4 REMOTE PROGRAMING

4.1 GPIB Command Index

This GPIB command index can be used as the index for Chapter 4.

4.2 GPIB Remote Programming

The spectrum analyzer is equipped with a GPIB (General Purpose Interface Bus) that complies with IEEE Standard 488.1-1978. This bus allows you to attach and use an external device to remotely control the spectrum analyzer.

4.2.1 GPIB

The GPIB is a high-performance interface bus used to connect measuring instruments to a computer. IEEE Standard 488.1-1978 defines the operations of the GPIB. Since the GPIB has a bus-configured interface, connected devices are designated by assigning them a specific address. You can connect up to 15 devices in parallel using a single bus. GPIB devices perform one or more of the following functions:

- Talker — Sends data to the bus. Only one active talker can exist on the GPIB bus.
- Listener — Receives data from the bus. Multiple active listeners can exist on the GPIB bus.
- Controller — Specifies which devices are designated as "talkers" or "listeners". Only one active controller can operate on the GPIB bus. Controllers used to control IFC and REN messages are referred to as system controllers.

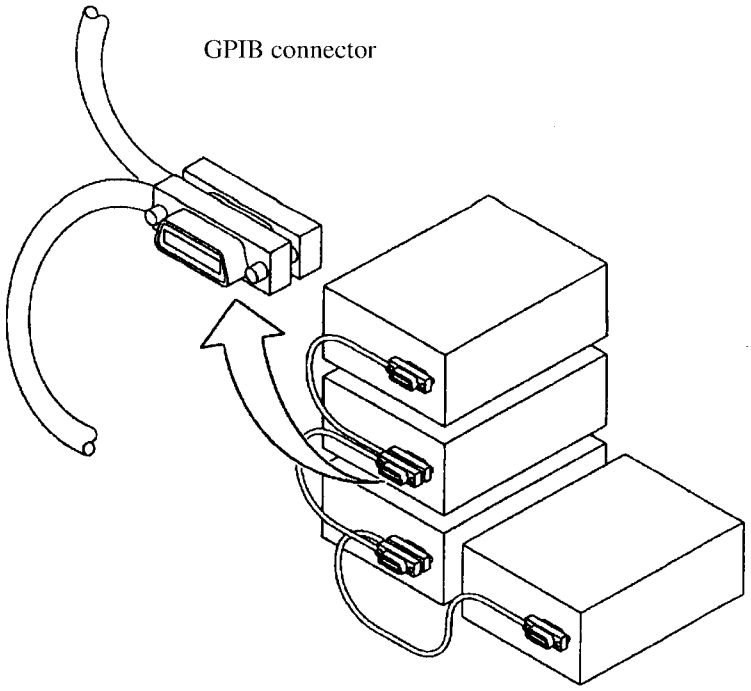
When there are multiple controllers attached to the bus, the system controller becomes the active controller by default. Other devices that can act as controllers operate as addressable devices when the system is activated.
The TCT (Take Control) interface message is used to set a controller other than the system controller as the active controller. After this setting is made, the system controller becomes inactive.
The controller controls the entire system by sending interface messages or device messages to each measuring instrument. The functions of the messages are:

- Interface message: messages used to control the GPIB bus
- Device message: messages used to control specific devices

4.2.2 GPIB Setup

(1) Connecting the GPIB

The following figure shows the standard GPIB connector and how it can be connected in parallel, or "stacked" with other connectors. Attach the GPIB connectors and secure them by tightening the screws to prevent them from coming apart during use.



The following conditions should be observed when using a GPIB interface:

- The total GPIB cable length in a single bus system must not be more than 20 m (you can calculate the current cable length using the formula total length = n x 2 m, where, n is the number of devices to be connected, including the GPIB controller).
- No more than 15 devices can be connected to a single bus system.
- There are no restrictions concerning the method of connection between cables. However, no more than three GPIB connectors should be connected to a single device, since more than this may damage the connector mounting due to excessive strain.

  (Example) The total cable length in a system with five devices should be 10m or less (2 m x 5 devices = 10 m). There is no restriction on the length of the cables between the individual devices as long as the total length does not exceed 10 m. However, if you connect 10 devices or more, make sure that at least some of the cables attaching the devices are less than 2 m so that the total is less than 20 m.

(2) Setting the GPIB Address

The GPIB Address dialog box is displayed.

1. Press **CONFIG** and ***GPIB***.
   The GPIB Address dialog box is displayed.

2. Use the data knob, the step keys, or the numeric keys to set the GPIB address as required.

3. Press **ENTER (Hz)** to set the address.

(3) Turning the display off

If the screen display is turned off, the speed of measurements made using GPIB control increases.

1. Press **CONFIG** and ***Annotation ON/OFF*** (OFF).
   OFF is selected, and all indications except for the trace are removed.

4.2.3 GPIB Interface Functions

Code	Description
SH1	Source handshake
AH1	Acceptor handshake
T6	Basic talker, serial polling, listener-specified talker cancel
TE0	Extended talker (not available)
L4	Basic listener function, talker-specified listener cancel
LE0	Extended listener (not available)
SR1	Service request function
RL1	Remote, local, local lockout
PP0	Parallel polling (not available)
DC1	Device clear
DT0	Device trigger (not available)
C0	System controller (not available)
E1	Using open-collector bus driver

4.2.4 Responses to Interface Messages

The IEEE Standard 488.1-1978 defines how the spectrum analyzer responds to interface messages. The responses are described in this section.

For information on how to send interface messages to the spectrum analyzer, refer to the instruction manual of the controller you are using.

(1) Interface Clear (IFC)

The IFC message is transmitted directly to the spectrum analyzer through a signal line. The message allows the spectrum analyzer to stop the operation of the GPIB bus. Although all input/output operation is stopped, the input/output buffer is not cleared. Note that the DCL is used to clear the buffer.

(2) Remote Enable (REN)

The REN message is transmitted directly to the spectrum analyzer through a signal line. If the spectrum analyzer is specified as a listener when the message is true, the spectrum analyzer is in remote mode. The spectrum analyzer remains in remote mode until the GTL message is received, REN becomes false, or you press the **LOCAL** key.

When the spectrum analyzer is in local mode, it ignores all received data, and key inputs (except for LOCAL key input) and when the spectrum analyzer is in LOCAL LOCKOUT mode, it ignores all key input.

(3) Serial Polling Enable (SPE)

When the spectrum analyzer is receiving a message from an external device, it is in serial polling mode. If the spectrum analyzer is specified as a talker in this mode, it sends status bytes instead of normal messages. The spectrum analyzer remains in the serial polling mode until the SPD (Serial Polling Disable) message or the IFC message is received.

When the spectrum analyzer sends an SRQ (Service Request) message to the controller, bit 6 (RQS bit) of the response data is set to 1 (true). When the spectrum analyzer has finished sending this message, the RQS bit reverts to 0 (false). The SRQ message is sent directly through a signal line.

(4) Device Clear (DCL)

When the spectrum analyzer receives a DCL message, it performs the following actions:

- Clears the input and output buffers.
- Resets syntax analysis, execution control, and response data generation.
- Cancels all commands that prevent the remote command from being executed next.
- Cancels commands that are paused to wait for other parameters.

When the spectrum analyzer receives the DCL message, it does not do the following:

- Changes data set or stored in the spectrum analyzer.
- Interrupt front panel operation.
- Modify or interrupt any spectrum analyzer operations being executed.
- Change any status bytes other than MAV (MAV becomes 0 when the output buffer is cleared).

(5) Selected Device Clear (SDC)

The SDC message operates in the same manner as the DCL message. However, it is executed only when the spectrum analyzer is a listener. In other cases, the SDC message is ignored.

(6) Go to Local (GTL)

The GTL message puts the spectrum analyzer into local mode. In local mode, all the operations normally accessible from the front panel are available.

(7) Local Lockout (LLO)

The LLO message puts the spectrum analyzer in the local lockout mode. If the spectrum analyzer is set to the remote mode when this is done, all operations normally available from the front panel are disabled (note that in the normal remote mode, you can perform front panel operations using the **LOCAL** key).

You can use one of the following three methods to set the spectrum analyzer to local mode from the local lockout mode:

- Send a GTL message to the spectrum analyzer
- Set the REN message to false (the local lockout mode will be canceled)
- Turn the spectrum analyzer power off and on again

4.2.5 Message Exchange Protocol

The spectrum analyzer receives program messages from controllers or other devices through the GPIB bus and generates response data. Program messages include commands, queries (commands used to query response data) and data. The procedure used to exchange these commands, queries and data is explained in this section.

(1) GPIB Buffers

The spectrum analyzer is equipped with the following two buffers:

(a) Input Buffer

The input buffer is used to store data temporarily for command analysis (it has a length of 1024 bytes so an input larger than this is ignored.)

Use either of the following two methods to clear this buffer:

- Turn the spectrum analyzer power on.
- Execute DCL or SDC.

(b) Output Buffer

The output buffer is used to store data which is going to be read from the controller (1024 bytes).

Use either of the following two methods to clear this buffer:

- Turn the spectrum analyzer power on.
- Execute DCL or SDC.

(2) Message Exchange

GPIB control between a controller and a device consists of two main elements; command message analysis (by the parser) and response data generation. These are explained below.

(a) Parser

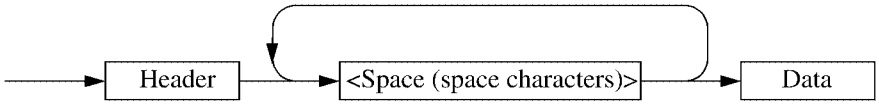
The parser receives command messages in the order of reception from the input buffer, analyzes the syntax, and determines what the received command is.

(b) Response Data Generation

When the parser determines what the query is, the spectrum analyzer generates data in the output buffer in response (that is, to output data a query must be sent immediately before the data).

4.2.6 Command Syntax

Command programs for the spectrum analyzer are defined using the following format:



(1) Header

Two types of header are available: the common command header and the simple header. The common command header has an asterisk (*) at the beginning of the mnemonic.
The simple header is a functionally independent command that has no hierarchical structure.
You can form a query command by attaching a "?" in the rear of a header.

(2) Space (Space Character)

You should separate the header from the data by one or more spaces, however spaces may be omitted.

(3) Data

When the command requires multiple data, data is separated by commas. A space may be inserted before or after each comma. For more information on data types, see Section 4.2.7 "Data Formats."

(4) Writing Multiple Commands

You can write multiple commands by separating them with semicolons in one line.

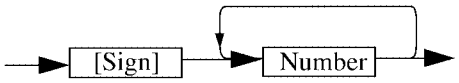
4.2.7 Data Formats

The spectrum analyzer uses the following data formats for the input and output data.

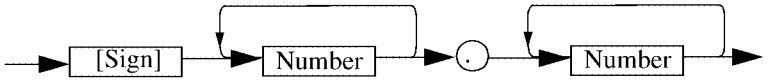
(1) Numeric Data

There are three numeric data formats, any of which can be used for input.
Some commands add units to the data when the data is input.
The following shows the three numeric data formats.

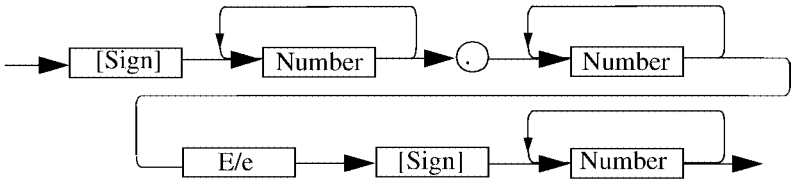
- Integer type: NR1 format



- Fixed-point type: NR2 format



- Floating-point type: NR3 format



(2) Units

The table below lists the units that you can use.

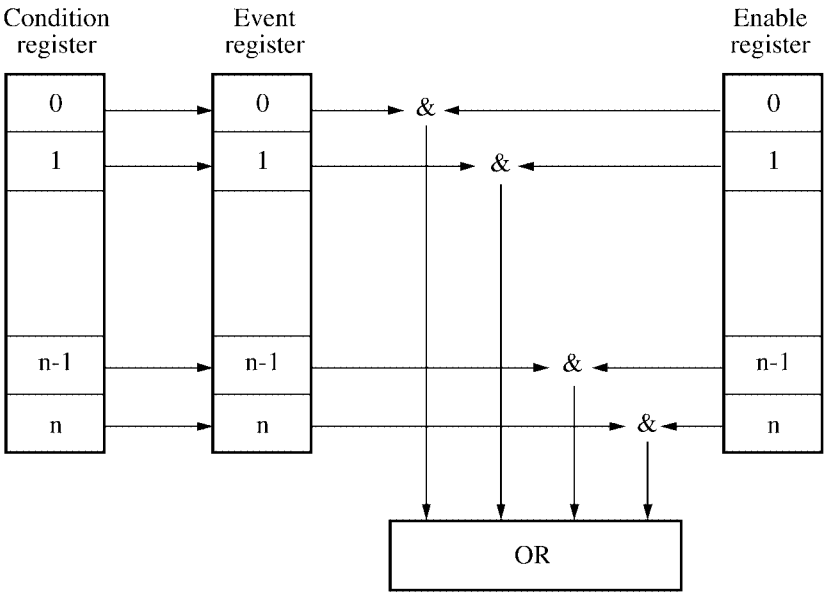
Unit	Exponential	Description
GZ	10^{9}	Frequency
MZ	10^{6}	Frequency
KZ	10^{3}	Frequency
HZ	10^{0}	Frequency
VOLT	10^{0}	Voltage
MV	10^{-3}	Voltage
UV	10^{-6}	Voltage
NV	10^{-9}	Voltage
MW	10^{-3}	Power
DB	10^{0}	dB correspondence
MA	10^{-3}	Electric Current
SC	10^{0}	Second
MS	10^{-3}	Second
US	10^{-6}	Second
PER	10^{0}	Percentage
%	10^{0}	Percentage

4.2.8 Status Byte

The spectrum analyzer has a hierarchical status register structure which complies with IEEE Standard 488.2-1987. This is used to send information on the status of various aspects of a device to the controller. This section explains the status byte and event assignments operation models.

