINter<1|2> is not significant. If no printer is configured an empty string is output.

Example: `":SYST:COMM:PRIN:ENUM:FIRS?"`

Features: *RST value: -
SCPI: device-specific

Modes: A, VA

:SYSTem:COMMunicate:PRINter<1|2>:ENUMerate:NEXT?

This command queries the name of the next printer installed under Windows NT. This command can only be sent after command `:SYSTem:COMMunicate:PRINter: ENUMerate:FIRSt?.` An empty string is output after all printer names have been output. The numeric suffix in PRINter<1|2> is not significant.

Example: `":SYST:COMM:PRIN:ENUM:NEXT?"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:SYSTem:COMMunicate:PRINter<1|2>:SELect <printer_name>

This command selects one of the printers installed under Windows NT. The name of the first printer is queried with `FIRSt?.` After that the names of other installed printers can be queried with `NEXT?`. The numeric suffix in PRINter<1|2> selects the device.

Parameter: <printer_name> ::= string which has been queried with commands `:SYSTem:COMMunicate :PRINter:ENUMerate:FIRSt? and NEXT?.`

Example: `":SYST:COMM:PRIN:SEL 'HP_DESKJET660'"`

Features: *RST value: -
SCPI: device-specific

Modes: R, A, VA

:SYSTem:DATE 1980 to 2099, 1 to 12, 1 to 31

This command is used to enter the date for the internal calendar.

Example: `" SYST:DATE 1994,12,1"`

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The sequence of entry is year, month, day.

:SYSTem:DISPlay:UPDate ON | OFF

This command switches on or off the update of all display elements.

Example: `" SYST:DISP:UPD ON`

Features: *RST value: OFF
SCPI: device specific

Modes: R, A, VA

:SYSTem:ERRor?

This command queries the earliest entry to the error queue, thus deleting it. .

Example: ":SYST:ERR?"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

Positive error numbers indicate device-specific errors, negative error numbers are error messages defined by SCPI (cf. Chapter 9). If the error queue is empty, the error number 0, "no error", is returned. This command is identical with the command :STATus:QUEue:NEXT?. This command is a query which is why it is not assigned an *RST value.

:SYSTem:PASSword[:CENable] <string>

This command enables access to the service functions by means of the password.

Example: ":SYST:PASS 'XXXX'"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

This command is an event which is why it is not assigned an *RST value and has no query.

:SYSTem:SET <block>

The query :SYSTem:SET? causes the data of the current instrument setting to be transmitted to the controller in binary form (SAVE function). The data can be read back into the instrument (RECALL function) by means of command :SYSTem:SET <block>. If the data records are stored on the instrument harddisk with SAVE/RECALL (:MMEMory:STORe bzw. :MMEMory:LOAD), it is possible to store the data in an external computer by means of :SYSTem:SET.

Example: ":SYST:SET "

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The receive terminator has to be set to EOI to ensure reliable transfer of data (setting SYST:COMM:GPIB:RTER EOI).

:SYSTem:PRESet

This command triggers an instrument reset.

Example: ":SYST:PRES"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The effect of this command corresponds to that of the *PRESET* key with manual control or to the *RST command.

:SYSTem:TIME 0 to 23, 0 to 59, 0 to 59

This command sets the internal clock.

Example: ":SYST:TIME 12,30,30"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

The sequence of entry is hour, minute, second.

:SYSTem:VERSion?

This command queries the number of the SCPI version, which is relevant for the instrument.

Example: ":SYST:VERS?"

Features: *RST value: –
SCPI: conforming

Modes: R, A, VA

This command is a query which is why it is not assigned an *RST value.

:SYSTem:BINFo?

This command queries all present modules with model index, modification index and HW code. Entries are separated by commas.

Return format: module1, model index1, modification index1, HW code1, module2, model index2, modification index2, HW code2, module3,...,moduleN, model indexN, modification indexN, HW codeN

Example: ":SYST:BINF?"

Features: *RST value: –
SCPI: device-specific

4Modes: R, A, VA

TRACe Subsystem

The TRACe subsystem controls access to the instrument's internal trace memory.

COMMAND	PARAMETERS	UNIT	COMMENT
TRACe			
[:DATA]	TRACE1 | TRACE2 | TRACE3 | TRACE4 | SINGle | SCAN | STATus | FINAL1 | FINAL2 | FINAL3 | FINAL4, @<block>|<numeric_value>...		
:COPY	TRACE1|TRACE2|TRACE3|TRACE4, TRACE1|TRACE2|TRACE3|TRACE4		
:FEED			
:CONTrol<1...4>	ALWays | NEVer		

:TRACe[:DATA] TRACE1| TRACE2| TRACE3| TRACE4| SINGle | SCAN | STATus | FINAL1 | FINAL2 | FINAL3 | FINAL4, <block> | <numeric_value>

This command transfers trace data from the controller to the instrument, the query reads trace data out of the instrument.

Example:
```
":TRAC TRACE1,"+A$   (A$: data list in the current format)
"TRAC? TRACE1"
```

Features: *RST value: -
SCPI: conforming

Modes: R, A, VA

The trace data are transferred in the current format (corresponding to the command `FORMat ASCii|REAL`). The device-internal trace memory is addressed using the trace names 'TRACE1' to 'TRACE4'.
The transfer of trace data from the controller to the instrument takes place by indicating the trace name and then the data to be transferred. In ASCII format, these data are values separated by commas. If the transfer takes place using the format real (REAL 32), the data are transferred in block format (see FORMat subsystem).

The parameter of the query is the trace name TRACE1 to TRACE4, it indicates which trace memory will be read out.

Saving and recalling trace data together with the device settings to/from the device-internal hard disk or to/from a floppy is controlled via the commands "`:MMEMory:STORe:STATe`" and "`:MMEMory:LOAD:STATe`" respectively. Trace data are selected with "`:MMEMory:SELect[:ITEM]:ALL`" or ""`:MMEMory:SELect[:ITEM]:TRACe`". Trace data in ASCII format (ASCII FILE EXPORT) are exported with the command "`:MMEM:STORe:TRACe`".

The transfer format for the trace data depends on the instrument setting

Analyzer mode (span >0 and zero span):

500 results are output in the unit selected for display.

Note: *With AUTO PEAK detector, only positive peak values can be read out.*
Trace data can be read into instrument with logarithmic display only in dBm, with linear display only in volts.

FORMAT REAL,32 is to be used as format for binary transmission.

Vector analyzer mode, digital demodulation

The number of data transferred (except for the symbol table) is determined by the following formula
number of results = result length * points per symbol

Max. 6400 results can be transferred (for example result length 1600, points per symbol 4).

In all cartesian diagrams (MAGNITUDE CAP BUFFER, MAGNITUDE, PHASE, FREQUENCY, REAL/IMAG, EYE[I], EYE[Q], ERROR VECT MAGNITUDE) test data are transferred in the unit selected for display.

FORMAT REAL,32 is to be used for binary transmission.

Note: *In the case of the eye pattern, results are simply superimposed in the display, ie the EYE representation is the same as the REAL/IMAG representation.*

In the polar diagrams (POLAR CONSTELL, POLAR VECTOR) the real and the imaginary component are transferred as a pair for each result.

FORMAT REAL,32 is to be used for binary transmission.

With the SYMB TABLES / ERRORS setting, the displayed symbols can be read out as traces. Trace assignment is as follows:

Full screen	Trace 1
Split screen, screen A:	Trace 1
Split screen, screen B:	Trace 2

One byte (8 bits) is read out for each symbol.

FORMAT UINT,8 is to be used for binary transmission.

Receiver

SINGle is possible only as a query for single measurements in the receiver mode. The values of all activated detectors are transferred separated by commas. The values of all activated detectors are transferred separated by commas. The values are transferred in the sequence the detectors are set with command `SENSe:DETector:RECeiver` (example: setting `DET:REC POS, AVER` yields `<value for POS detector>`, `<value for AVERage detector >` in response to query `TRAC:DATA? SINGL`).

SCAN is only possible in form of a query during scan measurements. The number of transmitted measurement results depends on the scan settings.

FORMAT REAL,32 is to be used as format setting for binary transmission.

Structure of transmitted data:

- 4 byte: trace status: bit 0 to 9 subscan; bit 10: last block of subscan; Bit 11: last block of last subscan of scan; Bit 12: last of all blocks (for multiple scans after the last scan)
- 4 bytes:number n of the transmitted measurement results of a trace
- 4 byte:trace1 active (0/1)
- 4 byte:trace2 active (0/1)
- 4 byte:trace3 active (0/1)
- 4 byte:trace4 active (0/1)
- n*4 bytes:measurement results of trace 1 if trace 1 is active
- n*4 bytes:measurement results of trace 2 if trace 2 is active
- n*4 bytes:measurement results of trace 3 if trace 3 is active

- n*4 bytes:measurement results of trace 4 if trace 4 is active
- n*1 byte:status information per measurement result:
  bit 0: underrange trace1; bit 1: underrange trace2;bit 2: underrange trace3;
  bit 3: underrange trace4;bit 4: overrange trace1 to trace4

STATus is only possible in form of a query during scan measurements. 1 byte status information per measurement result is transmitted:

bit 0: underrange trace1; bit 1: underrange trace2;bit 2: underrange trace3;
bit 3: underrange trace4;bit 4: overrange trace1 to trace4

FINAL1, FINAL2, FINAL3 and FINAL4 are only possible in form of a query. The final measurement values are transmitted.

:TRACe:COPY :TRACe:COPY TRACE1| TRACE2| TRACE3| TRACE4 ,
TRACE1| TRACE2| TRACE3| TRACE4

This command copies data from one trace to another. The second operand designates the source, the first operand the destination of the data to be copied.

Example: `":TRAC:COPY TRACE1,TRACE2"`

Features: *RST value:
SCPI: conforming

Modes: R, A, VA

This command is an event and therefore has no query and no *RST value assigned.

:TRACe:FEED:CONTrol<1 to 4> ALWays | NEVer

This command switches block data transmission during a scan on and off.

Example: `"TRAC:FEED:CONT ALW"`

Features: *RST Value: NEVer
SCPI: conforming

Mode: R

The block size dependens on the scan time, the trace number is not evaluated.

TRIGger Subsystem

The TRIGger subsystem is used to synchronize instrument actions with events. This makes it possible to control and synchronize the start of a sweep. An external trigger signal can be fed to the connector at the rear panel of the instrument. In split screen mode, a distinction is made between TRIGger1 (screen A) and TRIGger2 (screen B).

COMMAND	PARAMETERS	UNIT	COMMENT
TRIGger<1|2>			
[:SEQuence]			
:SOURce	IMMediate | LINE | EXTernal |VIDeo		
:LEVel			
[:EXTernal]	<numeric_value>	V|MV|UV	
:VIDeo	<numeric_value>	PCT	
:HOLDoff	POSitive|NEGative	S	
:SLOPe		--	

:TRIGger<1|2>[:SEQuence]:SOURce IMMediate | LINE | EXTernal | VIDeo

This command selects the trigger source for the start of a sweep.

Example: `":TRIG:SOUR EXT"`

Features: *RST value: IMMediate
SCPI: conforming

Modes: R, A, VA

The value IMMediate corresponds to the "FREE RUN" setting.

:TRIGger<1|2>[:SEQuence]:LEVel[:EXTernal] –5.0 to +5.0V

This command sets the level of the external trigger source.

Example: `":TRIG:LEV 2V"`

Features: *RST value: –5.0V
SCPI: conforming

Modes: R, A, VA

:TRIGger<1|2>[:SEQuence]:LEVel:VIDeo 0 to 100PCT

This command sets the level of the video trigger source.

Example: `":TRIG:LEV:VID 50PCT"`

Features: *RST value: 50 PCT
SCPI: device-specific

Modes: R, A, VA

:TRIGger<1|2>[:SEQuence]:HOLDoff -100s to 100s

This command defines the length of the trigger delay.

Example: `":TRIG:HOLD 500us"`

Features: *RST value: 0s
SCPI: conforming

Modes: R, A, VA

A negative delay time (pretrigger) can be set in the time domain (SPAN < 0 Hz) only. The maximum permissible range and the maximum effective resolution of the pretrigger are limited by the set sweep time (max range= - 499/500 x sweep time; max. resolution = sweep time/500). Pretriggering is not possible when the rms or the average detector is activated.

:TRIGger<1|2>[:SEQuence]:SLOPe POSitive | NEGative

This command selects the slope of the trigger signal.

Example: `":TRIG:SLOP NEG"`

Features: *RST value: POSitive
SCPI: conforming

Modes: R, A, VA

The selected trigger slope applies to all trigger signal sources.

UNIT Subsystem

The UNIT subsystem is used to switch the basic unit of setting parameters. In split screen mode, a distinction is made between UNIT1 (screen A) and UNIT2 (screen B).

COMMAND	PARAMETERS	UNIT	COMMENT
UNIT<1|2>			
:POWer	DBM | DBPW | DBPT | WATT | DBUV | DBMV | VOLT | DBUA | AMPere V | W | DB | PCT | UNITLESS | DBUV_MHZ | DBMV_MHZ | DBUA_MHZ | DBUV_M | DBUA_M | DBUV_MMHZ | DBUA_MMHZ		
:PROBe	<Boolean>		

:UNIT<1|2>:POWer DBM | DBPW | DBPT | WATT | DBUV | DBMV | VOLT | DBUA | AMPere | V | W | DB | PCT | UNITLESS | DBUV_MHZ |DBMV_MHZ | DBUA_MHZ | DBUV_M | DBUA_M | DBUV_MMHZ | DBUA_MMHZ

This command selects the default unit for input and output.

Example: `":UNIT:POW DBUV"`

Features: *RST value: DBM
SCPI: conforming

Modes: R, A

:UNIT<1|2>:PROBe ON | OFF

This command determines whether the coding of a probe which is connected to the front panel is taken into consideration (ON) or not (OFF).

Example: `":UNIT:PROB OFF"`

Features: *RST value: ON
SCPI: device-specific

Modes: R, A

Alphabetical List of Commands

In the following, all remote-control commands are listed with their parameters and page numbers. Generally, they are arranged alphabetically according to the keywords of the command.

Command	Parameter	Page
:ABORt		6.7
:HOLD		6.7
:CALCulate<1|2>:DELTamarker<1 to 4>:AOFF		6.9
:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed:RPOint:X	<numeric_value>	6.12
:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed:RPOint:Y		6.12
:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed:RPOint:Y:OFFSet	<numeric_value>	6.12
:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:FIXed[:STATe]	ON | OFF	6.12
:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:PNOise:RESult?		6.13
:CALCulate<1|2>:DELTamarker<1 to 4>:FUNCtion:PNOise[:STATe]	ON | OFF	6.13
:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:APEak		6.10
:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:LEFT		6.11
:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:NEXT		6.10
:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum[:PEAK]		6.10
:CALCulate<1|2>:DELTamarker<1 to 4>:MAXimum:RIGHt		6.10
:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum:LEFT		6.11
:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum:NEXT		6.11
:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum[:PEAK]		6.11
:CALCulate<1|2>:DELTamarker<1 to 4>:MINimum:RIGHt		6.11
:CALCulate<1|2>:DELTamarker<1 to 4>:MODE	ABSolute | RELative	6.9
:CALCulate<1|2>:DELTamarker<1 to 4>:STEP:AUTO	ON | OFF	6.13
:CALCulate<1|2>:DELTamarker<1 to 4>:STEP[:INCRement]	<numeric value>	6.13
:CALCulate<1|2>:DELTamarker<1 to 4>:TRACe	1 to 4	6.9
:CALCulate<1|2>:DELTamarker<1 to 4>:X	0 to MAX (frequency | sweep time | symbols)	6.9
:CALCulate<1|2>:DELTamarker<1 to 4>:X:RELative?		6.9
:CALCulate<1|2>:DELTamarker<1 to 4>:Y?		6.10
:CALCulate<1|2>:DELTamarker<1 to 4>[:STATe]	ON | OFF	6.8
:CALCulate<1|2>:CTHReshold	MIN to MAX (depending on current unit)	6.15
:CALCulate<1|2>:CTHReshold:STATe	ON | OFF	6.15
:CALCulate<1|2>:DLINe<1|2>	MINimum to MAXimum (depending on current unit)	6.14
:CALCulate<1|2>:DLINe<1|2>:STATe	ON | OFF	6.14
:CALCulate<1|2>:FEED	'XTIM:DDEM:MEAS' | 'XTIM:DDEM:REF' | 'XTIM:DDEM:ERR:MPH' | 'XTIM:DDEM:ERR:VECT' | 'XTIM:DDEM:SYMB' | 'TCAP'	6.17
:CALCulate<1|2>:FLINe<1|2>	0 GHz to f_{max}	6.16
:CALCulate<1|2>:FLINe<1|2>:STATe	ON | OFF	6.16
:CALCulate<1|2>:FORMat	MAGNitude | PHASe | UPHase | RIMag | FREQuency | IEYE | QEYE | TEYE | FEYE | COMP | CONS	6.18
:CALCulate<1|2>:FSK:DEViation:REFerence	<numeric_value>	6.19
:CALCulate<1|2>:RLINe	MINimum to MAXimum (depending on current unit)	6.15

Command	Parameter	Page
:CALCulate<1|2>:RLINe:STATe	ON | OFF	6.15
:CALCulate<1|2>:THReshold	MINimum to MAXimum (depending on current unit)	6.14
:CALCulate<1|2>:THReshold:STATe	ON | OFF	6.15
:CALCulate<1|2>:TLINe<1|2>	0 to 1000s	6.16
:CALCulate<1|2>:TLINe<1|2>:STATe	ON | OFF	6.16
:CALCulate<1|2>:LIMit<1 to 8>:ACTive?		6.21
:CALCulate<1|2>:LIMit<1 to 8>:CATalog?		6.22
:CALCulate<1|2>:LIMit<1 to 8>:CLEar[:IMMediate]		6.26
:CALCulate<1|2>:LIMit<1 to 8>:COMMent	<string>	6.27
:CALCulate<1|2>:LIMit<1 to 8>:CONTrol[:DATA]	<numeric_value>,<numeric_value>..	6.22
:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:DOMain	FREQuency | TIME	6.22
:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:MODE	RELative | ABSolute	6.23
:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:OFFSet	<numeric_value>	6.22
:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:SHIFt	<numeric_value>	6.23
:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:SPACing	LINear < LOGarithmic	6.23
:CALCulate<1|2>:LIMit<1 to 8>:CONTrol:UNIT[:TIME]	S | SYM	6.23
:CALCulate<1|2>:LIMit<1 to 8>:COPY	1 to 8 | <name>	6.27
:CALCulate<1|2>:LIMit<1 to 8>:DELete		6.27
:CALCulate<1|2>:LIMit<1 to 8>:FAIL?		6.26
:CALCulate<1|2>:LIMit<1 to 8>:LOWer[:DATA]	<numeric_value>	6.25
:CALCulate<1|2>:LIMit<1 to 8>:LOWer:MARGin	<numeric_value>	6.25
:CALCulate<1|2>:LIMit<1 to 8>:LOWer:MODE	RELative | ABSolute	6.26
:CALCulate<1|2>:LIMit<1 to 8>:LOWer:OFFSet	<numeric_value>	6.25
:CALCulate<1|2>:LIMit<1 to 8>:LOWer:SHIFt	<numeric_value>	6.26
:CALCulate<1|2>:LIMit<1 to 8>:LOWer:SPACing	LINear | LOGarithmic	6.26
:CALCulate<1|2>:LIMit<1 to 8>:LOWer:STATe	ON | OFF	6.25
:CALCulate<1|2>:LIMit<1 to 8>:NAME	1 to 8 | <string>	6.27
:CALCulate<1|2>:LIMit<1 to 8>:STATe	ON | OFF	6.21
:CALCulate<1|2>:LIMit<1 to 8>:TRACe	1 to 4	6.21
:CALCulate<1|2>:LIMit<1 to 8>:UNIT	DBM | DBPW | DBPT | WATT | DBUV | DBMV | VOLT | DBUA | AMPere | DB | DBUV_MHZ | DBMV_MHZ | DBUA_MHZ |DBUV_M | DBUA_M | DBUV_MHZ | DBUA_MHZ | UNITLESS	6.21
:CALCulate<1|2>:LIMit<1 to 8>:UPPer[:DATA]	<numeric_value>,<numeric_value>..	6.23
:CALCulate<1|2>:LIMit<1 to 8>:UPPer:MARGin	<numeric_value>	6.24
:CALCulate<1|2>:LIMit<1 to 8>:UPPer:MODE	RELative | ABSolute	6.24
:CALCulate<1|2>:LIMit<1 to 8>:UPPer:OFFSet	<numeric_value>	6.24
:CALCulate<1|2>:LIMit<1 to 8>:UPPer:SHIFt	<numeric_value>	6.24
:CALCulate<1|2>:LIMit<1 to 8>:UPPer:SPACing	LINear | LOGarithmic	6.25
:CALCulate<1|2>:LIMit<1 to 8>:UPPer:STATe	ON | OFF	6.24
:CALCulate<1|2>:MARKer<1 to 4>:AOFF		6.30
:CALCulate<1|2>:MARKer<1 to 4>:COUNt	ON | OFF	6.31
:CALCulate<1|2>:MARKer<1 to 4>:COUNt:FREQuency?		6.32
:CALCulate<1|2>:MARKer<1 to 4>:COUNt:RESolution	0.1 | 1 | 10 | 100 | 1000 | 10000 Hz	6.31
:CALCulate<1|2>:MARKer<1 to 4>:COUPled[:STATe]	ON | OFF	6.32
:CALCulate<1|2>:MARKer<1 to 4>:LOEXclude	ON | OFF	6.32
:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:APEak		6.33

Command	Parameter	Page
:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:LEFT		6.33
:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:NEXT		6.33
:CALCulate<1|2>:MARKer<1 to 4>:MAXimum[:PEAK]		6.33
:CALCulate<1|2>:MARKer<1 to 4>:MAXimum:RIGHt		6.33
:CALCulate<1|2>:MARKer<1 to 4>:MINimum:LEFT		6.34
:CALCulate<1|2>:MARKer<1 to 4>:MINimum:NEXT		6.34
:CALCulate<1|2>:MARKer<1 to 4>:MINimum[:PEAK]		6.34
:CALCulate<1|2>:MARKer<1 to 4>:MINimum:RIGHt		6.34
:CALCulate<1|2>:MARKer<1 to 4>:SCOupled[:STATe]	ON | OFF	6.32
:CALCulate<1|2>:MARKer<1 to 4>:TRACe	1 to 4	6.31
:CALCulate<1|2>:MARKer<1 to 4>:X	0 to MAX(frequency | sweep time | symbols)	6.31
:CALCulate<1|2>:MARKer<1 to 4>:X:SLIMits[:STATe]	ON | OFF	6.31
:CALCulate<1|2>:MARKer<1 to 4>:Y?		6.32
:CALCulate<1|2>:MARKer<1 to 4>[:STATe]	ON | OFF	6.30
:CALCulate<1|2>:MARKer<1 to 4>:STEP[:INCRement]	<numeric_value>	6.34
:CALCulate<1|2>:MARKer<1 to 4>:STEP:AUTO	ON | OFF	6.35
:CALCulate<1|2>:MARKer<1 to 4>:PEXCursion	<numeric_value>	6.35
:CALCulate<1|2>:MARKer<1 to 4>:READout	MPHase | RIMaginary	6.35
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:CENTer:	ON | OFF	6.46
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:CSTep:	ON | OFF	6.46
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:DDEMod:RESult?	MERM | MEPK | MEPS | PERM PEPK | PEPS | EVRM | EVPK | EVPS | IQOF | IQIM |ADR | FERR | FEPK | RHO | DEV | FSRM | FSPK | FSPS | DTTS	6.38
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:MSTep		6.46
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown	<numeric_value>	6.35
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown:FREQuency?		6.36
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown:RESult?		6.36
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NDBDown:STATe	ON | OFF	6.35
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NOISe:RESult?		6.36
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:NOISe[:STATe]	ON | OFF	6.36
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:POWer:RESult?	OBANdwidth | OBWidth	6.38
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:POWer:SELect?	OBANdwidth | OBWidth	6.38
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:POWer[:STATe]	OFF	6.39
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor	(60dB/3dB) | (60dB/6dB)	6.37
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor:FREQuency?		6.37
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor:RESult?		6.37
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SFACtor:STATe	ON | OFF	6.37
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:STARt		6.46
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:STOP		6.46
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:STRack[:STATe]	ON | OFF	6.37
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AOFF		6.45
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:AVERage	ON | OFF	6.45
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAX:AVER:RES?		6.40
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAX:PHOLd:RES?		6.40
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAXimum:RESult?		6.39
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MAXimum[:STATe]	ON | OFF	6.39

Command	Parameter	Page
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN:AVER:RES?		6.45
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN:PHOLd:RES?		6.45
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN:RESult?		6.44
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MEAN[:STATe]	ON | OFF	6.44
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle:AVER:RES?		6.43
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle:PHOL:RES?		6.43
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle:RESult?		6.42
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MIDDle[:STATe]	ON | OFF	6.42
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak:AVER:RES?		6.42
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak:PHOL:RES?		6.42
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak:RESult?		6.41
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:MPEak[:STATe]	ON | OFF	6.41
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PHOLd	ON | OFF	6.45
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak:AVER:RES?		6.41
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak:PHOL:RES?		6.41
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak:RESult?		6.40
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:PPEak[:STATe]	ON | OFF	6.40
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS:AVER:RES?		6.44
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS:PHOLd:RES?		6.44
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS:RESult?		6.43
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:RMS[:STATe]	ON | OFF	6.43
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:SUMMary:STATe	ON | OFF	6.39
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:ZOOM	<numeric_value>	6.36
:CALCulate<1|2>:MARKer<1 to 4>:FUNCtion:REFerence:	ON | OFF	6.47
:CALCulate<1|2>:MATH<1 to 4>[:EXPRession][:DEFine]	<expr>	6.48
:CALCulate<1|2>:MATH<1 to 4>:STATe	ON | OFF	6.48
:CALCulate<1|2>:MATH:MODE	LINear | LOGarithmic	6.48
:CALCulate<1|2>:PEAKsearch|PSEarch[:IMMediate]		6.49
:CALCulate<1|2>:PEAKsearch|PSEarch:MARGin	MINimum to MAXimum	6.49
:CALCulate<1|2>:PEAKsearch|PSEarch: METHod	SUBRange | PEAK	6.50
:CALCulate<1|2>:PEAKsearch|PSEarch:SUBRanges	1 to 500	6.49
:CALCulate<1|2>:UNIT:ANGLe	DEG | RAD	6.51
:CALCulate<1|2>:UNIT:POWer	DBM | V | W | DB | PCT | UNITLESS | DBPT | DBPW | WATT | DBUV | DBMV | VOLT | DBUA | AMPere | DBUV_MHZ | DBMV_MHZ | DBUA_MHZ | DBUV_M | DBUA_M | DBUV_MMHZ | DBUA_MMHZ	6.51
:CALCulate<1|2>:X:UNIT:TIME	S | SYM	6.51
:CALibration[:ALL]		6.52
:CALibration:BANDwidth| BWIDth[:RESolution]?		6.52
:CALibration:IQ?		6.52
:CALibration:LDETector?		6.53
:CALibration:LOSuppression?		6.53
:CALibration:PPEak?		6.53
:CALibration:SHORt?		6.53
:CALibration:STATe	ON | OFF	6.53
:DIAGnostic:INFO:CCOunt:ATTenuation?		6.55
:DIAGnostic:INFO:CCOunt:PRESelector<1..6>?		6.55

Command	Parameter	Page
:DIAGnostic:SERVice:FUNCtion	numeric_value>,<numeric_value>...	6.54
:DIAGnostic:SERVice:INPut[:SELect]	CALibration | RF	6.54
:DIAGnostic:SERVice:NSOurce	ON | OFF	6.54
:DISPlay:ANNotation:FREQuency	ON | OFF	6.57
:DISPlay:BARGraph:LEVel:LOWer?		6.65
:DISPlay:BARGraph:LEVel:UPPer?		6.65
:DISPlay:CMAP<1 to 13>:DEFault		6.57
:DISPlay:CMAP<1 to 13>:HSL	0.0 to 100.0 (tint),0.0 to 100.0 (saturation),0.0 to 100.0 (brightness)	6.58
:DISPlay:CMAP<1 to 13>:PDEFined	<color>	6.58
:DISPlay:FORMat	SINGle | SPLit	6.57
:DISPlay:LOGO	ON | OFF	6.57
:DISPlay:PROGram[:MODE]	ON | OFF	6.57
:DISPlay:PSAVe:HOLDoff		6.66
:DISPlay:PSAVe[:STATe]	ON | OFF	6.66
:DISPlay[:WINDow<1|2>]:MINFo	ON | OFF	6.59
:DISPlay[:WINDow<1|2>]:TEXT[:DATA]	<string>	6.59
:DISPlay[:WINDow<1|2>]:TEXT:STATe	ON | OFF	6.59
:DISPlay[:WINDow<1|2>]:TIME	ON | OFF	6.59
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:EYE:COUNt	1 to Result Length	6.65
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE	WRITe | VIEW | AVERage | MAXHold | MINHold | FRESult	6.64
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE:ANALog	ON | OFF	6.64
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE:CWRite	ON | OFF	6.64
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:MODE:HCONtinuous	ON | OFF	6.64
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>[:STATe]	ON | OFF	6.65
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:SYMBol	DOTS | BARS | OFF	6.65
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X:SPACing	LINear | LOGarithmic	6.61
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:RVALue	<numeric_value>	6.59
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM	ON | OFF	6.60
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM[:FREQ]:CENTer	<numeric_value>	6.60
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM[:FREQ]:STARt	<numeric_value>	6.60
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:X[:SCALe]:ZOOM[:FREQ]:STOP	<numeric_value>	6.60
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y:SPACing	LINear | LOGarithmic | PERCent	6.63
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]	10dB to 200dB	6.61
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:BOTTom	<numeric value>	6.63
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:MODE	ABSolute | RELative	6.61
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:PDIVision	<numeric_value>	6.63
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RLEVel	-200dBm to 200dBm	6.61
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RLEVel:OFFSet	-200dB to 200dB	6.62
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RPOSition	0 to 100 PCT	6.62
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RVALue	<numeric_value>	6.62
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:RVALue:AUTO	ON | OFF	6.62
:DISPlay[:WINDow<1|2>]:TRACe<1 to 4>:Y[:SCALe]:TOP	<numeric value>	6.63
:FORMat[:DATA]	ASCii | REAL | UINT [,32]	6.67
:FORMat:DEXPort:APPend[:STATe]	ON | OFF	6.68
:FORMat:DEXPort:COMMent	<string>	6.68
:FORMat:DEXPort:DSEParator	POINt|COMMa	6.68

Command	Parameter	Page
:FORMat:DEXPort:HEADer[:STATe]	ON | OFF	6.68
:HCOPy:ABORt		6.69
:HCOPy:DESTination<1|2>	'MMEM' | 'SYST:COMM:PRIN' | 'SYST:COMM:CLIP'	6.70
:HCOPy:DEVice:COLor	ON | OFF	6.70
:HCOPy:DEVice:LANGuage<1|2>	WMF | EWMF | GDI | BMP	6.70
:HCOPy[:IMMediate]		6.71
:HCOPy:ITEM:ALL		6.71
:HCOPy:ITEM:FFEed<1|2>:STATe	ON | OFF	6.71
:HCOPy:ITEM:LABel:TEXT	<string>	6.71
:HCOPy:ITEM:PFEed<1|2>:STATe	ON | OFF	6.71
:HCOPy:ITEM:WINDow<1|2>:TABLe:STATe	ON | OFF	6.72
:HCOPy:ITEM:WINDow<1|2>:TEXT	<string>	6.72
:HCOPy:ITEM:WINDow<1|2>:TRACe:CAINcrement	ON | OFF	6.72
:HCOPy:ITEM:WINDow<1|2>:TRACe:STATe	ON | OFF	6.72
:HCOPy:PAGE:DIMensions:FULL		6.73
:HCOPy:PAGE:DIMensions:QUADrant<1 to 4>		6.73
:HCOPy:PAGE:ORIentation<1|2>	LANDscape | PORTrait	6.73
:INITiate<1|2>:CONTinuous	ON | OFF	6.74
:INITiate<1|2>:CONMeas		6.74
:INITiate<1|2>:DISPlay	ON | OFF	6.74
:INITiate<1|2>[:IMMediate]		6.74
:INPut<1|2>:ATTenuation	0 to 70 | 80 dB	6.76
:INPut<1|2>:ATTenuation:AUTO	ON | OFF	6.76
:INPut<1|2>:ATTenuation:AUTO:MODE	NORMal | LNOise | LDIStorsion	6.76
:INPut<1|2>:ATTenuation:PROTection	ON | OFF	6.76
:INPut<1|2>:ATTenuation:STEPsize	1dB | 10dB	6.77
:INPut<1|2>:COUPling	AC | DC	6.77
:INPut<1|2>:GAIN:AUTO	ON | OFF	6.78
:INPut<1|2>:GAIN:STATe	ON | OFF	6.78
:INPut<1|2>:MIXer	-10 to -100 dBm	6.77
:INPut<1|2>:PRESelection:CATalog?		6.79
:INPut<1|2>:PRESelection:COUPling:HIGH:FREQuency	5MHz to 500MHz	6.80
:INPut<1|2>:PRESelection:COUPling:HIGH:SET	'name of the preselector set for high resolution bandwidth'	6.80
:INPut<1|2>:PRESelection:COUPling:LOW:FREQuency	10Hz to 5MHz	6.79
:INPut<1|2>:PRESelection:COUPling:LOW:SET	'name of the preselector set for low resolution bandwidth'	6.79
:INPut<1|2>:PRESelection:COUPling:MID:SET	'name of the preselector set for medium resolution bandwidth'	6.79
:INPut<1|2>:PRESelection:FILTer:BPASS[:FREQuency]	40MHz to 1GHz, 65MHz to 2GHz	6.82
:INPut<1|2>:PRESelection:FILTer:HPASS[:FREQuency]	100Hz to 5MHz	6.82
:INPut<1|2>:PRESelection:FILTer:LPASS[:FREQuency]	20KHz to 40MHz	6.82
:INPut<1|2>:PRESelection:FILTer[:STATe]	ON | OFF	6.82
:INPut<1|2>:PRESelection:USET:CLEar		6.82
:INPut<1|2>:PRESelection:USET:COMMent	'Comment on preselector set'	6.81
:INPut<1|2>:PRESelection:USET:LRANge[:DATA]	<numeric value>,<numeric value>,<numeric_value>	6.81
:INPut<1|2>:PRESelection:USET:MRANge[:DATA]	<numeric value>,<numeric	6.81

Command	Parameter	Page
	value>,<numeric_value>	
:INPut<1|2>:UPORt<1|2>:STATe	ON | OFF	6.77
:INPut<1|2>:UPORt<1|2>[:VALue]		6.77
:INSTrument:COUPle	NONE | MODE | X | Y | CONTrol | XY | XCONtrol | YCONtrol | ALL	6.85
:INSTrument<1|2>:NSELect	1 | 2 | 6	6.84
:INSTrument<1|2>[:SELect]	SANalyzer | DDEMod | RECeiver	6.84
:MMEMory:CATalog?	<string>	6.87
:MMEMory:CDIRectory	directory name	6.87
:MMEMory:CLear:ALL		6.92
:MMEMory:CLear:STATe	1,path	6.91
:MMEMory:COMMent	<string>	6.94
:MMEMory:COPY	path, file name	6.88
:MMEMory:DATA	<file name>,<block data>	6.88
:MMEMory:DELete	path, file name	6.88
:MMEMory:INITialize	'A:'	6.88
:MMEMory:LOAD:AUTO	1,path	6.89
:MMEMory:LOAD:STATe	path, file name	6.89
:MMEMory:MDIRectory	path	6.89
:MMEMory:MOVE	path, file name	6.90
:MMEMory:MSIS	'A:' | 'C:'	6.90
:MMEMory:NAME	path, file name	6.90
:MMEMory:RDIRectory	directory name	6.90
:MMEMory:SELect[:ITEM]:ALL		6.94
:MMEMory:SELect[:ITEM]:CSETup	ON | OFF	6.93
:MMEMory:SELect[:ITEM]:DEFault		6.94
:MMEMory:SELect[:ITEM]:GSETup	ON | OFF	6.92
:MMEMory:SELect[:ITEM]:HCOPy	ON | OFF	6.93
:MMEMory:SELect[:ITEM]:HWSettings	ON | OFF	6.92
:MMEMory:SELect[:ITEM]:LINes:ALL	ON | OFF	6.93
:MMEMory:SELect[:ITEM]:LINes[:ACTive]	ON | OFF	6.92
:MMEMory:SELect[:ITEM]:MACRos	ON | OFF	6.93
:MMEMory:SELect[:ITEM]:NONE		6.94
:MMEMory:SELect[:ITEM]:TRACe<1 to 4>	ON | OFF	6.92
:MMEMory:SELect[:ITEM]:TRANsducer[:ACTive]	ON | OFF	6.93
:MMEMory:SELect[:ITEM]:TRANsducer:ALL	ON | OFF	6.94
:MMEMory:STORe:FINal	path	6.91
:MMEMory:STORe:STATe	path, file name	6.91
:MMEMory:STORe:TRACe	1 to 4, path	6.91
:OUTPut:UPORt<1|2>[:VALue]	#B00000000 to #B11111111	6.95
:OUTPut:UPORt<1|2>:STATe	ON | OFF	6.95
:[SENSe<1|2>:]AVERage:AUTO	ON | OFF	6.96
:[SENSe<1|2>:]AVERage:COUNt	0 to 256	6.96
:[SENSe<1|2>:]AVERage[:STATe]	ON | OFF	6.96
:[SENSe<1|2>:]AVERage:TYPE	MAXimum | SCALar	6.97
:[SENSe<1|2>:]BANDwidth|BWIDth:DEMod	<num_value>	6.100
:[SENSe<1|2>:]BANDwidth|BWIDth:DEMod:COUPling	ON | OFF	6.100

Command	Parameter	Page
:[SENSe<1|2>:]BANDwidth|BWIDth:PLL	AUTO | HIGH | MEDium | LOW	6.100
:[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]	10 Hz to 500MHz	6.98
:[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:AUTO	ON | OFF	6.98
:[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:RATio	0.0001 to 1	6.99
:[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo	1Hz to 500MHz	6.99
:[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:AUTO	ON | OFF	6.99
:[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:RATio	0.001 to 1000 | SINe | PULSe | NOISe	6.99
:[SENSe<1|2>:]CORRection:TRANsducer:ACTive?		6.101
:[SENSe<1|2>:]CORRection:TRANsducer:CATalog?		6.101
:[SENSe<1|2>:]CORRection:TRANsducer:COMMent	<string>	6.102
:[SENSe<1|2>:]CORRection:TRANsducer:DATA	<freq>,<level>..	6.103
:[SENSe<1|2>:]CORRection:TRANsducer:DELete		6.103
:[SENSe<1|2>:]CORRection:TRANsducer:SCALing	LINear|LOGarithmic	6.102
:[SENSe<1|2>:]CORRection:TRANsducer:SELect	<name>	6.102
:[SENSe<1|2>:]CORRection:TRANsducer[:STATe]	ON | OFF	6.103
:[SENSe<1|2>:]CORRection:TRANsducer:UNIT	<string>	6.102
:[SENSe<1|2>:]CORRection:TSET:ACTive?		6.104
:[SENSe<1|2>:]CORRection:TSET:BREak	ON | OFF	6.104
:[SENSe<1|2>:]CORRection:TSET:CATalog?		6.103
:[SENSe<1|2>:]CORRection:TSET:COMMent	<string>	6.105
:[SENSe<1|2>:]CORRection:TSET:DELete		6.105
:[SENSe<1|2>:]CORRection:TSET:RANGe<1 to 10>	<freq>,<freq>,<name>..	6.105
:[SENSe<1|2>:]CORRection:TSET:SELect	<name>	6.104
:[SENSe<1|2>:]CORRection:TSET[:STATe]	ON | OFF	6.105
:[SENSe<1|2>:]CORRection:TSET:UNIT	<string>	6.104
:[SENSe<1|2>:]DEMod	OFF | AM | AMVideo | FM | PM	6.106
:[SENSe<1|2>:]DEMod:FILTer[:LPASs]:FREQuency	<numeric_value>	6.106
:[SENSe<1|2>:]DEMod:FILTer[:LPASs]:AUTO	ON | OFF	6.107
:[SENSe<1|2>:]DEMod:FILTer:HPASs:FREQuency	0HZ | 10HZ | 100HZ | 1KHZ	6.107
:[SENSe<1|2>:]AM:RANGe[:UPPer]	3PCT | 10PCT | 100PCT	6.107
:[SENSe<1|2>:]FM[DEViation]:RANGe:AUTO	ON | OFF	6.107
:[SENSe<1|2>:]FM[DEViation]:RANGe[:UPPer]	<numeric_value>	6.107
:[SENSe<1|2>:]PM[DEViation]:RANGe[:UPPer]	<numeric_value>	6.107
:[SENSe<1|2>:]DETector:CMEM[:STATe]	ON | OFF	6.108
:[SENSe<1|2>:]DETector<1 to 4>:FMEasurement	NEGative | POSitive | RMS | AVERage | ACVideo	6.109
:[SENSe<1|2>:]DETector<1 to 4>[:FUNCtion]	APEak |NEGative | POSitive | SAMPle | RMS | AVERage | ACVideo	6.108
:[SENSe<1|2>:]DETector<1 to 4>[:FUNCtion]:AUTO	ON | OFF	6.109
:[SENSe<1|2>:]DETector<1 to 4>:PSTRetch[:STATe]	ON | OFF	6.109
:[SENSe<1|2>:]DETector<1 to 4>:PSTRetch:AUTO	ON | OFF	6.109
:[SENSe<1|2>:]DETector:RECeiver[:FUNCtion]	POSitive | NEGative | RMS | AVERage | ACVideo	6.109
:[SENSe<1|2>:]DDEMod:FILTer:ALPHa	0.2 to 1	6.113
:[SENSe<1|2>:]DDEMod:FILTer:MEASurement	OFF | RCOSine | RRCosine | GAUSsian | B22 | B25 | B44 | QFM | FM95 | QFR | FR95 | QRM | RM95 | QRR | RR95 | A25Fm | EMES | EREF	6.113
:[SENSe<1|2>:]DDEMod:FILTer:REFerence	RCOSine | RRCosine | GAUSsian | B22 |	6.113

Command	Parameter	Page
	B25 | B44 | QFM | FM95 | QFR | FR95 | QRM | RM95 | QRR | RR95 | A25Fm | EMES | EREF	
:[SENSe<1|2>:]DDEMod:FORMat	QPSK | PSK | MSK | QAM | FSK	6.111
:[SENSe<1|2>:]DDEMod:FSK:NSTate	2 | 4	6.112
:[SENSe<1|2>:]DDEMod:MSK:FORMat	TYPE1 | TYPE2 | NORMal | DIFFerential	6.112
:[SENSe<1|2>:]DDEMod:NORMalize	ON | OFF	6.114
:[SENSe<1|2>:]DDEMod:PRATe	1 | 2 | 4 | 8 | 16	6.113
:[SENSe<1|2>:]DDEMod:PRESet	GSM | EDGe | NADC | TETRa | DCS1800 | PCS1900 | PHS | PDCup | PDCDown | APCO25CQPSK | APCO25C4FM | CDPD | DECT | CT2 | ERMes | MODacom | PWT | TFTS | F16 | F322 | F324 | F64 | FQCDma | F95Cdma | RQCDma | R95Cdma | FNADc | RNADc	6.117
:[SENSe<1|2>:]DDEMod:PSK:FORMat	NORMal | DIFFerential | N3Pi8	6.111
:[SENSe<1|2>:]DDEMod:PSK:NSTate	2 | 8	6.111
:[SENSe<1|2>:]DDEMod:QAM:NSTate	16	6.112
:[SENSe<1|2>:]DDEMod:QPSK:FORMat	NORMal | DIFFerential | OFFSet | DPI4	6.111
:[SENSe<1|2>:]DDEMod:SBANd	NORMal | INVerse	6.111
:[SENSe<1|2>:]DDEMod:SEARch:PULSe:STATe	ON | OFF	6.114
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:CATalog?		6.114
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:COMMent	<string>	6.115
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:DATA	<string>	6.115
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:DELete	<string>	6.116
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:MONLy	ON | OFF	6.116
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:NAME	<string>	6.115
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:OFFSet	<numeric_value>	6.114
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:PATTern	<string>	6.115
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:SELect	<string>	6.114
:[SENSe<1|2>:]DDEMod:SEARch:SYNC:STATe	ON | OFF	6.115
:[SENSe<1|2>:]DDEMod:SEARch:TIME	100 to 1600	6.116
:[SENSe<1|2>:]DDEMod:SRATe	160 Hz to 1.6 MHz	6.112
:[SENSe<1|2>:]DDEMod:TIME	1 to Frame Length	6.112
:[SENSe<1|2>:]FILTer:HPASs:FREQuency	100 Hz | 1 kHz | 10 kHz	6.118
:[SENSe<1|2>:]FILTer:HPASS[:STATe]	ON | OFF	6.118
:[SENSe<1|2>:]FILTer:NOTCh:STATe	ON | OFF	6.118
:[SENSe<1|2>:]FREQuency:CENTer	0 GHz to f_{max}	6.119
:[SENSe<1|2>:]FREQuency:CENTer:LINK	STARt | STOP | SPAN	6.119
:[SENSe<1|2>:]FREQuency:CENTer:STEP	0 to f_{max}	6.120
:[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK	SPAN | RBW | OFF	6.120
:[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor	1 to 100 PCT	6.120
:[SENSe<1|2>:]FREQuency[:CW|:FIXed]	f_{min} to f_{max}	6.123
:[SENSe<1|2>:]FREQuency[:CW|:FIXed]:STEP	f_{min} to f_{max}	6.123
:[SENSe<1|2>:]FREQuency:MODE	CW|FIXed | SWEep	6.122
:[SENSe<1|2>:]FREQuency:OFFSet	<numeric_value>	6.122
:[SENSe<1|2>:]FREQuency:RANGe	2 GHZ | 22 GHZ	6.120
:[SENSe<1|2>:]FREQuency:SPAN	0 GHz to f_{max}	6.120
:[SENSe<1|2>:]FREQuency:SPAN:FULL		6.121
:[SENSe<1|2>:]FREQuency:SPAN:LINK	CENTer | STOP | SPAN	6.121

Command	Parameter	Page
:[SENSe<1|2>:]FREQuency:STARt	0 GHz to f_{max}	6.121
:[SENSe<1|2>:]FREQuency:STARt:FLINe[:STATe]	ON | OFF	6.121
:[SENSe<1|2>:]FREQuency:STARt:LINK	CENTer | STOP | SPAN	6.121
:[SENSe<1|2>:]FREQuency:STOP	0 GHz to f_{max}	6.122
:[SENSe<1|2>:]FREQuency:STOP:FLINe[:STATe]	ON | OFF	6.122
:[SENSe<1|2>:]FREQuency:STOP:LINK	CENTer | STARt | SPAN	6.122
:[SENSe<1|2>:]POWer:BANDwidth|BWIDth	0 to 100PCT	6.124
:[SENSe<1|2>:]ROSCillator:SOURce	INTernal | EXTernal	6.125
:[SENSe<1|2>:]ROSCillator:EXTernal:FREQuency	1MHz to 16MHz	6.125
:[SENSe<1|2>:]ROSCillator[:INTernal]:TUNe	0 to 4095	6.125
:[SENSe<1|2>:]ROSCillator[:INTernal]:TUNe:SAVe		6.125
:[SENSe<1|2>:]SCAN<1 to 10>:BANDwidth:RESolution	f_{min} to f_{max}	6.127
:[SENSe<1|2>:]SCAN<1 to 10>:INPut:ATTenuation	dB_{min} to dB_{max}	6.127
:[SENSe<1|2>:]SCAN<1 to 10>:INPut:ATTenuation:AUTO	ON | OFF	6.128
:[SENSe<1|2>:]SCAN<1 to 10>:INPut:BIMPedance	150 | 600 | 10000 OHM	6.127
:[SENSe<1|2>:]SCAN<1 to 10>:INPut:GAIN	0dB | 10dB | 20dB | 30dB	6.128
:[SENSe<1|2>:]SCAN<1 to 10>:INPut:GAIN:AUTO	ON | OFF	6.128
:[SENSe<1|2>:]SCAN<1 to 10>:INPut:GAIN:STATe	ON | OFF	6.128
:[SENSe<1|2>:]SCAN<1 to 10>:INPut:TYPE	RF | BALanced	6.127
:[SENSe<1|2>:]SCAN:RANGes[:COUNt]	1 to 10	6.128
:[SENSe<1|2>:]SCAN<1 to 10>:STARt	f_{min} to f_{max}	6.126
:[SENSe<1|2>:]SCAN<1 to 10>:STEP	f_{min} to f_{max}	6.126
:[SENSe<1|2>:]SCAN<1 to 10>:STOP	f_{min} to f_{max}	6.126
:[SENSe<1|2>:]SCAN<1 to 10>:TIME	100 □s to 100 s	6.127
:[SENSe<1|2>:]SWEep:COUNt	0 to 32767	6.130
:[SENSe<1|2>:]SWEep:EGATe	ON | OFF	6.130
:[SENSe<1|2>:]SWEep:EGATe:HOLDoff	0 to 100s	6.131
:[SENSe<1|2>:]SWEep:EGATe:LENGth	0 to 100s	6.131
:[SENSe<1|2>:]SWEep:EGATe:LEVel	-5V to +5V	6.130
:[SENSe<1|2>:]SWEep:EGATe:POLarity	POSitive | NEGative	6.131
:[SENSe<1|2>:]SWEep:EGATe:TYPE	LEVel | EDGE	6.130
:[SENSe<1|2>:]SWEep:GAP	ON | OFF	6.131
:[SENSe<1|2>:]SWEep:GAP:LENGth	0 to 100s	6.132
:[SENSe<1|2>:]SWEep:GAP:PRETrigger	0 to 100s	6.131
:[SENSe<1|2>:]SWEep:GAP:TRGTogap	0 to 100s	6.132
:[SENSe<1|2>:]SWEep:SPACing	LINear | LOGarithmic | AUTO	6.132
:[SENSe<1|2>:]SWEep:TIME	<numeric_value>	6.129
:[SENSe<1|2>:]SWEep:TIME:AUTO	ON | OFF	6.129
:[SENSe<1|2>:]SWEep:TIME:FMEasurement	<numeric_value>	6.130
:[SENSe<1|2>:]TCAPture:LENGth	1024 | 2048 | 4096 | 8192 | 16384	6.117
:STATus:OPERation:CONDition?		6.134
:STATus:OPERation:ENABle	0 to 65535	6.134
:STATus:OPERation[:EVENt]?		6.134
:STATus:OPERation:NTRansition	0 to 65535	6.135
:STATus:OPERation:PTRansition	0 to 65535	6.135
:STATus:PRESet		6.135
:STATus:QUEStionable:CONDition?		6.135

Command	Parameter	Page
:STATus:QUEStionable:ENABle	0 to 65535	6.136
:STATus:QUEStionable[:EVENt]?		6.135
:STATus:QUEStionable:NTRansition	0 to 65535	6.136
:STATus:QUEStionable:PTRansition	0 to 65535	6.136
:STATus:QUEStionable:FREQuency:CONDition?		6.140
:STATus:QUEStionable:FREQuency:ENABle	0 to 65535	6.141
:STATus:QUEStionable:FREQuency[:EVENt]?		6.140
:STATus:QUEStionable:FREQuency:NTRansition	0 to 65535	6.141
:STATus:QUEStionable:FREQuency:PTRansition	0 to 65535	6.141
:STATus:QUEStionable:LIMit:CONDition?		6.137
:STATus:QUEStionable:LIMit:ENABle	0 to 65535	6.138
:STATus:QUEStionable:LIMit[:EVENt]?		6.137
:STATus:QUEStionable:LIMit:NTRansition	0 to 65535	6.138
:STATus:QUEStionable:LIMit:PTRansition	0 to 65535	6.138
:STATus:QUEStionable:LMARgin:CONDition?		6.138
:STATus:QUEStionable:LMARgin:ENABle	0 to 65535	6.139
:STATus:QUEStionable:LMARgin[:EVENt]?		6.138
:STATus:QUEStionable:LMARgin:NTRansition	0 to 65535	6.139
:STATus:QUEStionable:LMARgin:PTRansition	0 to 65535	6.139
:STATus:QUEStionable:POWer:CONDition?		6.136
:STATus:QUEStionable:POWer:ENABle	0 to 65535	6.137
:STATus:QUEStionable:POWer[:EVENt]?		6.136
:STATus:QUEStionable:POWer:NTRansition	0 to 65535	6.137
:STATus:QUEStionable:POWer:PTRansition	0 to 65535	6.137
:STATus:QUEStionable:SYNC:CONDition?		6.139
:STATus:QUEStionable:SYNC:ENABle	0 to 65535	6.140
:STATus:QUEStionable:SYNC[:EVENt]?		6.139
:STATus:QUEStionable:SYNC:NTRansition	0 to 65535	6.140
:STATus:QUEStionable:SYNC:PTRansition	0 to 65535	6.140
:STATus:QUEStionable:TRANsducer:CONDition?		6.141
:STATus:QUEStionable:TRANsducer:ENABle	0 to 65535	6.142
:STATus:QUEStionable:TRANsducer[:EVENt]?		6.141
:STATus:QUEStionable:TRANsducer:NTRansition	0 to 65535	6.142
:STATus:QUEStionable:TRANsducer:PTRansition	0 to 65535	6.142
:STATus:QUEue[:NEXT]?		6.142
:SYSTem:BINFo?		6.148
:SYSTem:COMMunicate:GPIB:RDEVice<1|2>:ADDRess	0 to 30	6.144
:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess	0 to 30	6.143
:SYSTem:COMMunicate:GPIB[:SELF]:RTERminator	LFEOI | EOI	6.144
:SYSTem:COMMunicate:PRINter<1|2>:ENUMerate:FIRSt?		6.145
:SYSTem:COMMunicate:PRINter<1|2>:ENUMerate:NEXT?		6.146
:SYSTem:COMMunicate:PRINter<1|2>:SELect	<printer_name>	6.146
:SYSTem:COMMunicate:SERial<1|2>:CONTrol:DTR	IBFull | OFF	6.144
:SYSTem:COMMunicate:SERial<1|2>:CONTrol:RTS	IBFull | OFF	6.144
:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:BAUD	<numeric_value>	6.144
:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:BITS	7 | 8	6.145
:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:PACE	XON | NONE	6.145

Command	Parameter	Page
:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:PARity[:TYPE]	EVEN | ODD | NONE	6.145
:SYSTem:COMMunicate:SERial<1|2>[:RECeive]:SBITs	1 | 2	6.145
:SYSTem:DATE	1980 to 2099, 1 to 12, 1 to 31	6.146
:SYSTem:DISPlay:UPDate	ON | OFF	6.146
:SYSTem:ERRor?		6.147
:SYSTem:PASSword[:CENable]	'password'	6.147
:SYSTem:PRESet		6.147
:SYSTem:SET	<block>	6.147
:SYSTem:TIME	0 to 23, 0 to 59, 0 to 59	6.148
:SYSTem:VERSion?		6.148
:TRACe:COPY	TRACE1 | TRACE2 | TRACE3 | TRACE4 ,TRACE1 | TRACE2 | TRACE3 | TRACE4	6.151
:TRACe[:DATA]	TRACE1| TRACE2| TRACE3| TRACE4| SINGle| SCAN| STATus| FINAL1| FINAL2| FINAL3| FINAL4, <block> | <numeric_value>	6.149
:TRACe:FEED:CONTrol<1 to 4>	ALWays | NEVer	6.151
:TRIGger<1|2>[:SEQuence]:HOLDoff	-100s to 100 s	6.153
:TRIGger<1|2>[:SEQuence]:LEVel:VIDeo	0 to 100 PCT	6.152
:TRIGger<1|2>[:SEQuence]:LEVel[:EXTernal]	–5.0 V to +5.0 V	6.152
:TRIGger<1|2>[:SEQuence]:SLOPe	POSitive | NEGative	6.153
:TRIGger<1|2>[:SEQuence]:SOURce	IMMediate | LINE | EXTernal | VIDeo	6.152
:UNIT<1|2>:POWer	DBM | DBPW | DBPT | WATT | DBUV | DBMV | VOLT | DBUA | AMPere | V | W | DB | PCT | UNITLESS | DBUV_MHZ | DBMV_MHZ | DBUA_MHZ | DBUV_M | DBUA_M | DBUV_MMHZ | DBUA_MMHZ	6.154
:UNIT<1|2>:PROBe	ON | OFF	6.154

Table of Softkeys with IEC/IEEE-Bus Command Assignment

Signal Analysis Mode

FREQUENCY Key Group

START	
START MANUAL	[SENSe<1|2>:]FREQuency:STARt <num_value>
CENTER FIXED	[SENSe<1|2>:]FREQuency:STARt:LINK CENTer
SPAN FIXED	[SENSe<1|2>:]FREQuency:STARt:LINK SPAN
STOP FIXED	[SENSe<1|2>:]FREQuency:STARt:LINK STOP
START AT FREQ LINE	[SENSe<1|2>:]FREQuency:STARt:FLINe[:STATe] ON | OFF FSET
FREQ AXIS LIN LOG	[SENSe<1|2>:]SWEep:SPACing LIN | LOG
STOP	
STOP MANUAL	[SENSe<1|2>:]FREQuency:STOP <num_value>
START FIXED	[SENSe<1|2>:]FREQuency:STOP:LINK STARt
CENTER FIXED	[SENSe<1|2>:]FREQuency:STOP:LINK CENTer
SPAN FIXED	[SENSe<1|2>:]FREQuency:STOP:LINK SPAN
SPAN FIXED	[SENSe<1|2>:]FREQuency:STOP:LINK SPAN
STOP AT FREQ LINE	[SENSe<1|2>:]FREQuency:STOP:FLINe[:STATe] ON | OFF FSET
FREQ AXIS LIN LOG	[SENSe<1|2>:]SWEep:SPACing LIN | LOG
FREQ/ CENTER	
CENTER MANUAL	[SENSe<1|2>:]FREQuency:CENTer <num_value>
START FIXED	[SENSe<1|2>:]FREQuency:CENTer:LINK STARt
SPAN FIXED	[SENSe<1|2>:]FREQuency:CENTer:LINK SPAN
STOP FIXED	[SENSe<1|2>:]FREQuency:CENTer:LINK STOP
FREQUENCY OFFSET	[SENSe<1|2>:]FREQuency:OFFSet <num_value>
FREQ AXIS LIN LOG	[SENSe<1|2>:]SWEep:SPACing LIN | LOG

STEP

AUTO 0.1 * SPAN	[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK SPAN; [SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor 10PCT
or	
AUTO 0.1 * RBW	[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK RBW; [SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor 10PCT
AUTO 0.5 * SPAN	[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK SPAN; [SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor 50PCT
or	
AUTO 0.5 * RBW	[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK RBW; [SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor 50PCT
AUTO X * SPAN	[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK SPAN; [SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor <num_value>
or	
AUTO X * RBW	[SENSe<1|2>:]FREQuency:CENTer:STEP:LINK RBW; [SENSe<1|2>:]FREQuency:CENTer:STEP:LINK:FACTor <num_value>
STEPSIZE MANUAL	[SENSe<1|2>:]FREQuency:CENTer:STEP <num_value>
STEPSIZE = CENTER	no corresponding IEC/IEEE-bus command

ZOOM/ SPAN

SPAN MANUAL	[SENSe<1|2>:]FREQuency:SPAN <num_value>
START FIXED	[SENSe<1|2>:]FREQuency:SPAN:LINK START
CENTER FIXED	[SENSe<1|2>:]FREQuency:SPAN:LINK CENTer
STOP FIXED	[SENSe<1|2>:]FREQuency:SPAN:LINK STOP
ZERO SPAN	[SENSe<1|2>:]FREQuency:SPAN 0HZ or [SENSe<1|2>:]FREQuency:MODE CW|FIXed
FULL SPAN	[SENSe<1|2>:]FREQuency:SPAN:FULL
LAST SPAN	no corresponding IEC/IEEE-bus command
ZOOM	DISPlay[:WINDow<1|2>]:TRACe<1...4>:X[:SCALe]:ZOOM ON| OFF
MOVE ZOOM WINDOW	DISPlay[:WINDow<1|2>]:TRACe<1...4>:X[:SCALe]:ZOOM[:FREQuency]:CENTer <num_value>
MOVE ZOOM START	DISPlay[:WINDow<1|2>]:TRACe<1...4>:X[:SCALe]:ZOOM[:FREQuency]:STARt <num_value>
MOVE ZOOM STOP	DISPlay[:WINDow<1|2>]:TRACe<1...4>:X[:SCALe]:ZOOM[:FREQuency]:STOP <num_value>
ZOOM OFF	DISPlay[:WINDow<1|2>]:TRACe<1...4>:X[:SCALe]:ZOOM OFF
FREQ AXIS LIN LOG	[SENSe<1|2>:]SWEep:SPACing LIN | LOG

LEVEL Key Group

UNIT/ REF

REF LEVEL	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RLEVel <num_value>
REF LEVEL OFFSET	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RLEVel:OFFSet <num_value>
GRID ABS/REL	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:MODE ABSolute|RELative
UNIT	no corresponding IEC/IEEE-bus command
dBm	CALCulate<1|2>:UNIT:POWer DBM
dBmV	CALCulate<1|2>:UNIT:POWer DBMV
dBµV	CALCulate<1|2>:UNIT:POWer DBUV
dBµA	CALCulate<1|2>:UNIT:POWer DBUA
dBpW	CALCulate<1|2>:UNIT:POWer DBPW
dB*/MHz	CALCulate<1|2>:UNIT:POWer DBUV_MHZ CALCulate<1|2>:UNIT:POWer DBUA_MHZ CALCulate<1|2>:UNIT:POWer DBMV_MHZ
VOLT	CALCulate<1|2>:UNIT:POWer VOLT
AMPERE	CALCulate<1|2>:UNIT:POWer AMPere
WATT	CALCulate<1|2>:UNIT:POWer WATT
PROBE CODE ON / OFF	UNIT:PROBe ON|OFF
RF ATTEN MANUAL	INPut<1|2>:ATTenuation <num_value>
ATTEN AUTO NORMAL	INPut<1|2>:ATTenuation:AUTO:MODE NORMal INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW NOISE	INPut<1|2>:ATTenuation:AUTO:MODE LNOise INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW DIST	INPut<1|2>:ATTenuation:AUTO:MODE LDIStortion INPut<1|2>:ATTenuation:AUTO ON
MIXER LEVEL	INPut<1|2>:MIXer <numeric value>
MAX LEVEL AUTO	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RVALue:AUTO ON
MAX LEVEL MANUAL	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RVALue:AUTO OFF DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RVALue <num_value>

RANGE

LOG 120 dB	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing LOGarithmic DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe] 120
LOG 100 dB	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing LOGarithmic DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe] 100
LOG 50 dB	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing LOGarithmic DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe] 50
LOG 20 dB	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing LOGarithmic DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe] 20
LOG 10 dB	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing LOGarithmic DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe] 10
LOG MANUAL	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing LOGarithmic DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe] <num_value>
LINEAR/dB	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing LINear
LINEAR/%	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y:SPACing PERCent
GRID ABS/REL	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:MODE ABSolute|RELative

INPUT Key

INPUT	
RF ATTEN MANUAL	INPut<1|2>:ATTenuation <num_value>
ATTEN AUTO NORMAL	INPut<1|2>:ATTenuation:AUTO:MODE NORMal INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW NOISE	INPut<1|2>:ATTenuation:AUTO:MODE LNOise INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW DIST	INPut<1|2>:ATTenuation:AUTO:MODE LDIStortion INPut<1|2>:ATTenuation:AUTO ON
MIXER LEVEL	INPut<1|2>:MIXer <numeric value>
INPUT SELECT	--
RF INPUT 50 OHM	INPut<1|2>:IMPedance 50
RF INPUT DC COUPLED	INPut<1|2>:COUPling DC INPut<1|2>:TYPE RF
RF INPUT AC COUPLED	INPut<1|2>:COUPling AC INPut<1|2>:TYPE RF
BAL INPUT 150 OHM	INPut<1|2>:BIMPedance 150 INPut<1|2>:TYPE BALanced
BAL INPUT 600 OHM	INPut<1|2>:BIMPedance 600 INPut<1|2>:TYPE BALanced
BAL INPUT 10 KOHM	INPut<1|2>:BIMPedance 10000 INPut<1|2>:TYPE BALanced

MARKER Key Group

NORMAL

MARKER 1..4	CALCulate<1|2>:MARKer<1...4>[:STATe] ON | OFF CALCulate<1|2>:MARKer<1...4>:X <numeric value>; CALCulate<1|2>:MARKer<1...4>:Y?
SIGNAL COUNT	CALCulate<1|2>:MARKer<1...4>:COUNt ON | OFF CALCulate<1|2>:MARKer<1...4>:COUNt:FREQuency?
MARKER ZOOM	CALCulate<1|2>:MARKer<1...4>:FUNCtion:ZOOM <num_value>
MARKER INFO	DISPlay:WINDow<1|2>:MINFo ON | OFF (indication)
ALL MARKER OFF	CALCulate<1|2>:MARKer<1...4>:AOFF
POWER MEAS SETTINGS	no corresponding IEC/IEEE-bus command
% POWER BANDWIDTH	[SENSe<1|2>:]POWer: BANDwidth|BWIDth <num_value>
OCCUPIED PWR BANDW	CALCulate<1|2>:MARKer<1...4>:FUNCtion:POWer:SELect OBANdwidth | OBWidth CALCulate<1|2>:MARKer<1...4>:FUNCtion:POWer:RESult? OBANdwidth| OBWidth CALCulate<1|2>:MARKer<1...4>:FUNCtion:POWer[:STATe] OFF
COUNTER RESOL	CALCulate<1|2>:MARKer<1...4>:COUNt:RESolution <numeric value>
SIGNAL TRACK	CALCulate<1|2>:MARKer<1...4>:FUNCtion:STRack[:STATe] ON | OFF
NOISE	CALCulate<1|2>:MARKer<1...4>:FUNCtion:NOISe[:STATe] ON | OFF CALCulate<1|2>:MARKer<1...4>:FUNCtion:NOISe:RESult?

STEP

STEPSIZE AUTO	CALCulate<1|2>:MARKer<1...4>:STEP:AUTO ON | OFF
STEPSIZE MANUAL	CALCulate<1|2>:MARKer<1...4>:STEP[:INCRement] <num_value>
MKR TO STEPSIZE	CALCulate<1|2>:MARKer<1...4>:FUNCtion:MSTep
DELTA TO STEPSIZE	--

DELTA

DELTA 1...4	CALCulate<1|2>:DELTamarker<1...4>[:STATe] ON | OFF CALCulate<1|2>:DELTamarker<1...4>:X <numeric value> CALCulate<1|2>:DELTamarker<1...4>:X:RELative? CALCulate<1|2>:DELTamarker<1...4>:Y?
PHASE NOISE	CALCulate<1|2>:DELTamarker<1...4>:FUNCtion:PNOise[:STATe] ON | OFF CALCulate<1|2>:DELTamarker<1...4>:FUNCtion:PNOise:RESult?
REFERENCE POINT	
REF POINT LEVEL	CALCulate<1|2>:DELTamarker<1..4>:FUNCtion:FIXed:RPOint:Y <num_value>
REF POINT LVL OFFSET	CALCulate<1|2>:DELTamarker<1..4>:FUNCtion:FIXed:RPOint:Y:OFFSet <num_value>
REF POINT FREQUENCY	CALCulate<1|2>:DELTamarker<1..4>:FUNCtion:FIXed:RPOint:X <num_value>
REF POINT TIME	CALCulate<1|2>:DELTamarker<1..4>:FUNCtion:FIXed:RPOint:X <num_value>
REFERENCE FIXED	CALCulate<1|2>:DELTamarker<1...4>:FUNCtion:FIXed[:STATe] ON | OFF
DELTA MKR ABS REL	CALCulate<1|2>:DELTamarker<1...4>:MODE ABSolute | RELative
ALL DELTA OFF	CALCulate<1|2>:DELTamarker<1...4>:AOFF
STEP	
STEPSIZE AUTO	CALCulate<1|2>:DELTamarker<1...4>:STEP:AUTO ON | OFF
MANUAL STEPSIZE	CALCulate<1|2>:DELTamarker<1...4>:STEP[:INCRement] <num_value>
DELTA TO STEPSIZE	--

SEARCH

PEAK	CALCulate<1|2>:MARKer<1...4>:MAXimum[:PEAK] CALCulate<1|2>:DELTamarker<1...4>:MAXimum[:PEAK]
NEXT PEAK	CALCulate<1|2>:MARKer<1...4>:MAXimum:NEXT CALCulate<1|2>:DELTamarker<1...4>:MAXimum:NEXT
NEXT PEAK RIGHT	CALCulate<1|2>:MARKer<1...4>:MAXimum:RIGHt CALCulate<1|2>:DELTamarker<1...4>:MAXimum:RIGHt
NEXT PEAK LEFT	CALCulate<1|2>:MARKer<1...4>:MAXimum:LEFT CALCulate<1|2>:DELTamarker<1...4>:MAXimum:LEFT
SUM MKR ON/OFF	CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary[:STATe] ON | OFF
SUMMARY MARKER	
RMS	CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:RMS[:STATe] ON | OFF CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:RMS:RESult? CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:RMS:AVERage:RESult? CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:RMS:PHOLd:RESult?
MEAN	CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:MEAN[:STATe] ON | OFF CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:MEAN:RESult? CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:MEAN:AVERage:RESult? CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:MEAN:PHOLd:RESult?
PEAK HOLD ON/OFF	CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:PHOLd ON | OFF
AVERAGE ON/OFF	CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:AVERage ON | OFF
SWEEP COUNT	[SENSe<1|2>:]SWEep:COUNt <num_value>
ALL SUM MKR OFF	CALCulate<1|2>:MARKer<1...4>:FUNCtion:SUMMary:AOFF
SEARCH LIM ON/OFF	CALCulate<1|2>:MARKer<1...4>:X:SLIMits[:STATe] ON | OFF
SELECT MARKER	no corresponding IEC/IEEE-bus command
ACTIVE MKR/DELTA	no corresponding IEC/IEEE-bus command
MIN	CALCulate<1|2>:MARKer<1...4>:MINimum[:PEAK] CALCulate<1|2>:DELTamarker<1...4>:MINimum[:PEAK]
NEXT MIN	CALCulate<1|2>:MARKer<1...4>:MINimum:NEXT CALCulate<1|2>:DELTamarker<1...4>:MINimum:NEXT
NEXT MIN RIGHT	CALCulate<1|2>:MARKer<1...4>:MINimum:RIGHt CALCulate<1|2>:DELTamarker<1...4>:MINimum:RIGHt
NEXT MIN LEFT	CALCulate<1|2>:MARKer<1...4>:MINimum:LEFT CALCulate<1|2>:DELTamarker<1...4>:MINimum:LEFT
EXCLUDE LO ON/OFF	CALCulate<1|2>:MARKer<1...4>:LOEXclude ON | OFF

PEAK EXCURSION

```
CALCulate<1|2>:MARKer<1...4>:PEXCursion <num_value>
```

N dB DOWN

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:NDBDown <num_value>
CALCulate<1|2>:MARKer<1...4>:FUNCtion:NDBDown:STATe ON | OFF
CALCulate<1|2>:MARKer<1...4>:FUNCtion:NDBDown:RESult?
CALCulate<1|2>:MARKer<1...4>:FUNCtion:NDBDown:FREQuency?
```

SHAPE FACT 60/3 dB

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor (60dB/3dB)
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor:STATe ON | OFF
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor:RESult?
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor:FREQuency?
```

SHAPE FACT 60/6 dB

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor (60dB/6dB)
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor:STATe ON | OFF
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor:RESult?
CALCulate<1|2>:MARKer<1...4>:FUNCtion:SFACtor:FREQuency?
```

MKR->

PEAK

```
CALCulate<1|2>:MARKer<1...4>:MAXimum[:PEAK]
CALCulate<1|2>:DELTamarker<1...4>:MAXimum[:PEAK]
```

MKR-> CENTER

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:CENTer
```

MKR-> REF LEVEL

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:REFerence
```

MKR-> REF LEVEL

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:REFerence
```

ADD TO PEAK LIST

no corresponding IEC/IEEE-bus command

MKR-> START

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:STARt
```

MKR-> STOP

```
CALCulate<1|2>:MARKer<1...4>:FUNCtion:STOP
```

MKR-> TRACE

```
CALCulate<1|2>:MARKer<1...4>:TRACe <numeric value>
CALCulate<1|2>:DELTamarker<1...4>:TRACe <numeric value>
```

SELECT MARKER

no corresponding IEC/IEEE-bus command

ACTIVE MKR/DELTA

no corresponding IEC/IEEE-bus command

LINES Key Group

D LINES

DISPLAY LINE 1/2	CALCulate<1|2>DLINe<1|2>:STATe ON | OFF CALCulate<1|2>DLINe<1|2> <num_value>
THRESHOLD LINE	CALCulate<1|2>THReshold ON | OFF CALCulate<1|2>THReshold <num_value>
REFERENCE LINE	CALCulate<1|2>RLINe:STATe ON | OFF CALCulate<1|2>RLINe <num_value>
FREQUENCY LINE 1/2	CALCulate<1|2>FLINe<1|2>:STATe ON | OFF CALCulate<1|2>FLINe<1|2> <num_value>
or	
TIME LINE 1/2	CALCulate<1|2>TLINe<1|2>:STATe ON | OFF CALCulate<1|2>TLINe<1|2> <num_value>
BASELINE CLIPPING	CALCulate<1|2>:CTHReshold:STATe ON | OFF CALCulate<1|2>:CTHReshold -82dBm

LIMITS

SELECT LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:NAME <string> CALCulate<1|2>:LIMit<1...8>:STATe ON | OFF
NEW LIMIT LINE	see EDIT LIMIT LINE
NAME	CALCulate<1|2>:LIMit<1...8>:NAME <string>
VALUES	no corresponding IEC/IEEE-bus command
INSERT VALUE	no corresponding IEC/IEEE-bus command
DELETE VALUE	no corresponding IEC/IEEE-bus command
SHIFT X LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:CONTrol:SHIFt <num_value>
SHIFT Y LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:UPPer:SHIFt <num_value> CALCulate<1|2>:LIMit<1...8>:LOWer:SHIFt <num_value>
SAVE LIMIT LINE	automatically executed during IEC/IEEE-bus operation

EDIT LIMIT LINE