(1) Status Register

The spectrum analyzer uses the status register model defined by IEEE Standard 488.2-1987. This consists of a condition register, an event register and an enable register.



(a) Condition Register

The condition register continuously monitors the status of devices, showing their latest status. However, this register is used internally, so no data can be written into or read out from this register.

(b) Event Register

The event register latches and retains the status information from the condition register (in some cases, it retains status changes).

Once the register is set, the condition is maintained until a query command reads out the information or the register is reset by means of the *CLS command.
No data can be written into the event register.

(c) Enable Register

The enable register specifies which bit in the event register is to be used as the valid status to generate a summary. The enable register is ANDed with the event register. The OR of the result of the AND operation is generated as a summary. The summary is written into the following status byte registers.
Any data can be written into the enable register.

The following three types of status registers are used in the spectrum analyzer:

- Status byte register
- Standard event register
- Standard operation status register

The arrangement of the status registers of the spectrum analyzer are shown in Figure 4-1. The status registers are shown in detail in Figure 4-2.

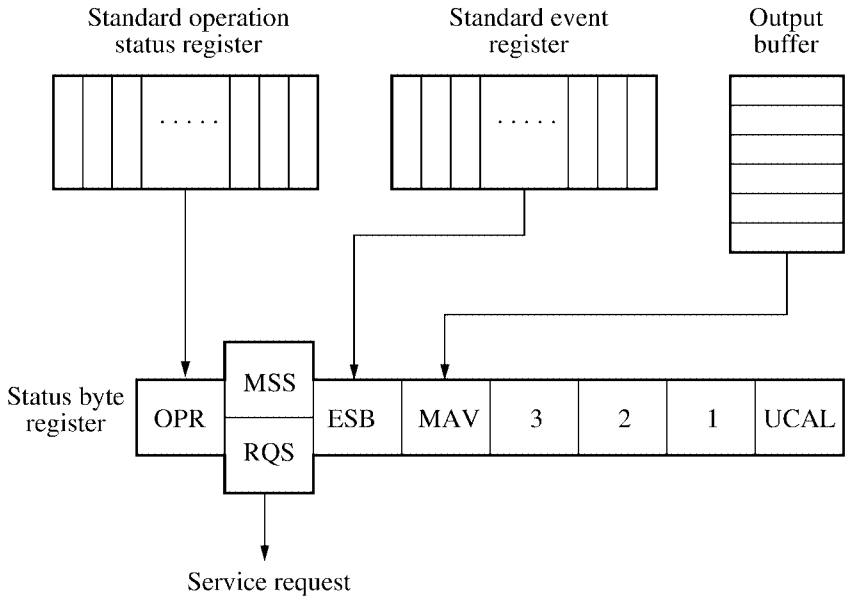


Figure 4-1 Arrangement of the Three Status Registers

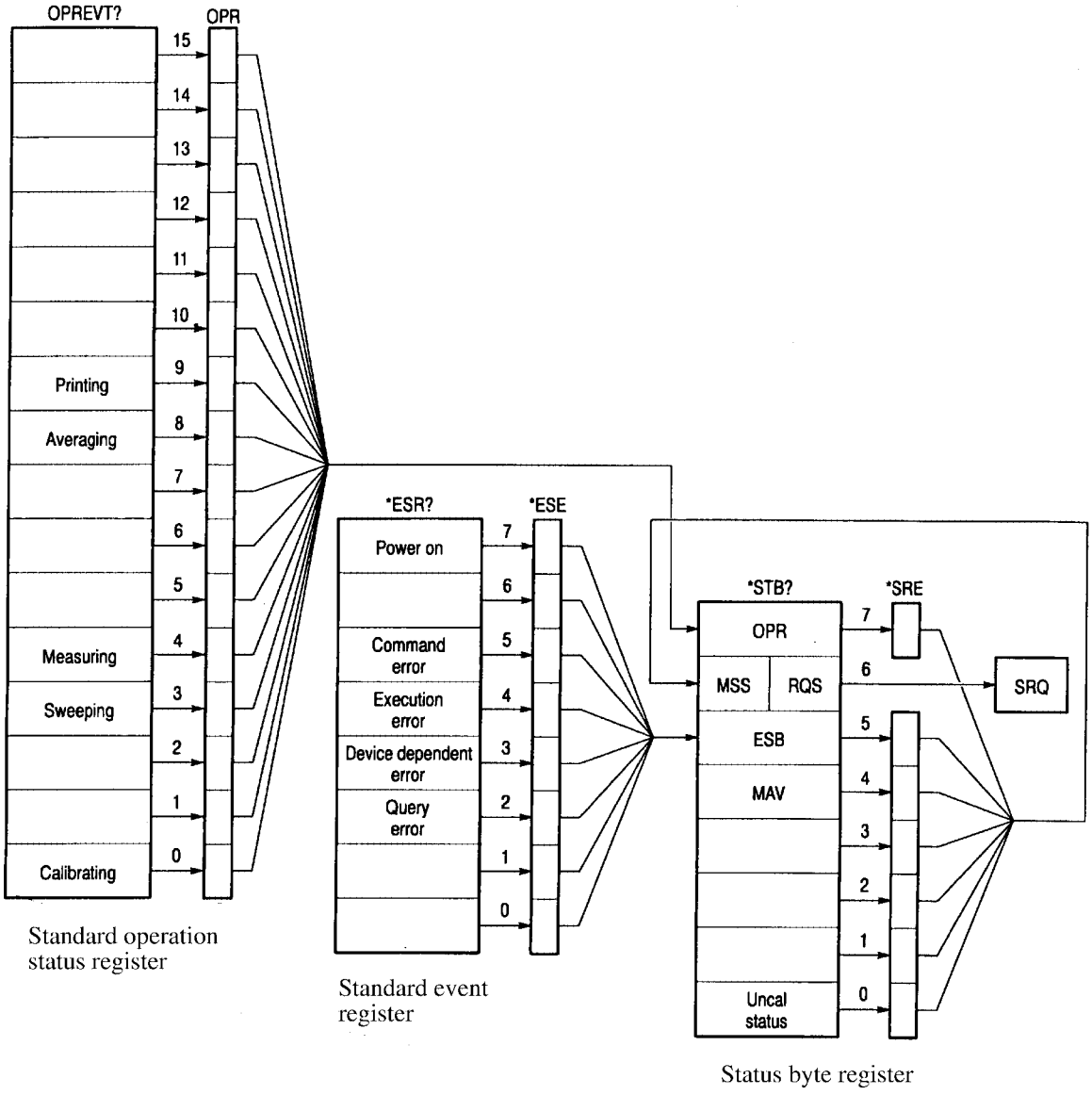


Figure 4-2 Details of the Three Status Registers

(2) Event Enable Register

Each event register has an enable register to determine which bit is available. The enable register sets the corresponding bit in decimal value.

- Set of Service Request Enable Register: *SRE
- Set of Standard Even Status Enable Register: *ESE
- Set of Operation Status Enable Register: OPR

Example: Only the Measuring bit in the operation status register is available.
The OPR bit of the status byte register is set to 1 when the Measuring bit of the operation status register is set to one.
PRINT @8;"OPR16" (An example of the program in N88BASIC)
OUTPUT 708;"*OPR16" (An example of the program for the HP200 and 300 series)

Example: The OPR (the summary of Operation Status Register) bit and ESB (the summary of Event Status Register) bit of the status byte register are available.
The MSS bit of the status byte register is set to 1 when the OPR bit or the ESB bit is set to one.
PRINT @8;"SRE160" (An example of the program in N88BASIC)
OUTPUT 708;"*SRE160" (An example of the program for the HP200 and 300 series)

(3) Standard Operation Status Register

Bit assignments for the event register (which represents the standard operation status) is listed below:

Bit	Functional definition	Description
15 to 10		This is always 0
9	Printing	This is set to 1 at the end of printing
8	Averaging	This is set to 1 when averaging is completed
7 to 5		This is always 0
4	Measuring	This is set to 1 at the end of sequence measurement
3	Sweeping	This is set to 1 when sweeping is completed
2 to 1		This is always 0
0	Calibrating	This is set to 1 when calibration data acquisition finishes

(4) Status Byte Register

The status byte register summarizes the information from the status register. In addition, a summary of the status byte register is sent to the controller as a service request. As a result, this register operates slightly differently from the status register. This section explains the status byte register.
The structure of the status byte register is shown in Figure 4-3.

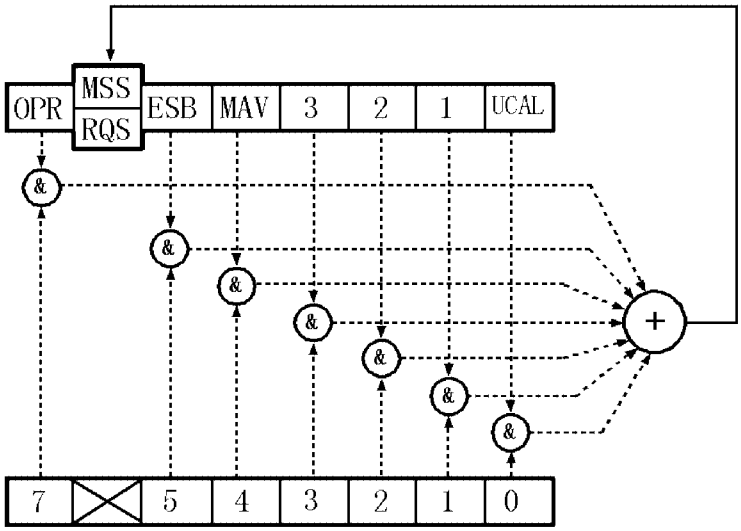


Figure 4-3 Structure of the Status Byte Register

This status byte register has the same functions as the status register, except for the following three points:

- The summary of the status byte register is written in bit 6 of the status byte register.
- Bit 6 of the enable register is always valid and cannot be changed.
- Bit 6 (MSS) of the status byte register writes the RQS of the service request.

The register responds to serial polling from the controller. On doing so, bits 0 to 5 and bit 7 of the status byte register and the RQS are read out, and then the RQS is reset to 0. Other bits are not cleared until each factor has been reset to 0.
When the *CLS and S2 commands are executed, the status byte register, the RQS bit, and the MSS bit can be cleared. Consequently, the SRQ line is now false.

4.2.8 Status Byte

The table below explains the meanings of the bits in the status byte register.

Bit	Functional definition	Description
7	OPR	The OPR bit is a summary of the standard operation status register.
6	MSS	The RQS bit is true when the MSS bit of the status byte register is set to 1. The MSS bit is the summary bit for the entire status data structure. The serial poll cannot read out the MSS bit. (However, the MSS bit is understood to be 1 when the RQS bit is 1.) To read the MSS bit, use the common command *STB?. The *STB? command can read out bit 0 to 5 and bit 7 of the status byte register and the MSS bit. In this case, neither the status byte register nor the MSS bit can be cleared. The MSS bit cannot become 0 until all the unmasked factors in the status register structure have been cleared.
5	ESB	The ESB bit is a summary of the standard event register.
4	MAV	Summary bit for the output buffer. This instrument does not use this bit.
3 to 1		This is always 0.
0	UCAL	This is set to 1 when an signal level error occurs because the sweep is too fast.

(5) Standard event register

The table below explains the meanings of the bits in the standard event register.

Bit	Functional definition	Description
7	Power on	This is set to 1 when the spectrum analyzer is switched on
6		This is always 0
5	Command Error	This is set to 1 when the parser finds a syntax error
4	Execution Error	This is set to 1 when the system fails to execute an instruction received as a GPIB command for some reason (such as out-of-range parameter)
3	Device Dependent Error	This is set to 1 when errors other than command errors, execution errors, or query errors occur
2	Query Error	This is set to 1 when no data exists or data has been deleted when the controller attempts to read out data from the spectrum analyzer
1	Request Control	Not supported in the spectrum analyzer
0	Operation Complete	Not supported in the spectrum analyzer

4.2.9 GPIB Command Codes

The following tables list the GPIB commands by function.

Listener Code Column: An asterisk (*) in the Listener Code Column indicates that the function requires numeric data together with the function code.
The sign /***/ in the Listener Code Column indicates that the function requires character string data together with the function code.
[ON], [ON,] and numeric data are omissible.
String data such as file name, label and so on can receive characters, which are found after the command and prior to the delimiter, as input values. However, when data begins with a "/", the characters between "/" and "/" are received as input.

Output Format Column: A comma (,) in the Output Format column indicates that multiple items are output.
ON/OFF or AUTO/MANUAL in the Output Format column indicates that the code outputs 1 or 0, respectively.
All frequencies are in hertz (Hz), and all times are in seconds. Levels are output in the currently displayed unit.