```
CALCulate<1|2>:LIMit<1...8>:UNIT DBM| DBPW| WATT| DBUV| VOLT|DBUA|
        AMPere| DB| DBUV MHZ| DBUA MHZ| DEG| RAD| S| HZ| PCT| UNITLESS
CALCulate<1|2>:LIMit<1...8>:COMMent 'string'
CALCulate<1|2>:LIMit<1...8>:TRACe <num_value>
CALCulate<1|2>:LIMit<1...8>:CONTrol[:DATA] <num_value>, <num_value>..
CALCulate<1|2>:LIMit<1...8>:CONTrol:DOMain   FREQuency|TIME
CALCulate<1|2>:LIMit<1...8>:CONTrol:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:CONTrol:MODE   RELative | ABSolute
CALCulate<1|2>:LIMit<1...8>:CONTrol:UNIT[:TIME] S | SYM
CALCulate<1|2>:LIMit<1...8>:CONTrol:SPACing LINear | LOGarithmic
CALCulate<1|2>:LIMit<1...8>:UPPer[:DATA] <num_value>, <num_value>..
CALCulate<1|2>:LIMit<1...8>:UPPer:STATe   ON | OFF
CALCulate<1|2>:LIMit<1...8>:UPPer:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:UPPer:MARGin <num_value>
CALCulate<1|2>:LIMit<1...8>:UPPer:MODE   RELative | ABSolute
CALCulate<1|2>:LIMit<1...8>:UPPer:SPACing LINear | LOGarithmic
CALCulate<1|2>:LIMit<1...8>:LOWer[:DATA] <num_value>,<num_value>..
CALCulate<1|2>:LIMit<1...8>:LOWer:STATe ON | OFF
CALCulate<1|2>:LIMit<1...8>:LOWer:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:LOWer:MARGin <num_value>
CALCulate<1|2>:LIMit<1...8>:LOWer:MODE   RELative | ABSolute
CALCulate<1|2>:LIMit<1...8>:LOWer:SPACing LINear | LOGarithmic
CALCulate<1|2>:LIMit<1...8>:FAIL?
CALCulate<1|2>:LIMit<1...8>:CLEar[:IMMediate]
```

COPY LIMIT LINE

```
CALCulate<1|2>:LIMit<1...8>:COPY 1...8 | <name>
```

DELETE LIMIT LINE

```
CALCulate<1|2>:LIMit<1...8>:DELete
```

X OFFSET

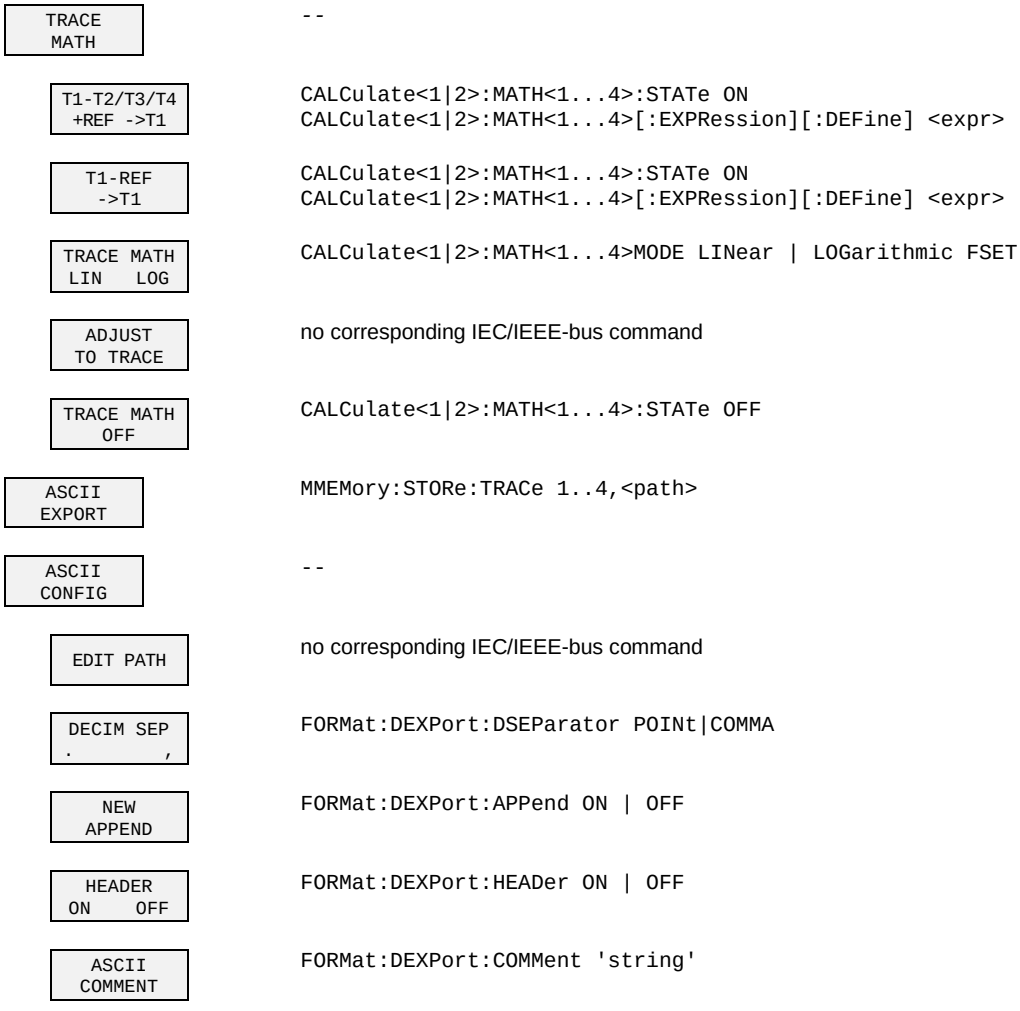
```
CALCulate<1|2>:LIMit<1...8>:CONTrol:OFFset <num_value>
```

Y OFFSET

```
CALCulate<1|2>:LIMit<1...8>:UPPer:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:LOWer:OFFset <num_value>
```

TRACE Key Group

TRACE 1	
CLEAR/ WRITE	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE WRITe
VIEW	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE VIEW
BLANK	DISPlay[:WINDow<1|2>]:TRACe<1...4>[:STATe] OFF
AVERAGE	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE AVERage or [SENSe<1|2>:]AVERage:MODE SCALe
MAX HOLD	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE MAXHold or [SENSe<1|2>:]AVERage:MODE MAX
MIN HOLD	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE MINHold or [SENSe<1|2>:]AVERage:MODE MIN
HOLD CONT ON/OFF	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE:HCONtinuous ON|OFF
SWEEP COUNT	[SENSe<1|2>:]SWEep:COUNt <num_value>
DETECTOR	--
AUTO SELECT	[SENSe<1|2>:]DETector[:FUNCtion]:AUTO ON | OFF
DETECTOR AUTOPEAK	[SENSe<1|2>:]DETector[:FUNCtion] APEak
DETECTOR MAX PEAK	[SENSe<1|2>:]DETector[:FUNCtion] POSitive
DETECTOR MIN PEAK	[SENSe<1|2>:]DETector[:FUNCtion] NEGative
DETECTOR SAMPLE	[SENSe<1|2>:]DETector[:FUNCtion] SAMPle
DETECTOR RMS	[SENSe<1|2>:]DETector[:FUNCtion] RMS
DETECTOR AVERAGE	[SENSe<1|2>:]DETector[:FUNCtion] AVERage
DETECTOR AC VIDEO	[SENSe<1|2>:]DETector[:FUNCtion] ACVideo
CAPT MEM ON OFF	[SENSe<1|2>:]DETector:CMEM[:STATe] ON | OFF FSET
COPY..	TRACe:COPY TRACE1| TRACE2| TRACE3| TRACE4 , TRACE1| TRACE2| TRACE3| TRACE4
ANALOG TR ON/OFF	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE:ANALog ON|OFF


TRACE MATH
--
T1-T2/T3/T4 +REF ->T1
CALCulate<1|2>:MATH<1...4>:STATe ON
CALCulate<1|2>:MATH<1...4>[:EXPRession][:DEFine] <expr>
T1-REF ->T1
CALCulate<1|2>:MATH<1...4>:STATe ON
CALCulate<1|2>:MATH<1...4>[:EXPRession][:DEFine] <expr>
TRACE MATH LIN LOG
CALCulate<1|2>:MATH<1...4>MODE LINear | LOGarithmic FSET
ADJUST TO TRACE
no corresponding IEC/IEEE-bus command
TRACE MATH OFF
CALCulate<1|2>:MATH<1...4>:STATe OFF
ASCII EXPORT
MMEMory:STORe:TRACe 1..4,<path>
ASCII CONFIG
--
EDIT PATH
no corresponding IEC/IEEE-bus command
DECIM SEP . ,
FORMat:DEXPort:DSEParator POINt|COMMA
NEW APPEND
FORMat:DEXPort:APPend ON | OFF
HEADER ON OFF
FORMat:DEXPort:HEADer ON | OFF
ASCII COMMENT
FORMat:DEXPort:COMMent 'string'

SWEEP Key Group

RUN/ COUPLING	
RES BW MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution] <num_value>
RES BW AUTO	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:AUTO ON | OFF
VIDEO BW MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo <num_value>
VIDEO BW AUTO	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:AUTO ON | OFF
SWEEP TIME MANUAL	[SENSe<1|2>:]SWEep:TIME <num_value>
SWEEP TIME AUTO	[SENSe<1|2>:]SWEep:TIME:AUTO ON | OFF
COUPLING DEFAULT	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:AUTO ON [SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:AUTO ON [SENSe<1|2>:]SWEep:TIME:AUTO ON
COUPLING RATIO	--
RBW / VBW SINE [1]	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:RATio SINe
RBW / VBW PULSE [.1]	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:RATio PULSe
RBW / VBW NOISE [10]	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:RATio NOISe
RBW / VBW MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:RATio <num_value>
SPAN / RBW AUTO [50]	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:RATio 0.02
SPAN / RBW MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:RATio <num_value>
CENTER FREQUENCY	[SENSe<1|2>:]FREQuency:CENTer 1.3 GHz
RBW UP	[SENSe<1|2>:]BWIDth[:RESolution]:AUTO OFF [SENSe<1|2>:]BWIDth[:RESolution] 1MHz
RBW DOWN	[SENSe<1|2>:]BWIDth[:RESolution]:AUTO OFF [SENSe<1|2>:]BWIDth[:RESolution] 1MHz
REF LEVEL UP	DISPlay[:WINDow<1|2>]:TRACe<1..4>:Y:RLEVel -60DBM
REF LEVEL DOWN	DISPlay[:WINDow<1|2>]:TRACe<1..4>:Y:RLEVel -60DBM
ATTEN OUP	INPut<1|2>:ATTenuation 40DB

ATTEN DOWN	INPut<1|2>:ATTenuation 40DB
PULS STRET ON OFF	[SENSe<1|2>:]DETector<1...4>:PSTRetch[:STATe] ON | OFF [SENSe<1|2>:]DETector<1...4>:PSTRetch:AUTO ON | OFF
EXT VIDEO ON OFF	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:EXTernal:STATe ON | OFF
AC VIDEO COUPLING	[SENSe<1|2>:]FILTer:HPASs:STATe ON | OFF [SENSe<1|2>:]FILTer:HPASs 10 kHz | 1 kHz | 100 Hz
VIDEO NOTCH ON OFF	[SENSe<1|2>:]FILTer:NOTCh:STATe ON | OFF
MAIN PLL BANDWIDTH	[SENSe<1|2>:]BANDwidth|BWIDth:PLL AUTO|HIGH|MEDium|LOW
TRIGGER	
FREE RUN	TRIGger<1|2>:[:SEQuence]:SOURce IMMediate
VIDEO	TRIGger<1|2>:[:SEQuence]:SOURce VIDeo TRIGger<1|2>:[:SEQuence]:LEVel:VIDeo <numeric value>
LINE	TRIGger<1|2>:[:SEQuence]:SOURce LINE
EXTERN	TRIGger<1|2>:[:SEQuence]:SOURce EXTernal TRIGger<1|2>:[:SEQuence]:LEVel[:EXTernal] -5.0...+5.0V
TRIGGER DELAY	TRIGger<1|2>:[:SEQuence]:HOLDoff <num_value>
SLOPE POS/NEG	TRIGger<1|2>:[:SEQuence]:SLOPe POSitive|NEGative
SCAN/ SWEEP	
CONTINUOUS SWEEP	INITiate<1|2>::CONTinuous ON; INITiate[:IMMediate]
SINGLE SWEEP	INITiate<1|2>::CONTinuous OFF; INITiate[:IMMediate]
SWEEP TIME AUTO	[SENSe<1|2>:]SWEep:TIME:AUTO ON | OFF
SWEEP TIME MANUAL	[SENSe<1|2>:]SWEep:TIME <num_value>
SWEEP COUNT	[SENSe<1|2>:]SWEep:COUNt <num_value>
GAP SWEEP ON/OFF	[SENSe<1|2>:]SWEep:GAP ON | OFF
GAP SWEEP SETTINGS	--

TRIGGER LEVEL	TRIGger<1|2>:[:SEQuence]:LEVel:VIDeo <numeric value>
PRE TRIGGER	[SENSe<1|2>:]SWEep:GAP:PRETrigger <num_value>
TRG TO GAP TIME	[SENSe<1|2>:]SWEep:GAP:TRGTogap <num_value>
GAP LENGTH	[SENSe<1|2>:]SWEep:GAP:LENGth <num_value>
GATE ON / OFF	[SENSe<1|2>:]SWEep:EGATe ON | OFF
GATE SETTINGS	--
GATE LEVEL	[SENSe<1|2>:]SWEep:EGATe:LEVel <num_value>
GATE MODE LEVEL/EDGE	[SENSe<1|2>:]SWEep:EGATe:TYPE LEVel | EDGE
GATE POL POS/NEG	[SENSe<1|2>:]SWEep:EGATe:POLarity POSitive | NEGative
GATE DELAY	[SENSe<1|2>:]SWEep:EGATe:HOLDoff <num_value>
GATE LENGTH	[SENSe<1|2>:]SWEep:EGATe:LENGth <num_value>
GATE ADJUST	
GATE LEVEL	[SENSe<1|2>:]SWEep:EGATe:LEVel <num_value>
GATE MODE LEVEL/EDGE	[SENSe<1|2>:]SWEep:EGATe:TYPE LEVel | EDGE
GATE POL POS/NEG	[SENSe<1|2>:]SWEep:EGATe:POLarity POSitive | NEGative
GATE DELAY	[SENSe<1|2>:]SWEep:EGATe:HOLDoff <num_value>
GATE LENGTH	[SENSe<1|2>:]SWEep:EGATe:LENGth <num_value>
SWEEP TIME MANUAL	[SENSe<1|2>:]SWEep:TIME <num_value>
RES BW MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution] <num_value>
VIDEO MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo <num_value>
VIDEO AUTO	[SENSe<1|2>:]BANDwidth|BWIDth:VIDeo:AUTO ON | OFF
SGL SWEEP DISP OFF	INITiate<1|2>::DISPlay ON | OFF

General Device Settings

DATA VARIATION Key Group

HOLD	
UNLOCK	no corresponding IEC/IEEE-bus command
LOCK DATA	no corresponding IEC/IEEE-bus command
LOCK ALL	no corresponding IEC/IEEE-bus command
STEP	if needed, the step width is entered in the subsystem of the corresponding parameter.
STEPSIZE AUTO	--
STEPSIZE MANUAL	--

SYSTEM Key Group

DISPLAY

FULL SCREEN	DISPlay:FORmat SINGle
SPLIT SCREEN	DISPlay:FORmat SPLit
ACTIVE SCREEN x	The screen is selected via the numeric suffix
SCREEN A BARGRAPH	INSTrument2[:SELect] RECeiver This softkey is only available in Receiver mode
SCREEN A SWEEP	INSTrument[:SELect] SANalyzer This softkey is only available in Receiver mode
SCREEN COUPLING	
MODE COUPLED	INSTrument<1|2>:COUPle MODE
HORIZONTAL SCALING	INSTrument<1|2>:COUPle X
VERTICAL SCALING	INSTrument<1|2>:COUPle Y
COUPLING CONTROL	INSTrument<1|2>:COUPle CONTrol
SCREENS UNCOUPLED	INSTrument<1|2>:COUPle NONE | ALL
CONFIG DISPLAY	
SELECT OBJECT	
BRIGHTNESS	DISPlay:CMAP:HSL <hue>,<sat>,<lum>
TINT	DISPlay:CMAP<1...13>:HSL <hue>,<sat>,<lum>
SATURATION	DISPlay:CMAP<1...13>:HSL <hue>,<sat>,<lum>
DEFAULT COLORS	DISPlay:CMAP<1...13>:DEFault
PREDEFINED COLORS	DISPlay:CMAP<1...13>:PDEFined BLACk | BLUE | BROWn | GREen | CYAN | RED | MAGenta | YELLow | WHITe |DGRAy| LGRAy | LBLUe | LGREen | LCYan | LRED| LMAGenta
LOGO ON/OFF	DISPlay:LOGO ON|OFF
FREQUENCY ON/OFF	DISPlay:ANNotation:FREQuency ON|OFF

DATA ENTRY FIELD	--
DATAENTRY X	no corresponding IEC/IEEE-bus command
DATAENTRY Y	no corresponding IEC/IEEE-bus command
DEFAULT POSITION	no corresponding IEC/IEEE-bus command
DATAENTRY OPAQUE	no corresponding IEC/IEEE-bus command
TIME ON OFF	DISPlay[:WINDow<1|2>]:TIME ON | OFF
DISPLAY COMMENT	DISPlay[:WINDow<1|2>]:TEXT[:DATA] <string> DISPlay[:WINDow<1|2>]:TEXT:STATe ON | OFF
SCR SAFER ON OFF	DISPlay:PSAVer[:STATe] ON | OFF
SCR SAFER TIME	DISPlay:PSAVer:HOLDoff <num_value>

CAL

CAL SHORT	CALibration:SHORt?
CAL TOTAL	CALibration[:ALL]?
CAL RES BW	CALibration:BANDwidth|BWIDth[:RESolution]?
CAL LOG	CALibration:LDETector?
CAL LO SUPP	CALibration:LOSuppression?
CAL I/Q	CALibration:IQ?
PRESEL PEAK	CALibration:PPEak?
CAL CORR ON/OFF	CALibration:STATe ON | OFF
CAL RESULTS	no corresponding IEC/IEEE-bus command

INFO	
FIRMWARE VERSION	*IDN?
HARDWARE+ OPTIONS	*OPT?
SELFTEST	*TST?
EXECUTE TEST	*TST?
SYSTEM MESSAGES	SYSTem:ERRor?
CLEAR MESSAGE	SYSTem:ERRor?
CLEAR ALL MESSAGES	--
UPDATE MESSAGES	--
OPTIONS	*OPT?
STATISTIC	no corresponding IEC/IEEE-bus command
ATT SWITCHES	DIAGnostic:INFO:CCOunt:ATTenuation<1|3>
RF PRESS SWITCHES	DIAGnostic:INFO:CCOunt::PRESelector<1..6>

CONFIGURATION Key Group

MODE

The submenus are described under the associated operating mode.

ANALYZER

```
INSTrument<1|2>:[:SELect]   SANalyzer
INSTrument<1|2>::NSELect   1
```

RECEIVER

```
INSTrument<1|2>:[:SELect]   RECeiver
INSTrument<1|2>::NSELect   6
```

VECTOR ANALYZER

```
INSTrument<1|2>:[:SELect]   DDEMod
INSTrument<1|2>:NSELect   3
```

AF DEMOD

```
[SENSe<1|2>:]DEModulation OFF | AM | FM | PM | AMVideo
```

FREQ RANGE 2 GHz

```
[SENSe<1|2>:]FREQuency:RANGe 2E9
```

FREQ RANGE 22 GHz

```
[SENSe<1|2>:]FREQuency:RANGe 22E9
```

SETUP

TRANSDUCER

```
--
```

TRANSDUCER FACTOR

```
[SENSe<1|2>:]CORRection:TRANsducer:SELect <name>
[SENSe<1|2>:]CORRection:TRANsducer[:STATe] ON | OFF
```

TRANSDUCER SET

```
[SENSe<1|2>:]CORRection:TSET:SELect <name>
[SENSe<1|2>:]CORRection:TSET[:STATe] ON | OFF
```

EDIT TRD FACTOR

```
[SENSe<1|2>:]CORRection:TRANsducer:SCALing LINear|LOGarithmic
```

TRD FACTOR NAME

```
[SENSe<1|2>:]CORRection:TRANsducer:SELect <name>
```

TRD FACTOR UNIT

```
[SENSe<1|2>:]CORRection:TRANsducer:UNIT <string>
```

TRD FACTOR VALUES

```
[SENSe<1|2>:]CORRection:TRANsducer:DATA <freq>,<level>..
```

INSERT LINE

no corresponding IEC/IEEE-bus command

DELETE LINE

no corresponding IEC/IEEE-bus command

SAVE TRD FACTOR

automatically executed during IEC/IEEE-bus operation

EDIT TRD SET

```
[SENSe<1|2>:]CORRection:TSET:BREak ON|OFF
```

TRANSD SET NAME

```
[SENSe<1|2>:]CORRection:TSET:SELect <name>
```

TRANSD SET UNIT

```
[SENSe<1|2>:]CORRection:TSET:UNIT <string>
```

TRANSD SET RANGES

```
[SENSe<1|2>:]CORRection:TSET:RANGe<1...10> <freq>,<freql>,<name>..
```

INSERT LINE	no corresponding IEC/IEEE-bus command
DELETE LINE	no corresponding IEC/IEEE-bus command
SAVE TRD SET	automatically executed during IEC/IEEE-bus operation
NEW FACT/SET	see EDIT TRD FACTOR or EDIT TRD SET
DELETE FACTOR/SET	`[SENSe<1|2>:]CORRection:TRANsducer:DELete` `[SENSe<1|2>:]CORRection:TSET:DELete`
PREAMP ON OFF	`INPut<1|2>:GAIN [:STATe] ON | OFF`
PREAMP GAIN	`INPut<1|2>:GAIN <numeric_value>`
DIR MODE ON OFF	`INPut<1|2>:PRESelection[:STATe] ON | OFF`
PRESELECT CONFIG	
COUPLED TO RBW	`INPut<1|2>:PRESelection:COUPling ON | OFF`
NORMAL	`INPut<1|2>:PRESelection:SET 'NORMAL'`
NARROW	`INPut<1|2>:PRESelection:SET 'NARROW'`
WIDE	`INPut<1|2>:PRESelection:SET 'WIDE'`
SELECT SET	
SELECT	`INPut<1|2>:PRESelection:USET[:SELect] 'name'`

NEW SET	`INPut<1|2>:PRESelection:USET:NAME 'name'`
	`INPut<1|2>:PRESelection:USET:COMMent 'comment'`
	`INPut<1|2>:PRESelection:USET:LRANge[:DATA]`
	`INPut<1|2>:PRESelection:USET:MRANge[:DATA]`

EDIT SET	see NEW SET
COPY SET	no corresponding IEC/IEEE-bus command
DELETE SET	`INPut<1|2>:PRESelection:USET[:SELect] 'name'` `INPut<1|2>:PRESelection:USET:CLEar`

RBW CPLD SETTING	`INPut<1|2>:PRESelection:COUPling:LOW:FREQuency`
	`INPut<1|2>:PRESelection:COUPling:LOW:SET 'set_name'`
	`INPut<1|2>:PRESelection:COUPling:MID:SET 'set_name'`
	`INPut<1|2>:PRESelection:COUPling:HIGH:FREQuency`
	`INPut<1|2>:PRESelection:COUPling:HIGH:SET 'set_name'`

SELECT FILTER	
LOW RANGE	INPut<1|2>:PRESelection:FILTer:LPASs[:FREQuency] <num_value> INPut<1|2>:PRESelection:FILTer:HPASs[:FREQuency] <num_value>
MID RANGE	INPut<1|2>:PRESelection:FILTer:BPASs[:FREQuency] <num_value>
FILTER SET MAN	INPut<1|2>:PRESelection:FILT[:STATe] ON | OFF
OPTIONS	no corresponding IEC/IEEE-bus command
REFERENCE INT/EXT	[SENSe<1|2>:]ROSCillator:SOURce INTernal|EXTernal
EXT REF FREQUENCY	[SENSe<1|2>:]ROSCillator:EXTernal:FREQuency <num_value>
SERVICE	--
INPUT RF	DIAGnostic:SERVice:INPut[:SELect] RF
INPUT CAL	DIAGnostic:SERVice:INPut[:SELect] CALibration
NOISE SOURCE	DIAGnostic:SERVice:NSOurce ON | OFF
REFERENCE ADJUST	--
REFERENCE	[SENSe<1|2>:]ROSCillator:[INTernal:]TUNe <num_value>
REFERENCE PROG	[SENSe<1|2>:]ROSCillator:[INTernal:]TUNe:SAVe
HARMONIC GENERATOR	DIAGnostic:SERVice:HGENerator FF | 10KHZ | 100KHZ | BALanced
ENTER PASSWORD	SYSTem:PASSword[:CENable] <string>
GENERAL SETUP	--
GPIB ADDRESS	SYSTem:COMMunicate:GPIB[:SELF]:ADDRess 0 à 30
USER PORT A/B	INPut<1|2>:UPORt<1|2>[:VALue]? INPut<1|2>:UPORt<1|2>:STATe ON | OFF OUTPut<1|2>:UPORt<1|2>:STATe ON | OFF OUTPut<1|2>:UPORt<1|2>[:VALue] <Binary>
COM PORT 1/2	SYSTem:COMMunicate:SERial<1|2>:CONTrol:DTR IBFull | OFF SYSTem:COMMunicate:SERial<1|2>:CONTrol:RTS IBFull | OFF SYSTem:COMMunicate:SERial<1|2>[:RECeive]:BAUD <numeric_value> SYSTem:COMMunicate:SERial<1|2>[:RECeive]:BITS 7 | 8 SYSTem:COMMunicate:SERial<1|2>[:RECeive]:PARity[:TYPE] EVEN| ODD| NONE SYSTem:COMMunicate:SERial<1|2>[:RECeive]:SBITs 1|2 SYSTem:COMMunicate:SERial<1|2>[:RECeive]:PACE XON | NONE

TIME	SYSTem:TIME 0 à 23, 0 à 59, 0 à 59
DATE	SYSTem:DATE <num>,<num>,<num>
MONITOR CONNECTED	--
KEY CLICK ON/OFF	--
FIRMWARE UPDATE	no corresponding IEC/IEEE-bus command

STATUS Key Group

LOCAL	Device message "Go to LOCAL (GTL)"

HARDCOPY Key Group

START

```
HCOPy[:IMMediate<1|2>]
```

SETTINGS

COPY SCREEN

```
HCOPy:ITEM:ALL
```

COPY TRACE

```
HCOPy:ITEM:WINDow<1|2>:TRACe:STATe   ON | OFF
```

COPY TABLE

```
HCOPy:ITEM:WINDow<1|2>:TABle:STATe   ON | OFF
```

SELECT QUADRANT

```
--
```

- UPPER LEFT

```
HCOPy:PAGE:DIMensions:QUADrant 1
```

- LOWER LEFT

```
HCOPy:PAGE:DIMensions:QUADrant 2
```

- UPPER RIGHT

```
HCOPy:PAGE:DIMensions:QUADrant 3
```

- LOWER RIGHT

```
HCOPy:PAGE:DIMensions:QUADrant 4
```

- FULL PAGE

```
HCOPy:PAGE:DIMensions:FULL
```

ENTER TEXT

```
--
```

- COMMENT SCREEN A/B

```
HCOPy:ITEM:WINDow<1|2>:TEXT   <string>
```

- TITLE

```
HCOPy:ITEM:LABel:TEXT   <string>
```

HARDCOPY DEVICE

```
HCOPy:DEVice:LANGuage WMF|EWMF |BMP |GDI
```

- SETTINGS DEVICE 1/2

```
HCOPy:DESTination<1|2>   <string>
MMEMory:NAME <file_name>
HCOPy:ITEM:FFEed<1|2>:STATe   ON|OFF
HCOPy:PAGE:ORIentation<1|2>   LANDscape | PORTrait
```

- ENABLE DEV1/DEV2

The numeric suffix after `HCOPy:IMMediate` denotes the first or second device.

COLOR ON/OFF

```
HCOPy:DEVice:COLor ON | OFF
```

TRC COLOR AUTO INC

```
HCOPy:ITEM:WINDow<1|2>:TRACe:CAINcrement ON | OFF
```

MEMORY Key Group

CONFIG

EDIT PATH	MMEMory:MSIS <device> MMEMory:CDIRectory <directory_name>
DELETE	MMEMory:DELete <file_name> MMEMory:RDIRectory <directory_name>
FORMAT DISK	MMEMory:INITialize <msus>
MAKE DIRECTORY	MMEMory:MDIRectory <directory_name>
RENAME	MMEMory:MOVE <file_source>,<file_destination>
SORT MODE	no corresponding IEC/IEEE-bus command
COPY	MMEMory:COPY <file_source>,<file_destination>

SAVE

EDIT NAME	MMEMory:STORe:STATe 1,<file_name>
EDIT PATH	The path is included in the file names
EDIT COMMENT	MMEMory:COMMent <string>
SELECT ITEMS TO SAVE	--
SELECT ITEMS	MMEMory:SELect[:ITEM]:GSETup ON|OFF MMEMory:SELect[:ITEM]:HWSettings ON|OFF MMEMory:SELect[:ITEM]:TRACe<1...4> ON|OFF MMEMory:SELect[:ITEM]:LINes[:ACTive] ON|OFF MMEMory:SELect[:ITEM]:LINes:ALL ON|OFF MMEMory:SELect[:ITEM]:CSETup ON|OFF MMEMory:SELect[:ITEM]:HCOPy ON|OFF MMEMory:SELect[:ITEM]:MACRos ON|OFF MMEMory:SELect[:ITEM]:TRANsducer[:ACTive] ON|OFF MMEMory:SELect[:ITEM]:TRANsducer:ALL ON|OFF
ENABLE ALL ITEMS	MMEMory:SELect[:ITEM]:ALL
DISABLE ALL ITEMS	MMEMory:SELect[:ITEM]:NONE
DEFAULT CONFIG	MMEMory:SELect[:ITEM]:DEFault
DATA SET LIST	--

DATA SET CLEAR

```
MMEMory:CLEar:STATe 1,<file_name>
```

DATA SET CLEAR ALL

```
MMEMory:CLEar:ALL
```

RECALL

EDIT NAME

```
MMEMory:LOAD:STATe 1,<file_name>
```

EDIT PATH

The path is included in the file names.

AUTO RECALL

```
MMEMory:LOAD:AUTO 1,<file_name>
```

SELECT ITEMS TO RECALL

```
--
```

SELECT ITEMS

```
MMEMory:SELect[:ITEM]:GSETup ON|OFF
MMEMory:SELect[:ITEM]:HWSettings ON|OFF
MMEMory:SELect[:ITEM]:TRACe<1...4> ON|OFF
MMEMory:SELect[:ITEM]:LINes[:ACTive] ON|OFF
MMEMory:SELect[:ITEM]:LINes:ALL ON|OFF
MMEMory:SELect[:ITEM]:CSETup ON|OFF
MMEMory:SELect[:ITEM]:HCOPy ON|OFF
MMEMory:SELect[:ITEM]:CDATa ON|OFF
MMEMory:SELect[:ITEM]:MACRos ON|OFF
MMEMory:SELect[:ITEM]:TRANsducer[:ACTive] ON|OFF
MMEMory:SELect[:ITEM]:TRANsducer:ALL ON|OFF
```

ENABLE ALL ITEMS

```
MMEMory:SELect[:ITEM]:ALL
```

DISABLE ALL ITEMS

```
MMEMory:SELect[:ITEM]:NONE
```

DEFAULT CONFIG

```
MMEMory:SELect[:ITEM]:DEFault
```

DATA SET LIST

```
--
```

DATA SET CLEAR

```
MMEMory:CLEar:STATe 1,<file_name>
```

DATA SET CLEAR ALL

```
MMEMory:CLEar:ALL
```

USER Key

USER

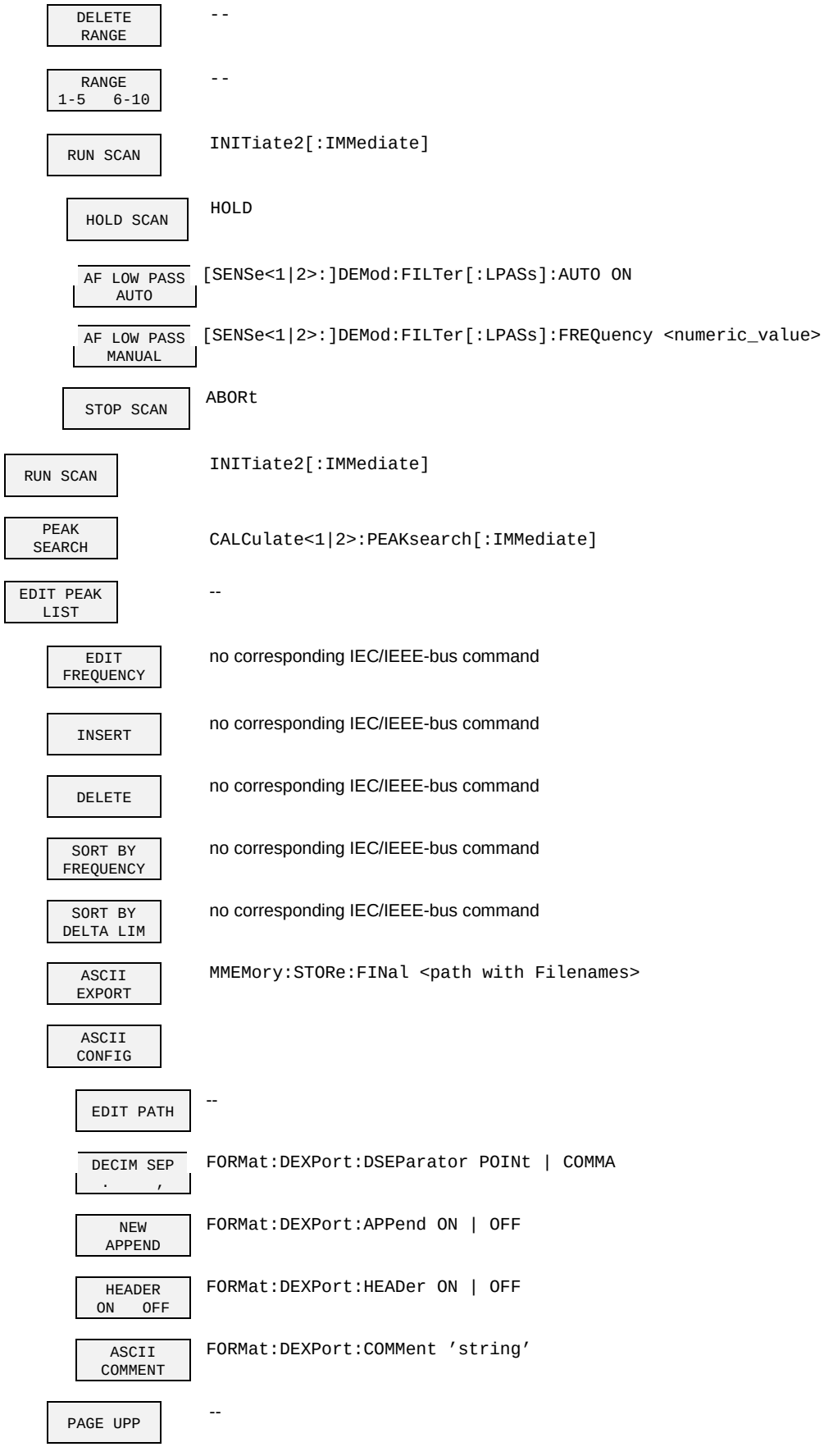
MACRO 1...7	no corresponding IEC/IEEE-bus command
DEFINE MACRO	no corresponding IEC/IEEE-bus command
RECORD ON/OFF	no corresponding IEC/IEEE-bus command
DEFINE PAUSE	no corresponding IEC/IEEE-bus command
DELETE MACRO	no corresponding IEC/IEEE-bus command
MACRO TITLE	no corresponding IEC/IEEE-bus command
SELECT MACRO	no corresponding IEC/IEEE-bus command

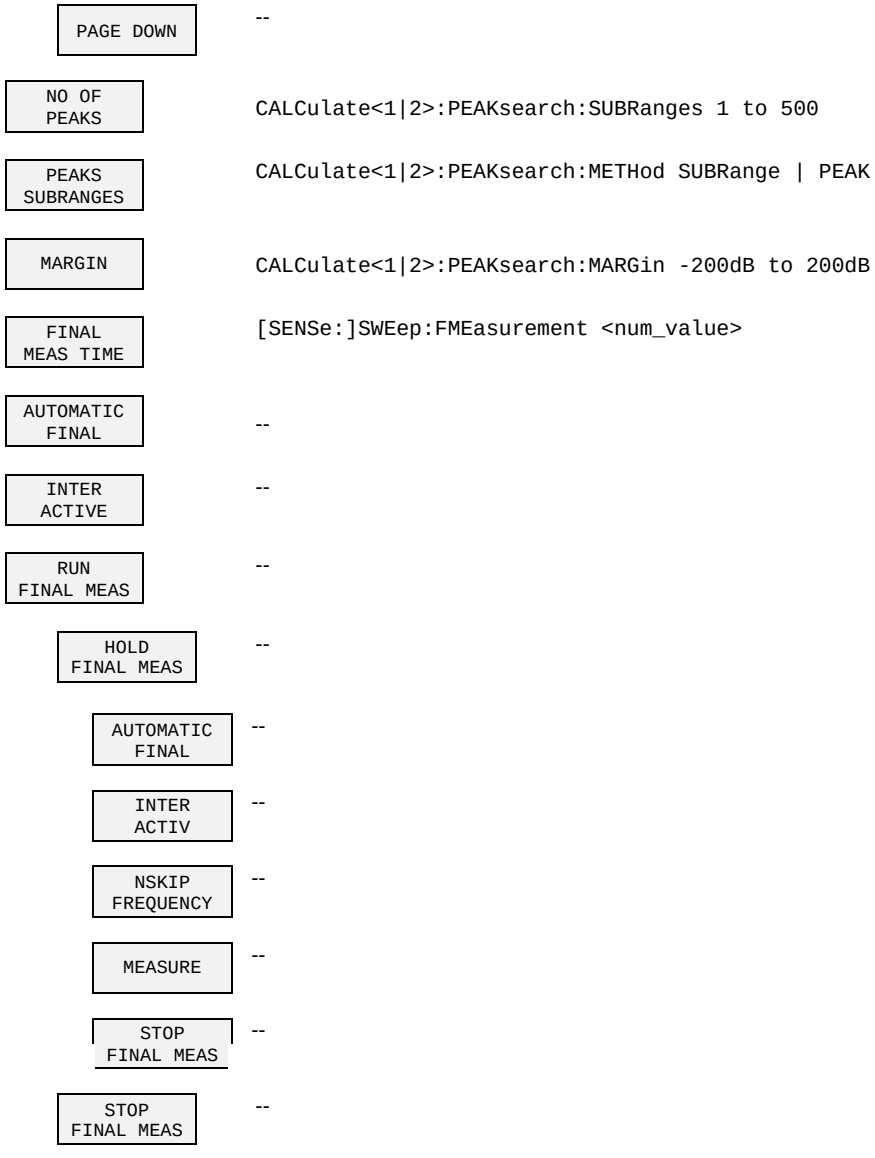
Receiver Mode

CONFIGURATION Key Group

MODE	
RECEIVER	INSTrument<1|2>:[:SELect] RECeiver
RECEIVER FREQUENCY	[SENSe<1|2>:]FREQuency[:CW|FIXed] <numeric_value>
ATTEN	--
ATTEN MANUAL	INPut<1|2>:ATTenuation <numeric_value>
0 DB MIN ON OFF	INPut<1|2>:ATTenuation:PROTection ON|OFF
AUTO RANGE ON OFF	INPut<1|2>:ATTenuation:AUTO ON|OFF
AUTOPREAMP ON OFF	INPut<1|2>:GAIN:AUTO ON|OFF
PREAMP GAIN	INPut<1|2>:GAIN <numeric_value>
PREAMP ON OFF	INPut<1|2>:GAIN:STATe ON|OFF
RES BW	[SENSe<1|2>:]BANDwidth:RESolution <numeric_value>
DETECTOR	--
MAX PEAK	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] POSitive
MIN PEAK	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] NEGative
AVERAGE	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] AVERage
RMS	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] RMS
AC VIDEO	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] ACVideo
MEAS TIME	[SENSe<1|2>:]SWEep:TIME <numeric_value>
AF DEMOD	--
AM	[SENSe<1|2>:]DEMod AM
AM VIDEO	[SENSe<1|2>:]DEMod AMVideo

FM	[SENSe<1|2>:]DEMod FM
PM	[SENSe<1|2>:]DEMod PM
AF FILTER	[SENSe<1|2>:]DEMod AM
AF LOW PASS AUTO	[SENSe<1|2>:]DEMod:FILTer[:LPASs]:AUTO ON
AF LOW PASS MANUAL	[SENSe<1|2>:]DEMod:FILTer[:LPASs]:FREQuency <numeric_value>
AF DC	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 0
AF HI PASS 10 Hz	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 10
AF HI PASS 100 Hz	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 100
AF HI PASS 1 kHz	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 1000
DEMOD ON OFF	[SENSe<1|2>:]DEMod OFF (automatically executing during AM|FM)
DEFINE SCAN	--
SCAN TABLE	[SENSe<1|2>:]FREQuency:STARt <numeric_value> [SENSe<1|2>:]FREQuency:STOP <numeric_value> [SENSe<1|2>:]SWEep:SPACing LINear | LOGarithmic | AUTO DISPlay[:WINDow<1|2>]:TRACe:Y[:SCALe]:TOP <numeric_value> DISPlay[:WINDow<1|2>]:TRACe:Y[:SCALe]:BOTTom <numeric_value> DISPlay[:WINDow<1|2>]:TRACe:X:SPACing LINear | LOGarithmic
ADJUST AXIS	--
SINGLE SCAN	INITiate2:CONTinuous OFF; INITiate[:IMMediate]
CONTINUOUS SCAN	INITiate2:CONTinuous ON; INITiate[:IMMediate]
SCAN RANGES	[SENSe<1|2>:]SCAN<1...10>:RANGes[:COUNt] 1 ... 10 [SENSe<1|2>:]SCAN<1...10>:STARt <numeric_value> [SENSe<1|2>:]SCAN<1...10>:STOP <numeric_value> [SENSe<1|2>:]SCAN<1...10>:STEP <numeric_value> [SENSe<1|2>:]SCAN<1...10>:BANDwidth:RESolution <numeric_value> [SENSe<1|2>:]SCAN<1...10>:TIME <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPUT:ATTenuation:AUTO <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPut:ATTenuation <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPUT:GAIN <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPUT:GAIN:STATe ON | OFF [SENSe<1|2>:]SCAN<1...10>:INPUT:GAIN:AUTO ON | OFF [SENSe<1|2>:]SCAN<1...10>:INPut:COUPling DC | AC [SENSe<1|2>:]SCAN<1...10>:INPut:TYPE RF | BALanced
INS BEFORE RANGE	--
INS AFTER RANGE	--


DELETE RANGE
--
RANGE 1-5 6-10
--
RUN SCAN
INITiate2[:IMMediate]
HOLD SCAN
HOLD
AF LOW PASS AUTO
[SENSe<1|2>:]DEMod:FILTer[:LPASs]:AUTO ON
AF LOW PASS MANUAL
[SENSe<1|2>:]DEMod:FILTer[:LPASs]:FREQuency <numeric_value>
STOP SCAN
ABORt
RUN SCAN
INITiate2[:IMMediate]
PEAK SEARCH
CALCulate<1|2>:PEAKsearch[:IMMediate]
EDIT PEAK LIST
--
EDIT FREQUENCY
no corresponding IEC/IEEE-bus command
INSERT
no corresponding IEC/IEEE-bus command
DELETE
no corresponding IEC/IEEE-bus command
SORT BY FREQUENCY
no corresponding IEC/IEEE-bus command
SORT BY DELTA LIM
no corresponding IEC/IEEE-bus command
ASCII EXPORT
MMEMory:STORe:FINal <path with Filenames>
ASCII CONFIG
EDIT PATH
--
DECIM SEP . ,
FORMat:DEXPort:DSEParator POINt | COMMA
NEW APPEND
FORMat:DEXPort:APPend ON | OFF
HEADER ON OFF
FORMat:DEXPort:HEADer ON | OFF
ASCII COMMENT
FORMat:DEXPort:COMMent 'string'
PAGE UPP
--


PAGE DOWN
--
NO OF PEAKS
CALCulate<1|2>:PEAKsearch:SUBRanges 1 to 500
PEAKS SUBRANGES
CALCulate<1|2>:PEAKsearch:METHod SUBRange | PEAK
MARGIN
CALCulate<1|2>:PEAKsearch:MARGin -200dB to 200dB
FINAL MEAS TIME
[SENSe:]SWEep:FMEasurement <num_value>
AUTOMATIC FINAL
--
INTER ACTIVE
--
RUN FINAL MEAS
--
HOLD FINAL MEAS
--
AUTOMATIC FINAL
--
INTER ACTIV
--
NSKIP FREQUENCY
--
MEASURE
--
STOP FINAL MEAS
--
STOP FINAL MEAS
--

FREQUENCY Key Group

START

```
[SENSe<1|2>:]FREQuency:STARt <numeric_value>
```

STOP

```
[SENSe<1|2>:]FREQuency:STOP <numeric_value>
```

FREQ/ CENTER

```
[SENSe<1|2>:]FREQuency:CENTer <numeric_value>
```

STEP

STEPSIZE MANUAL

```
[SENSe<1|2>:]FREQuency:CENTer:STEP <numeric_value>
```

STEPSIZE =FREQUENCY

no corresponding IEC/IEEE-bus command

ZOOM/ SPAN

```
--
```

LEVEL Key Group

UNIT/ REF

dBµV

```
CALCulate<1|2>:UNIT:POWer  DBMV
```

dBm

```
CALCulate<1|2>:UNIT:POWer DBM
```

dBµA

```
CALCulate<1|2>:UNIT:POWer DBUA
```

dBpW

```
CALCulate<1|2>:UNIT:POWer DBPW
```

dBpT

```
CALCulate<1|2>:UNIT:POWer DBPT
```

dBµV/m

```
CALCulate<1|2>:UNIT:POWer DBUV_M
```

dBµA/m

```
CALCulate<1|2>:UNIT:POWer DBUA_M
```

PROBE CODE ON / OFF

```
UNIT:PROBe  ON|OFF
```

RANGE

LOG 120 dB	DISPlay[:WINDow]:TRACe:Y[:SCALe] 120
LOG 100 dB	DISPlay[:WINDow]:TRACe:Y[:SCALe] 100
LOG 50 dB	DISPlay[:WINDow]:TRACe:Y[:SCALe] 50
LOG 20 dB	DISPlay[:WINDow]:TRACe:Y[:SCALe] 20
LOG 10 dB	DISPlay[:WINDow]:TRACe:Y[:SCALe] 10
LOG MANUAL	DISPlay[:WINDow]:TRACe:Y[:SCALe] <numeric_value>
GRID MAX LEVEL	DISPlay[:WINDow]:TRACe:Y[:SCALe]:TOP <numeric_value>
GRID MIN LEVEL	DISPlay[:WINDow]:TRACe:Y[:SCALe]:BOTTom <numeric_value>

INPUT Key

INPUT	
RF ATTEN MANUAL	INPut<1|2>:ATTenuation <numeric_value>
0 DB MIN ON OFF	INPut<1|2>:ATTenuation:PROTection ON | OFF
AUTO RANGE ON OFF	INPut<1|2>:ATTenuation:AUTO ON | OFF
AUTOPREAMP ON OFF	INPut<1|2>:GAIN:AUTO ON | OFF
PREAMP ON OFF	INPut<1|2>: GAIN:STATe ON | OFF
PREAMP GAIN	INPut<1|2>:GAIN <numeric_value>
INPUT SELECT	
RF INPUT 50 OHM	INPut<1|2>:IMPedance 50
RF INPUT DC COUPLED	INPut<1|2>:COUPling DC INPut<1|2>:TYPE RF
RF INPUT AC COUPLED	INPut<1|2>:COUPling AC INPut<1|2>:TYPE RF
BAL INPUT 150 OHM	INPut<1|2>:BIMPedance 150 INPut<1|2>:TYPE BALanced
BAL INPUT 600 OHM	INPut<1|2>:BIMPedance 600 INPut<1|2>:TYPE BALanced
BAL INPUT 10 KOHM	INPut<1|2>:BIMPedance 10000 INPut<1|2>:TYPE BALanced

MARKER Key Group

NORMAL

MARKER 1..4	CALCulate<1|2>:MARKer<1...4>[:STATe] ON | OFF; CALCulate<1|2>:MARKer<1...4>:X <numeric value>; CALCulate<1|2>:MARKer<1...4>:Y?
MARKER ZOOM	CALCulate<1|2>:MARKer<1...4>:FUNCtion:ZOOM <num_value>
PREV ZOOM RANGE	no corresponding IEC/IEEE-bus command
ZOOM OFF	DISPlay[:WINDow<1|2>]:TRACe<1...4>:X[:SCALe]:ZOOM OFF
MARKER INFO	DISPlay:WINDow<1|2>:MINFo ON | OFF (indication)
ALL MARKER OFF	CALCulate<1|2>:MARKer<1...4>:AOFF

STEP

STEPSIZE AUTO	CALCulate<1|2>:MARKer<1...4>:STEP:AUTO ON | OFF
STEPSIZE MANUAL	CALCulate<1|2>:MARKer<1...4>:STEP[:INCRement] <numeric_value>
MKR TO STEPSIZE	CALCulate<1|2>:MARKer<1...4>:FUNCtion:MSTep
DELTA TO STEPSIZE	--

DELTA

DELTA 1...4	CALCulate<1|2>:DELTamarker<1...4>[:STATe] ON | OFF CALCulate<1|2>:DELTamarker<1...4>:X <numeric value> CALCulate<1|2>:DELTamarker<1...4>:X:RELative? CALCulate<1|2>:DELTamarker<1...4>:Y?
REFERENCE POINT	--
REF POINT LEVEL	CALCulate<1|2>:DELTamarker<1..4>:FUNCtion:FIXed:RPOint:Y <num_value>
REF POINT LVL OFFSET	CALCulate<1|2>:DELTamarker<1..4>:FUNCtion:FIXed:RPOint:Y:OFFSet <num_value>
REF POINT FREQUENCY	CALCulate<1|2>:DELTamarker<1..4>:FUNCtion:FIXed:RPOint:X <num_value>
REFERENCE FIXED	CALCulate<1|2>:DELTamarker<1...4>:FUNCtion:FIXed[:STATe] ON | OFF
DELTA MKR ABS REL	CALCulate<1|2>:DELTamarker<1...4>:MODE ABSolute | RELative

ALL DELTA OFF	CALCulate<1|2>:DELTamarker<1...4>:AOFF
STEP	
STEPSIZE AUTO	CALCulate<1|2>:DELTamarker<1...4>:STEP:AUTO ON | OFF
MANUAL STEPSIZE	CALCulate<1|2>:DELTamarker<1...4>:STEP[:INCRement] <num_value>
DELTA TO STEPSIZE	--
SEARCH	
PEAK	CALCulate<1|2>:MARKer<1...4>:MAXimum[:PEAK] CALCulate<1|2>:DELTamarker<1...4>:MAXimum[:PEAK]
NEXT PEAK	CALCulate<1|2>:MARKer<1...4>:MAXimum:NEXT CALCulate<1|2>:DELTamarker<1...4>:MAXimum:NEXT
NEXT PEAK RIGHT	CALCulate<1|2>:MARKer<1...4>:MAXimum:RIGHt CALCulate<1|2>:DELTamarker<1...4>:MAXimum:RIGHt
NEXT PEAK LEFT	CALCulate<1|2>:MARKer<1...4>:MAXimum:LEFT CALCulate<1|2>:DELTamarker<1...4>:MAXimum:LEFT
TUNE TO MARKER	CALCulate<1|2>:MARKer<1...4>:FUNCtion:CENTer
MARKER TRACK	CALCulate<1|2>:MARKer<1...4>:COUPled:STATe
SETTINGS COUPLED	CALCulate<1|2>:MARKer<1...4>:SCOupled:STATe
SEARCH LIM ON/OFF	CALCulate<1|2>:MARKer<1...4>:X:SLIMits[STATe] ON | OFF
SELECT MARKER	no corresponding IEC/IEEE-bus command
ACTIVE MKR/DELTA	no corresponding IEC/IEEE-bus command
MIN	CALCulate<1|2>:MARKer<1...4>:MINimum[:PEAK] CALCulate<1|2>:DELTamarker<1...4>:MINimum[:PEAK]
NEXT MIN	CALCulate<1|2>:MARKer<1...4>:MINimum:NEXT CALCulate<1|2>:DELTamarker<1...4>:MINimum:NEXT
NEXT MIN RIGHT	CALCulate<1|2>:MARKer<1...4>:MINimum:RIGHt CALCulate<1|2>:DELTamarker<1...4>:MINimum:RIGHt
NEXT MIN LEFT	CALCulate<1|2>:MARKer<1...4>:MINimum:LEFT CALCulate<1|2>:DELTamarker<1...4>:MINimum:LEFT
PEAK EXCURSION	CALCulate<1|2>:MARKer<1...4>:PEXCursion <numeric_value>




```
CALCulate<1|2>:MARKer<1...4>:MAXimum[:PEAK]
CALCulate<1|2>:DELTamarker<1...4>:MAXimum[:PEAK]

CALCulate<1|2>:MARKer<1...4>:FUNCtion:CENTer
```

no corresponding IEC/IEEE-bus command

```
CALCulate<1|2>:MARKer<1...4>:COUPled:STATe

CALCulate<1|2>:MARKer<1...4>: SCOUPled:STATe

CALCulate<1|2>:MARKer<1...4>:FUNCtion:CSTep

CALCulate<1|2>:MARKer<1...4>:TRACe <numeric value>
CALCulate<1|2>:DELTamarker<1...4>:TRACe <numeric value>
```

no corresponding IEC/IEEE-bus command

no corresponding IEC/IEEE-bus command

LINES Key Group

D LINES

DISPLAY LINE 1/2	CALCulate<1|2>:DLINe<1|2>:STATe ON | OFF CALCulate<1|2>:DLINe<1|2> <numeric_value>
THRESHOLD LINE	CALCulate<1|2>:THReshold ON | OFF CALCulate<1|2>:THReshold <numeric_value>
REFERENCE LINE	CALCulate<1|2>:RLINe:STATe ON | OFF CALCulate<1|2>:RLINe <numeric_value>
FREQUENCY LINE 1/2	CALCulate<1|2>:FLINe<1|2>:STATe ON | OFF CALCulate<1|2>:FLINe<1|2> <numeric_value>

LIMITS

SELECT LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:NAME <string> CALCulate<1|2>:LIMit<1...8>:STATe ON | OFF
NEW LIMIT LINE	see EDIT LIMIT LINE
NAME	CALCulate<1|2>:LIMit<1...8>:NAME <string>
VALUES	no corresponding IEC/IEEE-bus command
INSERT VALUE	no corresponding IEC/IEEE-bus command
DELETE VALUE	no corresponding IEC/IEEE-bus command
SHIFT X LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:CONTrol:SHIFt <numeric_value>
SHIFT Y LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:UPPer:SHIFt <numeric_value> CALCulate<1|2>:LIMit<1...8>:LOWer:SHIFt <numeric_value>
SAVE LIMIT LINE	automatically executed during IEC/IEEE-bus operation
EDIT LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:UNIT DBM | DBPW | DBPT | DBUV| DBUA | DBUV_M | DBUA_M | HZ CALCulate<1|2>:LIMit<1...8>:TRACe <numeric_value> CALCulate<1|2>:LIMit<1...8>:COMMent 'string' CALCulate<1|2>:LIMit<1...8>:CONTrol[:DATA] <num_value>, <num_value>.. CALCulate<1|2>:LIMit<1...8>:CONTrol:OFFset <numeric_value> CALCulate<1|2>:LIMit<1...8>:CONTrol:SPACing LINear | LOGarithmic CALCulate<1|2>:LIMit<1...8>:UPPer[:DATA] <num_value>, <num_value>.. CALCulate<1|2>:LIMit<1...8>:UPPer:STATe ON | OFF CALCulate<1|2>:LIMit<1...8>:UPPer:OFFset <numeric_value> CALCulate<1|2>:LIMit<1...8>:UPPer:SPACing LINear | LOGarithmic CALCulate<1|2>:LIMit<1...8>:LOWer[:DATA] <num_value>,<num_value>.. CALCulate<1|2>:LIMit<1...8>:LOWer:STATe ON | OFF CALCulate<1|2>:LIMit<1...8>:LOWer:OFFset <numeric_value> CALCulate<1|2>:LIMit<1...8>:LOWer:SPACing LINear | LOGarithmic
COPY LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:COPY 1...8 | <name>

DELETE LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:DELete
X OFFSET	CALCulate<1|2>:LIMit<1...8>:DELete
Y OFFSET	CALCulate<1|2>:LIMit<1...8>:DELete

TRACE Key Group

TRACE 1

CLEAR/ WRITE	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE WRITe
VIEW	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE VIEW
BLANK	DISPlay[:WINDow<1|2>]:TRACe<1...4>[:STATe] OFF
MAX HOLD	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE MAXHold
MIN HOLD	DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE MINHold
SCAN COUNT	[SENSe<1|2>:]SWEep:COUNt <numeric_value>
DETECTOR	no corresponding IEC/IEEE-bus command
MIN PEAK	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] NEGative
MAX PEAK	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] POSitive
AVERAGE	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] AVERage
RMS	[SENSe<1|2>:]DETector:RECeiver[:FUNCtion] RMS
MIN PEAK	[SENSe:]DETector[:FUNCtion] NEGative
AC VIDEO	[SENSe:]DETector[:FUNCtion] ACVideo
FINAL MAX PEAK	[SENSe:]DETector:FMEasurement POSitive
FINAL AVERAGE	[SENSe:]DETector:FMEasurement AVERage
FINAL RMS	[SENSe:]DETector:FMEasurement RMS
FINAL MIN PEAK	[SENSe:]DETector:FMEasurement NEGative

FINAL AC VIDEO	[SENSe:]DETector:FMEasurement ACVideo
COPY	TRACe:COPY TRACE1| TRACE2| TRACE3| TRACE4 , TRACE1| TRACE2| TRACE3| TRACE4
T1-T2/T3/T4 +REF ->T1	CALCulate<1|2>:MATH<1...4>:STATe ON CALCulate<1|2>:MATH<1...4>[:EXPRession][:DEFine] <expr>
T1-REF ->T1	CALCulate<1|2>:MATH<1...4>:STATe ON CALCulate<1|2>:MATH<1...4>[:EXPRession][:DEFine] <expr>
ASCII EXPORT	MMEMory:STORe:TRACe 1..4,<path>
ASCII CONFIG	
EDIT PATH	no corresponding IEC/IEEE-bus command
DECIM SEP . ,	FORMat:DEXPort:DSEParator POINt|COMMA
NEW APPEND	FORMat:DEXPort:APPend ON | OFF
HEADER ON OFF	FORMat:DEXPort:HEADer ON | OFF
ASCII COMMENT	FORMat:DEXPort:COMMent 'string'
TRACE MATH OFF	CALCulate<1|2>:MATH<1...4>:STATe OFF

SWEEP Key Group

RUN/ COUPLING	INITiate[:IMMediate]
TRIGGER	
FREE RUN	TRIGger<1|2>:[:SEQuence]:SOURce IMMediate
EXTERN	TRIGger<1|2>:[:SEQuence]:SOURce EXTernal TRIGger<1|2>:[:SEQuence]:LEVel[:EXTernal] -5.0...+5.0V
SLOPE POS/NEG	TRIGger<1|2>:[:SEQuence]:SLOPe POSitive|NEGative
SCAN/ SWEEP	

SCAN TABLE	[SENSe<1|2>:]FREQuency:STARt <numeric_value> [SENSe<1|2>:]FREQuency:STOP <numeric_value> [SENSe<1|2>:]SWEep:SPACing LINear | LOGarithmic | AUTO DISPlay[:WINDow]:TRACe:Y[:SCALe]:TOP <numeric_value> DISPlay[:WINDow]:TRACe:Y[:SCALe]:BOTTom <numeric_value> DISPlay[:WINDow<1|2>]:TRACe:X:SPACing LINear | LOGarithmic
ADJUST AXIS	no corresponding IEC/IEEE-bus command
SINGLE SCAN	INITiate2:CONTinuous OFF; INITiate[:IMMediate]
CONTINUOUS SCAN	INITiate2:CONTinuous ON; INITiate[:IMMediate]
SCAN RANGES	[SENSe<1|2>:]SCAN<1...10>:RANGes[:COUNt] 1 ... 10 [SENSe<1|2>:]SCAN<1...10>:STARt <numeric_value> [SENSe<1|2>:]SCAN<1...10>:STOP <numeric_value> [SENSe<1|2>:]SCAN<1...10>:STEP <numeric_value> [SENSe<1|2>:]SCAN<1...10>:BANDwidth:RESolution <numeric_value> [SENSe<1|2>:]SCAN<1...10>:TIME <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPUT:ATTenuation:AUTO <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPut:ATTenuation <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPUT:GAIN <numeric_value> [SENSe<1|2>:]SCAN<1...10>:INPUT:GAIN:STATe ON | OFF [SENSe<1|2>:]SCAN<1...10>:INPUT:GAIN:AUTO ON | OFF [SENSe<1|2>:]SCAN<1...10>:INPut:COUPling DC | AC [SENSe<1|2>:]SCAN<1...10>:INPut:TYPE RF | BALanced
INS BEFORE RANGE	--
INS AFTER RANGE	--
DELETE RANGE	--
RANGE 1-5 6-10	--
RUN SCAN	INITiate2[:IMMediate]
HOLD SCAN	HOLD
CONT AT REC FREQ	INITiate2[:IMMediate]
CONT AT HOLD	no corresponding IEC/IEEE-bus command
STOP SCAN	no corresponding IEC/IEEE-bus command
STOP SCAN	ABORT

AF Demodulation Mode

CONFIGURATION Key Group

MODE	--
AF DEMOD	--
AM	[SENSe<1|2>:]DEMod AM
AM VIDEO	[SENSe<1|2>:]DEMod AMVideo
FM	[SENSe<1|2>:]DEMod FM
PM	[SENSe<1|2>:]DEMod PM
AF DISPLAY ON OFF	DISPlay[:WINDow<1|2>]:FEED 'AF' | 'VIDeo'
AF RANGE	--
FM RANGE AUTO	[SENSe<1|2>:]DEMod:FM[:DEViation]:RANGe:AUTO ON
FM RANGE MANUAL	[SENSe<1|2>:]DEMod:FM[:DEViation]:RANGe:UPPer <numeric_value>
PM RANGE MANUAL	[SENSe<1|2>:]DEMod:PM[:DEViation]:RANGe:UPPer <numeric_value>
AM RANGE 100%	[SENSe<1|2>:]DEMod:AM:RANGe:UPPer 100
AM RANGE 10%	[SENSe<1|2>:]DEMod:AM:RANGe:UPPer 10
AM RANGE 3%	[SENSe<1|2>:]DEMod:AM:RANGe:UPPer 3
AF FILTER	
AF LOW PASS AUTO	[SENSe<1|2>:]DEMod:FILTer[:LPASs]:AUTO ON
AF LOW PASS MANUAL	[SENSe<1|2>:]DEMod:FILTer[:LPASs]:FREQuency <numeric_value>
AF DC	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 0
AF HI PASS 10 Hz	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 10
AF HI PASS 100 Hz	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 100
AF HI PASS 1 kHz	[SENSe<1|2>:]DEMod:FILTer[:HPASs]:FREQuency 1000
AF BW COUPLED	[SENSe<1|2>:]BWIDth:DEMod:COUPling ON | OFF
AF BANDWIDTH	[SENSe<1|2>:]BWIDth:DEMod 100 kHz
DEMOD OFF	[SENSe<1|2>:]DEMod OFF

Vector Signal Analysis Mode

CONFIGURATION Key Group - Digital Demodulation

MODE --

VECTOR ANALYZER --

DIGITAL STANDARD
```
[SENSe<1|2>:]DDEMod:PRESet   PWT| F95Cdma| R95Cdma| GSM| EDGe | FNADc|
       RNADc| TETRa| DCS1800| PCS1900| PHS| PDCup | PDCDown |
       APCO25CQPSK | APCO25C4FM |CDPD| DECT| CT2| ERMes| MODacom |
       TFTS| F16| F322| F324| F64| FQCDMA| RQCDMA
```

DIGITAL DEMOD
```
INSTrument<1|2>:[:SELect]  DDEMod
[SENSe<1|2>:]DDEMod:FORMat   QPSK | PSK | MSK |QAM | FSK
[SENSe<1|2>:]DDEMod:QPSK:FORMat NORMal | DIFFerential | OFFSet | DPI4
[SENSe<1|2>:]DDEMod:PSK:NSTate   2 | 8
[SENSe<1|2>:]DDEMod:PSK:FORMat   NORMal | DIFFerential | N3Pi8
[SENSe<1|2>:]DDEMod:MSK:FORMat   TYPE1 | TYPE2 | NORMal | DIFFerential
[SENSe<1|2>:]DDEMod:QAM:NSTate   16
[SENSe<1|2>:]DDEMod:FSK:NSTate   2 | 4

CALCulate<1|2>:MARKer<1...2>:FUNCtion:DDEMod:RESult? MERM| MEPK| MEPS|
       PERM| PEPK| PEPS| EVRM| EVPK| EVPS| IQOF| IQIM| ADR| FERR| DEV|
       FSRM| FSPK| FSPS| RHO| FEPK
```

MODULATION PARAMETERS --

SYMBOL RATE
```
[SENSe<1|2>:]DDEMod:SRATe <num_value>
```

SIDE BAND NORM INV
```
[SENSe<1|2>:]DDEMod:SBANd NORMal|INVerse
```

MEAS FILTER
```
[SENSe<1|2>:]DDEMod:FILTer:MEASurement  OFF|RCOSine|RRCosine|GAUSsian
```

REFERENCE FILTER
```
[SENSe<1|2>:]DDEMod:FILTer:REFerence  RCOSine | RRCosine | GAUSsian
```

ALPHA/BT
```
[SENSe<1|2>:]DDEMod:FILTer:ALPHa <num_value>
```

FSK REF DEVIATION
```
CALCulate<1|2>:FSK:DEViation:REFerence <num_value>
```

NORMALIZE ON / OFF
```
[SENSe<1|2>:]DDEMod:NORMalize ON | OFF
```

MEAS RESULT --

MAGNITUDE CAP BUFFER
```
CALCulate<1|2>:FEED  'TCAP'
```

MEAS SIGNAL
```
CALCulate<1|2>:FEED  'XTIM:DDEM:MEAS'
```

MAGNITUDE
```
CALCulate<1|2>:FORMat MAGNitude
```

PHASE
```
CALCulate<1|2>:FORMat PHASe
```

FREQUENCY	CALCulate<1|2>:FORMat FREQuency
REAL/IMAG PART	CALCulate<1|2>:FORMat RIMag
EYE DIAG [FREQ]	CALCulate<1|2>:FORMat FEYE
EYE DIAG [I]	CALCulate<1|2>:FORMat IEYE
EYE DIAG [Q]	CALCulate<1|2>:FORMat QEYE
EYE DIAG TRELLIS	CALCulate<1|2>:FORMat TEYE
POLAR [IQ] VECTOR	CALCulate<1|2>:FORMat COMP
POLAR [IQ] CONSTELL	CALCulate<1|2>:FORMat CONS
SYMBOL DISPLAY	DISPLay[:WINDow<1|2>]:TRACe<1...4>:SYMBol DOTS | BARS |OFF
PHASE WRAP ON/OFF	CALCulate<1|2>:FORMat PHASe | UPHase
EYE LENGTH	DISPlay[:WINDow<1|2>]:TRACe<1...4>:EYE:COUNt <num_value>
REFERENCE SIGNAL	CALCulate<1|2>:FEED ‘XTIM:DDEM:REF’
MAGNITUDE	CALCulate<1|2>:FORMat MAGNitude
PHASE	CALCulate<1|2>:FORMat PHASe
FREQUENCY	CALCulate<1|2>:FORMat FREQuency
REAL/IMAG PART	CALCulate<1|2>:FORMat RIMag
EYE DIAG [FREQ]	CALCulate<1|2>:FORMat FEYE
EYE DIAG [I]	CALCulate<1|2>:FORMat IEYE
EYE DIAG [Q]	CALCulate<1|2>:FORMat QEYE
EYE DIAG TRELLIS	CALCulate<1|2>:FORMat TEYE
POLAR [IQ] VECTOR	CALCulate<1|2>:FORMat COMP
POLAR [IQ] CONSTELL	CALCulate<1|2>:FORMat CONS

SYMBOL DISPLAY	DISPLay[:WINDow<1|2>]:TRACe<1...4>:SYMBol DOTS | BARS |OFF
PHASE WRAP ON/OFF	CALCulate<1|2>:FORMat PHASe | UPHase
EYE LENGTH	DISPlay[:WINDow<1|2>]:TRACe<1...4>:EYE:COUNt <num_value>
ERROR SIGNAL	CALCulate<1|2>:FEED ‘XTIM:DDEM:ERR:MPH’
MAGNITUDE	CALCulate<1|2>:FORMat MAGNitude
PHASE	CALCulate<1|2>:FORMat PHASe
FREQUENCY	CALCulate<1|2>:FORMat FREQuency
REAL/IMAG PART	CALCulate<1|2>:FORMat RIMag
ERROR VECT MAGNITUDE	CALCulate<1|2>:FORMat MAGNitude CALCulate<1|2>:FEED ‘XTIM:DDEM:ERR:VECT’
POLAR [IQ] VECTOR	CALCulate<1|2>:FORMat COMP
POLAR [IQ] CONSTELL	CALCulate<1|2>:FORMat CONS
SYMBOL DISPLAY	DISPLay[:WINDow<1|2>]:TRACe<1...4>:SYMBol DOTS | BARS | OFF
SYMB TABLE / ERRORS	CALCulate<1|2>:FEED ‘XTIM:DDEM:SYMB’
MEMORY SIZE	[SENSe<1|2>:]TCAPture:LENGth 1024 | 2048 | 4096 | 8192 | 16384
FRAME LENGTH	[SENSe<1|2>:]DDEMod:SEARch:TIME <num_value>
RESULT LENGTH	[SENSe<1|2>:]DDEMod:TIME <num_value>
POINTS PER SYMBOL	[SENSe<1|2>:]DDEMod:PRATe 1| 2| 4| 8| 16
TRIGGER	See section "SWEEP - TRIGGER"
RANGE	See section "LEVEL - RANGE"
IF BANDWIDTH	--
IF BW AUTO	[SENSe<1|2>:]BANDwidth|BWIDth:RESolution:AUTO ON | OFF
IF BW MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth:RESolution <num_value>

FREQUENCY Key Group

CENTER	[SENSe<1|2>:]FREQuency:CENTer <num_value>
CENTER FREQUENCY	[SENSe<1|2>:]FREQuency:CENTer <num_value>
FREQUENCY OFFSET	[SENSe<1|2>:]FREQuency:OFFSet <num_value>

LEVEL Key Group

UNIT/ REF	--
REF LEVEL	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RLEVel <num_value>
REF LEVEL OFFSET	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RLEVel:OFFSet <num_value>
RF ATTEN MANUAL	INPut<1|2>:ATTenuation <num_value>
ATTEN AUTO NORMAL	INPut<1|2>:ATTenuation:AUTO:MODE NORMal INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW NOISE	INPut<1|2>:ATTenuation:AUTO:MODE LNOise INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW DIST	INPut<1|2>:ATTenuation:AUTO:MODE LDIStortion INPut<1|2>:ATTenuation:AUTO ON
MIXER LEVEL	INPut<1|2>:MIXer <numeric value>
RANGE	--
Y PER DIV	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:PDIVision <numeric value>
REF VALUE Y AXIS	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RVALue <num_value>
REF VALUE X AXIS	DISPlay[:WINDow<1|2>]:TRACe<1...4>:X[:SCALe]:RVALue <num_value>
REF VALUE POSITION	DISPlay[:WINDow<1|2>]:TRACe<1...4>:Y[:SCALe]:RPOSition 0...100PCT
TIME/DIV	[SENSE]:SWEep:TIME <num_value>
SCALE UNIT	--
Y UNIT LOG[dB]	CALCulate<1|2>:UNIT:POWer DB

Y UNIT LINEAR	CALCulate<1|2>:UNIT:POWer UNITLESS
Y UNIT DEG	CALCulate<1|2>:UNIT:ANGLe DEG
Y UNIT RAD	CALCulate<1|2>:UNIT:ANGLe RAD
Y UNIT DBM	CALCulate<1|2>:UNIT:POWer DBM
Y UNIT VOLT	CALCulate<1|2>:UNIT:POWer VOLT
Y UNIT WATT	CALCulate<1|2>:UNIT:POWer WATT
X UNIT TIME	CALCulate:X:<1|2>:UNIT:TIME S
X UNIT SYMBOL	CALCulate:X:<1|2>:UNIT:TIME SYMB

INPUT Key

INPUT

RF ATTEN MANUAL	INPut<1|2>:ATTenuation <num_value>
ATTEN AUTO NORMAL	INPut<1|2>:ATTenuation:AUTO:MODE NORMal; INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW NOISE	INPut<1|2>:ATTenuation:AUTO:MODE LNOise; INPut<1|2>:ATTenuation:AUTO ON
ATTEN AUTO LOW DIST	INPut<1|2>:ATTenuation:AUTO:MODE LDIStortion; INPut<1|2>:ATTenuation:AUTO ON
MIXER LEVEL	INPut<1|2>:MIXer <numeric value>
INPUT SELECT	
RF INPUT 50 OHM	INPut<1|2>:IMPedance 50
RF INPUT DC COUPLED	INPut<1|2>:COUPling DC INPut<1|2>:TYPE RF
RF INPUT AC COUPLED	INPut<1|2>:COUPling AC INPut<1|2>:TYPE RF
BAL INPUT 150 OHM	INPut<1|2>:BIMPedance 150 INPut<1|2>:TYPE BALanced
BAL INPUT 600 OHM	INPut<1|2>:BIMPedance 600 INPut<1|2>:TYPE BALanced
BAL INPUT 10 KOHM	INPut<1|2>:BIMPedance 10000 INPut<1|2>:TYPE BALanced

RF INPUT 50 OHM	INPut<1|2>:IMPedance 50
RF INPUT 75 OHM/RAM	INPut<1|2>:IMPedance:CORRection RAM
RF INPUT 75OHM/RAZ	INPut<1|2>:IMPedance:CORRection RAZ

MARKER Key Group

NORMAL	
MARKER 1 / 2	CALCulate<1|2>:MARKer<1|2>[:STATe] ON | OFF CALCulate<1|2>:MARKer<1|2>:X <numeric value> CALCulate<1|2>:MARKer<1|2>:Y?
POLAR MARKER R/I / MA/PH	CALCulate<1|2>:MARKer<1...2>:READout MPHase | RIMaginary
POLAR MARKER DEG/ RAD	CALCulate<1|2>:UNIT:ANGLe DEG | RAD
COUPLED MARKERS	CALCulate<1|2>:MARKer<1...2>:COUPled[:STATe] ON | OFF
MARKER INFO	DISPlay:WINDow<1|2>:MINFo ON | OFF (indication sur l'écran)
ALL MARKER OFF	CALCulate<1|2>:MARKer<1...2>:AOFF
DELTA	
DELTA 1/2	CALCulate<1|2>:DELTamarker<1...2>:AOFF
DELTA MKR ABS / REL	CALCulate<1|2>:DELTamarker<1...2>:MODE ABSolute | RELative
ALL DELTA OFF	CALCulate<1|2>:DELTamarker<1...2>[:STATe] ON | OFF CALCulate<1|2>:DELTamarker<1...2>:X <numeric value> CALCulate<1|2>:DELTamarker<1...2>:Y?
MARKER SEARCH	
PEAK	CALCulate<1|2>:MARKer<1...2>:MAXimum[:PEAK] CALCulate<1|2>:DELTamarker<1...2>:MAXimum[:PEAK]
MIN	CALCulate<1|2>:MARKer<1...2>:MINimum[:PEAK] CALCulate<1|2>:DELTamarker<1...2>:MINimum[:PEAK]
MAX |PEAK|	CALCulate<1|2>:MARKer<1...2>:MAXimum:APEak CALCulate<1|2>:DELTamarker<1...2>:MAXimum:APEak
SUMMARY ON OFF	CALCulate<1|2>:MARKer<1..2>:FUNCtion:SUMMary[:STATe] ON | OFF
SUMMARY MARKER	--

MAX |PEAK|	CALCulate<1|2>:MARKer<1..,2>:FUNCtion:SUMMary:MAXimum[:STATe] ON | OFF CALCulate<1|2>:MARKer<1...2>:FUNCtion:SUMMary:MAXimum:RESult? CALCulate<1|2>:MARKer<1...2>:FUNCtion:SUMMary:MAXimum:AVERage:RESult? CALCulate<1|2>:MARKer<1...2>:FUNCtion:SUMMary:MAXimum:PHOLd:RESult?
+PEAK	CALCulate<1|2>:MARKer<1...2>:FUNCtion:SUMMary:PPEak[:STATe] ON | OFF CALCulate<1|2>:MARKer<1...2>:FUNCtion:SUMMary:PPEak:RESult? CALCulate<1|2>:MARKer<1...2>:FUNCtion:SUMMary:PPEak:AVERage:RESult? CALCulate<1|2>:MARKer<1...2>:FUNCtion:SUMMary:PPEak:PHOLd:RESult?
-PEAK	CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MPEak[:STATe] ON | OFF CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MPEak:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MPEak:AVERage:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MPEak:PHOLd:RESult?
±PEAK/2	CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MIDDle[:STATe] ON | OFF CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MIDDle:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MIDDle:AVERage:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MIDDle:PHOLd:RESult?
RMS	CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:RMS[:STATe] ON | OFF CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:RMS:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:RMS:AVERage:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:RMS:PHOLd:RESult?
MEAN	CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MEAN[:STATe] ON | OFF CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MEAN:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MEAN:AVERage:RESult? CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:MEAN:PHOLd:RESult?
PEAK HOLD ON / OFF	CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:PHOLd ON | OFF
AVERAGE/HOLD ON / OFF	CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:AVERage ON | OFF
SWEEP COUNT	[SENSe<1|2>:]SWEep:COUNt <num_value>
ALL SUM MKR OFF	CALCulate<1|2>:MARKer<1|2>:FUNCtion:SUMMary:AOFF
SEARCH LIMIT ON/OFF	CALCulate<1|2>:MARKer<1|2>:X:SLIMits[:STATe] ON | OFF
ACTIVE MKR / DELTA	--

MKR ->

PEAK	CALCulate<1|2>:MARKer<1|2>:MAXimum[:PEAK] CALCulate<1|2>:DELTamarker<1...2>:MAXimum[:PEAK]
MKR → TRACE	CALCulate<1|2>:MARKer<1|4>:TRACe <numeric value>
SELECT MARKER	--
ACTIVE MKR / DELTA	no corresponding IEC/IEEE-bus command

LINES Key Group

D LINES	--
DISPLAY LINE 1/2	CALCulate<1|2>:DLINe<1|2>:STATe ON | OFF CALCulate<1|2>:DLINe<1|2> <num_value>
REFERENCE LINE	CALCulate<1|2>:RLINe:STATe ON | OFF CALCulate<1|2>:RLINe <num_value>
THRESHOLD LINE	CALCulate<1|2>:THReshold ON | OFF CALCulate<1|2>:THReshold <num_value>
TIME/SYMB 1/2	CALCulate<1|2>:TLINe<1|2>:STATe ON | OFF CALCulate<1|2>:TLINe<1|2> <num_value>
LIMITS	
SELECT LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:NAME <string> CALCulate<1|2>:LIMit<1...8>:STATe ON | OFF
NEW LIMIT LINE	see EDIT LIMIT LINE
NAME	CALCulate<1|2>:LIMit<1...8>:NAME <string>
VALUES	no corresponding IEC/IEEE-bus command
INSERT VALUE	no corresponding IEC/IEEE-bus command
DELETE VALUE	no corresponding IEC/IEEE-bus command
SHIFT X LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:CONTrol:SHIFt <num_value>
SHIFT Y LIMIT LINE	CALCulate<1|2>:LIMit<1...8>:UPPer:SHIFt <num_value> CALCulate<1|2>:LIMit<1...8>:LOWer:SHIFt <num_value>
SAVE LIMIT LINE	automatically executed during IEC/IEEE-bus operation

EDIT LIMIT LINE