Table 4-1 Frequency (1 of 3)

Function	Listener Code	Talker Request	
		Code	Output Format
Center frequency	CF *	CF?	Frequency
CF Step size	CS *	CS?	Frequency
CF Step AUTO	CA	CA?	0:Manual 1:Auto
Frequency offset ON OFF	FO [ON,]* FO OFF	FO? FOON?	Frequency 0:OFF 1:ON
Start frequency	FA *	FA?	Frequency
Stop frequency	FB *	FB?	Frequency
Frequency span	SP *	SP?	Frequency
Full span	FS	—	—
Zero span	ZS	—	—
Peak zoom	PKZOOM	—	—
Last span	LTSP LS	— —	— —
Preselector Auto Manual	PPA PPM*	PPM?	Integer (-127 to 127) (*1)

(*1) Preselector is available for the R3162/72/82 only.

Table 4-1 Frequency (2 of 3)

Function	Listener Code	Talker Request	
		Code	Output Format
Frequency setting mode Frequency input mode Channel type 1 input mode Channel type 2 input mode	 FINPMD FREQ FINPMD CH1 FINPMD CH2	FINPMD?	 0:Frequency 1:CH Type 1 2:CH Type 2
Start channel offset	FACHO *	FACHO?	Frequency
Stop channel offset	FBCHO *	FBCHO?	Frequency
Carrier channel setting	CH *	CH?	Integer (channel number)
Center channel setting	CFCH *	CFCH?	Integer (channel number)
Start channel setting	FACH *	FACH?	Integer (channel number)
Stop channel setting	FBCH *	FBCH?	Integer (channel number)
Channel type 1 Table 1 input Table 2 input Table 3 input	(*1) CHED1 *,*,*,*,* CHED2 *,*,*,*,* CHED3 *,*,*,*,*	 —— —— ——	 —— —— ——
Table 1 for Channel type 1 Enable Disable	 CHTBL1 ENBL CHTBL1 DSBL	CHTBL1?	 0:Enable 1:Disable
Table 2 for Channel type 1 Enable Disable	 CHTBL2 ENBL CHTBL2 DSBL	CHTBL2?	 0:Enable 1:Disable
Table 3 for Channel type 1 Enable Disable	 CHTBL3 ENBL CHTBL3 DSBL	CHTBL3?	 0:Enable 1:Disable
Channel type 2 Input Deletion	(*2) CHEDIN *,*,*,* CHEDDEL	 —— ——	 —— ——
Internal mixer External mixer	MXI MXE	MXR?	0:INT (Internal) 1:EXT(External)
Signal Ident ON OFF	SIGID ON SIGID OFF	SIGID?	0:OFF 1:ON
Image Suppress ON OFF	IMGSP ON IMGSP OFF	IMGSP?	0:OFF 1:ON
Band selection	BND *	BND?	Integer
Band lock ON OFF	BNDLC ON BNDLC OFF	BNDLC?	0:OFF 1:ON

(*1) Specify the asterisks in order of the start channel number, stop channel number, start frequency, channel spacing and channel offfset.

(*2) Specify the asterisks in order of the channel number, carrier frequency, start frequency and stop frequency.

Table 4-1 Frequency (3 of 3)

Function		Listener Code	Talker Request	
			Code	Output Format
Average loss		AGL *	AGL?	Level
	ON	AGL ON [*]	AGLON?	0:OFF
	OFF	AGL OFF		1:ON
Loss vs Freq	ON	LVF ON	LVF?	0:OFF
	OFF	LVF OFF		1:ON
Loss vs Freq input		LVFIN *,*,* (*3)	——	——
Loss vs Freq deletion		LVFDEL	——	——

(*3) Specify asterisks (*) in the order of frequency, level (n-) and level (n+).

Table 4-2 Level

Function		Listener Code	Talker Request	
			Code	Output Format
Reference level		RL *	RL?	Level
ATT		AT*	AT?	Level
ATT auto		AA	AA?	0:Manual 1:Auto
XdB/Div		DD *	DD?	0: 10dB 1: 5dB 2: 2dB 3: 1dB
Linear × 1		LL1	—	—
Level offset	ON	RO [ON,]*	RO? ROON?	Level 0:OFF 1:ON
	OFF	RO OFF		
Hi Sens	ON	HS[ON]	HS?	0: OFF 1: ON
	OFF	HS OFF		
Input	50 Ω	OHM50	OHM?	0:50Ω
	75 Ω	OHM75		1:75Ω (*1)
Display unit				
	dBm	AUNITS DBM	AUNITS?	0:dBm
	dBmV	AUNITS DBMV		1:dBmV
	dBμV	AUNITS DBUV		2:dBμV
	Volts	AUNITS V		5:V
	Watts	AUNITS W		6:W

Table 4-3 BW

Function		Listener Code	Talker Request	
			Code	Output Format
RBW		RB*	RB?	Frequency (*3)
RBW auto		BA	BA?	0:Manual 1:Auto
VBW		VB*	VB?	Frequency (*2)
VBW auto		VA	VA?	0:Manual 1:Auto
Couple ALL auto		AL	AL?	0:Manual 1:Auto
Wide RBW	ON	WRBW ON	WRBW?	0:OFF 1:ON
	OFF	WRBW OFF		

(*1) Only the R3132 can use both 50 Ω and 75 Ω as the input impedance.
(*2) When the wide RBW is turned on, the VBW is set at 0 Hz.
(*3) The RBW is set to 0 Hz when OPT73 (FM Demod) is turned on, and Range is set to 500 kHz/ or more.

Table 4-4 Sweep

Function	Listener Code	Talker Request	
		Code	Output Format
Sweep time	SW* ST*	SW? ST?	Sweep Time Sweep Time
SWP auto	AS	AS?	0:Manual 1:Auto
Sweep mode	——	SWM?	0:Normal & Full 1:Normal & Win 20:Single & Full 21:Single & Win
Sweep mode :Normal	CONTS SN	—— ——	—— ——
Sweep mode :Single	SI SNGLS	—— ——	—— ——
Take Sweep (Used to wait until the sweep is complete.)	TS	——	——
Sweep Reset & Start	SR	——	——
Gate Sweep ON OFF	GTSWP ON GTSWP OFF	GTSWP?	0:OFF 1:ON
Gate position	GTPOS *	GTPOS?	Time
Gate width	GTWID *	GTWID?	Time
Gate Slope - Gate Slope +	GTSLP - GTSLP FALL GTSLP + GTSLP RISE	GTSLP?	0:+ 1:-
Gate Source	——	GTSRC?	3:TV-V 4:TV-H 5:EXT
Gate Source EXT Trigger TV-V TV-H	GTSRC EXT GTEX* GTSRC TVV GTSRC TVH GTTVH*	—— GTEX? —— —— GTTVH?	—— Level (Real number) —— —— No. (Integer)
External Gate IN ON	GEX ON	GEX?	0:OFF 1:ON
OFF	GEX OFF	GEXON?	0:OFF 1:ON

Table 4-5 Trigger

Function	Listener Code	Talker Request	
		Code	Output Format
Trigger Mode	——	TRGSRC?	0:Free RUN 1:Line 2:Video 3:TV-V 4:TV-H 5:Ext.
FREE RUN	TRGSRC FREE	——	——
LINE Trigger	TRGSRC LINE	——	——
VIDEO Trigger	TRGSRC VIDEO VI *	—— VI?	—— %(Integer)
External Trigger	TRGSRC EXT EX *	—— EX?	—— Level (Real number)
TV-V Trigger	TRGSRC TVV	——	——
TV-H Trigger	TRGSRC TVH TVH *	—— TVH?	—— No.(Integer)
Trigger Slope + Trigger Slope -	TRIGSLP+ TRIGSLP RISE TRIGSLP- TRIGSLP FALL	TRIGSLP?	0: + 1: -
TV system NTSC PAL&SECAM	TVHNT TVHPS	TVSYS?	1: NTSC 0: PAL&SECAM
Trigger Delay	TRGDT *	TRGDT?	Time

Table 4-6 Trace (1 of 2)

Function	Listener Code	Talker Request	
		Code	Output Format
Trace A	——	TA?	HI LOW 0: 0: Write 1:NORM 1: View 2:A-DL→A 2: Blank 3:A-B→A 3: Max Hold 4:B-A→A 4: Min Hold 5: 5: Averaging 6: 6: Power AVG
A write	AW	——	——
A view	AV	——	——
A blank	AB	——	——
A max hold A max hold OFF	AMAX ON AMAX OFF	AMAX?	0:OFF 1:ON
A min hold A min hold OFF	AMIN ON AMIN OFF	AMIN?	0:OFF 1:ON
A Averaging times	SWPCNT * AG *	SWPCNT? AG?	Average times Average times
Start	AAVG ON AGR	AAVG?	0:OFF 1:ON
Stop	AAVG OFF AGS		
Pause Continue	AGP AGC	AGP?	0:Continue 1:Pause
1 time continuous	AGSGL AGCNT	AGSGL?	0:Continuous 1:1 time
Power Average A ON	APAVG ON	APAVG?	0: OFF 1: ON
OFF	APAVG OFF		
Pause Continue	AGP AGC	AGP?	0: Continue 1: Pause
1 time continuous	AGSGL AGCNT	AGSGL?	0: Continuous 1: 1 time
Math A A-B→A B-A→A A-DL→A	ABA BAA ADLA	—— —— ——	—— —— ——
Trace B	——	TB?	0:Write 1:View 2:Blank 3:Max Hold 4:Min Hold 5:Averaging 6: Power AVG
B write	BW	——	——
B view	BV	——	——
B blank	BB	——	——
B max hold B max hold OFF	BMAX ON BMAX OFF	BMAX?	0:OFF 1:ON

Table 4-6Trace (2 of 2)

Function	Listener Code	Talker Request	
		Code	Output Format
B min hold B min hold OFF	BMIN ON BMIN OFF	BMIN?	0:OFF 1:ON
B Averaging times	SWPCNT * BG *	SWPCNT? BG?	Average times Average times
Start	BAVG ON BGR	BAVG?	0:OFF 1:ON
Stop	BAVG OFF BGS		
Pause Continue	BGP BGC	BGP?	0:Continue 1:Pause
1 time continuous	BGSGL BGCNT	BGSGL?	0:Continuous 1:1 time
Power Average B ON	BPAVG ON	BPAVG?	0: OFF
OFF	BPAVG OFF		1: ON
Pause Continue	BGP BGC	BGP?	0: Continue 1:Pause
1 time continuous	BGSGL BGCNT	BGSGL?	0: Continuous 1: 1 time
Store A → B Store B → A	BSTORE ASTORE	— —	— —
Number of trace points 501 Number of trace points 1001	TPS TPL	TP?	0:501 1:1001
Director Mode A Normal Positive Negative Sample	 DET NRM DET POS DET NEG DET SMP	DET?	 0:Normal 1:Positive 2:Negative 3:Sample
Director Mode B Normal Positive Negative Sample	 DETB NRM DETB POS DETB NEG DETB SMP	DETB?	 0:Normal 1:Positive 2:Negative 3:Sample

Table 4-7 Pass/Fail

Function	Listener Code	Talker Request	
		Code	Output Format
Pass/Fail judgement ON OFF	PFC ON PFC OFF	PFC?	0:OFF 1:ON
Reading the judgment result	——	PFJ?	0:Pass 1:Fail
Reading the judgment result (in detail)	——	OPF?	0:Pass 1:Fail(Upper) 2:Fail(Lower) 3:Fail(Both) 4:Error
Upper Fail Point Lower Fail Point	—— ——	FPU? FPL?	n<DLM> f1,l1<DLM>... n<DLM> f1,l1<DLM>...(*1)
Frequency Domain input selection Time Domain input selection	LIMTYP FREQ LIMTYP TIME	LIMTYP?	0:Freq (*2) 1:Time
X-position mode: Absolute Relative (Left) Relative(Center)	 LIMPOS ABS LIMPOS REL LIMPOS LFT LIMPOS CENT	LIMPOS?	 0:Absolute 1:Relative(Left) 2:Relative(Center)
Y-position mode: Absolute Relative(Top) Relative(Bottom) Relative(Center)	 LIMAPOS ABS LIMAPOS REL LIMAPOS TOP LIMAPOS BOTM LIMAPOS CENT(*3)	LIMAPOS?	 0:Absolute 1:Relative(Top, Center) 2:Relative(Bottom)
X-offset	LIMSFT *	LIMSFT?	Frequency/Time
Y-offset	LIMASFT *	LIMASFT?	Level
Limit Line 1 ON OFF Data input Data erase	 LMTA ON LMTA OFF LMTAIN *,* LMTADEL	 LMTA? —— ——	 0:OFF 1:ON (*2)(*4) (*2)
Limit Line 2 ON OFF Data input Data erase	 LMTB ON LMTB OFF LMTBIN *,* LMTBDEL	 LMTB? —— ——	 0:OFF 1:ON (*2) (*2)

(*1) n = Number of points fn, ln = Frequency (Time), Level <DLM> = Delimiter
(*2) To use this function, first select Domain using the LIMTYP command.
(*3) Enabled when OPT73 (FM Demod) is turned on. Relative (Top) and Relative (Bottom) cannot be used.
(*4) To enter limit line data used with OPT73 (FM Demod), first turn FM Demod on.
To enter limit line data used with OPT73 sensitivity measurements, first turn sensitivity on.