```
CALCulate<1|2>:LIMit<1...8>:UNIT  DBM| DBPW| WATT| DBUV | DBMV | VOLT |
                                  DBUA | AMPere| DB|
                                  DBUV_MHZ| DBMV_MHZ | DBUA_MHZ|
                                  DEG| RAD| S| HZ| PCT
CALCulate<1|2>:LIMit<1...8>:TRACe <num_value>
CALCulate<1|2>:LIMit<1...8>:COMMent 'string'
CALCulate<1|2>:LIMit<1...8>:CONTrol[:DATA] <num_value>,<num_value>..
CALCulate<1|2>:LIMit<1...8>:CONTrol:DOMain   FREQuency|TIME
CALCulate<1|2>:LIMit<1...8>:CONTrol:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:CONTrol:MODE   RELative | ABSolute
CALCulate<1|2>:LIMit<1...8>:CONTrol:UNIT[:TIME] S | SYM
CALCulate<1|2>:LIMit<1...8>:CONTrol:SPACing LINear | LOGarithmic
CALCulate<1|2>:LIMit<1...8>:UPPer[:DATA] <num_value>,<num_value>..
CALCulate<1|2>:LIMit<1...8>:UPPer:STATe   ON | OFF
CALCulate<1|2>:LIMit<1...8>:UPPer:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:UPPer:MARGin <num_value>
CALCulate<1|2>:LIMit<1...8>:UPPer:MODE   RELative | ABSolute
CALCulate<1|2>:LIMit<1...8>:UPPer:SPACing LINear | LOGarithmic
CALCulate<1|2>:LIMit<1...8>:LOWer[:DATA] <num_value>,<num_value>..
CALCulate<1|2>:LIMit<1...8>:LOWer:STATe ON | OFF
CALCulate<1|2>:LIMit<1...8>:LOWer:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:LOWer:MARGin <num_value>
CALCulate<1|2>:LIMit<1...8>:LOWer:MODE   RELative | ABSolute
CALCulate<1|2>:LIMit<1...8>:LOWer:SPACing LINear | LOGarithmic
CALCulate<1|2>:LIMit<1...8>:FAIL?
CALCulate<1|2>:LIMit<1...8>:CLEar[:IMMediate]
```

COPY LIMIT LINE

```
CALCulate<1|2>:LIMit<1...8>:COPY 1...8 | <name>
```

DELETE LIMIT LINE

```
CALCulate<1|2>:LIMit<1...8>:DELete
```

X OFFSET

```
CALCulate<1|2>:LIMit<1...8>:CONTrol:OFFset <num_value>
```

Y OFFSET

```
CALCulate<1|2>:LIMit<1...8>:UPPer:OFFset <num_value>
CALCulate<1|2>:LIMit<1...8>:LOWer:OFFset <num_value>
```

TRACE Key Group

TRACE

```
--
```

CLEAR/WRITE

```
DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE WRITe
```

VIEW

```
DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE VIEW
```

BLANK

```
DISPlay[:WINDow<1|2>]:TRACe<1...4>[:STATe] OFF
```

CONTINUOUS WRITE

```
DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE:CWRite ON|OFF
```

AVERAGE

```
DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE  AVERage or
[SENSe<1|2>:]AVERage:MODE SCALe
```

MAX HOLD

```
DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE MAXHold or
[SENSe<1|2>:]AVERage:MODE MAX
```

MIN HOLD

```
DISPlay[:WINDow<1|2>]:TRACe<1...4>:MODE MINHold or
[SENSe<1|2>:]AVERage:MODE MIN
```

SWEEP COUNT

```
[SENSe<1|2>:]SWEep:COUNt <num_value>
```

SWEEP Key Group

RUN/ COUPLING	--
IF BW AUTO	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution]:AUTO ON | OFF
IF BW MANUAL	[SENSe<1|2>:]BANDwidth|BWIDth[:RESolution] <num_value>
SCAN/ SWEEP	
CONTINUOUS SWEEP	INITiate<1|2>::CONTinuous ON; INITiate[:IMMediate]
SINGLE SWEEP	INITiate<1|2>::CONTinuous OFF; INITiate[:IMMediate]
SWEEP COUNT	[SENSe<1|2>:]SWEep:COUNt <num_value>
RESULT LENGTH	[SENSe<1|2>:]DDEMod:TIME <num_value>
TRIGGER	
FREE RUN	TRIGger<1|2>:[:SEQuence]:SOURce IMMediate
VIDEO	TRIGger<1|2>:[:SEQuence]:SOURce VIDeo TRIGger<1|2>:[:SEQuence]:LEVel:VIDeo <numeric value>
EXTERN	TRIGger<1|2>:[:SEQuence]:SOURce EXTernal TRIGger<1|2>:[:SEQuence]:LEVel:EXTernal <numeric value>
TRIGGER OFFSET	TRIGger[:SEQuence]:HOLDoff <numeric value>
SLOPE POS NEG	TRIGger[:SEQuence]:SLOPe POSitive|NEGative
MEAS ONLY IF SYNC'D	:[SENSe<1|2>:]DDEMod:SEARch:SYNC:MONLy ON | OFF
FIND BURST ON OFF	[SENSe<1|2>:]DDEMod:SEARch:PULSe:STATe ON | OFF
FIND SYNC ON OFF	[SENSe<1|2>:]DDEMod:SEARch:SYNC:STATe ON | OFF
SYNC OFFSET	[SENSe<1|2>:]DDEMod:SEARch:SYNC:OFFSET <num_value>
SYNC PATTERN	--
SELECT PATTERN	[SENSe<1|2>:]DDEMod:SEARch:SYNC:SELect <pattern_name>

NEW SYNC PATTERN	[SENSe<1|2>:]DDEMod:SEARch:SYNC:PATTern <string>
NAME	[SENSe<1|2>:]DDEMod:SEARch:SYNC:NAME <pattern_name>
COMMENT	[SENSe<1|2>:]DDEMod:SEARch:SYNC:COMMent <string>
VALUE	[SENSe<1|2>:]DDEMod:SEARch:SYNC:DATA <string> [SENSe<1|2>:]DDEMod:SEARch:SYNC:PATTern <string>
SAVE PATTERN	automatically executed during IEC/IEEE-bus operation
EDIT SYNC PATTERN	[SENSe<1|2>:]DDEMod:SEARch:SYNC:PATTern <string>
DELETE PATTERN	--

Contents - Chapter 7 "Remote Control - Programming Examples"

7 Programming Examples 7.1

Programming via IEC/IEEE Bus 7.1

Including IEC-Bus Library for QuickBasic 7.1

Initialization and Default Status 7.1

Initiate Controller 7.1

Initiate Instrument 7.1

Transmission of Instrument Setting Commands 7.2

Switchover to Manual Control 7.2

Reading out Instrument Settings 7.2

Positioning a Marker and Displaying Values 7.2

Command synchronization 7.3

Service Request 7.4

Programming via the RSIB Interface 7.6

Visual Basic 7.6

Visual Basic for Applications (Winword and Excel) 7.9

C / C++ 7.10

7 Programming Examples

The examples explain the programming of the instrument and can serve as a basis to solve more complex programming tasks.

QuickBASIC has been used as programming language. However, the programs can be translated into other languages.

Programming via IEC/IEEE Bus

Including IEC-Bus Library for QuickBasic

```
REM ----------- Include IEC-bus library for quickbasic ----------------
'$INCLUDE: 'c:\qbasic\qbdecl4.bas'
```

Initialization and Default Status

The IEC bus as well as the settings of the instrument are brought into a defined default status at the beginning of every program. Subroutines "InitController" and "InitDevice" are used to this effect.

Initiate Controller

```
REM ------------ Initiate controller -----------
REM InitController
iecaddress% = 20                            'IEC-bus address of the
                                            'instrument
CALL IBFIND("DEV1", analyzer%)              'Open port to the instrument
CALL IBPAD(analyzer%, iecaddress%)          'Inform controller on instrument
                                            'address
CALL IBTMO(analyzer%, 11)                   'Response time to 1 sec
REM ***************************************************************
```

Initiate Instrument

The IEC-bus status registers and instrument settings of the instrument are brought into the default status.

```
REM ------------ Initiate instrument --------------
REM InitDevice
CALL IBWRT(analyzer%, "*CLS")           'Reset status register
CALL IBWRT(analyzer%, "*RST")           'Reset instrument
REM***************************************************************
```

Transmission of Instrument Setting Commands

Center frequency, span, and reference level of the instrument are set in this example.

```
REM -------- Instrument setting commands -------------
CALL IBWRT(analyzer%, "FREQUENCY:CENTER 120MHz")   'Center frequency 120 MHz
CALL IBWRT(analyzer%, "FREQUENCY:SPAN 10MHZ")      'Span 10 MHz
CALL IBWRT(analyzer%, "DISPLAY:TRACE:Y:RLEVEL -10dBm")
                                                   'Reference level -10dBm
REM ************************************************************************
```

Switchover to Manual Control

```
REM -------- Switch instrument over to manual control -----------
CALL IBLOC(analyzer%)                          'Set instrument to Local state
REM ************************************************************************
```

Reading out Instrument Settings

The settings made in example 3 are read out here. The abbreviated commands are used.

```
REM --------- Reading out instrument settings ---------
CFfrequency$ = SPACE$(20)             'Provide text variables (20 characters)
CALL IBWRT(analyzer%, "FREQ:CENT?")  'Request center frequency
CALL IBRD(analyzer%, CFfrequency$)   'Read value

CFspan$ = SPACE$(20)                  'Provide text variables (20 characters)
CALL IBWRT(analyzer%, "FREQ:SPAN?")  'Request span
CALL IBRD(analyzer%, CFspan$)        'Read value

RLevel$ = SPACE$(20)                  'Provide text variables (20 characters)
CALL IBWRT(analyzer%, "DISP:TRAC:Y:RLEV?")
                                      'Request reference level
CALL IBRD(analyzer%, RLevel$)        'Read value

REM -------- Display values on the screen -----------
PRINT "Center frequency: "; CFfrequency$,
PRINT "Span:             "; CFspan$,
PRINT "Reference level:  "; RLevel$,
REM************************************************************************
```

Positioning a Marker and Displaying Values

```
REM -------- Examples of marker functions -----------
CALL IBWRT(analyzer%, "CALC:MARKER ON;MARKER:MAX")
                                  'Activate marker1 and start peak search
MKmark$ = SPACE$(30)              'Provide text variables (30 characters)
CALL IBWRT(analyzer%, "CALC:MARK:X?;Y?")    'Request frequency and level
CALL IBRD(analyzer%, MKmark$)               'Read value

REM -------- Display values on the screen -----------
PRINT "Center frequency / level "; MKmark$,
REM ************************************************************************
```

Command synchronization

The possibilities for synchronization implemented in the following example are described in Chapter 5, Section "Command Order and Command Synchronization".

```
REM -------- Examples of command synchronization ---------
REM The command INITiate[:IMMediate] starts a single sweep if the command
REM INIT:CONT OFF was previously sent. It should be ensured that the next
REM command is only then executed when the entire sweep is complete.

CALL IBWRT(analyzer%, "INIT:CONT OFF")

REM -------- First possibility: Use of *WAI ---------
CALL IBWRT(analyzer%, "ABOR;INIT:IMM; *WAI")

REM -------- Second possibility: Use of *OPC? ---------
OpcOk$ = SPACE$(2)                  'Space for *OPC? - Provide response
CALL IBWRT(analyzer%, "ABOR;INIT:IMM; *OPC?")
REM -------- here the controller can service other instruments ---------
CALL IBRD(analyzer%, OpcOk$)        'Wait for "1" from *OPC?

REM -------- Third possibility: Use of *OPC ---------
REM In order to be able touse the service request function in conjugation
REM with a National Instruments GPIB driver, the setting "Disable
REM Auto Serial Poll" must be changed to "yes" by means of IBCONF!

CALL IBWRT(analyzer%, "*SRE 32")     'Permit service request for ESR
CALL IBWRT(analyzer%, "*ESE 1")      'Set event-enable bit for
                                     'operation-complete bit
ON PEN GOSUB OpcReady                'Initialization of the
                                     'service request routine
PEN ON
CALL IBWRT(analyzer%, "ABOR;INIT:IMM; *OPC")

REM Continue main program here
STOP                                 'End of program

OpcReady:
REM As soon as the sweep has ended, this subroutine is activated
REM Program suitable reaction to the OPC service request.
ON PEN GOSUB OpcReady                'Enable service request routine again
RETURN
REM ************************************************************************
```

Service Request

The service request routine requires an extended initialization of the instrument in which the respective bits of the transition and enable registers are set.
In order to use the service request function in conjugation with National Instruments GPIB driver, the setting "Disable Auto Serial Poll" must be changed to "yes" by means of IBCONF!

```
REM ---- Example of initialization of the SRQ in the case of errors  ------
CALL IBWRT(analyzer%, "*CLS")                    'Reset Status Reporting System
CALL IBWRT(analyzer%,"*SRE 168")                 'Permit service request for
                                                 'STAT:OPER,STAT:QUES and ESR
                                                 'register
CALL IBWRT(analyzer%,"*ESE 60")                  'Set event-enable bit for
                                                 'command, execution, device-
                                                 'dependent and query error
CALL IBWRT(analyzer%,"STAT:OPER:ENAB 32767")     'Set OPERation enable bit for
                                                 'all events
CALL IBWRT(analyzer%,"STAT:OPER:PTR 32767")      'Set appropriate OPERation
                                                 'Ptransition bits
CALL IBWRT(analyzer%,"STAT:QUES:ENAB 32767")     'Set questionable enable bits
                                                 'for all events
CALL IBWRT(analyzer%,"STAT:QUES:PTR 32767")      'Set appropriate questionable
                                                 'Ptransition bits
ON PEN GOSUB Srq                                 'Initialization of the service
                                                 'request routine
PEN ON
REM Continue main program here
STOP
```

A service request is then processed in the service request routine.
Note: the variables userN% and userM% must be pre-assigned usefully!

```
Srq:
REM ------------ Service request routine -----------
DO
   SRQFOUND% = 0
   FOR I% = userN% TO userM%                     'Poll all bus users
        ON ERROR GOTO nouser                     'No user existing
        CALL IBRSP(I%, STB%)                     'Serial poll, read status byte
        IF STB% > 0 THEN                         'This instrument has bits set
                                                 'in the STB
          SRQFOUND% = 1
          IF (STB% AND 16)  > 0 THEN GOSUB Outputqueue
          IF (STB% AND 4)   > 0 THEN GOSUB Failure
          IF (STB% AND 8)   > 0 THEN GOSUB Questionablestatus
          IF (STB% AND 128) > 0 THEN GOSUB Operationstatus
          IF (STB% AND 32)  > 0 THEN GOSUB Esrread
        END IF
nouser:
   NEXT I%
LOOP UNTIL SRQFOUND% = 0
ON ERROR GOTO error handling
ON PEN GOSUB Srq: RETURN                         'Enable SRQ routine again
                                                 'End of SRQ routine
```

Reading out the status event registers, the output buffer and the error/event queue is effected in subroutines.

```
REM -------- Subroutines for the individual STB bits ------
Outputqueue:                                  'Reading the output buffer
Message$ = SPACE$(100)                        'Make space for response
CALL IBRD(analyzer%, Message$)
PRINT "Message in output buffer :"; Message$
RETURN

Failure:                                      'Read error queue
ERROR$ = SPACE$(100)                          'Make space for error variable
CALL IBWRT(analyzer%, "SYSTEM:ERROR?")
CALL IBRD(analyzer%, ERROR$)
PRINT " Error text :"; ERROR$
RETURN


Questionablestatus:                       'Read questionable status register
Ques$ = SPACE$(20)                        'Preallocate blanks to text variable
CALL IBWRT(analyzer%, "STATus:QUEStionable:EVENt?")
CALL IBRD(analyzer%, Ques$)
PRINT "Questionable Status:"; Ques$
RETURN

Operationstatus:                          'Read operation status register
Oper$ = SPACE$(20)                        'Preallocate blanks to text variable
CALL IBWRT(analyzer%, "STATus:OPERation:EVENt?")
CALL IBRD(analyzer%, Oper$)
PRINT "Operation Status:"; Oper$
RETURN


Esrread:                                  'Read event status register
Esr$ = SPACE$(20)                         'Preallocate blanks to text variable
CALL IBWRT(analyzer%, "*ESR?")            'Read ESR
CALL IBRD(analyzer%, Esr$)
IF (VAL(Esr$) AND 1) > 0 THEN PRINT "Operation complete"
IF (VAL(Esr$) AND 4) > 0 THEN GOTO Failure
IF (VAL(Esr$) AND 8) > 0 THEN PRINT "Device dependent error"
IF (VAL(Esr$) AND 16) > 0 THEN GOTO Failure
IF (VAL(Esr$) AND 32) > 0 THEN GOTO Failure
IF (VAL(Esr$) AND 64) > 0 THEN PRINT "User request"
IF (VAL(Esr$) AND 128) > 0 THEN PRINT "Power on"
RETURN
REM ************************************************************************

REM ------------- Error routine ----------------
Error handling:
PRINT "ERROR                                  'Output error message
STOP                                          'Stop software
```

Programming via the RSIB Interface

The following hints apply to both the 16-bit and the 32-bit DLL versions (RSIB.DLL and RSIB32.DLL) unless an explicit distinction is made.

The RSIB interface supports links to max. 16 measuring instruments at the same time.

Visual Basic

Programming Hints:

- Access to RSIB.DLL functions

  To generate Visual Basic control applications, the file RSIB.BAS for 16 bit basic programs or RSIB32.BAS for 32 bit basic programs (C:/R_S/INSTR/RSIB)is added to a project to enable call-up of the RSIB.DLL or RSIB32.DLL functions.

- Generating a response buffer

  Prior to calling the functions `RSDLLibrd()` and `RSDLLilrd()`, a string of sufficient length must be generated. This is possible either by defining the string or using the command `Space$()`.
  Generating a string of the length 100:
  - `Dim Response as String * 100`
  - `Dim Response as String`
    `Response = Space$(100)`

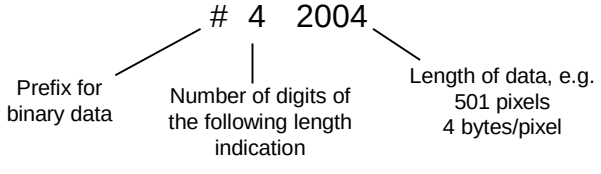
  If a response is to be output as a string from the measuring instrument, the appended blanks can be removed using the Visual Basic Function RTrim().

  Example:
  ```
  Response = Space$(100)
  Call RSDLLibrd(ud, Response, ibsta, iberr, ibcntl)
  Response = RTrim(Response)
  ' Output of Response
  ```

- Reading trace data in real format

  This mechanism is operational only for 16-bit Basic versions. With 32-bit Basic programs the header and useful data must be read by means of two separate function calls.
  Using the function declarations in the file RSIB.BAS or RSIB32.BAS the responses of the device can be assigned to one string only. If the data are to be read into an array with float values, the header and the useful data must be read out with separate function calls.

  Example of a header

  In order to enable the trace data to be directly read into a float array, a special function declaration must be created.

  ```
  Declare Function RSDLLilrdTraceReal Lib "rsib32.dll" Alias "RSDLLilrd"
  (ByVal ud%, Rd As Single, ByVal Cnt&, ibsta%, iberr%, ibcntl&) As Integer
  ```

Example:

```
Dim ibsta As Integer            ' Status variable
Dim iberr As Integer            ' Error variable
Dim ibcntl As Long              ' Count variable
Dim ud As Integer               ' Handle for measuring instrument
Dim Result As String            ' Buffer for simple results
Dim Digits As Byte              ' Number of digits of length indication
Dim TraceBytes As Long          ' Length of trace data in bytes
Dim TraceData(501) As Single    ' Buffer for floating point
                                ' Binary data

' Set up connection to instrument
ud = RSDLLibfind("89.10.38.97", ibsta, iberr, ibcntl)

' Query trace data in real format
Call RSDLLibwrt(ud, "FORM:DATA REAL,32", ibsta, iberr, ibcntl)
Call RSDLLibwrt(ud, "TRACE? TRACE1", ibsta, iberr, ibcntl)

'Read number of digits of length indication
Result = Space$(20)
Call RSDLLilrd(ud, Result, 2, ibsta, iberr, ibcntl)
Digits = Val(Mid$(Result, 2, 1))

'Read length indication
Result = Space$(20)
Call RSDLLilrd(ud, Result, Digits, ibsta, iberr, ibcntl)
TraceBytes = Val(Left$(Result, Digits))    'and store

' Read out trace data
Call RSDLLilrdTraceReal(ud, TraceData(0), TraceBytes, ibsta, iberr,ibcntl)
```

Programming examples:

- In this example, the start frequency of the instrument is queried.

```
Dim ibsta As Integer    ' Status variable
Dim iberr As Integer    ' Error variable
Dim ibcntl As Long      ' Count variable
Dim ud As Integer       ' Handle for measuring instrument
Dim Response As String  ' Response string

' Set up connection to measuring instrument
ud = RSDLLibfind("89.10.38.97", ibsta, iberr, ibcntl)
If (ud < 0) Then
   ' Error treatment
End If

' Send query command
Call RSDLLibwrt(ud, "FREQ:START?", ibsta, iberr, ibcntl)

' Provide space for response
Response = Space$(100)

' Read response from measuring instrument
Call RSDLLibrd(ud, Response, ibsta, iberr, ibcntl)
```

- In this example, a Save/Recall of the instrument setups is performed.

```
Dim ibsta As Integer    ' Status variable
Dim iberr As Integer    ' Error variable
Dim ibcntl As Long      ' Count variable
Dim ud As Integer       ' Handle for measuring instrument
Dim Cmd As String       ' Command string

' Set up connection to measuring instrument
ud = RSDLLibfind("89.10.38.97", ibsta, iberr, ibcntl)
If (ud < 0) Then
   ' Error treatment
End If

' Request instrument settings
Cmd = "SYST:SET?"
Call RSDLLibwrt(ud, Cmd, ibsta, iberr, ibcntl)

' Store instrument response in file
Call RSDLLibrdf(ud, "C:\db.sav", ibsta, iberr, ibcntl)

' Reset instrument
Call RSDLLibwrt(ud, "*RST", ibsta, iberr, ibcntl)

' and restore the previous settings
' to this end disable the END message
Call RSDLLibeot(ud, 0, ibsta, iberr, ibcntl)
' first send off command
Call RSDLLibwrt(ud, "SYST:SET ", ibsta, iberr, ibcntl)
' enable the END message again
Call RSDLLibeot(ud, 1, ibsta, iberr, ibcntl)
' and send the data
Call RSDLLibwrtf(ud, "C:\db.sav", ibsta, iberr, ibcntl)
```

Visual Basic for Applications (Winword and Excel)

Programming hints:

The programming language Visual Basic for Applications (VBA) is supported as a macro language by various manufacturers. The programs Winword and Excel use this language for the versions Winword 97 or Excel 5.0 and higher.
For macros created with Visual Basic for Applications, the same hints are valid as for Visual Basic Applications.

Programming example:

- Using the macro `QueryMaxPeak` a single sweep with subsequent query of the maximum peak is performed. The result is entered in a Winword or Excel document.

```
Sub QueryMaxPeak()

    Dim ibsta As Integer    ' Status variable
    Dim iberr As Integer    ' Error variable
    Dim ibcntl As Long      ' transferred characters
    Dim ud As Integer       ' Unit Descriptor (handle)for instrument
    Dim Response As String  ' Response string

    ' Set up connection to measuring instrument
    ud = RSDLLibfind("89.10.38.97", ibsta, iberr, ibcntl)
    If (ud < 0) Then
      Call MsgBox("Device with address 89.10.38.97 could" & _
                  "not be found", vbExclamation)
      End
    End If

    ' Determine maximum peak in the range 1-2MHZ
    Call RSDLLibwrt(ud, "*RST", ibsta, iberr, ibcntl)
    Call RSDLLibwrt(ud, "INIT:CONT OFF", ibsta, iberr, ibcntl)
    Call RSDLLibwrt(ud, "FREQ:START 1MHZ", ibsta, iberr, ibcntl)
    Call RSDLLibwrt(ud, "FREQ:STOP 2MHZ", ibsta, iberr, ibcntl)
    Call RSDLLibwrt(ud, "INIT:IMM;*WAI", ibsta, iberr, ibcntl)
    Call RSDLLibwrt(ud, "CALC:MARK:MAX;Y?", ibsta, iberr, ibcntl)
    Response = Space$(100)
    Call RSDLLibrd(ud, Response, ibsta, iberr, ibcntl)
    Response = RTrim(Response) ' Cut off space

    ' Insert value in current document (Winword)
    Selection.InsertBefore (Response)
    Selection.Collapse (wdCollapseEnd)

    ' Terminate connection to measuring instrument
    Call RSDLLibonl(ud, 0, ibsta, iberr, ibcntl)

End Sub
```

The entry of the peak value in the Winword document can be replaced as follows for Excel:

```
    ' Insert value in current document (Excel)
    ActiveCell.FormulaR1C1 = Response
```

C / C++

Programming Hint:

- Access to the functions of the RSIB32.DLL (Windows platforms)

  The functions of the `RSIB32.DLL` are declared in the header file `RSIB.H`. The DLL functions can be linked to a C/C++ program in different ways.

  1. Enter one of the supplied import libraries (`RSIB.LIB or RSIB32.LIB`) into the linker options.
  2. Load the library using the function `LoadLibrary()` during runtime and determine the function pointers of the DLL functions using `GetProcAddress()`. Before the end of the program, the `RSIB.DLL` must be unloaded again using the function `FreeLibrary()`.

  When import libraries are used, the DLL is automatically loaded immediately before the application is started. At the end of the program, the DLL is unloaded again unless it is still used by other applications.

- Access to librsib.so functions (Unix platforms)

  The functions of `librsib.so` are declared in the header file `RSIB.H`. Upper/lower case characters for file names are typically observed under Unix. The library functions are linked to a C/C++ program by entering the `-lrsib` linker option.

  The *shared library* `librsib.so` is automatically loaded on starting the application. The accessibility (for example via standard path) of the library must be ensured. Refer to the beginning of this main chapter under „Unix Environment“.

- Query of strings

  If instrument responses are to be further processed as strings, a zero termination must be appended.

  Example:

```
char buffer[100];
...
RSDLLibrd( ud, buffer, &ibsta, &iberr, &ibcntl );
buffer[ibcntl] = 0;
```

Programming example:

In the following C program, a single sweep is started on the instrument with the IP address 89.1.1.200 and a marker is set to the maximum level. Before the maximum level can be determined, the sweep must be terminated. Synchronization to the end of the sweep is made by triggering a service request at the end of the sweep with command "*OPC" (operation complete). The control program waits for the SRQ with the function RSDLLWaitSrq(). Then the maximum level is determined ("CALC:MARK:MAX") and the level is read out ("Y?"). Prior to readout a check is made with serial poll whether data are available (MAV bit set in status register

```
#define MAX_RESP_LEN  100

short           ibsta, iberr;
unsigned long   ibcntl;
short           ud;
short           srq;
char            Maxlevel[MAX_RESP_LEN];
char            spr;
// Determining the handle for the instrument
ud = RSDLLibfind( "89.1.1.200", &ibsta, &iberr, &ibcntl );

// If instrument exists
if ( ud >= 0 ) {

  // Setting timeout for RSDLLWaitSrq() to 10 seconds
  RSDLLibtmo( ud, 10, &ibsta, &iberr, &ibcntl );

  // Activate SRQ generation through event status register (ESR)
  // and enable ESB bit of SRE register
  RSDLLibwrt( ud, "*ESE 1;*SRE 32", &ibsta, &iberr, &ibcntl );

  // Set single sweep, trigger sweep
  // and generate SRQ at the end of the sweep with "*OPC"
  RSDLLibwrt( ud, "INIT:CONT off;INIT;*OPC", &ibsta, &iberr, &ibcntl );

  // Wait for SRQ (end of sweep)
  RSDLLWaitSrq( ud, &srq, &ibsta, &iberr, &ibcntl );

  // Clear RQS/MSS bit
  RSDLLibrsp( ud, &spr, &ibsta, &iberr, &ibcntl );

  // If sweep is terminated
  if (srq) {

    // Set marker to first maximum and query level
    RSDLLibwrt( ud, "CALC:MARK:MAX;Y?", &ibsta, &iberr, &ibcntl );
    // Check if data are available (MAV bit in status register set)
     RSDLLibrsp( ud, &spr, &ibsta, &iberr, &ibcntl );
     if (spr & 0x10) {
        // then read out data
        RSDLLilrd( ud, MaxPegel, MAX_RESP_LEN, &ibsta, &iberr, &ibcntl );
    }
  }
  // end connection to instrument
  RSDLLibonl (ud, 0, &ibsta, &iberr, &ibcntl ) ;
 }

else {
  ; // Error - instrument not found
}
```

Contents - Chapter 8 "Maintenance and Instrument Interfaces"

8 Maintenance and Instrument Interfaces 8.1

Maintenance 8.1

Mechanical Maintenance 8.1

Electrical Maintenance 8.1

Testing the Level Measuring Accuracy 8.1

Testing the Frequency Accuracy 8.1

Instrument Interfaces 8.2

IEC Bus Interface 8.2

Interface Characteristics 8.2

Bus Lines 8.3

Interface Functions 8.4

IEC Bus Messages 8.4

Interface Messages 8.4

Instrument Messages 8.5

RS-232-C Interface 8.6

Interface Characteristics 8.6

Signal Lines 8.6

Interface Functions 8.7

Transmission Parameters 8.7

Handshake 8.8

RSIB Interface 8.10

Windows Environment 8.10

Unix Environment 8.11

RSIB Interface Functions 8.12

Variables ibsta, iberr, ibcntl 8.12

List of Interface Functions 8.13

Description of Interface Functions 8.14

User Interface (USER) 8.21

Printer Interface (LPT) 8.22

Measurement Converters (PROBE CODE) 8.23

Probe Connector (PROBE POWER) 8.23

AF-Output (AF OUTPUT) 8.24

IF Output 21.4 MHz (21.4 MHz OUT) 8.24

Video Output (VIDEO OUT) 8.24

Reference Output/Input (EXT REF IN/OUT) 8.24

Sweep Output (SWEEP) 8.24

External Trigger Input (EXT TRIGGER/GATE) 8.24

Noise Source Control (NOISE SOURCE) 8.24

External Keyboard (KEYBOARD) 8.25

Mouse Connector (MOUSE) 8.25

Monitor Connector (MONITOR) 8.25

External Video Filter Input/Output (VIDEO FILTER EXT IN/OUT) 8.26

Unstretched Video Output (VIDEO OUT UNSTRETCHED) 8.26

IF Output 741.4 MHz (741.4 MHz OUT) 8.26

20-MHz Reference Output (REF 20 MHz OUT) 8.26
Preselector Control..... 8.26
Demodulation Output (AM/FM OUT) 8.26

8 Maintenance and Instrument Interfaces

The following chapter contains information on the maintenance of the FSET and on the instrument interfaces.

The address of our support center and a list of all Rohde & Schwarz service centers can be found at the beginning of this manual.

Maintenance

Mechanical Maintenance

The FSET does not require any mechanical maintenance. The front panel can be cleaned from time to time using a damp, soft cloth.

Electrical Maintenance

Testing the Level Measuring Accuracy

Due to the total calibration facility by means of the built-in calibration generator, high long-term stability of the level measuring characteristics is guaranteed. It is recommended to check the measuring accuracy every two years according to the performance test. Reprogramming of the correction data by a R&S servicing shop is necessary, if tolerances are exceeded.

Testing the Frequency Accuracy

The frequency accuracy of the reference oscillator is to be checked once a year according to section 5. It is not necessary to perform this test if the instrument is operated using an external reference.

Test utility: Frequency counter of signal generator

Test setup: Measurement with frequency counter
Connect frequency counter to the REF OUT/IN socket at the rear of the instrument.

Measurement with signal generator
Apply signal with 1GHz, -10dBm to RFin.

Measurement: Settings on FSET:

[CENTER 1000 MHz**]**
[SPAN 0 MHz**]**
[REF REF LEVEL -10 dBm**]**
[MARKER COUNT ON
COUNTER RESOL 0.1 Hz**]**

Activate internal frequency counter (marker function)

Instrument Interfaces

IEC Bus Interface

The standard instrument is equipped with an IEC/IEEE Bus connector. An IEEE 488 interface connector is located on the rear panel of the FSET. An external controller for remote control of the instrument can be connected via the IEEE 488 interface connector using a shielded cable.

A further optional IEC/IEEE-bus interface can be installed in the FSET and is assigned to the computer. This interface can be controlled by standard software (R&S-Basic, QUICK-Basic, etc.). It enables the remote control of the instrument via an external link to the two IEC/IEEE-bus interfaces and in addition the control of other equipment via the IEC/IEEE-bus connector at the rear of the unit (eg control of a complete test setup).

The following section describes the first IEC Bus interface over which the instrument can be remotely controlled. The characteristics of the PC2A/PC-AT controller interface depend upon the user-installed software running on the processor and is, therefore, not described here.

Interface Characteristics

- 8-bit parallel data transfer
- bidirectional data transfer
- three-line handshake
- high data transfer rate of max. 350 kbyte/s
- up to 15 instruments can be connected
- maximal length of the interconnecting cables 15 m (single connection, 2m)
- wired-OR connection if several instruments are connected in parallel.

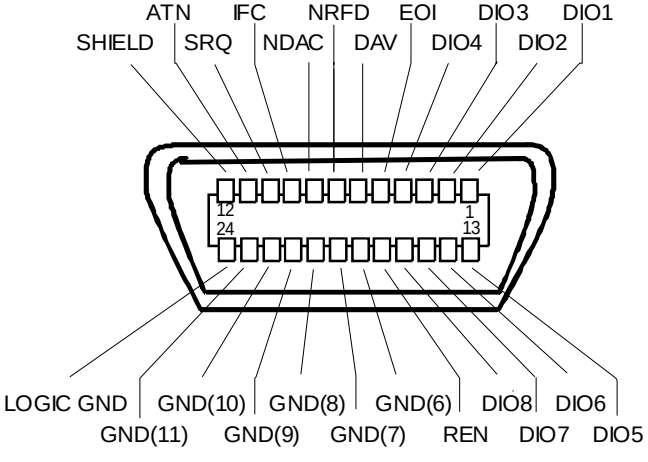


Fig. 8-1 Pin assignment of IEC-Bus interface

Bus Lines

1. Data bus with 8 lines DIO 1 to DIO 8.
The transmission is bit-parallel and byte-serial in the ASCII/ISO code. DIO1 is the least significant, DIO8 the most significant bit.

2. Control bus with 5 lines.
IFC (Interface Clear),
active low resets the interfaces of the devices connected to the default setting.

ATN (Attention),
active low signals the transmission of interface messages
inactive high signals the transmission of device messages.

SRQ (Service Request),
active low enables a device connected to send a service request to the controller.

REN (Remote Enable),
active low permits the switch over to remote control.

EOI (End or Identify),
has two functions in connection with ATN:
active low marks the end of data transmission when ATN=high
active low triggers a parallel poll when ATN=low.

3. Handshake bus with three lines.
DAV (Data Valid),
active low signals a valid data byte on the data bus.

NRFD (Not Ready For Data),
active low signals that one of the devices connected is not ready for data transfer .

NDAC (Not Data Accepted),
active low as long as the device connected is accepting the data present on the data bus.

Interface Functions

Instruments which can be remote controlled via the IEC bus can be equipped with different interface functions. Table 8-1 lists the interface functions appropriate for the instrument.

Table 8-1 Interface functions

Control character	**Interface function**
SH1	Handshake source function (source handshake), full capability
AH1	Handshake sink function (acceptor handshake), full capability
L4	Listener function, full capability, deaddressed by MTA.
T6	Talker function, full capability, ability to respond to serial poll, deaddressed by MLA
SR1	Service request function (Service Request), full capability
PP1	Parallel poll function, full capability
RL1	Remote/Local switch over function, full capability
DC1	Reset function (Device Clear), full capability
DT1	Trigger function (Device Trigger), full capability
C12	Controller function, capability to send interface messages, to accept or relinquish the controller function

IEC Bus Messages

The messages transferred via the data lines of the IEC bus can be divided into two groups:

– **interface messages**
and
– **instrument messages.**

Interface Messages

Interface messages are transferred on the data lines of the IEC Bus when the "ATN" control line is active (LOW). They are used for communication between controller and instruments and can only be sent by the controller which currently has control of the IEC Bus.

Universal Commands

The universal commands are encoded 10 - 1F hex. They affect all instruments connected to the bus without addressing.

Table 8-2 Universal Commands

Command	QuickBASIC command	Effect on the instrument
DCL (Device Clear)	IBCMD (controller%, CHR$(20))	Aborts the processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument settings.
IFC (Interface Clear)	IBSIC (controller%)	Resets the interfaces to the default setting.
LLO (Local Lockout)	IBCMD (controller%, CHR$(17))	The LOC/IEC ADDR key is disabled.
SPE (Serial Poll Enable)	IBCMD (controller%, CHR$(24))	Ready for serial poll.
SPD (Serial Poll Disable)	IBCMD (controller%, CHR$(25))	End of serial poll.
PPU (Parallel Poll Unconfigure)	IBCMD (controller%, CHR$(21))	End of the parallel-poll state.

Addressed Commands

The addressed commands are encoded 00 - 0F hex. They are only effective for instruments addressed as listeners.

Table 8-3 Addressed Commands

Command	QuickBASIC command	Effect on the instrument
SDC (Selected Device Clear)	IBCLR (device%)	Aborts the processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument setting.
GTL (Go to Local)	IBLOC (device%)	Transition to the "Local" state (manual control).
PPC (Parallel Poll Configure)	IBPPC (device%, data%)	Configure instrument for parallel poll. Additionally, the QuickBASIC command executes PPE/PPD.

Instrument Messages

Instrument messages are transferred on the data lines of the IEC bus when the "ATN" control line is not active. ASCII code is used.
Structure and syntax of the instrument messages are described in Chapter 5. The commands are listed and explained in detail in Chapter 6.

RS-232-C Interface

The standard FSET is equipped with two serial interfaces (RS-232-C). The RS-232-C interfaces are two independent, remote-control channels, both of which may be active at the same time. The programming and interrogation commands affect the same instrument hardware. The interfaces can be set up manually in the *SETUP-GENERAL SETUP* menu in the *COM PORT1/2* table.

Each of the active RS-232-C interfaces is assigned to one of the 9-pin connectors located on the rear panel. Interface 1 is assigned to the connector COM1 and Interface 2 is assigned to the connector COM2.

Interface Characteristics

- Serial data transmission in asynchronous mode
- Bidirectional data transfer via two separate lines
- Transmission rate selectable from 110 to 19200 baud
- Logic '0' signal from+3 V to +15 V
- Logic '1' signal from -15 V to -3 V
- An external instrument (controller) can be connected.
- Software handshake (XON, XOFF)
- Hardware handshake

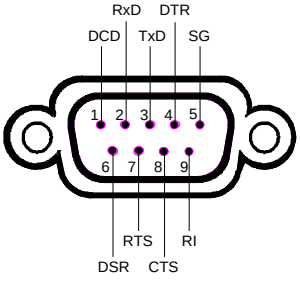


Fig. 8-2 Pin assignment of the RS-232-C interface

Signal Lines

1. Data lines

The data transmission is bit-serial in ASCII code starting with the LSB. Two lines, **RxD** and **TxD** are necessary as the minimum requirement for transmission; however, no hardware handshake is possible. For handshaking, only the XON/XOFF software handshake protocol can be used .

RxD (Receive Data)
Input, LOW = logic '1', HIGH = logic '0'.
Data line, local terminal receives data from remote station.

TxD (Transmit Data)
Output, LOW = logic '1', HIGH = logic '0'.
Data line, local terminal transmits data to remote station.

2. Control lines

DCD (Data Carrier Detect),
Not used in FSET.

DTR (Data Terminal Ready),
Output, active LOW,
Indicates that the local terminal is ready to receive data.

DSR (Data Set Ready),
Input, active LOW,
Indicates that the remote station is ready to receive data.

RTS (Request To Send),
Output, active LOW.
Indicates that the local terminal wants to transmit data.

CTS (Clear To Send),
Input, active LOW.
Used to tell the local terminal that the remote station is ready to receive data.

RI (Ring Indicator),
Not used in FSET.

Interface Functions

For interface control, several strings are defined and control characters are reserved which are based upon IEC-\IEEE bus control.

Table 8-4 Control strings or control characters of the RS-232 interface

Control string or character	Interface function
"@REM"	Switch over to remote
"@LOC"	Switch over to local
<Ctrl Q> 11 Hex	Enables character output
<Ctrl S> 13 Hex	Inhibits character output
0D Hex, 0A Hex	Terminator <CR>, <LF>

Transmission Parameters

To ensure error-free data transmission, the parameters of the instrument and the controller must have the same settings. The parameters are defined in the *SETUP-GENERAL SETUP* menu.

Transmission rate (baud rate) the following transmission rates can be set in the analyzer: 110, 300, 600, 1200, 2400, 4800, 9600, 19200.

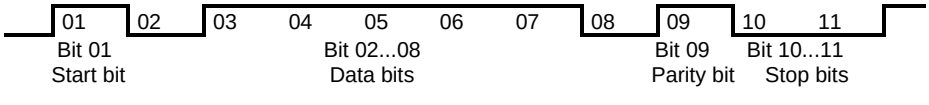
Data bits Data transmission is in 7- or 8-bit ASCII code. The first bit transmitted is the LSB (least significant bit).

Start bit Each data byte starts with a start bit. The falling edge of the start bit indicates the beginning of the data byte.

Parity bit In order to detect errors, a parity bit may be transmitted. No parity, even parity or odd parity may be selected. In addition, the parity bit can be set to logic '0' or to logic '1'.

Stop bits The transmission of a data byte is terminated by 1, 1,5 or 2 stop bits.

Example:
Transmission of character 'A' (41 hex) in 7-bit ASCII code, with even parity and 2 stop bits:



Handshake

Software handshake

In the software handshake mode of operation, the data transfer is controlled using the two control characters XON / XOFF.

The instrument uses the control character XON to indicate that it is ready to receive data. If the receive buffer is full, it sends the XOFF character via the interface to the controller. The controller then interrupts the data output until it receives another XON from the instrument. The controller indicates to the instrument that it is ready to receive data in the same way.

Cable required for software handshake

The connection of the instrument with a controller for software handshake is made by crossing the data lines. The following wiring diagram applies to a controller with a 9-pin or 25-pin configuration.

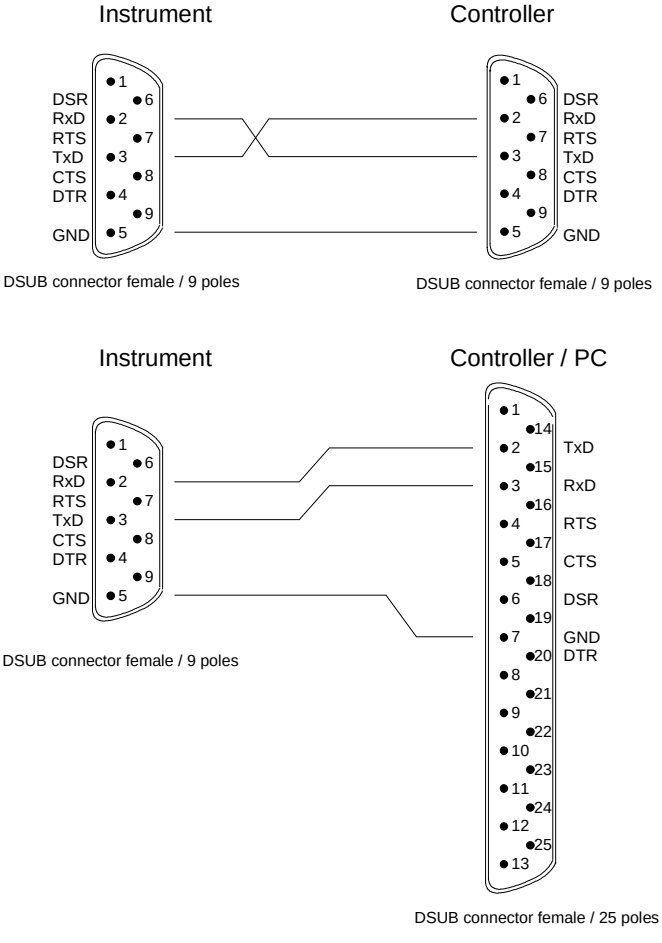


Fig. 8-3 Wiring of the data lines for software handshake

Hardware handshake

For hardware handshake, the instrument indicates that it is ready to receive data via the lines DTR and RTS. A logic '0' on both lines means 'ready' and a logic '1' means 'not ready'. The RTS line is always active (logic '0') as long as the serial interface is switched on. The DTR line thus controls the readiness of the instrument to receive data.

The readiness of the remote station to receive data is reported to the instrument via the CTS and DSR line. A logic '0' on both lines activates the data output and a logic '1' on both lines stops the data output of the instrument. The data output takes place via the TxD line.

Cable for hardware handshake

The connection of the instrument to a controller is made with a so-called zero modem cable. Here, the data, control and acknowledge lines must be crossed. The following wiring diagram applies to a controller with a 9-pin or 25-pin configuration.

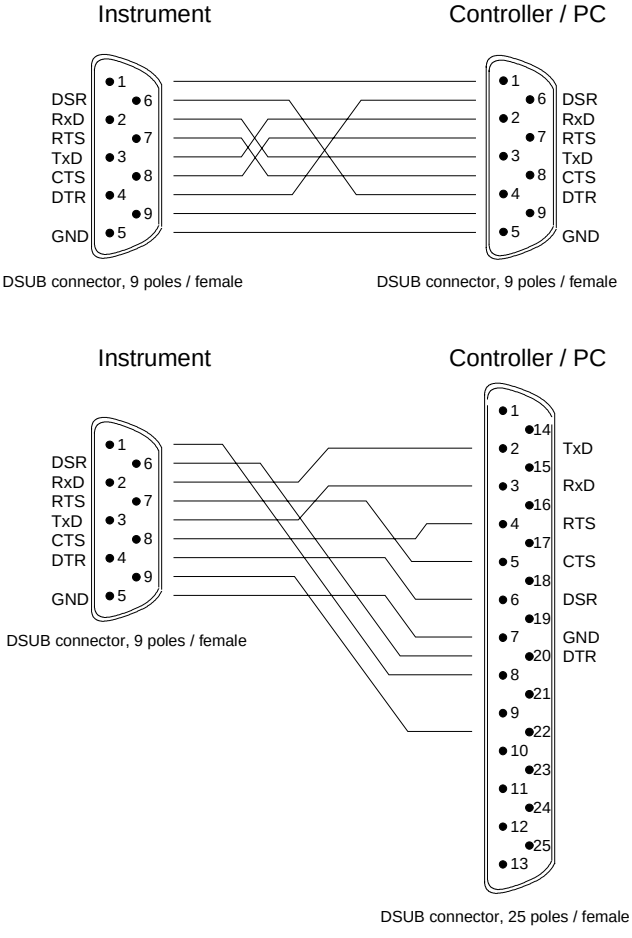


Fig. 8-4 Wiring of the data, control and acknowledge lines for hardware handshake

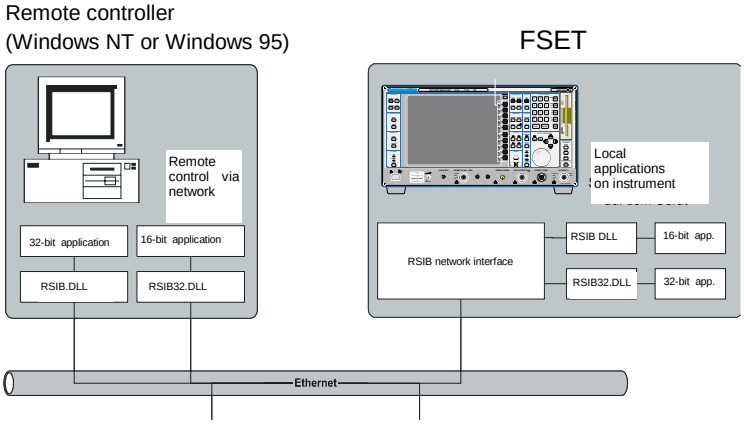
RSIB Interface

The RSIB interface allows the control of FSET by Visual C++ and VisualBasic programs but also by using the Windows applications WinWord and Excel. The functions for the programming of control applications are provided by DLLs `RSIB32.DLL` (for 32-bit applications) and `RSIB.DLL` (for 16-bit applications).

A Unix operating system can be installed on an external computer in addition to a Windows operating system. In this case, the control applications are created either in C or C++. The supported Unix operating systems include:

- Sun Solaris 2.6 Sparc Station
- Sun Solaris 2.6 Intel Platform
- Red Hat Linux 6.2 x86 Processors

Control applications may run locally on the measuring instrument as well as on a remote controller in a network. With local control, the name `'@local'` is specified for link setup by means of function `RSDLLibfind()`. If `'@local'` is not specified, the `RSIB.DLL` interprets the name as an IP address and attempts to set up a link to the device via the Winsock interface.



Windows Environment

To access the measuring instruments via the RSIB interface the DLLs should be installed in the corresponding directories:

- `RSIB.DLL` in Windows NT `system` directory or control application directory.
- `RSIB32.DLL` in Windows NT `system32` directory or control application directory.

The DLLs are already installed in the corresponding directories on the measuring instruments.

For the different programming languages, files exist containing the declarations for the DLL functions and the error code definitions.

Visual Basic (16 bit):	`'RSIB.BAS'`	`(C:/R_S/Instr/RSIB)`
Visual Basic (32 bit):	`'RSIB32.BAS'`	`(C:/R_S/Instr/RSIB)`
C:	`'RSIBC.H'`	`(C:/R_S/Instr/RSIB)`

Thus, the RSIB directory features a program 'RSIBCNTR.EXE' with SCPI commands which can be sent to the instrument by the RSIB interface. This program can be used as a test for the function of the interface. Transit-time module VBRUN3200.DLL in the path or the Windows directory are required.

The control is performed with Visual C++ or Visual Basic programs. The local link to the internal controller is established with the name '@local'.

Via VisualBasic: internal controller: ud = RSDLLibfind ("@local", ibsta, iberr, ibcntl)
remote controller: ud = RSDLLibfind ('82.1.1.200', ibsta, iberr, ibcntl)

The return to manual operation can be performed via the front panel (*LOCAL* key) or the RSIB interface.

Via RSIB:
```
...
ud = RSDLLibloc (ud, ibsta, iberr, ibcntl);
...
```

Unix Environment

In order to access the measuring equipment via the RSIB interface, copy the `librsib.so.X.Y` file to a directory for which the control application has read rights. `X.Y` in the file name indicates the version number of the library, for example `1.0`.

The `librsib.so.X.Y` library is created as a *shared library*. The applications using the library have nothing to do with versions. They simply link the library with the `lrsib` option. The following instructions have to be observed so that linking can be successfully performed and the library can be found during the program run:

File link:

- Use the operating system command *In* to create a file with the link name `librsib.so` and pointing to `librsib.so.X.Y` in a directory for which the control application has read rights. Example:
  `$ ln –s /usr/lib/librsib.so.1.0 /usr/lib/librsib.so`

Linker options for creating applications:

- `-lrsib` : import library
- `-Lxxx` : path information where the import library can be found. This is where the above file link has been created. Example: `-L/usr/lib`.

Additional linker options for creating applications (only under Solaris):

- `-Rxxx`: path information where the library is searched for during the program run: `-R/usr/lib`.

Run-time environment:

- Set environment variable `LD_RUN_PATH` to the directory in which the file link has been created. This is necessary only if `librsib.so` cannot be found in the standard search path of the operating system and the `-R` linker option (only Solaris) was not specified.

For C/C++ programming, the declarations of the library functions and the definition of error codes are contained in:

C/C++: `'RSIB.H'(C:\R_S\Instr\RSIB)`

RSIB Interface Functions

This following sections describe all functions of libraries 'RSIB.DLL'or 'RSIB32.DLL' or "librsib.so" by means of which control applications can be written.

Variables ibsta, iberr, ibcntl

Same as with the National Instrument interface, successful execution of a command can be checked by means of the variables ibsta, iberr and ibcntl. To this end, references to the three variables are transferred to all RSIB functions. In addition, the status word ibsta is returned as a function value by all functions.

Status word ibsta
All functions send back a status word that provides information on the status of the RSIB interface. The following bits are defined:

Bit name	Bit	Hex code	Description
ERR	15	8000	This bit is set if an error occurs during a function call. If this bit is set, iberr contains an error code which specifies the error.
TIMO	14	4000	This bit is set if a timeout occurs during a function call. A timeout may occur in the following situations: • while waiting for an SRQ with the function RSDLLWaitSrq(). • if no acknowledgment is received for data sent to an instrument with RSDLLibwrt() or RSDLLilwrt(). • if no response from server to a data request with function RSDLLibrd() or RSDLLilrd().
CMPL	8	0100	This bit is set if the reply of the IEC/IEEE-bus parser is completely read. If a reply of the parser is read with the function RSDLLilrd() and the buffer length is not sufficient, the bit is cleared.

Error variable iberr
If the ERR bit (8000h) is set in the status word, iberr contains an error code that specifies the error. The RSIB has error codes of its own independent of the National Instrument interface.

Error	Error code	Description
IBERR_DEVICE_REGISTER	1	RSIB.DLL cannot register any new device.
IBERR_CONNECT	2	Link to the device has failed.
IBERR_NO_DEVICE	3	An interface function was called with an invalid device handle.
IBERR_MEM	4	No free memory available.
IBERR_TIMEOUT	5	Timeout has occurred.
IBERR_BUSY	6	The RSIB interface is blocked by a function not yet completed. Windows is not blocked, for example, by function RSDLLibrd() if data are still to be transmitted in response to this function. In this case a new call is possible. Further calls are however rejected by RSIB.DLL with error code IBERR_BUSY.
IBERR_FILE	7	Error in reading from or writing to a file.
IBERR_SEMA	8	Error upon creating or assigning a semaphore (only under Unix)

Count variable - ibcntl
The variable ibcntl is updated with the number of bytes transmitted on every read and write function call.

List of Interface Functions

The DLL functions are matched to the interface functions for IEC/IEEE-bus programming from National Instruments. Functions supported by the DLLs are listed in the following table.

Function	Description
RSDLLibfind()	Provides a handle for accessing a device.
RSDLLibwrt()	Sends a string terminated with a null to a device.
RSDLLilwrt()	Sends a specific number of bytes to a device.
RSDLLibwrtf()	Sends the contents of a file to a device.
RSDLLibrd()	Reads data from a device into a string.
RSDLLilrd()	Reads a specific number of bytes from a device.
RSDLLibrdf()	Reads data from a device into a file.
RSDLLibtmo()	Sets timeout period for RSIB functions.
RSDLLibsre()	Switches a device to local or remote.
RSDLLibloc()	Switches a device temporarily to local.
RSDLLibeot()	Enables/disables END message in write operations.
RSDLLibrsp()	Starts a serrial poll and reads the status byte
RSDLLibonl()	Switches the instrument online/offline
RSDLLTestSrq()	Checks if a device has generated an SRQ.
RSDLLWaitSrq()	Waits until a device generates an SRQ.
RSDLLSwapBytes	Swaps the byte sequence for binary numeric display (only required for non-Intel platforms)

Description of Interface Functions

RSDLLibfind()

The function provides a handle for access to the device with the name udName.

VB format: `Function RSDLLibfind (ByVal udName$, ibsta%, iberr%, ibcntl&) As Integer`

C-format: `short FAR PASCAL RSDLLibfind( char far *udName, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

C format (Unix): `short RSDLLibfind( char *udName, short *ibsta, short *iberr, unsigned long *ibcntl)`

Parameter: `udName` IP address of device

Example: `ud = RSDLLibfind ("89.10.38.97", ibsta, iberr, ibcntl)`

The function must be called before all other interface functions.

The function supplies a handle as a return value. The handle must be specified in all functions for accessing the instrument. If no instrument with the name udName is found, the handle will take on a negative value.

The local link on the instrument is established with the name '@local'. For setting up a link via a network, on the other hand, the IP address of the instrument must be given (eg '89.1.1.200')

RSDLLibwrt

The function sends data to the instrument with the handle ud.

VB format: `Function RSDLLibwrt (ByVal ud%, ByVal Wrt$, ibsta%, iberr%, ibcntl&) As Integer`

C format: `short FAR PASCAL RSDLLibwrt( short ud, char far *Wrt, short far *ibsta, short far *iberr, unsigned long far *ibcntl )`

C format (Unix): `short RSDLLibwrt( short ud, char *Wrt, short *ibsta, short *iberr, unsigned long *ibcntl )`

Parameter:

`ud`	Device handle
`Wrt`	String sent to the device.

Example: `RSDLLibwrt(ud, "SENS:FREQ:STAR?", ibsta, iberr, ibcntl)`

The function allows setting and query commands to be sent to the measuring instruments. The function `RSDLLibeot()` is used for defining whether the data are interpreted as a complete command.

RSDLLilwrt

The function sends `Cnt` bytes to the instrument with the handle `ud`.

VB format: `Function RSDLLilwrt (ByVal ud%, ByVal Wrt$, ByVal Cnt&, ibsta%, iberr%, ibcntl&) As Integer`

C format: `short FAR PASCAL RSDLLilwrt( short ud, char far *Wrt, unsigned long Cnt, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

C format (Unix): `short RSDLLilwrt( short ud, char *Wrt, unsigned long Cnt, short *ibsta, short *iberr, unsigned long *ibcntl)`

Parameter:

`ud`	Device handle
`Wrt`	String sent to the IEC/IEEE-bus parser.
`Cnt`	Number of bytes sent to the instrument.

Example: `RSDLLilwrt (ud, '......', 100, ibsta, iberr, ibcntl)`

This function sends data to an instrument same as function `RSDLLibwrt()` but with the difference that binary data, too, can be sent. The length of the data is determined not by a zero-terminated string but by the definition of `Cnt` bytes. If the data are to be terminated with EOS (0Ah), the EOS byte must be appended to the string.

RSDLLibwrtf

The function sends the contents of a `file` to the instrument with the handle `ud`.

VB format: `Function RSDLLibwrtf (ByVal ud%, ByVal file$, ibsta%, iberr%, ibcntl&) As Integer`

C format: `short FAR PASCAL RSDLLibwrt( short ud, char far *Wrt, short far *ibsta, short far *iberr, unsigned long far *ibcntl )`

C format (Unix): `short RSDLLibwrtf( short ud, char *Wrt, short *ibsta, short *iberr, unsigned long *ibcntl )`

Parameter:

`ud`	Device handle
`file`	File whose contents are sent to the instrument.

Example: `RSDLLibrdf(ud, "C:\db.sav", ibsta, iberr, ibcntl)`

The function allows setting and query commands to be sent to the measuring instruments. The function `RSDLLibeot()` is used for defining whether the data are interpreted as a complete command.

RSDLLibrd()

The function reads data from the instrument into the string Rd with the handle ud.

VB format: `Function RSDLLibrd (ByVal ud%, ByVal Rd$, ibsta%, iberr%, ibcntl&) As Integer`

C format: `short FAR PASCAL RSDLLibrd( short ud, char far *Rd, short far *ibsta, short far *iberr, unsigned long far *ibcntl )`

C format (Unix): `short RSDLLibrd( short ud, char *Rd, short *ibsta, short *iberr, unsigned long *ibcntl )`

Parameter:

ud	Device handle
Rd	String into which the read data are copied.

Example: `RSDLLibrd (ud, Rd, ibsta, iberr, ibcntl)`

This function fetches replies of the IEC/IEEE-bus parser in response to a query command.

For this, a string of sufficient length must be generated when programming in Visual Basic. This can be accomplished in the definition of the string or with the command Space$().

Generation of a string of the length 100:

- `Dim Rd as String * 100`
- ```
  Dim Rd as String
  Rd = Space$(100)
  ```

RSDLLilrd

The function reads Cnt bytes from the instrument with the handle ud.

VB format: `Function RSDLLilrd (ByVal ud%, ByVal Rd$, ByVal Cnt&, ibsta%, iberr%, ibcntl&) As Integer`

C format: `short FAR PASCAL RSDLLilrd( short ud, char far *Rd, unsigned long Cnt, short far *ibsta, short far *iberr, unsigned long far *ibcntl )`

Parameter:

ud	Device handle
cnt	Maximum number of bytes copied to target string Rd by the DLL.

Example: `RSDLLilrd (ud, RD, 100, ibsta, iberr, ibcntl)`

The function reads data from an instrument same as function RSDLLibrd() but with the difference that here the maximum number of bytes is allowed to be copied to target string Rd to be defined with Cnt. Writing beyond a string can thus be prevented with this function. The number of bytes cut off is lost.

RSDLLibrdf()

Reads data from the instrument into the file file with the handle ud.

VB format: `Function RSDLLibrdf (ByVal ud%, ByVal file$, ibsta%, iberr%, ibcntl&) As Integer`

C format: `short FAR PASCAL RSDLLibrdf( short ud, char far *file, short far *ibsta, short far *iberr, unsigned long far *ibcntl )`

C format (Unix): `short RSDLLibrd( short ud, char *file, short *ibsta, short *iberr, unsigned long *ibcntl )`

Parameter:

ud	Device handle
file	File into which the read data are written.

Example: `RSDLLibrdf (ud, c:\db.sav, ibsta, iberr, ibcntl)`

This function serves to read replies of the IEC/IEEE-bus parser that are larger than 64 KB. The file name may include the drive and path.
  ```

RSDLLibtmo

The function defines the timeout for an instrument. The default value for the timeout is 5 seconds.

VB format: `Function RSDLLibtmo (ByVal ud%, ByVal tmo%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLibtmo( short ud, short tmo, short far *ibsta, short far *iberr, unsigned long far *ibcntl )`

C format (Unix): `short RSDLLibtmo( short ud, short tmo, short *ibsta, short *iberr, unsigned long *ibcntl )`

Parameter:		
	`ud`	Device handle
	`tmo`	Time in seconds

Example: `RSDLLibtmo (ud, 10, ibsta, iberr, ibcntl)`

Timeout can occur in the following cases:

- Waiting for an SRQ with the function `RSDLLWaitSrq()`.
- Waiting for an acknowledgment for data sent to an instrument with `RSDLLibwrt()` or `RSDLLilwrt()`.
- Waiting for a reply to a data request made with function `RSDLLibrd()` or `RSDLLilrd()`.

RSDLLibsre

The function switches the instrument to LOCAL or REMOTE mode.

VB format: `Function RSDLLibsre (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLibsre( short ud, short v, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

C format (Unix): `short RSDLLibsre( short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)`

Parameter:		
	`ud`	Device handle
	`v`	Instrument status
		0 - local
		1 - remote

Example: `RSDLLibsre (ud, 0, ibsta, iberr, ibcntl)`

RSDLLibloc

The function switches the instrument temporarily to LOCAL mode.

VB format: `Function RSDLLibloc (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLibloc( short ud, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

C format (Unix): `short RSDLLibloc( short ud, short *ibsta, short *iberr, unsigned long *ibcntl)`

Parameter: ud Device handle

Example: `RSDLLibloc (ud, ibsta, iberr, ibcntl)`

After switchover the instrument can be manually operated from the front panel. On the next access to the instrument by means of one of the functions of the RSIB.DLL, the instrument is switched again to the REMOTE state.

RSDLLibeot

The function enables the END message after write operations or disables it.

VB format: `Function RSDLLibeot (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLibeot( short ud, short v, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

C format (Unix): `short RSDLLibsre( short ud, short v, short *ibsta, short *iberr, unsigned long *ibcntl)`

Parameter: ud Device handle.

v 0 - no END message

1 - send END message

Example: `RSDLLibeot (ud, 1, ibsta, iberr, ibcntl)`

If the END message is disabled, the data of a command can be sent by means of several successive write function calls. The END message must be enabled prior to the last data block.

RSDLLibrsp

The function carries out a Serial Poll and outputs the instrument status.

VB format: `Function RSDLLibrsp(ByVal ud%, spr%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLibrsp( short ud, char far* spr, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

Parameter: ud Device handle.

spr Reference to status byte

Example: `RSDLLibrsp(ud, spr, ibsta, iberr, ibcntl)`

RSDLLibonl

The function switches the instrument to the 'online' or 'offline' state. When switching to ‘offline’ the interface is enabled and the device handle made invalid. The next call of RSDLLibfind sets up the communication again.

VB format: `Function RSDLLibonl (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLibonl( short ud, short v, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

Parameter:

ud	Device handle.
v	Status of instrument
	0 - local
	1 - remote

Example: `RSDLLibonl(ud, 0, ibsta, iberr, ibcntl)`

RSDLLTestSRQ

The function checks the status of the SRQ bit.

VB format: `Function RSDLLTestSrq (ByVal ud%, Result%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLTestSrq( short ud, short far *result, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

C format (Unix): `short RSDLLTestSrq( short ud, short *result, short *ibsta, short *iberr, unsigned long *ibcntl)`

Parameter:

ud	Device handle
`result` returns	Reference to an integer value in which the library the status of the SRQ bit.
	0 - no SRQ
	1 - SRQ active, instrument has output a service request

Example: `RSDLLTestSrq (ud, result%, ibsta, iberr, ibcntl)`

The function corresponds to the function `RSDLLWaitSrq` but with the difference that `RSDLLTestSRQ` returns immediately the current status of the SRQ bit whereas `RSDLLWaitSrq` waits for an SRQ to occur.

RSDLLWaitSrq

The function waits until the instrument triggers an SRQ with the handle ud.

VB format: `Function RSDLLWaitSrq (ByVal ud%, Result%, ibsta%, iberr%, ibcntl&) As Integer`

C format: `void FAR PASCAL RSDLLWaitSrq( short ud, short far *result, short far *ibsta, short far *iberr, unsigned long far *ibcntl)`

C format (Unix): `short RSDLLWaitSrq( short ud, short *result, short *ibsta, short *iberr, unsigned long *ibcntl)`

Parameter:

`ud`	Device handle
`result` returns	Reference to an integer value in which the library the status of the SRQ bit.
	0 - no SRQ occurred within the timeout
	1 - SRQ occurred within the timeout

Parameter: `RSDLLWaitSrq( ud, result, ibsta, iberr, ibcntl );`

The function waits until one of the two following events occurs:

- The instrument triggers an SRQ.
- No SRQ occurs during the timeout defined with `RSDLLibtmo()`.

RSDLLSwapBytes

This function changes the display of binary numbers on non-Intel platforms.

VB format: Not provided at present since it is required only on non-Intel platforms.

C format: `void FAR PASCAL RSDLLSwapBytes( void far *pArray, const long size, const long count)`

C format (Unix): `void RSDLLSwapBytes( void *pArray, const long size, const long count)`

Parameter:

`pArray`	Array in which modifications are made
`size`	Size of a single element in `pArray`
`count`	Number of elements in `pArray`

Example: `RSDLLSwapBytes( Buffer, sizeof(float), ibcntl/sizeof(float))`

This function swaps the display of various elements from *Big Endian* to *Little Endian* and vice versa. It is expected that a coherent storage area of elements of the same file type (`size` byte) is transferred to `pArray`. This function has no effect on Intel platforms.