Table 4-8 Display

Function	Listener Code	Talker Request	
		Code	Output Format
Display Line Level	DL *	DL? DLON?	Level 0:OFF
ON	DL ON [,*]		1:ON
OFF	DL OFF		
Reference Line Level	RLN	RLN?	Level
ON	RLN ON [,*]	RLNON?	0:OFF
OFF	RLN OFF		1:ON
Window ON	WDO ON	WDO?	0:OFF
OFF	WDO OFF		1:ON
Window center position	WLX *	WLX?	Frequency
Window width	WDX *	WDX?	Frequency
Window Sweep ON	WDOSWP ON	WDOSWP?	0:OFF
OFF	WDOSWP OFF		1:ON
Zoom	MLTSCR ZM	MLTSCR?	0:OFF
F/T	MLTSCR FT		1:ZOOM
T/T	MLTSCR TT		2:F/T
ZOOM OFF(Screen Reset)	MLTSCR OFF		3:T/T
Zoom position	ZMPOS *	ZMPOS?	Frequency/Time
Width	ZMWID *	ZMWID?	Frequency/Time
Activating the upper screen	SCRSEL TRA	SCRSEL?	0:Upper
Activating the lower screen	SCRSEL TRB		1:Lower

Table 4-9 MKR (1 of 2)

Function	Listener Code	Talker Request	
		Code	Output Format
Marker ON OFF	MN * (*1) MKOFF MO	MN?	0: OFF 1: Normal 2: Delta
Normal Marker (ΔMarker OFF)	MK * MKN *	—— ——	—— ——
ΔMarker ON	MKD *	——	——
Marker frequency	——	MF?	Frequency (Time)(*1)
Marker Level	——	ML?	Level(*1)
Frequency + Level	——	MFL?	Frequency (Time), Level(*1)
Normal marker absolute value Frequency Level	 —— ——	 MDF1? MDL1?	 Normal MKR frequency Normal MKR level
ΔMarker absolute value Frequency Level	 —— ——	 MDF2? MDL2?	 Delta MKR frequency Delta MKR level
FixedΔMarker ON OFF	 FX ON FX OFF	FX?	 0:OFF 1:ON
MKR step Size	MPM *	MPM?	Frequency (Time)
MKR step auto	MPA	MPA?	0:Manual 1:Auto
Signal Track ON OFF	 SG ON SG OFF	SG?	 0:OFF 1:ON
MKR Couple ON OFF	CPLMK [ON] CPLMK OFF	CPLMK?	0:OFF 1:ON
MKR move A Trace B Trace	MKTRACE TRA MKTRACE TRB	MKTRACE?	0:Blank 1:A Trace 2:B Trace
Peak Search	PS	——	——
Next Peak Next Peak Left Next Peak Right	NXP NXL NXR	—— —— ——	—— —— ——
Min Search	MIS	——	——
Max-Min Search	MMS	——	——
Continuous Peak? ON OFF	 CP ON CP OFF	CP?	 0:OFF 1:ON

(*1):When using the delta mode, the frequency or level difference is used.

Table 4-9MKR (2 of 2)

Function		Listener Code	Talker Request	
			Code	Output Format
Peak ΔY div		DY *	DY?	ΔY(real value)
Peak range				
	Normal	PSN	PKRNG?	0:All
	Upper side	PSU		1:Upper
	Lower side	PSL		2:Lower
Malti Marker	ON	MLT ON	MLT?	0:OFF
	OFF	MLT OFF		1:ON
Moving the active marker		MK *	------	------
		MKN *	------	------
		MN *	------	------
Malti Marker No1	ON	MLN1 *	——	——
	OFF	MLF1	——	——
Malti Marker No2	ON	MLN2 *	------	------
	OFF	MLF2	------	------
Malti Marker No3	ON	MLN3 *	——	——
	OFF	MLF3	——	——
Malti Marker No4	ON	MLN4 *	------	------
	OFF	MLF4	------	------
Malti Marker No5	ON	MLN5 *	——	——
	OFF	MLF5	——	——
Malti Marker No6	ON	MLN6 *	------	------
	OFF	MLF6	------	------
Malti Marker No7	ON	MLN7 *	——	——
	OFF	MLF7	——	——
Malti Marker No8	ON	MLN8 *	------	------
	OFF	MLF8	------	------
Malti Marker No9	ON	MLN9 *	——	——
	OFF	MLF9	——	——
Malti Marker No10	ON	MLN10 *	------	10 frequencies + ΔMKR
	OFF	MLF10	------	10 levels + ΔMKR
Malti Marker Frequency		——	MLSF?	n<DLM> f1<DLM>... (*1)
Malti Marker Level		——	MLSL?	n<DLM> l1<DLM>... (*2)
Peak List	Frequency	PLS FREQ	------	------
	Level	PLS LEVEL	------	------
	OFF	PLS OFF	------	------
Peak list query		——	PKLST?	n<DLM> f1,l1<DLM>... (*3)

(*1) n = 11 (fixed) fn = 10 different frequencies + ΔMKR, <DLM> = Delimiter
(*2) n = 11 (fixed) ln = 10 different levels + ΔMKR, <DLM> = Delimiter
(*3) n = Number of Peaks fn, ln = Frequency (time), level <DLM> = Delimiter

Table 4-10 MKR →

Function	Listener Code	Talker Request	
		Code	Output Format
MKR → CF	MKCF MC	— —	— —
MKRΔ → CF	MTCF	———	———
MKR → REF	MKRL MR	——— ———	——— ———
PEAK → CF	PKCF	—	—
PEAK → REF	PKRL	———	———
MKRΔ → SPAN	MTSP DS	——— ———	——— ———
MKR → CF Step	MKCS M0	— —	— —
MKRΔ → CF Step	MTCS M1	——— ———	——— ———
MKR → MKR Step	MKMKS M2	— —	— —
MKRΔ → MKR Step	MTMKS M3	——— ———	——— ———

Table 4-11 Meas (1 of 3)

Function	Listener Code	Talker Request	
		Code	Output Format
Noize/Hz	NI *	NI?	Frequency
dBm/Hz ON dBμV/√Hz ON dBc/Hz ON Noise/Hz OFF Noise/Hz value	NIM NIU NIC NIF ——	NION? NIRES?	0:OFF 1:dBm/Hz 2:dBμV/√Hz 3:dBc/Hz Level
XdB Down width	MKBW *	MKBW?	Level
XdB Down left right	XDB XDL XDR	—— —— ——	—— —— ——
XdB relative XdB abs. left XdB abs. right	DC0 DC1 DC2	DC?	0:Relative 1:Absolute (left side) 2:Absolute (right side)
Continuous dB down ON OFF	CDB ON CDB OFF	CDB?	0:OFF 1:ON
3rd Order meas	PKTHIRD	——	——
AM Modulation (%AM) AM Modulation OFF AM modulation factor	AMMOD [ON] AMMOD OFF ——	AMMODON? AMMOD?	0:OFF 1:ON Value (%)
AM video modulation factor (%AM Video) AM video modulation factor OFF AM video modulation factor value	VIDMOD [ON] VIDMOD OFF ——	VIDMODON? VIDMOD?	0:OFF 1:ON Value (%)
FM frequency deviation (FM Meas) FM frequency deviation OFF FM frequency deviation value Modulation frequency input ON OFF	FMMEAS [ON] FMMEAS OFF —— FMMODF [ON,]* FMMODF OFF	FMMEASON? FMMEAS? FMMODF? FMMODFON?	0:OFF 1:ON Frequency Frequency 0:OFF 1:ON
Sound Mode ON (AM or FM) ON (AM) ON (FM) OFF	 SON SAM SFM SOF	 SD?	 0:OFF 1:ON(AM) 2:ON(FM)
Volume	SDV *	SDV?	Volume (Integer)
Pause time	PU *	PU?	Time
Squelch ON Squelch OFF	SQE [ON,]* SQE OFF	SQE? SQEON?	Level 0:OFF 1:ON

Table 4-11 Meas (2 of 3)

Function		Listener Code	Talker Request	
			Code	Output Format
Phase noise measurement	C/N measurement mode			
	ON	CNIS ON	CNISON?	0: OFF
	OFF	CNIS OFF		1: ON
	Offset frequency data read-out	-----	CNIS?	n<DLM> f1, l1 <DLM>...(*1)
	Table input	CNOFSIN *	-----	Offset frequency
	Deleting the table	CNOFSDEL	-----	-----
	Signal track ON	CNSIG ON	CNSIG?	0: OFF
	OFF	CNSIG OFF		1: ON
	Average number	CNAVG [ON,] *	CNAVG?	Integer (2 to 999)
		CNAVG OFF	CNAVGON?	0: OFF
				1: ON
Phase jitter measurement	Phase jitter measurement mode			
	ON	PJIT ON	PJITON?	0: OFF
	OFF	PJIT OFF		1: ON
	Result value readout	-----	PJIT?	Carrier level, total SSB noise and phase jitter
	Start offset frequency	PJSRTO *	PJSRTO?	Offset frequency
	Stop offset frequency	PJSTPO *	PJSTPO?	Offset frequency
	Signal track ON	PJSIG ON	PJSIG?	0: OFF
	OFF	PJSIG OFF		1: ON
	Average number	PJAVG [ON,] *	PJAVG?	Integer (2 to 999)
		PJAVG OFF	PJAVGON?	0: OFF
				1: ON

(*1) n=Set number
fn=Offset frequency
ln=Level
<DLM>=Delimiter

Table 4-11 Meas (3 of 3)

Function		Listener Code	Talker Request	
			Code	Output Format
IM measurement	IM measurement mode			
	ON	IMM ON	IMMON?	0: OFF
	OFF	IMM OFF		1: ON
	Reference wave data read-out	-----	IMMREF?	Frequency, Level
	Delta frequency readout	-----	IMMDF?	Delta frequency
	Distortion signal data read-out	-----	IMMRES?	n<DLM>LL1, LJ1, UL1, UJ1<DLM>...(*1)
	Degree setting	IMODR *	IMODR?	Degree (3, 5, 7, 9)
	Criteria input			
	3 rd order	IMLS3 *	IMLS3?	Level
	5 th order	IMLS5 *	IMLS5?	Level
	7 th order	IMLS7 *	IMLS7?	Level
	9 th order	IMLS9 *	IMLS9?	Level
	Pass/Fail judgment			
	ON	IMPFC ON	IMPFC?	0 : OFF
	OFF	IMPFC OFF		1 : ON
	Average number	IMAVG * IMAVG [ON,] * IMAVG OFF	IMAVG? IMAVGON?	Integer (2 to 999) 0: OFF 1: ON
	Hi Sens (IM Meas)			
	ON	IMHS ON	IMHS?	0 : OFF
	OFF	IMHS OFF		1 : ON

(*1) n: Result set number corresponding to the degree
LLn: Level difference in the lower frequency signal
LJn: Pass/Fail judgment result for the lower frequency signal
0: Pass
1: Fail
-1: Judgment off
ULn: Level difference for the upper frequency signal
UJn: Pass/Fail judgment result for the upper frequency signal

Table 4-12 Auto Tune

Function	Listener Code	Talker Request	
		Code	Output Format
Auto Tune	TN	—	—

Table 4-13 Counter

Function	Listener Code	Talker Request	
		Code	Output Format
Resolution : 1kHz : 100Hz : 10Hz : 1Hz	CN0 CN1 CN2 CN3	CN?	0:1kHz 1:100Hz 2:10Hz 3:1Hz
Counter ON OFF	COUNT ON COUNT OFF	COUNT?	0:OFF 1:ON
Counter value	——	CNRES?	Frequency

Table 4-14 Power (1 of 3)

Function	Listener Code	Talker Request	
		Code	Output Format
Channel Power	PWCH	PWCH? PWCHON?	Level 0:OFF 1:ON
Parameter setup Default Manual Define → Default	 PWCHST USR PWCHST MNL PWCHST DEF	PWCHST?	 0:(Unused) 1:Default 2:Manual
Total Power	PWTOTAL	PWTOTAL? PWTOTALON?	Level 0:OFF 1:ON
Average Power	PWAVG	PWAVG? PWAVGON?	Level 0:OFF 1:ON
Average time	PWTM *	PWTM?	Integer (1 to 999)
Window center position	WLX *	WLX?	Frequency
Window width	WDX *	WDX?	Frequency
Power OFF	PWM	——	——
OBW Execution OBW OFF	OBW [ON] OBW OFF	OBWON?	0:OFF 1:ON
OBW measurement value	——	OBW?	Center,OBW
OBW %	OBWPER *	OBWPER?	OBW%
OBW real-time execution	OBWEXE	——	——

Table 4-14Power (2 of 3)

Function	Listener Code	Talker Request	
		Code	Output Format
Parameter setup Default Manual Define → Default	 OBWST USR OBWST MNL OBWST DEF	OBWST?	0:(Unused) 1:Default 2:Manual
ACP Execution ACP OFF	ACP [ON] ACP OFF	ACPON?	0:OFF 1:ON
ACP measurement value	——	ACP?	n<DLM> f1L,l1L, f1H,l1H<DLM>...(*1)
Reference power value	——	ACPREF?	Level
BS Window ON OFF	ACPBSW ON ACPBSW OFF	ACPBSW?	0:OFF 1:ON
ACP real-time execution	ACPEXE	——	——
Carrier Bandwidth	CARRBS *	CARRBS ?	Frequency
CS/BS table input erase	CSBSIN *,* CSBSDEL	—— ——	—— ——
ACP screen FULL SEPA CARRIER	ACPSCR FULL ACPSCR SEPA ACPSCR CARR	ACPSCR?	0:1 screen 1:Separate screen 2:1 screen (Carrier measurement)
Symbol Rate 1/T	SYMRT *	SYMRT?	Frequency
Roll Off Factor	RFACT *	RFACT?	Real number
√Nyquist filter ON OFF	NQST ON NQST OFF	NQST?	0:OFF 1:ON
Graph ON OFF	ADG [ON] ADG OFF	ADG?	0:OFF 1:ON
Parameter setup Default Manual Define → Default	 ACPST USR ACPST MNL ACPST DEF	ACPST?	0:(Unused) 1:Default 2:Manual
Spectrum mask execution Spectrum mask OFF	SPM [ON] SPM OFF	SPMON?	0:OFF 1:ON
Parameter setup Default Manual Define → Default	 SPMST USR SPMST MNL SPMST DEF	SPMST?	0:(Unused) 1:Default 2:Manual
Spectrum mask result	——	SPM?	ref<DLM>n<DLM>f1L,l1L, f1H,l1H<DLM>...(*2)