Different types of processor architecture store data in different byte sequences. For example, Intel processors store data in the reverse order of Motorola processors. Comparison of byte sequences:

Byte sequence	Use in	Display in memory	Description
Big Endian	Motorola processors, network standard	Most significant byte at least significant address	The *most significant* byte is at the left end of the word.
Little Endian	Intel processors	Least significant byte at least significant address	The *most significant* byte is at the right end of the word.

User Interface (USER)

The user interface, located on the rear panel of the FSET, is a 25 pin Cannon connector which provides access to the two user ports (Port A and Port B). Each port is 8 bits wide (A0 - A7 and B0 -B7) and can be configured either as output or as input. The voltage levels are TTL levels (Low < 0,4 V, High > 2 V).

In addition, an internal +5 V power supply voltage is provided. The maximum load current is 100 mA.

The pin assignments for the USER connector can be seen in the following diagram:

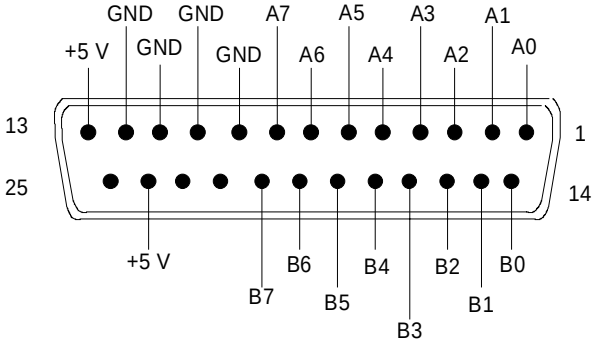
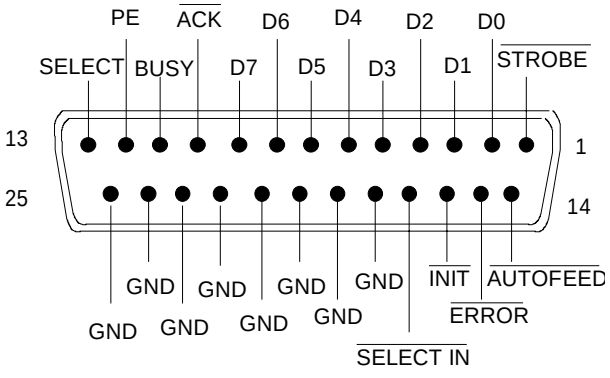


Fig. 8-5 Pin assignments for the USER connector.

The configuration of the user ports takes place in the *SETUP* menu (SETUP key) in the *GENERAL SETUP* sub-menu.

Printer Interface (LPT)

The 25-pin LPT connector on the rear panel of the FSET is provided for the connection of a printer.. The LPT interface is compatible with the CENTRONICS printer interface.



Pin	Signal	Input (I) Output (O)	Description
1	STROBE	O	Pulse for transmitting a data byte, min. 1µs pulse width (active LOW)
2	D0	O	Data Line 0
3	D1	O	Data Line 1
4	D2	O	Data Line 2
5	D3	O	Data Line 3
6	D4	O	Data Line 4
7	D5	O	Data Line 5
8	D6	O	Data Line 6
9	D7	O	Data Line 7
10	ACK	I	Indicates that the printer is ready to receive the next byte. (active LOW)
11	BUSY	I	Signal is active when the printer cannot accept data. (active HIGH)
12	PE	I	Signal is active when the paper tray is empty. (active HIGH)
13	SELECT	I	Signal is active when the printer is selected. (active HIGH)
14	AUTOFEED	O	When signal is active, the printer automatically performs a linefeed after each line. (active LOW)
15	ERROR	I	This signal is high when the printer has no paper, is not selected or has an error status. (active LOW)
16	INIT	O	Initialise the printer. (active LOW)
17	SELECT IN	O	If signal is active, the codes DC1/DC3 are ignored by the printer. (active LOW).
18 - 25	GND		Ground connection.

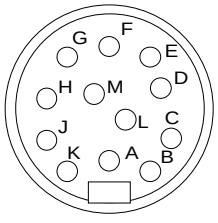
Fig. 8-6 Pin assignments for the LPT connector.

Measurement Converters (PROBE CODE)

The PROBE CODE connector is used for supplying power to measurement converters and the providing the correct conversion factor coding to the FSET. Using it, the conversion factors for high-impedance probes, current converters and antennas can be encoded in 10dB steps. In addition, the quantity to be measured (field strength, current and voltage) is also passed to the FSET. Active converters can be supplied with ±10 V via the PROBE CODE connector. The following R&S accessories are deliverable with the applicable coding:

- · Wide band dipole 20...80 MHz HUF-Z2
- · HF current converter 100 kHz..30 MHz ESH2-Z1
- · VHF current converter 20...300 MHz ESV-Z1
- · Current converter 20 Hz...100 MHz EZ-17
- · Preamplifier 20...1000 MHz ESV-Z2.

The pins of the PROBE CODE connector are assigned as follows:



Pin	Signal
A	ground
B	+10 V, max. 50 mA
C	µV/m (elec. field strength)
D	µA
E	10 dB
F	20 dB
G	40 dB
H	80 dB
K	- 10 V, max. 50 mA
M	factor sign inversion

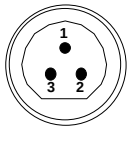
Fig 8-7 Pin assignment for the 12-pin Tuchel connector.

A 12-pin connector is used for coding (manufacturer: Tuchel, R&S part number 0018.5362.00, Tuchel type number: T3635/2). The input pins for implementing the code are connected to ground.

Example: An antenna for measurement of the electromagnetic field strength has an antenna factor of 10 dB, i.e., a field strength of 10 dBµV/m produces a voltage at the RF input of 0 dBµV.
-> Pins C and E are connected to ground.

Probe Connector (PROBE POWER)

To allow the connection of probes, the FSET provides the PROBE POWER power connector. It delivers the power supply voltages +15 V and -12,6 V and ground.
The connector is also suited for powering the high-impedance probes from Hewlett Packard.



Pin	Signal
1	GND
2	-12,6 V; max 150 mA
3	+15 V; max 150 mA

AF-Output (AF OUTPUT)

A miniature telephone jack can be used at the AF OUTPUT connector to connect an external loudspeaker, a headphone set or, e.g., a LF voltmeter. The internal resistance is 10 ohms and the output voltage can be controlled with the volume controller. When a jack is plugged in, the internal loudspeaker is automatically turned off.

IF Output 21.4 MHz (21.4 MHz OUT)

The 21,4 MHz IF signal of the FSET is available at the IF 21.4 MHz OUT BNC connector. The bandwidth corresponds to the selected bandwidth for a resolution bandwidth between 2 kHz and 10 MHz. For a resolution bandwidth below 2 kHz, the bandwidth of the output is 5 kHz.
The signal level at the IF output is 0 dBm for signals which correspond to the selected reference level.

Video Output (VIDEO OUT)

the video output delivers a signal which corresponds to the displayed video signal. The level scaling LIN or LOG is adapted to the display in the signal analysis mode and to the selected detector (LIN with average-, AC video- and RMS-detector, LOG with peak detector) in the receiver mode. The level range is always 100 dB.

Reference Output/Input (EXT REF IN/OUT)

When the FSET is operated with the internal reference, the internal 10 MHz reference signal is also available at the REF IN/OUT connector and thus provides the capability of, e.g., synchronisation of external instruments to the FSET. The level is 1 V RMS at 50 ohms source impedance.
For operation with an external reference, this connector becomes an input connector. The internal reference oscillator is then synchronised to the reference applied to the connector. The reference frequency can be between 1 and 16 MHz in 1 MHz steps. The necessary level is > 0 dBm.
The selection of internal or external references takes place in the *SETUP* menu.

Sweep Output (SWEEP)

The SWEEP BNC connector delivers a voltage of 5 V during the active sweep. At the start of the sweep the voltage switches from 0 V to 5 V, at the end of the sweep it switches from 5 V to 0 V.

External Trigger Input (EXT TRIGGER/GATE)

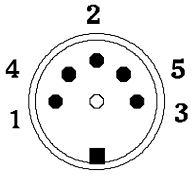
The EXT TRIG/GATE connector is used for controlling the measurement via an external signal.
Trigger voltage range:: -5 V ... +5 V

Noise Source Control (NOISE SOURCE)

Using the NOISE SOURCE connector, an external noise source can be switched on/off, in order, e.g., to measure the noise figure of Units Under Test (UUTs). Usual noise sources require a +28 V signal to be turned on. At 0 V, they are turned off. These switching voltages are delivered by the connector.

External Keyboard (KEYBOARD)

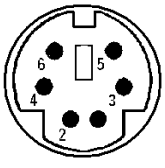
A 5-pin DIN connector is provided to allow connecting an external keyboard. Because of its low interference radiation, the PSA-Z1 keyboard is recommended (Order No. 1009.5001.31). However, any other multi-function keyboard may also be used.



Pin	Signal
1	Keyboard Clock
2	Data
3	Free
4	Ground
5	+5-V-Power Supply

Fig. 8-8 Pin assignments of the KEYBOARD connector.

Mouse Connector (MOUSE)



Pin	Signal
1	MOUSEDATA
2	NC
3	MOUSEGND
4	MOUSEVD5
5	MOUSECLK
6	NC

Fig 8-9 Pin assignments for the MOUSE connector.

Monitor Connector (MONITOR)

Pin	Signal	Pin	Signal
1	R	9	NC
2	G	10	GND
3	B	11	MID0 (NC)
4	MID2 (NC)	12	MID1 (NC)
5	NC	13	HSYNC
6	R-GND	14	VSYNC
7	G-GND	15	NC
8	B-GND		

Fig. 8-10 Pin assignments of the MONITOR connector.

External Video Filter Input/Output (VIDEO FILTER EXT IN/OUT)

With these connectors the user can loop in his own video filters into the video signal path in front of the pulse stretcher. The level scaling at the connectors is suitable for bandwidths from 2 kHz to 10 MHz RBW, the logarithmic envelope of the IF signal is independent of the display level scaling (linear or logarithmic). For bandwidths from 20 MHz to 500 MHz RBW the connectors provide video signals corresponding to the display level scaling (LIN/LOG). The maximum bandwidth of the video signal corresponds to the set video bandwidth.

Unstretched Video Output (VIDEO OUT UNSTRETCHED)

The video output unstretched provides the logarithmic envelope of the IF signal for bandwidths from 2 kHz to 10 MHz RBW, independent of the display level scaling (linear or logarithmic). The bandwidth of the video signal always corresponds to half the IF bandwidth and is not limited by the video filter in the measurement path. For bandwidths from 20 MHz to 500 MHz RBW the video output provides a video signal corresponding to the display level scaling (LIN/LOG). The bandwidth of the video signal corresponds to the set video bandwidth.

IF Output 741.4 MHz (741.4 MHz OUT)

The IF signal 741.4 MHz of FSET is provided at the BNC connector 741.4 MHz OUT. For resolution bandwidths between 20 MHz and 500 MHz the bandwidth corresponds to the selected bandwidth. For resolution bandwidths below 20 MHz the bandwidth of the output is 500 MHz.

The level at the IF output is 5 dBm for signals corresponding to the set reference level in the range from -60 dBm to + 30 dBm.

20-MHz Reference Output (REF 20 MHz OUT)

A reference to operate the self-test generator of RF Preselector FSET-Z.. is provided at this connector. The level is approx. 3 V. The signal is provided if the harmonic generator is switched on.

Preselector Control

The control signal to operate the RF Preselector FSET-Z.. are provided at this connector.

Demodulation Output (AM/FM OUT)

The demodulated AF signals are provided at this connector. Scaling corresponds to the set measurement range. The demodulated AF signal is also present at this connector with the RF level shown on the internal display.

Contents - Chapter 9 "Error Messages"

9 List of Error Messages

SCPI-Specific Error Messages 9.1
Command Error - Faulty command; sets bit 5 in the ESR register. 9.1
Execution Error - Error on execution of a command; sets bit 4 in the ESR register 9.4
Device Specific Error; sets bit 3 in the ESR register 9.7
Query Error - Error in data request; sets bit 2 in the ESR register 9.7

9 List of Error Messages

The following list contains the error messages for errors occurring in the instrument. The meaning of negative error codes is defined in SCPI, positive error codes mark errors specific of the instrument.

Error messages are entered in the error/event queue of the status reporting system in the remote control mode and can be queried with the command `SYSTem:ERRor?`. The answer format of FSET to the command is as follows:

<error code>, "<error text with queue query>; <remote control command concerned>"

The indication of the remote control command with prefixed semicolon is optional.

Example:

The command "`TEST:COMMAND`" generates the following answer to the query `SYSTem:ERRor? :`

-113,"Undefined header;TEST:COMMAND"

The table contains the error code in the left-hand column. In the right-hand column the error text being entered into the error/event queue or being displayed is printed in bold face. Below the error text, there is an explanation as to the respective error.

SCPI-Specific Error Messages

No Error

Error code	**Error text in the case of queue poll** Error explanation
0	**No error** This message is output if the error queue does not contain any entries.

Command Error - Faulty command; sets bit 5 in the ESR register.

Error code	**Error text in the case of queue poll** Error explanation
-100	**Command Error** The command is faulty or invalid.
-101	**Invalid Character** The command contains an invalid sign. Example: A header contains an ampersand, "**SENSe&**".
-102	**Syntax error** The command is invalid. Example: The command contains block data the instrument does not accept.
-103	**Invalid separator** The command contains an impermissible sign instead of a separator. Example: A semicolon is missing after the command.

Continuation: Command Error

Error code	Error text in the case of queue poll Error explanation
-104	**Data type error** The command contains an invalid value indication. Example: ON is indicated instead of a numeric value for frequency setting.
-105	**GET not allowed** A Group Execute Trigger (GET) is within a command line.
-108	**Parameter not allowed** The command contains too many parameters. Example: Command **SENSe:FREQuency:CENTer** permits only one frequency indication.
-109	**Missing parameter** The command contains too few parameters. Example: The command **SENSe:FREQuency:CENTer** requires a frequency indication.
-110	**Command header error** The header of the command is faulty.
-111	**Header separator error** The header contains an impermissible separator. Example: the header is not followed by a "White Space", "***ESE255**"
-112	**Program mnemonic too long** The header contains more than 12 characters.
-113	**Undefined header** The header is not defined for the instrument. Example: ***XYZ** is undefined for every instrument.
-114	**Header suffix out of range** The header contains an impermissible numeric suffix. Example: **SENSe3** does not exist in the instrument.
-120	**Numeric data error** The command contains a faulty numeric parameter.
-121	**Invalid character in number** A number contains an invalid character. Example: An "A" in a decimal number or a "9" in an octal number.
-123	**Exponent too large** The absolute value of the exponent is greater than 32000.
-124	**Too many digits** The number includes too many digits.
-128	**Numeric data not allowed** The command includes a number which is not allowed at this position. Example: The command **INPut:COUPling** requires indication of a text parameter.
-130	**Suffix error** The command contains a faulty suffix.
-131	**Invalid suffix** The suffix is invalid for this instrument. Example: nHz is not defined.
-134	**Suffix too long** The suffix contains more than 12 characters.

Continuation: Command Error

Error code	Error text in the case of queue poll Error explanation
-138	**Suffix not allowed** A suffix is not allowed for this command or at this position of the command. Example: The command ***RCL** does not permit a suffix to be indicated.
-140	**Character data error** The command contains a faulty text parameter
-141	**Invalid character data** The text parameter either contains an invalid character or it is invalid for this command. Example: Write error with parameter indication;**INPut:COUPling XC.**
-144	**Character data too long** The text parameter contains more than 12 characters.
-148	**Character data not allowed** The text parameter is not allowed for this command or at this position of the command. Example: The command ***RCL** requires a number to be indicated.
-150	**String data error** The command contains a faulty string.
-151	**Invalid string data** The command contains a faulty string. Example: An END message has been received prior to the terminating apostrophe.
-158	**String data not allowed** The command contains a valid string at a position which is not allowed. Example: A text parameter is set in quotation marks, **INPut:COUPling "DC"**
-160	**Block data error** The command contains faulty block data.
-161	**Invalid block data** The command contains faulty block data. Example: An END message was received prior to reception of the expected number of data.
-168	**Block data not allowed** The command contains valid block data at an impermissible position. Example: The command ***RCL** requires a number to be indicated.
-170	**Expression error** The command contains an invalid mathematical expression.
-171	**Invalid expression** The command contains an invalid mathematical expression. Example: The expression contains mismatching parentheses.
-178	**Expression data not allowed** The command contains a mathematical expression at an impermissible position.
-180	**Macro error** A faulty macro has been defined, or an error has occurred during execution of a macro.
-181	**Invalid outside macro definition** A macro parameter placeholder was encountered outside of a macro definition.
-183	**Invalid inside macro definition** A macro definition is syntactically wrong**.**
-184	**Macro parameter error** A command inside the macro definition has the wrong number or type of parameters.

Execution Error - Error on execution of a command; sets bit 4 in the ESR register

Error code	Error text in the case of queue poll Error explanation
-200	**Execution error** Error on execution of the command.
-201	**Invalid while in local** The command is not executable while the device is in local due to a hard local control. Example: The device receives a command which would change the rotary knob state, but the device is in local so the command can not be executed.
-202	**Settings lost due to rtl** A setting associated with hard local control was lost when the device changed to LOCS from REMS or to LWLS from RWLS.
-210	**Trigger error** Error on triggering the device.
-211	**Trigger ignored** The trigger (GET, *TRG or trigger signal) was ignored because of device timing considerations. Example: The device was not ready to respond.
-212	**Arm ignored** An arming signal was ignored by the device.
-213	**Init ignored** Measurement initialisation was ignored as another measurement was already in progress.
-214	**Trigger deadlock** The trigger source for the initiation of measurement is set to GET and subsequent measurement is received. The measurement cannot be started until a GET is received, but the GET would cause an interrupted-error)
-215	**Arm deadlock** The trigger source for the initiation of measurement is set to GET and subsequent measurement is received. The measurement cannot be started until a GET is received, but the GET would cause an interrupted-error.
-220	**Parameter error** The command contains a faulty or invalid parameter.
-221	**Settings conflict** There is a conflict between setting of parameter value and instrument state.
-222	**Data out of range** The parameter value lies out of the permissible range of the instrument.
-223	**Too much data** The command contains too many data. Example: The instrument does not have sufficient storage space.
-224	**Illegal parameter value** The parameter value is invalid. Example: The text parameter is invalid , **`TRIGger:SWEep:SOURce TASTe`**

Continuation: Execution Error

Error code	Error text in the case of queue poll Error explanation
-230	**Data corrupt or stale** The data are incomplete or invalid. Example: The instrument has aborted a measurement.
-231	**Data questionable** The measurement accuracy is suspect.
-240	**Hardware error** The command cannot be executed due to problems with the instrument hardware.
-241	**Hardware missing** Hardware is missing. Example: An option is not fitted.
-250	**Mass storage error** A mass storage error occured.
-251	**Missing mass storage** The mass storage is missing. Example: An option is not installed.
-252	**Missing media** The media is missing. Example: There is no floppy in the floppy disk drive.
-253	**Corrupt media** The media is corrupt. Example: The floppy is bad or has the wrong format.
-254	**Media full** The media is full. Example: There is no room on the floppy.
-255	**Directory full** The media directory is full.
-256	**File name not found** The file name cannot be found on the media.
-257	**File name error** The file name is wrong. Example: An attempt is made to copy to a duplicate file name.
-258	**Media protected** The media is protected. Example: The write-protect tab on the floppy is present.
-260	**Expression error** The expression contains an error.
-261	**Math error in expression** The expression contains a math error. Example: Divide-by-zero.

Continuation: Execution Error

Error code	Error text in the case of queue poll Error explanation
-270	**Macro error** Error on the execution of a macro.
-271	**Macro syntax error** The macro definition contains a syntax error.
-272	**Macro execution error** The macro definition contains an error.
-273	**illegal macro label** An illegal macro label is defined in the *DMC command. Example: The label is too long. The label is identical with the common command header or contains an invalid header syntax.
-274	**Macro parameter error** The macro definition improperly uses a macro parameter placeholder.
-275	**Macro definition too long** The macro definition is too long.
-276	**Macro recursion error** The command sequence defined by the macro is trapped in a program loop. Example: The event that would allow the loop to be exited does not occur.
-277	**Macro redefinition not allowed** The macro label defined in the *DMC command is already defined elsewhere.
-278	**Macro header not found** The macro label in the *GMC? query is not yet defined.
-280	**Program error** Error on the execution of a down-loaded program.
-281	**Cannot create program** The program cannot be created.
-282	**illegal program name** The name of the programm is illegal. Example: The name relates to a non-existing program.
-283	**illegal variable name** The inputted variable does not exist in the program.
-284	**Program currently running** The desired operation is not possible while the program is running. Example: A running program cannot be deleted.
-285	**Program syntax error** The down-loaded program contains a syntax error.
-286	**Program runtime error**

Device Specific Error; sets bit 3 in the ESR register

Error code	Error test in the case of queue poll Error explanation
-300	**Device-specific error** FSET-specific error not defined in greater detail.
-310	**System error** This error message suggests an error within the instrument. Please inform the R&S Service.
-311	**Memory error** Error in the instrument memory.
-312	**PUD memory lost** Loss of the protected user data stored using the *PUD command.
-313	**Calibration memory lost** Loss of the non-volatile calibration data stored using the *CAL? command.
-314	**Save/recall memory lost** Loss of the non-volatile data stored using the *SAV command.
-315	**Configuration memory lost** Loss of the non-volatile configuration data stored by the instrument.
-330	**Self-test failed** The selftest could not be executed.
-350	**Queue overflow** This error code is entered in the queue instead of the actual error code if the queue is full. It indicates that an error has occurred but not been accepted. The queue can accept 5 entries.

Query Error - Error in data request; sets bit 2 in the ESR register

Error code	Error text in the case of queue poll Error explanation
-400	**Query error** General error occurring when data are requested by a query.
-410	**Query INTERRUPTED** The query has been interrupted. Example: After a query, the instrument receives new data before the response has been sent completely.
-420	**Query UNTERMINATED** The query is incomplete. Example: The instrument is addressed as a talker and receives incomplete data.
-430	**Query DEADLOCKED** The query cannot be processed. Example: The input and output buffers are full, the instrument cannot continue operation.
-440	**Query UNTERMINATED after indefinite response** A query is in the same command line after a query which requests an indefinite response.

10 Index

Note:

- *The softkeys are listed alphabetically under the keyword "Softkey".*
- *For each softkey, the page in chapter 6 containing the description of the corresponding remote command is quoted in addition.*
- *The assignment between IEEE-bus commands and softkeys is described in Chapter 6, Section "Table of Softkeys with IEC/IEEE-Bus Command Assignment".*
- *Chapter 6 contains an alphabetical list of all IEEE-bus commands*

0 ... 9

0 DB MIN....4.189, *4.223*
16QAM....*4.276*
2FSK....*4.276*
4FSK....*4.276*
8PSK....*4.276*

A

Abort
hardcopy....*4.61*
macro....*4.59*
AC video detector....*4.154, 4.192*
Addressed command....*8.5*
ADJUST AXIS....*4.202*
Administrator identification....*1.41*
AF
demodulation....*4.20, 4.21, 4.265*
filter....*4.197, 4.270*
AF LOWPASS MANUAL....*4.197, 4.271*
Alpha (roll-off factor)....*4.280, 4.282*
Alphanumeric parameter
editing....*3.18*
Amplitude droop (sum error)....*4.297*
Analog trace....*4.158*
Analysis bandwidth....*4.303*
Ascii #....*5.14*
Attenuation....*4.104*
Attenuator....*4.188*
AUI connector....*1.62*
Autopeak detector....*4.155*
Average....*4.149*
carrier power....*4.318*
continuous sweep....*4.150*
single scan....*4.256*
single sweep....*4.150*
sweep count....*4.150*
Average detector....*4.156*
Average value
display....*4.130*
Averaging....*4.332*
Averaging of summary markers....*4.319*
Axis labelling....*3.8*

B

Bandwidth
analog....*4.334*
IF....*4.334*
occupied....*4.115*
resolution....*4.164*
video....*4.165*
Bandwidth/symbol period product....*4.280, 4.282*
Beeper....*4.57*
Block data....*5.14*
BNC connector....*1.61*
Boolean parameter....*5.13*
BPSK....*4.276*
BT....*4.280, 4.282*
Burst search....*4.338*

C

Calibration....*4.10*
Capture buffer....*4.284, 4.300*
CDPD....*4.277*
CD-ROM installation....*1.54*
Center frequency....*4.90, 4.303*
Characters, special....*6.2*
Clear/Write....*4.148, 4.255*
Clock....*4.57*
Colon....*5.14*
COM1/2 interface....*4.54*
Comma....*5.14*
Command
#....*5.14*
addressed....*8.5*
alphabetical list....*6.155*
colon....*5.14*
comma....*5.14*
header....*5.10*
line....*5.12*
long form....*5.11*
overlapping execution....*5.17*
query....*5.12*
question mark....*5.12, 5.14*
quotation mark....*5.14*
recognition....*5.16*
sequence....*5.17*
short form....*5.11*
structure....*5.9*
suffix....*5.11*
synchronization....*5.17*
syntax elements....*5.14*
univeral....*8.5*
white space....*5.14*
Common commands....*6.4*
CONDition register part....*5.19*

Configuration *4.23*
recall *4.77*
reset *4.2*
save *4.68*
Constellation diagram *4.289*
Copy
file *4.71*
limit line *4.141*
trace *4.152*
Coupling *4.163*
default settings *4.167*
define *4.171*
CT2 ..... *4.277*
Cursor keys *3.15*

D

D Lines *4.135, 4.322*
D8PSK *4.276*
Data set
creation *4.77*
partial *4.75*
recall *4.77*
save *4.72*
Date of internal real-time clock *4.57*
dB/MHz* *4.101*
dBµA/MHz *4.101*
dBµA/mMHz *4.101*
dBµV/MHz *4.101*
dBµV/mMHz *4.101*
dBmV/MHz *4.101*
DCL *5.16*
DCS1800 ..... *4.277*
Decision point *4.290, 4.291*
DECT standard *4.276*
Default
commands *6.1*
instrument settings *4.2*
Default setup
scan *4.199*
Delete
file *4.71*
line of peak list *4.208*
line of transducer table *4.29*
macro *4.84*
message *4.17*
transducer factor/set *4.26*
Delta marker *4.313*
step size *4.122, 4.235*
Demodulation
AF *4.265*
AM *4.195, 4.266*
display *4.268*
FM *4.195, 4.266*
PM *4.196, 4.267*
Demodulator, digital *4.276*
Demodulator,standard settings *4.277*
Detector
AC video *4.154, 4.192*
auto select *4.155*
autopeak *4.155*
average *4.156, 4.191*
max peak *4.155*
min peak *4.155*
RMS *4.156, 4.191*
sample *4.156*
Device
reset (overall) *4.2*
status *4.14*
DIFOVL *3.4*
Digital demodulator *4.276*
Digital modulation methods *4.275*
Digital standard *4.277*
Directory
create *4.71*
rename *4.71*
Disable
front panel *3.20*
keyboard *3.20*
Disk formatting *4.71*
Display
configuration *4.3*
demodulation *4.268*
energy-saving *1.40*
mean value *4.130*
rms value *4.130*
Display mode
full screen *3.9, 4.4*
split screen *3.8, 4.4*
Display range *4.306*
Display width (eye diagram) *4.288, 4.290, 4.291*
DMSK *4.276*
Double dagger *5.14*
DQPSK *4.276*
Driver
printer *1.46*
software *1.57*
Dynamic range *4.305*

E

Electrostatic discharge *1.35*
ENABle register part *5.19*
Energy saving mode *1.40*
Enhancement labels *3.6*
Entry
abortion *3.17*
alphanumeric parameters *3.18*
mouse control *3.23*
numeric parameter *3.17*
table *3.19*
termination *3.17*
window *3.16*
ERMES standard *4.277*
Error
constellation diagram *4.294*
frequency *4.293*
frequency (FSK) *4.294, 4.295*
magnitude *4.293*
magnitude (FSK) *4.294, 4.295*
magnitude of error vector *4.294*
phase *4.293*
real and imaginary part *4.294*
vector diagram *4.294*
Error messages *9.1*
Error signal *4.293*
Error-queue query *5.32*
ESE (event status enable register) *5.22*
ESR (event status register) *5.22*
Ethernet Adapter *1.61*
EVENt register part *5.19*
Event status enable register (ESE) *5.22*
Event status register (ESR) *5.22*
Ext Trig/Gate input *8.24*
External noise source *4.49*
External triggering *4.337*
ExtRef *3.4*
Eye diagram *4.287, 4.290, 4.291*
Eye length *4.288*

F

FFT-Filter ... 4.169
File
  copy ... 4.71
  delete ... 4.71
  rename ... 4.71
  sort ... 4.71
Filter
  input ... 4.280
  reference ... 4.280
Final results ... 4.256
Find burst ... 4.338
Firmware
  options (enabling) ... 4.48
  options (info) ... 4.15
  update ... 1.56, 4.58
  version ... 4.14
FLEX ... 4.277
Frame length ... 4.301
Free-running trigger ... 4.336
Frequency
  axis labelling ... 3.8
  carrier ... 4.303
  counter ... 4.114
  line ... 4.134, 4.243
  measurement window ... 4.86, 4.217
  offset ... 4.91, 4.303
  receiver ... 4.188
  setting accuracy ... 4.303
  start ... 4.86
  stop ... 4.88
  zoom ... 4.96
Frequency error ... 4.293
Frequency error (FSK) ... 4.294, 4.295, 4.299
Frequency error (sum error) ... 4.297
Frequency setting ... 4.303
Frequency-demodulated signal ... 4.286, 4.290, 4.291
Front panel
  disable ... 3.20
  keyboard emulation ... 3.22, 3.24
FSK ... 4.277
  deviation ... 4.280, 4.282, 4.299
FSK deviation error ... 4.299
FTP operation ... 1.72
Full screen ... 3.9, 4.4
Function test ... 1.40
Fuse ... 1.37

G

GET (Group Execute Trigger) ... 5.16
GPIB
  address ... 4.52
  interface ... 8.2
Group mid-points ... 4.283
GSM ... 4.277

H

Handshake ... 8.8
Hardcopy
  abort ... 4.61
  comments ... 4.65
  format ... 4.67
  items ... 4.63
  position ... 4.64
  printer ... 4.66
  settings ... 4.62
  start ... 4.60
Hardware
  configuration ... 4.15
  indication of settings ... 3.5
  installed options ... 4.15
Header ... 5.10
Help line editor ... 3.18

I

I/Q diagram ... 4.289
I/Q imbalance (sum error) ... 4.298
I/Q offset (sum error) ... 4.297
IEC/IEEE-bus
  address ... 4.52
  interface ... 8.2
  interface functions ... 8.4
IF bandwidth ... 4.334
IFOVLD ... 3.4
Indication
  hardware settings ... 3.5
  instrument settings ... 3.6
  marker information ... 3.5
  screen ... 3.4
Inphase component ... 4.289
Input
  attenuator ... 4.104, 4.188
  Ext Trig/Gate ... 8.24
  external reference ... 8.24
Input buffer ... 5.15
Input filter ... 4.280
Installation
  CD-ROM ... 1.54
  firmware update ... 1.56, 4.58
  keyboard ... 1.43
  monitor ... 1.44
  mouse ... 1.42
  network printer ... 1.52
  printer ... 1.46
  rack ... 1.36
  Windows NT software ... 1.57
Instrument functions ... 4.1
Instrument settings
  indication ... 3.6
  power up ... 4.78
  preset ... 4.2, 4.78
Interface
  IEC/IEEE-bus ... 8.2
  printer ... 8.22
  RS-232-C ... 8.6
  RSIB ... 8.10
Interrupt ... 5.31
IST flag ... 5.22

K

Key
  CAL ... 4.10
  CENTER ... 4.90, 4.217, 4.303
  CONFIG ... 4.70
  COUPLING ... 4.163, 4.263, 4.334
  cursor ... 3.15
  FREQ ... 4.90, 4.217, 4.303
  D LINES ... 4.134, 4.243, 4.321
  DELTA ... 4.118, 4.232, 4.313
  DISPLAY ... 4.3
  HOLD ... 3.20
  INFO ... 4.14
  INPUT ... 4.104, 4.222

LIMITS 4.138, 4.245, 4.324
LOCAL 4.59
MENU 3.13
MKR 4.132, 4.241
MKR -> 4.320
MODE 4.20, 4.273
NORMAL 4.109, 4.227, 4.310
numeric keypad 3.14
PRESET 4.2, 6.147
RANGE 4.102, 4.221
RECALL 4.77
REF 4.98, 4.219, 4.305
roll-key 3.15
RUN 4.163, 4.263, 4.334, 6.74
SAVE 4.72
SCAN 4.175, 4.263, 4.335
SEARCH 4.123, 4.236
SETTINGS 4.62
SETUP 4.23
SPAN 4.94
START (frequency) 4.86, 4.217
START (hardcopy) 4.60, 6.71
STEP 3.21
STOP 4.88, 4.217
SWEEP 4.175, 4.263, 4.335
TRACE 4.331
TRACE1 to 4 ..... 4.147, 4.253
TRIGGER 4.173, 4.264, 4.336
UNIT 4.98, 4.219, 4.305
USER 4.82
ZOOM 4.94
Keyboard
connection 1.43
connector 8.25
disable 3.20
external 3.22

L

Level
attenuation 4.104
line 4.136, 4.244, 4.322
maximum 4.99
mixer 4.106
range 4.102, 4.221
reference 4.98
unit 4.100, 4.219
Limit line 4.138, 4.323
copy 4.141, 4.248, 4.326
delete 4.141, 4.248, 4.326
edit 4.142, 4.249
save 4.146, 4.252, 4.330
select 4.139, 4.247, 4.324
shift 4.146, 4.252
value 4.145, 4.252, 4.330
Line
display 1,2 ..... 4.136, 4.244, 4.322
frequency 1, 2 ..... 4.136, 4.244
limit 4.139, 4.247, 4.323
reference 4.136, 4.244, 4.322
threshold 4.136, 4.244, 4.322
time 1, 2 ..... 4.137, 4.322
LO LvD 3.4
LO Lvl 3.4
LO unl 3.4
Login/logout (NT controller) 1.41
Lower case 6.1
LPT interface 8.22

M

Macro
abort 4.59
definition 4.83
start 4.82
Magnitude peak 4.317
Magnitude (capture buffer) 4.285
Magnitude (measurement signal) 4.286
Magnitude (reference signal) 4.286
Magnitude error 4.293, 4.294, 4.295
Magnitude error (FSK) 4.299
Magnitude error (sum error) 4.297
Maintenance 8.1
Manual operation 3.1
return to 4.59, 5.4
Marker 4.310
center 4.132
coupled 4.311
delta 4.313
indication 3.5
info 4.113, 4.230
Marker -> 4.320
n-dB-down 4.128
normal 4.109, 4.227, 4.310
peak 4.124, 4.132, 4.237, 4.241
phase indication 4.311
polar diagram 4.311
search 4.314
search limit 4.127, 4.240, 4.315
signal track 4.112
step size 4.117, 4.231
zoom 4.112
Max hold 4.151, 4.255, 4.333
Max peak detector 4.155
Mean 4.318
Mean power (GSM burst) 4.130
Measurement
recall 4.77
save 4.68
window 4.3
Measurement (digital demodulation) 4.284
Measurement converters 4.23
connector 8.22, 8.23
Measurement curve
minimum-value averaging 4.333
peak-value averaging 4.333
Measurement filter 4.280
Measurement signal
phase 4.286
Measurement signal (digital demodulation) 4.285
Memory
battery-backed-up 1.40
configuration 4.70
Memory size 4.300
Menu switching 3.12
MICROSOFT NET operation 1.66
Min hold 4.151, 4.256, 4.333
Min peak detector 4.155
Minimum
search 4.125, 4.238
Minimum-value averaging 4.333
Mixer level 4.106
MODACOM 4.277
Mode 4.20
signal analysis 4.85, 4.86
vector signal analysis 4.272
Modification level of modules 4.15
Modulation error 4.292
Modulation parameters 4.279
Monitor
connecting 4.57

connection ... 1.44
connector ... 8.25
Mouse
connection ... 1.42
connector ... 8.25
control ... 3.23
MSK ... 4.276

N

NADC (IS54) ... 4.277
Network operating system
FTP ... 1.72
MICROSOFT NET ... 1.66
NOVELL NETWARE ... 1.66
TCP/IP ... 1.71
Network printer connection ... 1.52
Noise power density measurement ... 4.115
Noise source, external ... 4.49
Normalization ... 4.283
Notch filter ... 4.170
NOVELL NETWARE operation ... 1.66
NT controller ... 1.41
NTRansition register part ... 5.19
Number of sweeps ... 4.319
Numeric keypad ... 3.14
Numeric parameter
editing ... 3.17
Numerical values (command) ... 5.13
Nyquist filter ... 4.280, 4.282

O

OCXO ... 3.4
Offset frequency ... 4.91
Operating mode
AF demodulation ... 4.195
receiver ... 4.187
signal analysis ... 4.86
Option
FSE-B16 - Ethernet Adapter ... 1.61
FSE-B17 - Second IEC/IEEE bus interface ... 1.58
OQPSK ... 4.276
Output
AF ... 8.22, 8.24
buffer ... 5.17
IF ... 8.24
LOG VIDEO OUT ... 8.24
noise source control ... 8.24
reference ... 8.24
sweep ... 8.24
video out ... 8.24
Output formats (digital demodulation) ... 4.285
OVLD ... 3.4