(*1) n = Number of points
fnL= nth frequency (time) Low
lnL= nth level Low
fnH= nth frequency (time) High
lnH= nth level High
<DLM> = Delimiter

(*2) ref = Reference power value
n = Number of points
fnL= nth frequency (time) Low
lnL= nth level Low
fnH= nth frequency (time) High
lnH= nth level High
<DLM> = Delimiter

Table 4-14Power (3 of 3)

Function	Listener Code	Talker Request	
		Code	Output Format
Result display mode REL ABS	SPMMOD REL SPMMOD ABS	SPMMOD?	0:REL 1:ABS
Spurious measurement execution Freq Time Spurious measurement OFF	SPURI FREQ SPURI TIME SPURI OFF	SPURION?	0:OFF 1:Freq 2:Time
Spurious measurement result	——	SPURI?	n<DLM>, m1<DLM>,f1,l1,j1<DLM>,…, fm1,lm1,jm1<DLM>, m2<DLM>,f1,l1,j1<DLM>,…, fm2,lm2,jm2<DLM>, : : : mn<DLM>,f1,l1,j1<DLM>,…, fmn,lmn,jmn<DLM>(*1)
Table selection	SPRTBL*	SPRTBL?	Integer(*2)
Table input Freq Time	SPRIN*(*3) SPRFIN*(*3) SPRTIN*(*4)	—— —— ——	—— —— ——
Table deletion	SPRDEL(*2)	——	——
Sweep count specification ON OFF	SPRCNT [ON,]* SPRCNT OFF	SPRCNT? SPRCNTON?	Integer 0:OFF 1:ON
Pass/fail judgment value LOW UP	SPRJ LOW SPRJ UP	SPRJ?	0:LOW 1:UP
Single Measure ON OFF	SIMS ON SIMS OFF	SIMS?	0: OFF 1: ON

(*1) n=Number of measurement points (0 thru 15)
m=Number of spurious signals (0 thru 10)
f=Spurious frequency
l=Spurious level
j=Judgment result (0: Pass, 1: Fail)
<DLM>=Delimiter
(*2) (*3) (*4)Perform the corresponding operation when the spurious measurement mode is turned on.
(*3) Specify the asterisk in order of the start frequency, stop frequency, RBW, sweep time and limit level.
(*4) Specify the asterisk in order of the center frequency, RBW, sweep time and limit level.

Table 4-15 EMC

Function	Listener Code	Talker Request	
		Code	Output Format
EMC Trace Detection :QP :PEAK :Normal	 EMCDET QP EMCDET PEAK EMCDET NRM	EMCDET?	 0:Normal 1:QP 3:PEAK
QP BW 200Hz QP BW 9kHz QP BW 120kHz QP BW 1MHz QP BW auto	QP0 (*1) QP1 QP2 QP3 (*2) QPAUTO QA	QPAUTO? QA?	0:AUTO 1:200Hz 2:9kHz 3:120kHz 4:1MHz
Antenna Selection Dipole (TP1722) Log-periodic (UHALP9107) Biconical (BBA9106) Bilog (EMC03142) User correction Antenna OFF	 ANT0 AN0 ANT1 AN1 ANT2 AN2 ANT3 AN3 ANT4 AN4 ANT OFF AF	ANT?	 0:OFF 1:Dipole 2:Log-periodic 3:Biconical 4:Bilog 5:User correction
User correction ON OFF	CR ON CR OFF	—— ——	—— ——
Table input	CRIN *,* (*3)	——	——
Table erase	CRDEL	——	——
Antenna mode Level mode	CR ANT CR LVL	CR?	0:Antenna 1:level

(*1) QP BW 200 Hz is available only when the Narrow Band RBW option is installed.
(*2) Available when EMC Trace Detection is set to Peak.
(*3) The asterisks "*,*" represent a frequency and its level. Specify them in order.

Table 4-16 CAL

Function	Listener Code	Talker Request	
		Code	Output Format
CAL ALL	CLALL	—	—
Total gain	CLGAIN	—	—
IF step AMP	CLSTEP	—	—
RBW switch	CLRBW	—	—
Log linearity	CLLOG	—	—
AMPTD OFS	CLMAG	—	—
PBW	CLPBW	—	—
CAL Signal Level	CLN *	CLN?	Level
CAL 10M Reference Coarse CAL 10M Reference Fine CAL 10M Reference Default CAL 10M Reference Store	CLCREF * CLFREF * CLDREF CLSREF	CLCREF? CLFREF? — —	Integer (0 to 255) Integer (0 to 255)
f-collection ON f-collection OFF	FC ON FC OFF	FC?	0:OFF 1:ON
CAL collection ON CAL collection OFF	CC ON CC OFF	CC?	0:OFF 1:ON

Table 4-17 Save Recall

Function		Listener Code	Talker Request	
			Code	Output Format
Save Reg.		SVn (*2)	——	——
Save File		SV File name (*1)	——	——
Delete Reg.		DELn (*2)	——	——
Delete File		DEL File name (*1)	——	——
Recall Reg.		RCn (*2)	——	——
Recall File		RC File name (*1)	——	——
Save Item	Setup ON Setup OFF	SVSET ON SVSET OFF	SVSET?	0:OFF 1:ON
	Trace ON Trace OFF	SVTRC ON SVTRC OFF	SVTRC?	0:OFF 1:ON
	Antenna ON Antenna OFF	SVANT ON SVANT OFF	SVANT?	0:OFF 1:ON
	Normalize ON Normalize OFF	SVNRM ON SVNRM OFF	SVNRM?	0:OFF 1:ON (*3)
	Limit Line 1 ON 2 ON 1/2 ON OFF	SVLIM 1 SVLIM 2 SVLIM 3 SVLIM ON SVLIM OFF	SVLIM?	0:OFF 1: 1 ON 2: 2 ON 3: 1/2 ON
	Loss:Freq ON Loss:Freq OFF	SVOLSS ON SVOLSS OFF	SVOLSS?	0:OFF 1:ON (*4)
	Level ON Level OFF	SVLVL ON SVLVL OFF	SVLVL?	0:OFF 1:ON
	Channel ON Channel OFF	SVCH ON SVCH OFF	SVCH?	0:OFF 1:ON
	Spurious ON Spurious OFF	SVSPR ON SVSPR OFF	SVSPR?	0:OFF 1:ON

(*1) Add FD: or RAM: to the file name to explicitly indicate a drive name.
(Example) Write a file name such as FD: FILE00.DAT and RAM: REG00.DAT. (.DAT can be omitted.)
(*2) n = Save/Del/Recall number
A value of 0 to 99 can apply to n.
(*3) (Available when the TG option is installed.)
(*4) Enabled if the external mixer is installed.

Table 4-18 Config

Function	Listener Code	Talker Request	
		Code	Output Format
Title ON erase	 LON /**/ LOF	LB? ——	Label ——
Printer Command Select ESC/P PCL ESC/P Raster	 PRTCMD ESC PRTCMD PCL PRTCMD ESCR	PRTCMD?	 0:ESC/P 1:PCL 2:ESC/P Raster
Gray level set Gray B/W Small B/W Large Color S-Color Small S-Color Large	PRT GRY PRT MOS PRT MOL PRT COL PRT SCOLS PRT SCOLL	PRT?	0:Gray 1:B/W Small 2:B/W Large 3:Color 4:S-Color Small 5:S-Color Large
Paper Feed ON OFF	 PFEED ON PFEED OFF	PFEED?	 0: OFF 1: ON
BMP output mode select Color S-Color Gray B/W	 HCIMAG COL HCIMAG SCOL HCIMAG GRY HCIMAG MON	HCIMAG?	 0:Color 1:Gray 2:B/W 3:S-Color
File compression ON OFF	HCCMPRS ON HCCMPRS OFF	HCCMPRS?	0:OFF 1:ON
File Number	HCFILE *	HCFILE?	Number
Reading Bitmap file	——	BMP?	Binary data<EOI>
Device Select Printer Floppy	 HCDEV PRT HCDEV FDD	HCDEV?	 0:Printer 1:Floppy
Print ON	HCOPY	——	——
10 MHz internal reference signal source 10 MHz external reference signal source	RFI RFE	FREF?	0:INT 1:EXT
Number of trace points 501 Number of trace points 1001	TPS TPL	TP?	0:501 1:1001

Table 4-19 Preset

Function	Listener Code	Talker Request	
		Code	Output Format
Preset	IP *RST	—— ——	—— ——

Table 4-20 Test

Function	Listener Code	Talker Request	
		Code	Output Format
Selftest	——	*TST?	Error No.

Table 4-21 GPIB

Function	Listener Code	Talker Request	
		Code	Output Format
A Trace input/output(ASCII)	TAA	TAA?	DDDDD<DLM> × TRP(*1)
A Trace input/output(BINARY)	TBA	TBA?	2Byte × TRP
B Trace input/output(ASCII)	TAB	TAB?	DDDDD<DLM> × TRP
B Trace input/output(BINARY)	TBB	TBB?	2Byte × TRP
Status byte clear	*CLS	——	——
STB read	——	*STB?	Integer (0 to 255)
SRE read/write	*SRE *	*SRE?	Integer (0 to 255)
ESR read	——	*ESR?	Integer (0 to 255)
ESE read/write	*ESE *	*ESE?	Integer (0 to 255)
OSR read	——	OPREVT?	Integer (0 to 65535)
OSER read	OPR	OPR?	Integer (0 to 65535)
SRQ interruptON	S0	——	——
SRQ interrupt OFF	S1	——	——
SRQ status clear	S2	——	——
Service request mask	RQS *	RQS?	Integer (0 to 255)
Delimiter CR LF <EOI>	DL0	——	——
LF	DL1	——	——
<EOI>	DL2	——	——
CR LF	DL3	——	——
LF <EOI>	DL4	——	——

(*1) TRP = Number of trace points (501/1001)
<DLM> = Delimiter

Table 4-22 Others

Function	Listener Code	Talker Request	
		Code	Output Format
Display ON Display OFF	ANNOT ON ANNOT OFF	ANNOT?	0:OFF 1:ON
Device ID output	——	*IDN?	Maker name, Device type, Serial No., Revision(*1)
Error number output	——	ERRNO?	Integer
Date setting	SETDATE DATE	SETDATE?	DATE(*2)
Time setting	SETTIME TIME	SETTIME?	TIME(*3)

(*1) (Example) ADVANTEST,R3132,123456789,A00
(*2) DATE in YYMMDD format.
(*3) TIME in HHMMSS format.

Table 4-23 FM Demodulation (OPT73) (1 of 2)

Function		Listener Code	Talker Request	
			Code	Output Format
FM Demod	ON OFF	FMDEM ON FMDEM OFF	FMDEM?	0: OFF (*1) 1: ON
FM Demod Range	Hz/div	FMRNG *	FMRNG?	0: 1 kHz/div 1: 2.5 kHz/div 2: 5 kHz/div 3: 10 kHz/div 4: 25 kHz/div 5: 50 kHz/div 6: 100 kHz/div 7: 250 kHz/div 8: 500 kHz/div 9: 1 MHz/div 10: 2.5 MHz/div 11: 5 MHz/div 12: 10 MHz/div 13: 25 MHz/div 14: 50 MHz/div
Sensitivity	ON OFF	FMSEN ON FMSEN OFF	FMSEN?	0: OFF (*1) 1: ON
Sensitivity	Unit	——	FMSUN?	0: Hz/s/ 1: Hz/ms/

(*1) When FM Demod (Sensitivity) is turned on, the units of the following function's output levels are in frequency (Sensitivity): Marker level, Upper (Lower) Fail Point, display line, and reference line.

Table 4-23 FM Demodulation (OPT73) (2 of 2)

Function		Listener Code	Talker Request	
			Code	Output Format
Sensitivity	Range	— FMSRNG *	FMSRNGS? FMSRNG?	R1,R2,R3,R4 (Frequency) 0: Range 1 (Minimum) 1: Range 2 2: Range 3 3: Range 4 (Maximum)
Sensitivity	Aperture	FMAPR *	FMAPR?	Real number (1% to 100%)
Deviation	ON OFF	FMDEV ON FMDEV OFF	FMDEV?	0: OFF 1: ON
Deviation	(P-P)/2 +Peak -Peak Repetition Freq.	—— —— —— ——	FMAVG? FMPPK? FMNPK? FMRPF?	Frequency Frequency Frequency Frequency
Linearity	ON OFF	FMLIN ON FMLIN OFF	FMLIN?	0: OFF 1: ON
Linearity Setup mode	ON OFF (Quit) Auto Adj Sample Points Offset Adj Slope Adj	FMLMD ON FMLMD OFF FMLA FMLSMP * FMLOFS * FMLSLP *	FMLMD? —— FMLSMP? FMLOFS? FMLSLP?	0: OFF 1: ON —— Integer (2 to 100) Frequency Frequency
Linearity Error	Max Min	— —	FMLMAX? FMLMIN?	Frequency Frequency
Calibration	All Range Only	FMALL FMONLY	—— ——	—— ——

Table 4-24 TG (OPT74)

Function	Listener Code	Talker Request	
		Code	Output Format
TG on TG off	TG TGF	TG?	0:OFF 1:ON
TG level	TGL *	TGL?	Level
Frequency Cal AUTO Manual	TGA TGM *	— TGM?	— Frequency
Normalize ON Normalize OFF	NORM ON ANORM ON NORM OFF ANORM OFF	NORM? ANORM?	0:OFF 1:ON 0:OFF 1:ON
Normalize Execute	NORM EX AR	—— ——	—— ——
Save Item Normalize ON Normalize OFF	SVNRM ON SVNRM OFF	SVNRM?	0:OFF 1:ON

Table 4-25 Entry

Function	Listener Code	Talker Request	
		Code	Output Format
Numeric value entry	0	—	—
	1	—	—
	2	—	—
	3	—	—
	4	—	—
	5	—	—
	6	—	—
	7	—	—
	8	—	—
	9	—	—
Decimal point	.	------	------
Negative sign	-	—	—
Positive sign	+	------	------
Exponent	EXP	—	—
	E	------	------
GHz	GZ	—	—
MHz	MZ	------	------
KHz	KZ	—	—
Hz	HZ	------	------
mW	MW	—	—
DB relationship	DB	------	------
mA	MA	—	—
Second	SC	------	------
Millisecond	MS	—	—
	MSEC	—	—
Microsecond	US	------	------
	USEC	—	—
Nanosecond	NSEC	------	------
Enter	ENT	—	—
Volt	VOLT	------	------
Millivolt	MV	—	—
Microvolt	UV	------	------
Nanovolt	NV	—	—
%	PER	------	------
	%	—	—

4.2.10 Example Programs

This section describes remote control examples used with GPIB port.