P

Parallel poll ... 5.32
Parallel poll enable register (PPE) ... 5.22
Parameter
block data ... 5.14
boolean ... 5.13
numerical values ... 5.13
string ... 5.14
text ... 5.14
Password
service function ... 4.51
Windows NT ... 1.41
Path ... 4.70
Pattern ... 4.344
name ... 4.345
selection ... 4.344
PCS1900 ... 4.277
PDC ... 4.277
Peak ... 4.317
detector ... 4.153, 4.257
excursion ... 4.125, 4.238
hold ... 4.318
search ... 4.124, 4.237
value averaging ... 4.333
Phase shift ... 4.286
Phase error ... 4.293
Phase error (sum error) ... 4.297
Phase of the measurement ... 4.286
Phase wrap ... 4.286
PHS ... 4.277
Point of decision ... 4.302
Points of decision (highlighting) ... 4.290
Points per symbol ... 4.302
Polar diagram ... 4.289
marker ... 4.311
Polar marker ... 4.311
Power mean ... 4.130
Power measurement
bandwidth, occupied ... 4.116
PPE (parallel poll enable register) ... 5.22
PREAMP ON/OFF ... 4.190, 4.223
Preamplification ... 4.33
Preamplifier ... 4.34
amplifier setting ... 4.34
Preselection ... 4.33
Preselector FSET-Zx ... 4.35
consistency check ... 4.44
copy ... 4.38
coupled to RBW ... 4.37
delete ... 4.39
direct mode ... 4.36
display ... 4.37, 4.43
edit ... 4.40, 4.41, 4.42, 4.43
error messages ... 4.45
filter ... 4.44
New Set ... 4.40
RBW Coupled Setting ... 4.39
selection ... 4.36
set SET1NARR ... 4.47
set SET1WIDE ... 4.47
set SET2NORM ... 4.47
standard preselector sets ... 4.46
standard set NARROW ... 4.46
standard set NORMAL ... 4.46
standard set WIDE ... 4.46
warning ... 4.45
Preset ... 4.2
Pre-trigger ... 4.185
Printer connection ... 1.46, 8.22
Printing
abort ... 4.61
Configuration ... 4.62
start ... 4.60
Probe Code connector ... 8.22, 8.23
Probe Power connector ... 8.22, 8.23
PTRansition register part ... 5.19
Pulse stretcher ... 4.167, 4.168

Q

QCDMA 4.277
QPSK 4.276
Quadrature component 4.289
Quasi analog display 4.158
Query 5.12, 5.32
Question mark 5.12, 5.14
Quotation mark 5.14

R

Rack installation 1.36
Range 4.306
  level 4.102, 4.221
Real-time clock 4.57
Recall data set 4.77
Receiver frequency 4.21, 4.187, 4.188, 4.217
Receiver 4.21, 4.187
Recording the correction data 4.10
Reference
  deviation 4.280, 4.282
  external 4.48
  filter 4.280
  level 4.304
  level offset 4.306
  line 4.136, 4.244
  position 4.308
  signal (digital demodulation) 4.285
  value 4.304
  value (position) 4.308
  value X/Y axis 4.307
Reference deviation 4.296
Reference level 4.98
  offset 4.99
Reference line 4.322
Reference signal (digital demodulator) 4.280, 4.281
Remote control
  basics 5.1
  IEC-bus 5.4
  indication 4.59, 5.3
  RS-232-C 5.5
  RSIB 5.6, 8.10
  switch over 5.3
Rename
  directory 4.71
  file 4.71
Reset
  device 4.2
  status reporting system 5.33
Resolution bandwidth 4.164, 4.334
Result length 4.301, 4.335
RF attenuation 4.104, 4.188
  auto 4.106
  auto low distortion 4.106
  auto low noise 4.106
RF input 4.104
  configuration 4.309
RF Input 4.222
Rho factor (sum error) 4.298
RJ45 (star topology) 1.62
RMS detector 4.156, 4.191
Rms value 4.318
  display 4.130
Roll-key 3.15
Roll-off factor 4.280, 4.282
RS-232-C
  interface 8.6
  interface functions 8.7
  transmission parameters 8.7
RSIB interface 8.10

S

Sample detector 4.156
Save
  configuration 4.68
  data set 4.72
  limit line 4.146, 4.252, 4.330
  measurement 4.68
Scale unit 4.308
Scaling
  vertical 4.306
Scan 4.198
  default setup 4.199
  editing 4.203
  entry 4.199
  run 4.203
  count 4.256
SCPI
  conformance information 6.1
  introduction 5.9
Screen 3.2, 4.4, 4.5
  full screen 3.9
  indications 3.4
  split screen 3.9
Search
  limit 4.127, 4.240
  minimum 4.125, 4.238
  peak 4.124, 4.237
Self test 4.16
Serial poll 5.31
Service functions 4.49
Service pack 1.57
Service request (SRQ) 5.21, 5.31
  indication 4.59
Service request enable register (SRE) 5.21
SETTINGS COUPLED 4.238, 4.242
Setup 4.23
  general 4.52
Softkey
  % POWER BANDWIDTH 4.115, 6.124
  + PEAK 4.317, 6.40
  ±PEAK/2 ..... 4.318, 6.42
  0 DB MIN 4.189, 4.223, 6.76
  AC VIDEO 4.193, 4.258, 6.108, 6.109
  AC VIDEO COUPLING 4.167, 4.169, 6.118
  ACTIVE MKR / DELTA 4.124, 4.237, 4.315, 4.320
  ACTIVE SCREEN A/B 4.4, 4.5
  ADD TO PEAK LIST 4.133, 4.242
  ADJUST AXIS 4.202
  ADJUST TO TRACE 4.158, 4.159
  AF BW COUPLED 4.267, 6.100
  AF DC 4.197, 4.271, 6.107
  AF DEMOD 4.195, 4.266, 6.106
  AF DEMOD BANDWIDTH 4.267, 6.100
  AF DISPLAY 4.268, 6.58
  AF FILTER 4.197, 4.270
  AF HIPASS 10Hz /100Hz / 1kHz 4.197, 4.271, 6.107
  AF LOWPASS MANUAL 4.197, 4.271,6.106
  AF LOWPASS AUTO 4.197, 4.271, 6.107
  AF RANGE 4.268
  ALL DELTA OFF 4.119, 4.233, 4.313, 6.9
  ALL MARKER OFF 4.112, 4.230, 4.312, 6.30
  ALL SUM MKR OFF 4.131, 4.319, 6.45
  ALPHA/BT 4.280, 4.282, 6.113
  AM 4.195, 4.266, 6.106
  AM RANGE 3% /10% /100% 4.269, 6.107
  AM VIDEO 4.195, 4.266
  AMPERE 4.102, 6.51
  ANALOG TR ON/OFF 4.158, 6.64
  ANALYZER 4.20, 4.85, 4.86, 6.84
  APPEND NEW 4.160, 4.260, 6.68
  area 3.11

ASCII COMMENT 4.161, 4.260, 6.68
ASCII CONFIG 4.160, 4.259
ASCII EXPORT 4.160, 4.209, 4.259, 6.91
ATT SWITCHES 4.18, 6.55
ATTEN 4.188
ATTEN AUTO LOW DIST 4.106, 6.76
ATTEN AUTO LOW NOISE 4.106, 6.76
ATTEN AUTO NORMAL 4.106, 6.76
ATTEN MANUAL 4.189, 6.76
ATTEN STEP 1dB/10dB 4.108, 6.77
ATTEN UP/DOWN 4.167
AUTO 0.1 * RBW 4.92, 6.120
AUTO 0.1 * SPAN 4.92, 6.120
AUTO 0.5 * RBW 4.93, 6.120
AUTO 0.5 * SPAN 4.93, 6.120
AUTO RANGE ON/OFF 4.189, 4.223, 6.76
AUTO RECALL 4.78, 6.89
AUTO SELECT 4.155, 6.109
AUTO x * RBW 4.93, 6.120
AUTO x * SPAN 4.93, 6.120
AUTOMATIC FINAL 4.212, 4.214
AUTOPREAMP ON/OFF 4.189, 4.223, 6.78
AVERAGE ..... 4.149, 4.193, 4.257, 4.332, ... 6.64, 6.97, 6.108, 6.109
AVERAGE ON/OFF 4.130, 4.319, 6.45
BAL INPUT 10 OHM 4.106, 4.108, 4.226, 6.83
BAL INPUT 150 OHM 4.106, 4.108, 4.226, 6.83
BAL INPUT 600 OHM 4.106, 4.108, 4.226, 6.83
BASELINE CLIPPING 4.137, 6.15
BLANK 4.149, 4.255, 4.331, 6.65
BRIGHTNESS 4.7, 6.58
CAL CORR ON/OFF 4.12, 6.53
CAL I/Q 4.11, 6.52
CAL LO SUPP 4.11, 6.53
CAL LOG 4.11, 6.53
CAL RES BW 4.11, 6.52
CAL RESULTS 4.13
CAL SHORT 4.11, 6.53
CAL TOTAL 4.11, 6.52
CAPT MEM 4.157, 6.108
CENTER FIXED 4.87, 4.88, 4.95, 6.121, 6.122
CENTER FREQUENCY 4.167, 6.119
CENTER MANUAL 4.90
CLEAR ALL MESSAGES 4.17
CLEAR MESSAGE 4.17, 6.147
CLEAR/WRITE 4.148, 4.331, 6.64
CLEAR/WRITE 4.255
COLOR ON/ OFF 4.63, 6.70
COM PORT 1/2 ..... 4.54, 6.144, 6.145
COMMENT (SYNC PATTERN) 4.345, 6.115
COMMENT SCREEN A/B 4.65, 6.72
CONFIG 4.160, 4.259
CONFIG DISPLAY 4.7
CONT AT HOLD 4.204
CONT AT REC FREQ 4.204
CONTINUOUS SCAN 4.202, 6.74
CONTINUOUS SWEEP 4.175, 4.335, 6.74
CONTINUOUS WRITE 4.332, 6.64
COPY 4.71, 6.88
COPY (TRACE) 4.152, 4.258, 6.151
COPY LIMIT LINE 4.141, 4.248, 4.326, 6.27
COPY SCREEN 4.63, 6.71
COPY TABLE 4.63, 6.72
COPY TRACE 4.63, 6.72
COUNTER RESOL 4.114, 6.31
COUPLED MARKER 4.311, 6.32
COUPLED TO RBW 4.37, 6.79
COUPLING CONTROL 4.6, 6.85
COUPLING DEFAULT 4.167, 6.98, 6.129
COUPLING RATIO 4.171
D LINES 4.243
DATA ENTRY FIELD 4.9
DATA SET CLEAR 4.74, 6.91
DATA SET CLEAR ALL 4.74, 6.92
DATA SET LIST 4.73
DATAENTRY OPAQUE 4.9
DATAENTRY X 4.9
DATAENTRY Y 4.9
DATE 4.57, 6.146
dB*/MHz 4.101, 6.51
dBµA 4.101, 4.220, 6.51
dBµA/m 4.220, 6.51
dBµV 4.101, 4.220, 6.51
dBµV/m 4.220, 6.51
dBm 4.101, 4.220, 6.51
dBmV 4.101, 6.51
dBpT 4.220, 6.51
dBpW 4.101, 4.220, 6.51
DECIM SEP 4.160, 4.260, 6.68
DEFAULT COLORS 4.8, 6.57
DEFAULT CONFIG 4.76, 6.94
DEFAULT POSITION 4.9
DEFINE MACRO 4.83
DEFINE PAUSE 4.84
DEFINE SCAN 4.200
DELETE 4.71, 6.88, 6.90
DELETE (peak list) 4.208
DELETE FACTOR/SET 4.26, 6.103, 6.105
DELETE LIMIT LINE 4.141, 4.248, 4.326, 6.27
DELETE LINE 4.42
DELETE MACRO 4.84
DELETE PATTERN 4.344
DELETE RANGE 4.203
DELETE SET 4.39, 6.82
DELETE VALUE (limit line) 4.146, 4.252, 4.330
DELTA 1 to 4 ..... 4.118, 4.232, 6.8, 6.9, 6.10
DELTA 1/2 ..... 4.313, 6.8, 6.9, 6.10
DELTA ABS REL 4.119, 4.313, 4.233, 6.9
DELTA TO STEPSIZE 4.117, 4.231
DEMOD OFF 4.196, 4.267, 6.106
DETECTOR 4.155, 4.192, 4.257
DETECTOR AC VIDEO 4.156, 6.108
DETECTOR AUTOPEAK 4.155, 6.108
DETECTOR AVERAGE 4.156, 6.108
DETECTOR MAX PEAK 4.155, 6.108
DETECTOR MIN PEAK 4.155, 6.108
DETECTOR RMS 4.156, 6.108
DETECTOR SAMPLE 4.156, 6.108
DIGITAL DEMOD 4.276, 6.84, 6.111, 6.112, 6.117
DIGITAL STANDARD 4.277, 6.117
DIR MODE ON/OFF 4.36, 6.78
DISABLE ALL ITEMS 4.76, 4.81, 6.94
DISPLAY COMMENT 4.8, 6.59
DISPLAY LINE 1/2 ..... 4.136, 4.244, 4.322, 6.14
D-LINES 4.322
EDIT COMMENT 4.73, 6.94
EDIT FREQUENCY 4.208
EDIT LIMIT LINE ..... 4.143, 4.250, 4.328, ... 6.21, 6.22, 6.23, 6.24, 6.25, 6.26, 6.27
EDIT NAME 4.73, 4.78, 6.89, 6.91
EDIT PATH 4.70, 4.73, 4.78, 4.160, 4.260, 6.87, 6.90
EDIT PEAK LIST 4.208
EDIT SET 4.40, 6.80
EDIT SYNC PATTERN 4.345, 6.115
EDIT TRD FACTOR 4.27, 6.102
EDIT TRD SET 4.30, 6.104
ENABLE ALL ITEMS 4.76, 4.81, 6.94
ENABLE DEV1 / DEV2 ..... 4.67
ENABLE OPTION 4.48
ENTER PASSWORD 4.51, 6.147
ENTER TEXT 4.65
ERROR SIGNAL 4.293, 6.17
ERROR SIGNAL (FSK demodulation) 4.294, 6.17
ERROR VECT MAGNITUDE 4.294, 6.17

EXCLUDE LO ON/OFF 4.125, 6.32
EXECUTE TESTS 4.16, 6.6
EXT REF FREQUENCY 4.48, 6.125
EXT VIDEO ON / OFF 4.167, 4.168, 6.99
EXTERN (trigger) 4.174, 4.264, 4.337, 6.152
EYE DIAG [FREQ] (FSK demodulation) 4.290, 4.291, 6.18
EYE DIAG [I] 4.287, 6.18
EYE DIAG [Q] 4.287, 6.18
EYE DIAG TRELLIS 4.287, 6.18
EYE LENGTH 4.288, 6.65
EYE LENGTH (FSK demodulation) .. 4.290, 4.291, 6.65
FILTER SET/MAN 4.37, 6.82
FINAL AC VIDEO 4.258, 6.109
FINAL AVERAGE 4.258, 6.109
FINAL MAX PEAK 4.258, 6.109
FINAL MEAS TIME 4.212, 6.130
FINAL MIN PEAK 4.258, 6.109
FINAL RESULTS 4.256, 6.64
FINAL RMS 4.258, 6.109
FIND BURST 4.338, 6.114
FIND SYNC 4.339, 6.115
FIRMWARE UPDATE 4.58
FIRMWARE VERSION 4.14, 6.5
FM 4.195, 4.266, 6.106
FM RANGE AUTO 4.269, 6.107
FM RANGE MANUAL 4.269, 6.107
FORMAT DISK 4.71, 6.88
FRAME LENGTH 4.301, 6.116
FREE RUN 4.173, 4.264, 4.336, 6.152
FREQ AXIS LIN/LOG 4.87, 6.132
FREQ RANGE 2GHZ/22GHZ 4.22, 6.120
FREQUENCY (error) 4.293
FREQUENCY (FSK demodulation) 4.290, 4.291
FREQUENCY (MSK demodulation) 4.286
FREQUENCY LINE 1/2 ..... 4.136, 4.244, 6.16
FREQUENCY OFFSET 4.91, 4.303, 6.122
FREQUENCY ON/OFF 4.8, 6.57
FSK REF DEVIATION 4.280, 4.282, 6.19
FREQUENCY (error) 4.294, 4.295
MAGNITUDE (error) 4.294, 4.295
FULL PAGE 4.64, 6.73
FULL SCREEN 4.4, 6.57
FULL SPAN 4.95, 6.121
GAP LENGTH 4.186, 6.132
GAP SWEEP ON/OFF 4.184, 6.131
GAP SWEEP SETTINGS 4.185
GATE ADJUST 4.181
GATE DELAY 4.180, 6.131
GATE EXTERN 4.181
GATE LENGTH 4.180, 6.131
GATE LEVEL 4.179, 6.130
GATE MODE LEVEL/EDGE 4.179, 6.130
GATE ON / OFF 4.178, 6.130
GATE POL 4.179, 6.131
GATE RF POWER 4.181
GATE SETTINGS 4.179
GENERAL SETUP 4.52
GPIB ADDRESS 4.52, 6.143
GRID ABS/REL 4.99, 4.103, 6.61
GRID MAX LEVEL 4.221
GRID MIN LEVEL 4.221
HARDCOPY DEVICE 4.66, 6.70
HARDWARE+OPTIONS 4.15, 6.5
HARMONIC GENERATOR 4.51, 6.55
HEADER ON/OFF 4.160, 4.260, 6.68
HOLD CONT ON/OFF 4.152, 6.64
HOLD FINAL MEAS 4.214
HOLD SCAN 4.204, 6.7
HORIZONTAL SCALING 4.6, 6.85
IF BW AUTO 4.334, 6.98
IF BW MANUAL 4.334, 6.98
INPUT CAL 4.49, 6.54
INPUT RF 4.49, 6.54
INPUT SELECT 4.106, 4.107, 4.225
INS AFTER RANGE 4.203
INS BEFORE RANGE 4.203
INSERT (peak list) 4.208
INSERT VALUE (limit line) 4.146, 4.252, 4.330
INSERT LINE 4.42
INTERACTIVE 4.213, 4.214
KEY CLICK ON/OFF 4.57
LAST SPAN 4.95
LINE (trigger) 4.173, 6.152
LINEAR/% 4.103, 6.63
LINEAR/dB 4.103, 6.63
LOCK ALL 3.20
LOCK DATA 3.20
LOG 10dB/20dB/50dB/100dB/120dB ..... 4.102, 4.221, ... 6.61, 6.63
LOG MANUAL 4.103, 4.221, 6.61, 6.63
LOGO ON/OFF 4.8, 6.57
LOW RANGE 4.41, 4.44, 6.81, 6.82
LOWER LEFT 4.64, 6.73
LOWER RIGHT 4.64, 6.73
MACRO 1 to 7 ..... 4.83
MACRO TITLE 4.84
MAGNITUDE 4.286, 6.18
MAGNITUDE (error) 4.293
MAGNITUDE (FSK demodulation) 4.290
MAGNITUDE CAP BUFFER 4.285, 6.17
MAIN PLL BANDWIDTH 4.169, 6.100
MAKE DIRECTORY 4.71, 6.89
MARGIN 4.212, 6.49
MARKER 1 to 4 ..... 4.110, 4.228, 6.30, 6.31, 6.32
MARKER 1/2 ..... 4.310, 6.30, 6.31, 6.32
MARKER INFO ..... 4.113, 4.230, 4.312, 6.13, 6.36, ... 6.37, 6.38, 6.39, 6.41, 6.42, 6.43, 6.44, 6.59
MARKER TRACK 4.238, 4.242, 6.32
MARKER ZOOM 4.112, 4.230, 6.36
MAX |PEAK| 4.315, 4.317, 6.10, 6.33, 6.39
MAX HOLD 4.151, 4.255, 4.333, 6.64, 6.97
MAX LEVEL AUTO 4.99, 6.62
MAX LEVEL MANUAL 4.99, 6.62
MAX PEAK 4.192, 4.257, 6.108, 6.109
MEAN 4.130, 4.318, 6.44
MEAS FILTER 4.280, 6.113
MEAS ONLY IF SYNC'D 4.337, 6.116
MEAS RESULT 4.284
MEAS SIGNAL 4.285, 6.17
MEAS TIME 4.194, 6.129
MEASURE 4.214
MEMORY SIZE 4.300, 6.117
MID RANGE 4.41, 4.44, 6.81, 6.82
MIN 4.125, 4.238, 4.315, 6.11, 6.34
MIN HOLD 4.151, 4.256, 4.333, 6.64, 6.97
MIN PEAK 4.192, 4.257, 6.108, 6.109
MIXER LEVEL 4.106, 6.77
MKR -> CENTER 4.132, 6.46
MKR -> CF STEPSIZE 4.133, 4.242, 6.46
MKR -> REF LEVEL 4.133, 6.47
MKR -> START 4.133, 6.46
MKR -> STOP 4.133, 6.46
MKR -> TRACE 4.133, 4.242, 4.320, 6.9, 6.31
MKR TO STEPSIZE 4.117, 4.231, 6.46
MODE COUPLED 4.6, 6.85
MODULATION PARAMETERS 4.279
MONITOR CONNECTED 4.57
MOVE ZOOM START 4.97, 6.60
MOVE ZOOM STOP 4.97, 6.60
MOVE ZOOM WINDOW 4.96, 6.60
N dB DOWN 4.128, 6.35
NAME (limit line) 4.144, 4.251, 4.328, 6.27
NAME (SYNC PATTERN) 4.345, 6.115
NARROW 4.37, 6.80

NEW FACT/SET 4.27, 4.30, 4.40, 6.102, 6.104
NEW LIMIT LINE 4.143, 4.250, 4.328
NEW SYNC PATTERN 4.345, 6.115
NEXT MIN 4.125, 4.238, 6.11, 6.34
NEXT MIN LEFT 4.125, 4.238, 6.11, 6.34
NEXT MIN RIGHT 4.125, 4.238, 6.11, 6.34
NEXT PEAK 4.124, 4.237, 6.10, 6.33
NEXT PEAK LEFT 4.124, 4.237, 6.11, 6.33
NEXT PEAK RIGHT 4.124, 4.237, 6.10, 6.33
NO OF PEAKS 4.212, 6.49
NOISE 4.115, 6.36
NOISE SOURCE 4.49, 6.54
NORMAL 4.37, 6.80
NORMALIZE 4.283, 6.114
OCCUPIED PWR BANDW 4.116, 6.38, 6.39
OPTIONS 4.15, 4.48, 6.5
PEAK 4.124, 4.237, 4.315, 6.10, 6.33
-PEAK 4.317, 6.41
PEAK EXCURSION 4.125, 4.238, 6.35
PEAK HOLD ON/OFF 4.130, 4.318, 6.45
PEAK SEARCH 4.207, 6.49
PEAKS/SUBRANGES 4.212, 6.50
PHASE 4.286, 6.18
PHASE (error) 4.293, 6.18
PHASE NOISE 4.121, 6.13
PHASE WRAP ON/OFF 4.286, 6.18
PM 4.196, 4.267, 6.106
PM RANGE MANUAL 4.269, 6.107
POINTS PER SYMBOL 4.302, 6.113
POLAR [IQ] CONSTELL 4.289, 6.18
POLAR [IQ] CONSTELL (error) 4.294, 6.18
POLAR [IQ] VECTOR 4.289, 6.18
POLAR [IQ] VECTOR (error) 4.294, 6.18
POLAR MARKER DEG/ RAD 4.311, 6.51
POLAR MARKER R/I / MA/PH 4.311, 6.35
POWER MEAS SETTINGS 4.115
PRE TRIGGER (gap sweep) 4.185, 6.131
PREAMP GAIN 4.34, 4.190, 4.224, 6.78
PREAMP ON/OFF 4.33, 4.190, 4.223, 6.78
PREDEFINED COLORS 4.8, 6.58
PRESEL PEAK 4.12, 6.53
PRESELECT ON/OFF 4.33, 6.78
PRESELECT CONFIG 4.36
PREVIOUS ZOOM 4.230
PROBE CODE ON/OFF 4.102, 4.220, 6.154
PULS STRET AUTO/OFF 4.167, 4.168, 6.109
RANGE 4.306
RANGES 1-5/6-10 ..... 4.203
RBWCPLD SETTING 4.39, 6.80
RBW / VBW MANUAL 4.172, 6.99
RBW / VBW NOISE 4.172, 6.99
RBW / VBW PULSE 4.172, 6.99
RBW / VBW SINE [1] 4.171, 6.99
REAL/IMAG PART 4.286, 6.18
REAL/IMAG PART (error) 4.294, 6.18
RECEIVER 4.21, 4.187, 6.84
RECEIVER FREQUENCY 4.188, 6.119, 6.123
RECORD ON/OFF 4.84
REF LEVEL 4.99, 4.305, 6.61
REF LEVEL OFFSET 4.99, 4.306, 6.62
REF LEVEL UP/DOWN 4.167
REF POINT FREQUENCY 4.120, 4.234, 6.12
REF POINT LEVEL 4.120, 4.234, 6.12
REF POINT LVL OFFSET 4.120, 4.234, 6.12
REF POINT TIME 4.120, 6.12
REF VALUE POSITION 4.308, 6.62
REF VALUE X AXIS 4.307, 6.59
REF VALUE Y AXIS 4.307, 6.62
REFERENCE 4.50, 6.125
REFERENCE FILTER 4.280, 6.113
REFERENCE FIXED 4.119, 4.233, 6.12
REFERENCE INT/EXT 4.48, 6.125
REFERENCE LINE 4.136, 4.244, 4.322, 6.15
REFERENCE POINT 4.120, 4.234, 6.12
REFERENCE PROG 4.50, 6.125
REFERENCE SIGNAL 4.285, 6.17
RENAME 4.71, 6.90
RES BW 4.191, 6.98
RES BW AUTO 4.164, 6.98
RES BW MANUAL 4.165, 6.98
RESTORE 4.58
RESULT LENGTH 4.301, 4.335, 6.112
RF ATTEN MANUAL 4.104, 4.222, 6.76
RF INPUT DC COUPLED 4.106, 4.108, 4.226, 6.77
RF POWER (trigger) 4.174, 6.152
RF PRES SWITCH 4.18
RMS 4.130, 4.193, 4.257, 4.318, 6.43, 6.108, 6.109
RUN FINAL MEAS 4.213
RUN SCAN 4.204, 6.74
SATURATION 4.8, 6.58
SAVE LIMIT LINE 4.146, 4.252, 4.330
SAVE PATTERN 4.346
SAVE SET 4.43
SCALE UNIT 4.308
SCAN COUNT 4.256, 6.130
SCAN RANGES 4.202, 6.126, 6.127, 6.128
SCAN TABLE 4.200, 6.61, 6.63, 6.121, 6.122, 6.132
SCR. SAVER ON/OFF 4.8, 6.66
SCR. SAVER TIME 4.8, 6.66
SCREEN A BARGRAPH 4.4, 4.5
SCREEN A SWEEP 4.4, 4.5
SCREEN COUPLING 4.6
SCREENS UNCOUPLED 4.6, 6.85
SEARCH LIMIT ON/OFF 4.127, 4.240, 4.315, 6.31
SELECT (preselector) 4.38, 6.80
SELECT FILTER 4.43
SELECT ITEMS 4.76, 4.81, 6.92, 6.93
SELECT LIMIT LINE 4.139, 4.247, 4.324, 6.21, 6.27
SELECT MACRO 4.84
SELECT MARKER 4.124, 4.237, 4.315
SELECT OBJECT 4.7
SELECT PATTERN 4.344
SELECT QUADRANT 4.64
SELECT SET 4.37
SELFTEST 4.16, 6.6
SERVICE 4.49, 6.54
SET NAME 4.41, 6.81
SETTINGS COUPLED 4.238, 4.242
SETTINGS DEVICE 1/2 ..... 4.66, 6.70
SGL SWEEP DISP OFF 4.176, 6.74
SHAPE FACT 60/3dB 4.128, 6.37
SHAPE FACT 60/6dB 4.128, 6.37
SHIFT X LIMIT LINE 4.146, 4.252, 4.330, 6.23
SHIFT Y LIMIT LINE 4.146, 4.252, 4.330, 6.24, 6.26
SIDE BAND NORM / INV 4.280, 6.111
SIGNAL COUNT 4.114, 6.31, 6.32
SIGNAL TRACK 4.112, 6.37
SINGLE SCAN 4.202, 6.74
SINGLE SWEEP 4.176, 4.335, 6.74
SKIP FREQUENCY 4.214
SLOPE POS/NEG 4.174, 4.264, 4.337, 6.153
SORT BY DELTA LIMIT 4.209
SORT BY FREQUENCY 4.208
SORT MODE 4.71
SPAN / RBW AUTO [50] 4.172, 6.99
SPAN / RBW MANUAL 4.172, 6.99
SPAN FIXED 4.86, 4.89, 4.91, 6.119, 6.121, 6.122
SPAN MANUAL 4.94, 6.120
SPLIT SCREEN 4.4, 6.57
START AT FREQ LINE 4.87, 6.121
START FIXED 4.88, 4.91, 4.95, 6.119, 6.121, 6.122
START MANUAL 4.86, 6.121
STATISTICS 4.18, 6.55
STEPSIZE = CENTER 4.93, 4.218

STEPSIZE AUTO 4.117, 6.13, 6.35
STEPSIZE MANUAL 4.93, 4.117, 4.218, 6.13, 6.34, 6.120
STOP AT FREQ LINE 4.89, 6.122
STOP FINAL MEAS 4.214
STOP FIXED 4.86, 4.91, 4.95, 6.119, 6.121
STOP MANUAL 4.88, 6.122
STOP SCAN 4.204, 6.7
SUM MKR 4.316, 6.39
SUM MKR ON/OFF 4.129
SUMMARY MARKER 4.129, 4.316
SWEEP COUNT 4.151, 4.176, 4.332, 4.335, 6.130
SWEEP COUNT 4.131, 4.319
SWEEP TIME AUTO 4.166, 6.129
SWEEP TIME MANUAL 4.166, 6.129
SYMB TABLE / ERRORS 4.29, 6.17
SYMBOL DISPLAY 4.290, 6.65
SYMBOL DISPLAY (FSK demodulation) 4.290, 4.291
SYMBOL RATE 4.279, 6.112
SYNC OFFSET 4.340, 6.114
SYNC PATTERN 4.344
SYSTEM MESSAGES 4.17, 6.147
T1-REF 4.158, 4.159, 4.259, 6.48
T1-T2+REF 4.158, 4.159, 4.259, 6.48
T1-T3+REF 4.158, 4.159, 4.259, 6.48
THRESHOLD LINE 4.136, 4.244, 4.322, 6.14, 6.15
TIME 4.57, 6.148
TIME LINE 1/2 ..... 4.137, 4.322, 6.16
TIME ON/OFF 4.8, 6.59
TINT 4.7, 6.58
TITLE 4.65, 6.71
TRACE MATH 4.158
TRACE MATH LIN/LOG 4.158, 4.159
TRACE MATH OFF 4.158, 4.159, 4.259, 6.48
TRANSD SET RANGES 4.32, 6.105
TRANSD SET UNIT 4.31, 6.104
TRANSDUCER FACTOR 4.25, 6.103
TRANSDUCER SET 4.25, 6.104, 6.105
TRC COLOR AUTO INC 4.63, 6.72
TRD FACTOR NAME 4.28, 6.102
TRD FACTOR UNIT 4.28, 6.102
TRD FACTOR VALUES 4.29, 6.103
TRD SET NAME 4.31, 6.104
TRG TO GAP TIME 4.186, 6.132
TRIGGER DELAY 4.174, 6.153
TRIGGER LEVEL (gap sweep) 4.185, 6.152
TRIGGER OFFSET 4.337, 6.153
TUNE TO MARKER 4.237, 6.46
UNIT 4.100, 6.154
UNLOCK 3.20
UPDATE 4.58
UPDATE MESSAGES 4.17
UPPER LEFT 4.64, 6.73
UPPER RIGHT 4.64, 6.73
USER PORT A/B 4.53, 6.77, 6.95
VALUE (SYNC PATTERN) 4.346, 6.115
VALUES (limit line) 4.145, 4.252, 4.330, 6.25
VECTOR ANALYZER 4.21, 4.273, 6.84
VERTICAL SCALING 4.6, 6.85
VID NOTCH ON/OFF 4.170, 6.118
VIDEO (trigger) 4.173, 4.336, 6.152
VIDEO BW AUTO 4.165, 6.99
VIDEO BW MANUAL 4.166, 6.99
VIEW 4.148, 4.255, 4.331, 6.64
VOLT 4.102, 6.51
WATT 4.102, 6.51
WIDE 4.37, 6.80
X OFFSET 4.141, 4.248, 4.326, 6.22
Y OFFSET 4.141, 4.248, 4.326, 6.24, 6.25
Y PER DIV 4.306, 6.63
ZERO SPAN 4.94, 6.120
ZOOM 4.96, 6.60
ZOOM OFF 4.97, 4.230, 6.60

Span 4.94
  measurement window 4.86, 4.217
Special characters 6.2
Split screen 3.9, 4.4
SRE (service request enable register) 5.21
SRQ (service request) 5.21, 5.31
Start frequency 4.217
  sweep 4.86
Start-Up 1.34
Status byte (STB) 5.21
Status information 3.4
  DIFOVL 3.4
  ExtRef 3.4
  IFOVLD 3.4
  LO LvD 3.4
  LO Lvl 3.4
  LO unl 3.4
  MAX / REF LVL 3.4
  OCXO 3.4
  OVLD 3.4
  UNCAL 3.4
Status Questionable
  ACPLimit register 5.24
  FREQuency register 5.25
  LIMit register 5.26
  LMARgin register 5.27
  POWer register 5.28
  SYNC register 5.29
  TRANsducer register 5.30
Status register
  CONDition part 5.19
  ENABle part 5.19
  ESE 5.22
  ESR 5.22
  EVENt part 5.19
  NTRansition part 5.19
  overview 5.20
  PPE 5.22
  PTRansition part 5.19
  SRE 5.21
  STATus OPERation 5.23
  STATus QUEStionable 5.24
  STB 5.21
  structure 5.18
  sum bit 5.19
Status reporting system 5.18
  resetting values 5.18, 5.33, 6.133
STB (status byte) 5.21
Step size
  center frequency 4.92
  delta marker 4.122, 4.235
  marker 4.117, 4.231
  receiver frequency 4.218
  setting 3.21
Stop frequency 4.217
  sweep 4.88
String 5.14
Suffix 5.11
Sum bit 5.19
Sum level 4.304
Summary Marker 4.316
Supply voltage, external noise source 4.49
Sweep
  continuous 4.335
  count 4.151, 4.176, 4.319, 4.332
  coupling 4.163
  gap 4.183
  gap length 4.186
  gated 4.177
  mode 4.175
  single 4.176, 4.335
  time 4.166

Switching on/off ... 1.38
Switching operations ... 4.18
Symbol decision point ... 4.290, 4.291
Symbol rate ... 4.279
Symbol table ... 4.296
Sync offset ... 4.340
Sync pattern ... 4.344
Sync pattern, editing ... 4.345
Sync search ... 4.339
Synchronization sequence ... 4.339
Syntax elements
command ... 5.14
System messages ... 4.17

T

Table entry ... 3.19
TCP/IP operation ... 1.71
Test receiver ... 4.187
TETRA ... 4.277
Text parameter ... 5.14
TFTS ... 4.277
Thick Ethernet ... 1.62
Thin Ethernet ... 1.61
Threshold line ... 4.322
Time lines ... 4.322
Time of internal real-time clock ... 4.57
Trace ... 4.147, 4.254
copy ... 4.152, 4.258
detector ... 4.153, 4.191
export ... 4.160, 4.259
mathematics ... 4.158
Transducer ... 4.23
entry ... 4.26
set ... 4.30
switch on ... 4.24
Transmission parameters of RS-232-C ... 8.7
Trellis diagram ... 4.287
Trigger ... 4.336
delay ... 4.174, 4.337
edge, pos./neg. ... 4.337
external ... 4.174, 4.337
free run ... 4.173
free running ... 4.336
free-running ... 4.264
freilaufend ... 4.264
gap sweep ... 4.185
line ... 4.173
offset ... 4.337
pre-trigger ... 4.185
RF power ... 4.174
slope ... 4.174, 4.264
threshold ... 4.336
video ... 4.173, 4.336

U

UNCAL ... 3.4
Unit circle ... 4.283
Unit, ... 4.308
Units ... 4.100
setting ... 4.219
Universal command ... 8.5
UNLD ... 3.4
Upper case ... 6.1
User port
configuration ... 4.53
interface ... 8.21

V

Vector analyzer mode ... 4.21
Vector diagram ... 4.289
Vector error (sum error) ... 4.297
Vector Signal Analyzer ... 4.272
Video bandwidth ... 4.165
Video filter
external ... 4.167, 4.168
notch filter ... 4.170
Video trigger ... 4.336
View ... 4.148, 4.255

W

WCPE ... 4.277
White space ... 5.14
Windows NT ... 1.41
administrator ... 1.41
login ... 1.41
password ... 1.41

Z

Zero span ... 4.94
Zoom ... 4.96, 4.149
amplitude ... 4.149, 4.255