4.2.10.1 Sample Programs for Setting or Reading Measurement Conditions

---

CAUTION ***Visual Basic 4.0 (referred to as VB henceforth) is used in the sample programs shown here. Also, National Instruments-made GPIB board (referred to as NI-made for brevity henceforth) is used for the GPIB control board; NI-made driver is used for the control driver.***

- Program examples using VB

Example VB-1: Setting the center frequency after performing an analyzer master reset

```
Call ibclr(spa)                          ' Performs a Device Clear.

Call ibwrt(spa, "IP")                    ' preset
Call ibwrt(spa, "CF 30MZ")               ' Set the center frequency to 30 MHz.
```

Example VB-2: Setting the start frequency to 300 kHz, setting the stop frequency to 800 kHz and adding 50 kHz to the frequency offset.

```
Call ibclr(spa)                          ' Performs a Device Clear.
Call ibwrt(spa, "FA 300KZ")              ' Set the start frequency to 300 kHz.
Call ibwrt(spa, "FB 800KZ")              ' Set the stop frequency to 800 kHz.
Call ibwrt(spa, "FO 50KZ")               ' Add 50 kHz to the frequency offset.
```

Example VB-3: Setting the reference level to 87 dBμV (in 5 dB/div) and the RBW to 100 kHz

```
Call ibclr(spa)                          ' Performs a Device Clear.

Call ibwrt(spa, "AUNITS DBUV")           ' Set the level unit to dBμV.
Call ibwrt(spa, "RL 87DB")               ' Set the reference level to 87 dB (μV).
Call ibwrt(spa, "DD 5DB")                ' Set the vertical gradation to 5 dB/div.
Call ibwrt(spa, "RB 100KZ")              ' Set the RBW to 100 kHz.
```

Example VB-4: Setting the instrument using variables

```
Dim A As String
Dim B As String
Dim C As String

A = "10"                                 ' Set the character string.
B = "2"
C = "20"

Call ibclr(spa)                          ' Performs a Device Clear.

Call ibwrt(spa, "CF " & A & "MZ")        ' Set the start frequency to A MHz.
Call ibwrt(spa, "SP " & B & "MZ")        ' Set the span frequency to B MHz.
Call ibwrt(spa, "AT " & C & "DB")        ' Set the ATT to C dB.
```

Example VB-5: Saving set values in Register 5 and recalling them from Register 5

```
Dim LabelBuff As String                    ' Character string buffer for the label

LabelBuff = "SPECTRUM Analyzer"            ' Set the label.

Call ibclr(spa)                            ' Performs a Device Clear.

Call ibwrt(spa, "CF 30MZ")                 ' Set the parameter.
Call ibwrt(spa, "SP 1MZ")
Call ibwrt(spa, "DET POS")
Call ibwrt(spa, "LON " & LabelBuff)        ' Set the label.

Call ibwrt(spa, "SV 5")                    ' Save the data in Register 5.

Call ibwrt(spa, "CF 1GZ")                  ' Change the set parameters.
Call ibwrt(spa, "SP 200MZ")

Call ibwrt(spa, "RC 5")                    'Recall the data from Register 5.
```

Example VB-6: Enter Limit line1 in the table and turn Limit line 1 on

```
Call ibclr(spa)                            ' Perform a device clear.

Call ibwrt(spa, "LMTADEL")                 ' Clear the table used for Limit Line 1.
Call ibwrt(spa, "AUNITS DBUV")             ' Set the unit of level to dBµV.

Call ibwrt(spa, "LMTAIN 25MZ, 49.5DB")     ' Enter data use by Limit Line 1.
Call ibwrt(spa, "LMTAIN 35MZ, 49.5DB")
Call ibwrt(spa, "LMTAIN 35MZ, 51.5DB")
Call ibwrt(spa, "LMTAIN 55MZ, 51.5DB")
Call ibwrt(spa, "LMTAIN 55MZ, 54.3DB")
Call ibwrt(spa, "LMTAIN 65MZ, 54.3DB")
Call ibwrt(spa, "LMTAIN 65MZ, 57.0DB")
Call ibwrt(spa, "LMTAIN 68MZ, 57.0DB")
Call ibwrt(spa, "LMTAIN 68MZ, 60.0DB")
Call ibwrt(spa, "LMTAIN 75MZ, 60.0DB")
Call ibwrt(spa, "LMTAIN 75MZ, 62.5DB")
Call ibwrt(spa, "LMTAIN 82MZ, 62.5DB")
Call ibwrt(spa, "LMTAIN 82MZ, 64.7DB")

Call ibwrt(spa, "FA 0MZ")                  ' Start frequency of 0 MHz
Call ibwrt(spa, "FB 100MZ")                ' Stop frequency of 100 MHz
Call ibwrt(spa, "LMTA ON")                 ' Turn Limit line 1 on.
```

Example VB-7: Sample Program of the Gated Sweep

```
Call ibclr(spa)                            ' Perform a device clear.

Call ibwrt(spa, "GTSRC EXT")               ' Set the Gate signal source to EXT.
Call ibwrt(spa, "GTSLP +")                 ' Set the Gate signal slope to plus (+).
Call ibwrt(spa, "GTWID 10MS")              ' Set the window width of the gated sweep to 10 msec.
Call ibwrt(spa, "GTPOS 10US")              ' Set the window position of the gated sweep to 10 µsec.
Call ibwrt(spa, "GTSWP ON")                ' Turn the gated sweep on.
```

4.2.10.2 Sample Programs for Reading Data

In order to output measurement data or settings, use the "xx?" command. This ensures that the data is read when the device is in the talker mode. Available output formats are listed in the table below. The delimiter positioned at the end of data can be specified from 5 types (refer to "Others" in the GPIB code list). Once set, "xx?" command continues to operate until it is changed.

	Output Format
Frequency	± D.DDDDDDDDDDD E ± DD CR LF ↑ ↑ ↑ ↑ 1 2 3 4 • Data size (1 to 3) is a maximum of 19 bytes, and the unit is Hz.
	Example Specify "CF?" and output as center frequency.
Level	± D.DDDDDDD E ± DD CR LF ↑ ↑ ↑ ↑ 1 2 3 4 • Data size (1 to 3) is a maximum of 19 bytes, and the unit corresponds to each UNIT setting.
	Example Specify "ML?" and output as marker level.
Time	± D.DDD E ± DD CR LF ↑ ↑ ↑ ↑ 1 2 3 4 • Data size (1 to 3) is a maximum of 19 bytes, and the unit is sec.
	Example Specify "SW?" and output sweep time.
	DDDD CR LF ↑ ↑ 2 4 • The maximum byte of the data size corresponds to the maximum size of the output data.
	Example ON/OFF status or Averaging count is output.

<Supplement> 1= Sign (a space for plus sign; "-" for minus sign)
2= Mantissa of data
3= Exponent of data
4= Delimiter (CR/LF in initial setting can be changed with "DLn" code.)

Example VB-8: Output the marker level

```
Dim sep As Integer

Call ibclr(spa)                                    ' Perform a device clear.

Call ibwrt(spa, "CF 30MZ")                         ' Set the parameter.
Call ibwrt(spa, "SP 1MZ")
Call ibwrt(spa, "MK 30MZ")                         ' The marker frequency is set to 30 MHz.
Call ibwrt(spa, "TS")

Call ibwrt(spa, "ML?")                             ' Query command for the marker level.

Rdbuff = Space(30)                                 ' Allocate a total of 30 bytes to the buffer area.

Call ibrd(spa, Rdbuff)                             ' Read the data (30 bytes Max.).
sep = InStr(1, Rdbuff, vbCrLf, 0)                  ' Check the number of character to the delimiter.
RichTextBox1.Text = "MarkerLevel = " & Left(Rdbuff, sep - 1)
                                                   ' Outputs the data on the screen.

An example display:
MarkerLevel = -8.818750000000E+01
```

Example VB-9: Reading the center frequency and displaying it

```
Dim sep As Integer

Call ibclr(spa)                                    ' Performs a Device Clear.

Call ibwrt(spa, "CF?")                             ' Query command for the center frequency.

Rdbuff = Space(30)                                 ' Allocate the buffer memory space to 30 bytes.
Call ibrd(spa, Rdbuff)                             ' Read the data (30 bytes Max.)
sep = InStr(1, Rdbuff, vbCrLf, 0)                  ' Check the number of characters prior to the delimiter.
RichTextBox1.Text = "CenterFreq = " & Left(Rdbuff, sep - 1)
                                                   ' Display the data on the screen.

An example display:
CenterFreq = +3.000000000000E+07
```

Example VB-10: Reading the level and display unit and displaying them

```
Dim sep As Integer

Call ibclr(spa)                                    ' Performs a Device Clear.

Call ibwrt(spa, "RL?")                             ' Query command for the reference level.

Rdbuff = Space(30)                                 ' Allocate the buffer memory space to 30 bytes.
Call ibrd(spa, Rdbuff)                             ' Read the data (30 bytes Max.) from the spectrum analyzer.
sep = InStr(1, Rdbuff, vbCrLf, 0)                  ' Check the number of characters prior to the delimiter.
RichTextBox1.Text = "RefLevel = " & Left(Rdbuff, sep - 1)
                                                   ' Display the data on the screen.

Call ibwrt(spa, "AUNITS?")                         ' Query command for the level unit

Rdbuff = Space(3)
Call ibrd(spa, Rdbuff)
sep = InStr(1, Rdbuff, vbCrLf, 0)                  ' Check the number of characters prior to the delimiter.
RichTextBox1.Text = RichTextBox1.Text & vbCrLf & "UNIT = " & Left(Rdbuff, sep - 1)
                                                   ' Display the previous result, followed by a return mark and the
                                                   ' most recent result.

An example display:
RefLevel = +0.000000000000E+00
UNIT = 0
```

Example VB-11: Executing the 6 dB-down operation, reading the frequency and level and displaying them

```
Dim sep As Integer

Call ibclr(spa)                                    ' Performs a Device Clear.

Call ibwrt(spa, "CF 30MZ")                         ' Set the parameter.
Call ibwrt(spa, "SP 20MZ")

Call ibwrt(spa, "MKBW 6DB")                        ' Set a 6 dB down measurement.
Call ibwrt(spa, "PS")                              ' Peak search.
Call ibwrt(spa, "XDB")                             ' Perform the 6 dB down measurement.
Call ibwrt(spa, "MFL?")                            ' Query command for the marker level and frequency.

Rdbuff = Space(50)                                 ' Allocate the buffer memory space to 50 bytes.
Call ibrd(spa, Rdbuff)                             ' Read the data (50 bytes Max.) from the spectrum analyzer.

sep = InStr(1, Rdbuff, vbCrLf, 0)                  ' Check the number of characters prior to the delimiter.

RichTextBox1.Text = "Marker Freq & Level = " & Left(Rdbuff, sep - 1)
                                                   ' Display the data on the screen.

An example display:
Marker Freq & Level = +2.000000000000E+05, +1.023437500000E+00
```

Example VB-12: Measuring OBW and displaying it

```
Dim LENG1 As Integer, LENG2 As Integer
Dim OBW As String
Dim FC As String
Dim searchchar As String

Call ibclr(spa)                                  ' Perform a device clear.

Call ibwrt(spa, "CF 30MZ")                       ' Send the command already set.
Call ibwrt(spa, "SP 1MZ")
Call ibwrt(spa, "MK 30MZ")
Call ibwrt(spa, "OBW ON")
Call ibwrt(spa, "TS")

Call ibwrt(spa, "OBW?")                          ' Send the query command.
Rdbuff = Space(60)                               ' Allocate the area to the read buffer.
Call ibrd(spa, Rdbuff)                           ' Read the read buffer (the maximum number of bytes to be output
                                                 ' is determined by the buffer area size).

' Formatting output character string
LENG1 = InStr(1, Rdbuff, Chr(44), 0)             ' Search for the first comma.
FC = Mid(Rdbuff, 1, LENG1 - 1)                   ' Read the character prior to the comma.

DoEvents

LENG2 = InStr((LENG1 + 1), Rdbuff, Chr(13), 0) 'Determine the last data by searching for the delimiter.
OBW = Mid(Rdbuff, (LENG1 + 1), (LENG2 - LENG1-1))
                                                 'Read the data between the second comma and the delimiter.

RichTextBox1.Text = "OBW = " & OBW & vbCrLf & "Fc = " & FC & vbCrLf
                                                 ' Display the data on the screen.

An example display:
OBW = +9.810000000000E+05
FC = +3.000250000000E+07
```

Example VB-13: Reading and displaying the three largest peak levels

```
Dim pk1 As String, pk2 As String, pk3 As String

Call ibclr(spa)                                    ' Perform a device clear.
Call ibwrt(spa, "CF 0MZ")                          ' Apply the settings.
Call ibwrt(spa, "SP 100MZ")

Call ibwrt(spa, "PS")                              ' Search for the peak.
Call ibwrt(spa, "ML?")                             ' Query command to search for the marker level
Rdbuff = Space(25)                                 ' Reserve buffer memory space.
Call ibrd(spa, Rdbuff)                             ' Receives the output.
pk1 = LeftB(Rdbuff, (InStrB(1, Rdbuff, Chr(13), 1) - 1))
                                                   ' Read the data between the starting point and the delimiter.

Call ibwrt(spa, "NXP")                             ' Search for the next peak.
Call ibwrt(spa, "ML?")
Rdbuff = Space(25)
Call ibrd(spa, Rdbuff)
pk2 = LeftB(Rdbuff, (InStrB(1, Rdbuff, Chr(13), 1) - 1))
                                                   ' Read the data between the starting point and the delimiter.

Call ibwrt(spa, "NXP")
Call ibwrt(spa, "ML?")
Rdbuff = Space(25)
Call ibrd(spa, Rdbuff)
pk3 = LeftB(Rdbuff, (InStrB(1, Rdbuff, Chr(13), 1) - 1))
                                                   ' Read the data between the starting point and the delimiter.

RichTextBox1.Text = "1st PK = " & pk1 & vbCrLf & "2nd PK = " & pk2 & vbCrLf & "3rd PK = " & pk3 & vbCrLf
                                                   ' Display the data on the screen.
```

An example display:
1st PK = -8.5539062500001E+01
2nd PK = -7.0046875000001E+01
3rd PK = -8.655468750000E+01

4.2.10.3 Sample Programs for Inputting or Outputting Trace Data

Trace data on the screen includes data for 501 or 1001 points on the frequency axis. For inputting and outputting data, it is necessary to transfer data for 501 or 1001 points from the left side (start frequency) in order. Each point level is expressed by an integer from 1792 to 14592 (however, if the trace exceeds the upper limit of the vertical scale, a value greater than 14592 is transferred).

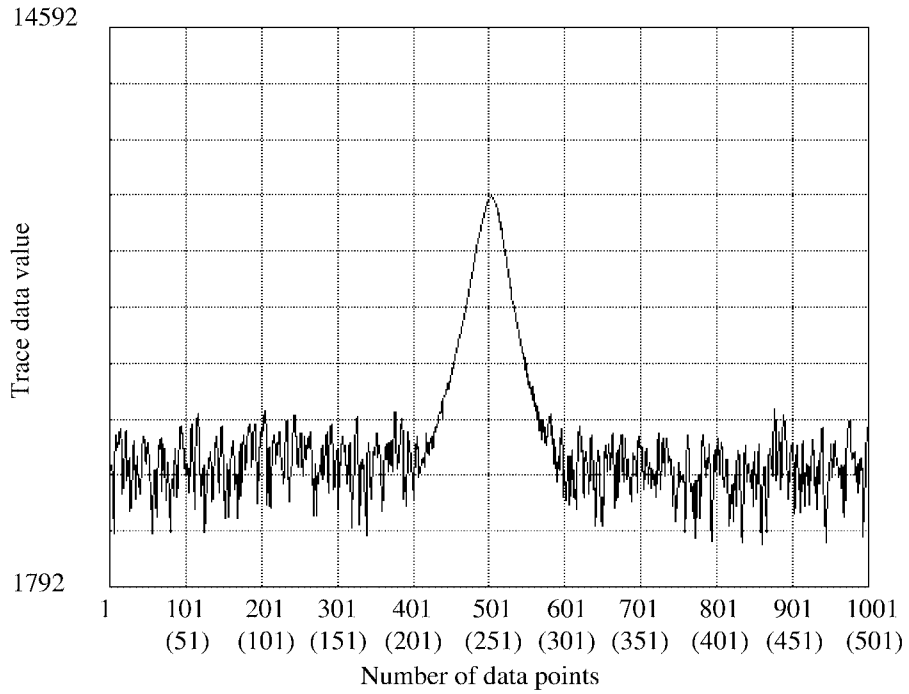


Figure 4-4 Relationship between Screen Graticule and Trace Data

Trace data can be input or output in either ASCII or binary format.

Table 4-26 Trace Accuracy Specification Codes

GPIB Code	Description
TPS	Sets the number of measurement points to 501.
TPL	Sets the number of measurement points to 1001.

Table 4-27 I/O format

<table>
<tr><th>I/O format</th><th colspan="3">Description</th></tr>
<tr><td rowspan="3">ASCII format</td><td colspan="3">DDDDD CR LF
↑ ↑
Data for one point ↑ Delimiter
Five-byte data without header</td></tr>
<tr><td></td><td>Input GPIB code</td><td>Output GPIB code</td></tr>
<tr><td>Memory A
Memory B</td><td>TAA
TAB</td><td>TAA?
TAB?</td></tr>
<tr><td rowspan="4">Binary format</td><td colspan="3">DD DD ..... DD DD + EOI
↑ ↑ ↑ ↑ ↑
Delimiter
Low-order byte for 1st point
Low-order byte for1001/501st point
High-order byte for 1st point
High-order byte for 1001/501st point

Each point data is divided into two parts: high-and low-order bytes. EOI signal is attached at the end of the data for continuous 1001/501 points.</td></tr>
<tr><td></td><td>Input GPIB code</td><td>Output GPIB code</td></tr>
<tr><td>Memory A
Memory B</td><td>TBA
TBB</td><td>TBA?
TBB?</td></tr>
</table>

Example VB-14: Read the trace data in ASCII format

```
Dim tr(1000) As String                          ' Allocate an array in the buffer for 1001 points.
Dim i As Integer
Dim res As String

Call ibclr(spa)                                 ' Perform a device clear.

Call ibwrt(spa, "DL0")                          ' CR LF EOI
Call ibwrt(spa, "DET NEG")                      ' Set it to the negative detector.
Call ibwrt(spa, "TAA?")

For i = 0 To 1000 Step 1                        ' Repeat the operation for 1001 points.
  tr(i) = Space(7)                              ' Allocate a total of 7 bytes (5 bytes for the data, and 2 bytes for
                                                ' delimiters).
  Call ibrd(spa, tr(i))                         ' Read the data.
                                                ' Output it to the screen.
  res = res & "tr(" & Str(i) & ") = " & Left(tr(i), 5) & vbCrLf
  DoEvents
Next i

RichTextBox1.Text = res
```

Example VB-15: Read the A memory data in binary format

```
Dim tr(1000) As Integer                         ' Allocates an array in the buffer for 1001 points.
Dim i As Integer
Dim res As String

Call ibclr(spa)                                 ' Performs a device clear.
Call ibconfig(0, IbcEndBitIsNormal, 0)          ' Sets the GPIB-board software so that the End bit of the Ibsta
                                                ' variables is set to 1 only when EOI has been received.
Call ibconfig(spa, IbcReadAdjust, 1)            ' Sets the spectrum analyzer so that the upper byte is swapped for
                                                ' the lower byte during a read operation.

Call ibwrt(spa, "DL2")                          ' Sets the delimiter to EOI only.
Call ibwrt(spa, "DET NEG")                      ' Sets it to the negative detector.
Call ibwrt(spa, "TBA?")                         ' Query for Trace A in binary data

Call ibrdi(spa, tr(), 1001 * 2)                 ' Reads binary data for 1001 points.

For i = 0 To 1000 Step 1                        ' Repeats the operation for 1001 points.

  res = res & Str(tr(i)) & vbCrLf               ' Outputs it to the screen.
  DoEvents

Next i
RichTextBox1.Text = res

Call ibwrt(spa, "DL0")                          ' Sets the delimiter to the CR, LF and EOI.
Call ibconfig(0, IbcEndBitIsNormal, 1)          ' Resets the GPIB software to the standard settings.
Call ibconfig(spa, IbcReadAdjust, 0)
```

Example VB-16: Enter data into A memory in ASCII mode
(When the 501 point mode is set, change 1001 and 1000 to 501 and 500, respectively.)

```
Dim trdata(1000) As Integer
Dim i As Integer

trdata(0) = 1792                          ' Provide a temporary data used to test the input (*).
For i = 1 To 1000 Step 1
  trdata(i) = Str(Val(trdata(i - 1)) + 12)
  DoEvents
Next i                                    ' When there is the data, the steps between the place marked with
                                          ' (* ) and this point are not required.

Call ibclr(spa)                           ' Perform a device clear.
Call ibwrt(spa, "AB")                     ' Set Trace A to BLANK.
Call ibwrt(spa, "TAA")                    ' Set Trace A in ASCII.

For i = 0 To 1000 Step 1                  ' Repeats the operation for 1001 points.
  Call ibwrt(spa, CStr(trdata(i)))        ' Sends the value after it has been converted to the ASCII data.
  DoEvents
Next i

Call ibwrt(spa, "AV")                     ' Set Trace A to VIEW.
```

4.2.10.4 Example program using the TS (Take Sweep) command

Example VB-17: An ACP measuremant is taken and then the measurement result is read (using the TS command).

```
Dim state As Integer
Dim sep1 As Integer, sep2 As Integer
Dim i As Integer, j As Integer
Dim cnt As Integer
Dim LvlH As String, LvlL As String
Dim FrqH As String, FrqL As String

Call ibclr(spa)                                 'A device clear is carried out.

Call ibwrt(spa, "SI")                           'Set the single mode.
Call ibwrt(spa, "CF 1500MZ")                    'Set the center frequency to 1500 MHz.
Call ibwrt(spa, "SP 250KZ")                     'Set the frequency span to 250 kHz.
Call ibwrt(spa, "RB 1KZ")                       'Set RBW to 1 kHz.
Call ibwrt(spa, "VB 3KZ")                       'Set VBW to 3 kHz.
Call ibwrt(spa, "ST 20SC")                      'Set the sweep time to 20 sec.
Call ibwrt(spa, "CSBSDEL")                      'Clear the channel space and bandwidth previously set.
Call ibwrt(spa, "CSBSIN 50KZ,21KZ")             'Set CS to 50 kHz, and BS to 21 kHz.
Call ibwrt(spa, "OPR 256")                      'The Avereaging bit of the operation register is set
                                                'to ENABLE.
Call ibwrt(spa, "*CLS")                         'Clear STATUS byte
Call ibwrt(spa, "S0")                           'SRQ is enabled.
Call ibwrt(spa, "ACP ON")                       'Start the ACP measurement.


For j = 1 To 10 Step 1
 Call ibwrt(spa, "TS")                          'Execute one sweep.
 Call ibwrt(spa, "ACP?")                        'The ACP measurement result has been requested.
 Rdbuff = Space(3)                              'Assign an area for 1 byte of integer and 2 bytes of delimiter prior
                                                 to reading the result.
 Call ibrd(spa, Rdbuff)                         'Read the data.
 cnt = CInt(Rdbuff)                             'Convert the contents of the buffer into integer type data.

 For i = 1 To cnt Step 1
  Rdbuff = Space(81)                            'Assign an area of 81 bytes (Real number x 4 + ',' x 3 + CRLF).
  Call ibrd(spa, Rdbuff)                        'Read the data.

  sep1 = InStr(1, Rdbuff, ",", 0)               'Search for the first comma starting from the top of the buffer.
  FrqL = Left(Rdbuff, sep1 - 1)                 'Read the string between the top and the character string.
  sep2 = InStr(sep1 + 1, Rdbuff, ",", 0)        'Search for the next comma.
  LvlL = Mid(Rdbuff, sep1 + 1, sep2 - sep1 - 1) 'Read the string between the first and second separators (com-
                                                 mas).

  sep1 = InStr(sep2 + 1, Rdbuff, ",", 0)        'Search for the third comma.
  FrqH = Mid(Rdbuff, sep2 + 1, sep1 - sep2 - 1) 'Read the string between the second and third separators (com-
                                                 mas).
  sep2 = InStr(sep1, Rdbuff, Chr(13), 0)        'Search for the terminator (CR).
  LvlL = Mid(Rdbuff, sep1 + 1, sep2 - sep1 - 1) 'Read the string between the separators (third comma and CR).

                                                'Displays the result on the screen.
  RichTextBox1.Text = RichTextBox1.Text & FrqL & "Hz;" & LvlL & vbCrLf
  RichTextBox1.Text = RichTextBox1.Text & FrqH & "Hz;" & LvlL & vbCrLf

 Next i
 DoEvents
Next j
```

4.2.10.5 Program Examples Using the Status Byte

Example VB-18: Execute single sweeping and wait until its finished (when not using SRQ)

```
Dim state As Integer

Call ibclr(spa)                          ' Performs a Device Clear.
Call ibwrt(spa, "SI")                    ' Turn the single sweep mode on.
Call ibwrt(spa, "OPR8")                  ' Enables Sweep-end bit of operation status register
Call ibwrt(spa, "*CLS")                  ' Clear the status byte.
Call ibwrt(spa, "SI")                    ' Begin sweeping.

Do

  Call ibwrt(spa, "*STB?")               ' Query command to read the status byte.
  Rdbuff = Space(8)                      ' Reserve a maximum of 8 bytes including the delimiter.
  Call ibrd(spa, Rdbuff)                 ' Read the data.
  state = Val(Rdbuff)                    ' Convert the character string into numeric values.

  DoEvents                               ' Check the loop for other events currently taking place.
Loop Until (state And 128)               ' Exit from the loop if the sweep-end bit is set to 1.
```

Example VB-19: Reading the peak frequency and level at the end of a single sweep (when using SRQ)

```
Dim boardID As Integer
Dim I As Integer
Dim res As Integer
Dim CFLEV As String

boardID = 0                              ' Set the board ID.

Call ibclr(spa)                          ' Performs a Device Clear.

Call ibwrt(spa, "SI")                    ' Turn the single sweep mode on.

Call ibwrt(spa, "*CLS")                  ' Clear the status byte.
Call ibwrt(spa, "OPR 8")                 ' Enables the Sweep-end bit of the operation status register
Call ibwrt(spa, "*SRE 128")              ' Enables the Operation status bit of the status byte.
Call ibwrt(spa, "S0")                    ' Specify Send mode for the SRQ signal.

For I = 1 To 10 Step 1                   ' A loop of 10 times
  Call ibwrt(spa, "SI")                  ' Begin sweeping
  Call WaitSRQ(boardID, res)             ' Wait until SRQ interruption occurs.
  Call ibrsp(spa, res)                   ' Execute serial polling.

  Call ibwrt(spa, "PS")                  ' Execute the peak search.
  Call ibwrt(spa, "MFL?")                ' Query for marker frequency and level

  Rdbuff = Space(43)                     ' Reserve 43 bytes.
  Call ibrd(spa, Rdbuff)                 ' Read the data.

  CFLEV = Left(Rdbuff, InStr(1, Rdbuff, Chr(13), 0) - 1)
  RichTextBox1.Text = RichTextBox1.Text & "Freq ,Lebel = " & CFLEV & vbCrLf
                                         ' Display data on the screen and start a new line.

  DoEvents                               ' Execute other events in Windows if any.
Next I
```

4.2.10.6 Example Program Used to Read Screen Data

Example VB-20: Outputting the current screen data in bitmap format and saving it into the file (bitmap.bmp)

NOTE: ***Depending on the copy image, compression of files and screen status, the amount of bitmap data varies. A data file of up to 300 KB can be output.***

```
Tmo%=14                                 ' A timeout of 30 sec.
Call ibtmo(spa,tmo%)                    ' A timeout of 30 seconds is set.
Call ibwrt(spa,"DL2")                   ' Selects only EOI as a delimiter.

Call ibwrt(spa,"HCIMAG SCOL")           ' Sets a simple color image to make a copy.
Call ibwrt(spa,"HCCMPRS OFF")           ' Turns the compression mode off.
Call ibwrt(spa,"BMP?")                  ' Requests the bitmap data output.
Call ibrdf(spa,"bitmap.bmp")            ' Saves the bitmap data into the file.

Call ibwrt(spa,"DL0")                   ' Changes the delimiter back to CR, LF and EOI.
```

4.3 RS-232 Remote Control Function

Most controllers (such as personal computers) do not have a GPIB interface, but the R3132 series can still be controlled using the RS-232 interface.

4.3.1 GPIB and RS-232 Compatibility

The control codes and functions are the same as those used for serial control, except for those which especially refer to the GPIB interface.

4.3.2 Features of RS-232 Remote Control

The following functions can be controlled by serial control.

- Measurement conditions setup: Measurement conditions each can be input in much the same as the key operation on the front panel.
- Output of the setup status: Both the setup status and data can be read out.
- Status: Status bytes which show the current status of the spectrum analyzer can be read out in the same way GPIB readouts.

4.3.3 Parameter Setup Window

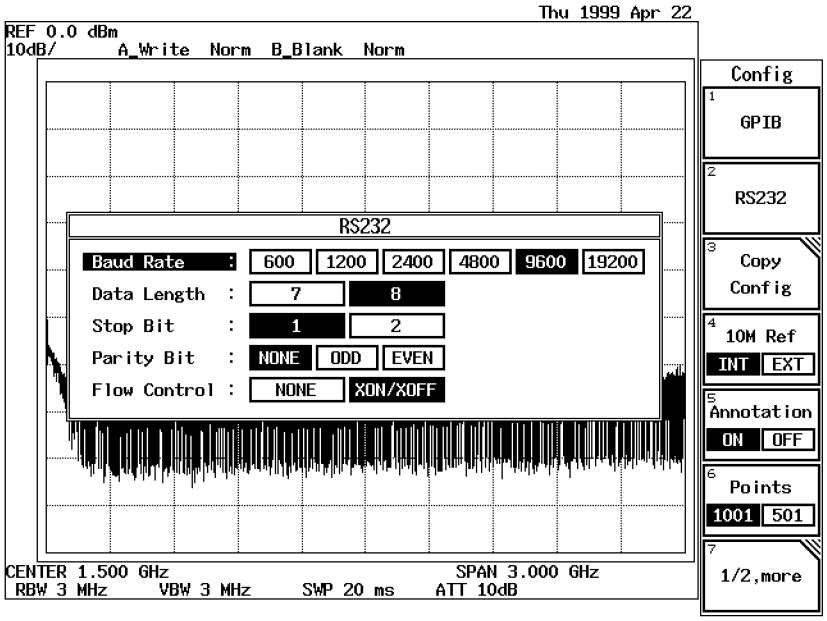


Figure 4-5 Parameter Setup

1. Baud Rate: Select from 600, 1200, 2400, 4800, 9600 or 19200.
2. Data length: Select seven bits or eight bits as the number of data bits.
3. Stop bit: Select one or two bits.
4. Parity check: Select from NONE, ODD or EVEN.
5. Flow control: Selects whether or not to use flow control XON/XOFF.

4.3.4 Interface connection

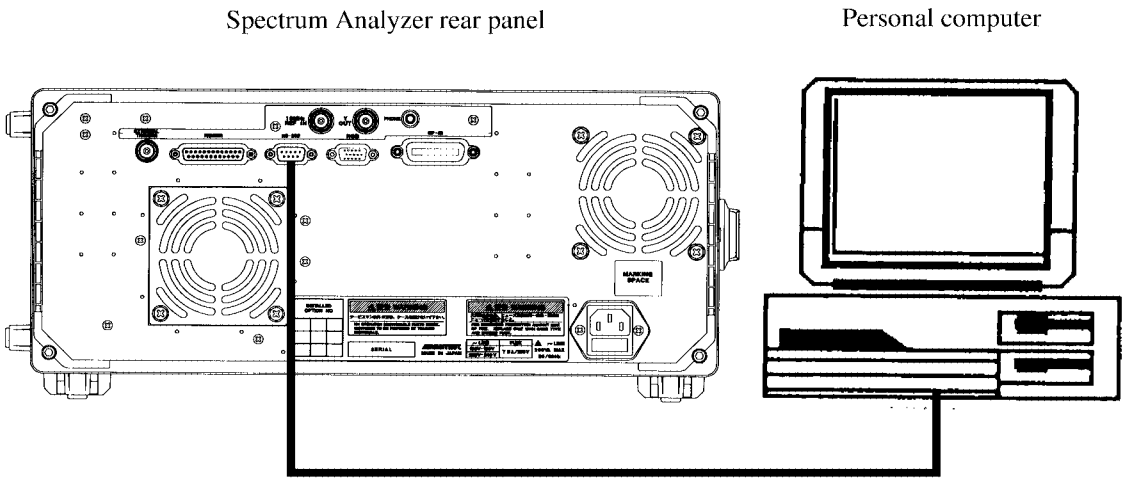


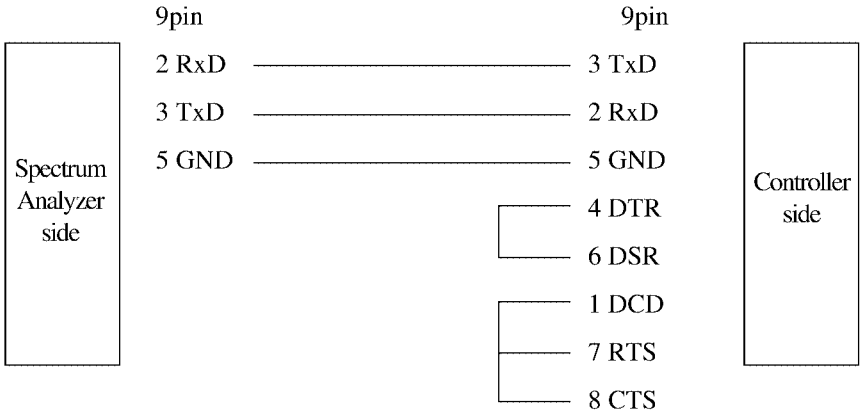
Figure 4-6 Connection Between the Controller and the Spectrum Analyzer

Although the spectrum analyzer uses only three pins, the controller side needs more connections for input and output.

NOTE:

1. When you send or receive data using the cable connections shown in Figure 4-6, set XON/XOFF to valid (ON).

2. DCD, DTR and DSR are not used in the spectrum analyzer. When you use CTS and RTS, use a cable with cross-connection to connect the controller to the spectrum analyzer. Flow control is not performed using CTS or RTS. Set XON/XOFF to valid (ON) to perform flow control.



Pin No.(9pin)	Signal name	Contents
1	DCD:Data Carrier Detector	Receive carrier detection
2	RxD: Receive Data	Receive data
3	TxD: Transmit Data	Transmission data
4	DTR: Data Terminal Ready	Data terminal ready
5	GND: Ground	Signal ground
6	DSR: Data set Ready	Data set ready
7	RTS: Request To Send	Request signal for sending
8	CTS: Clear to Send	Clear signal for sending
9	CI:	N.C

Figure 4-7 Cable Wiring Diagram

4.3.5 Data Format

Transmission messages between the spectrum analyzer and the controller are in ASCII code character strings and followed by carriage returns (CR) and line feeds (LF).

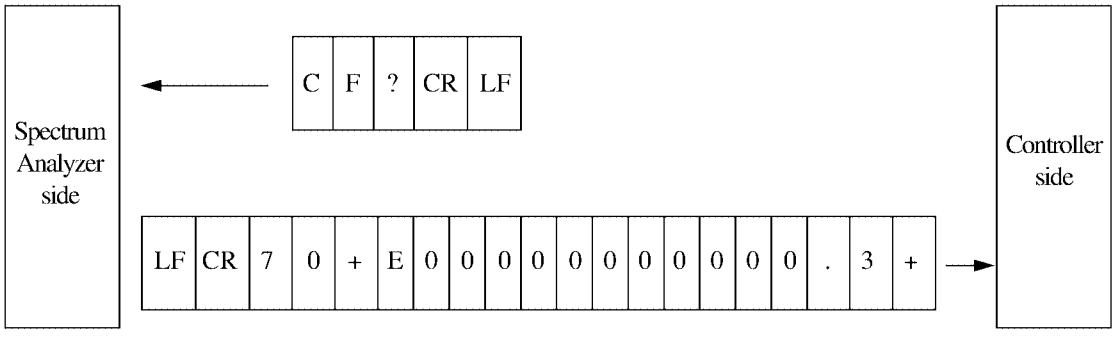


Figure 4-8 Data Format

NOTE:

1. ***Transmission data must be in ASCII code.***
2. ***Delimit the data from the controller with CR or CR and LF. Query data and the GPIB delimiters are the same. Therefore, send DL0 or DL3 after serial port was opened (refer to the example of RS-232 remote program).***

Data transmission example: Personal computers can recognize both CF 30.0MZ CR and CF 30.0MZ CR LF.

The format for query data is +3.00000000000E+07 CR LF (send DL0 or DL3). The output data of this RS-232 and GPIB are the same number of characters except delimiters (CR and LF).

4.3.6 Differences Between RS-232 and GPIB

- Command code

  Trace data can be input or output in the ASCII format only.

CAUTION ***The following commands are unavailable: TBA and TBB.***

4.3.7 Panel Control

During remote control operation, spectrum analyzer panel control is affected as follows.

- The remote lamp does not light.
- The key panel is not disabled.

CAUTION ***If any settings are changed during remote control, the operation of the spectrum analyzer may become unstable.***

4.3.8 Remote Control Usage Examples

The following examples show typical remote control commands, and are written in "Microsoft Quick Basic" (licensed by Microsoft Corporation).

The Open command statement OPEN" COM1: 9600, N, 8, 1, ASC" FOR RANDOM AS #1 shown below has the following characteristics: baud rate is 9600 bps, no parity, 8 bit data length, stop bit of 1, ASCII format and random access mode.

Example: This program is used to check the status byte register to see if the sweep has been completed.

```
OPEN "COM1:9600,N,8,1,ASC" FOR RANDOM AS #1
PRINT #1, "DL3"                          '  CR and LF are set as the GPIB delimiter.
PRINT #1, "SI"                           '  Single sweep is performed.
PRINT #1, "OPR8"                         '  Sweep completion bit in the GPIB operation register is set.
PRINT #1, "CLS"                          '  Clearing the status bytes.
PRINT #1, "SI"                           '  Single sweep is performed.
MEAS.LOOP:
PRINT #1, "*STB?"                        '  Read out the status bytes.
INPUT #1, STAT
IF (STAT AND 128) = 0 THEN GOTO MEAS.LOOP
PRINT #1, "PS"                           '  Peak search.
PRINT #1, "ML?"                          '  Read out the peak level.
INPUT#1,MLEVEL
PRINT MLEVEL
CLOSE #1
END
```