

**MS8604A**  
**Digital Mobile Radio**  
**Transmitter Tester**  
**Operation Manual**  
**Vol. 2**

**(GPIB Remote Control)**

**Second Edition**

**Measuring Instruments Division**  
**Measurement Group**

**ANRITSU CORPORATION**

**NOV.**  
**1994**

**M-W0682AE-2.0**

MS8604A Digital Mobile Radio  
Transmitter Tester  
Operation Manual  
Vol. 2  
(GPIB Remote Control)

December 1993 (First Edition)

October 1994 (Second Edition)

---

Copyright © 1994 by Anritsu Corporation

All rights reserved. No part of this manual may be reproduced  
without the prior written permission of Anritsu Corporation.

The contents of this manual may be changed without prior notice.

Printed in Japan

# SECTION 1

## GENERAL

This section outlines the GPIB functions of the MS8604A Digital Mobile Transmitter Tester.

### TABLE OF CONTENTS

1.1	GENERAL .....	1-3
1.2	Functions of GPIB .....	1-3
1.3	Functions of the GPIB with two ports .....	1-3
1.4	Examples of system configuration using GPIB 1/GPIB 2 .....	1-4
1.5	Specifications .....	1-5
1.5.1	Interface function .....	1-5
1.5.2	Device message list .....	1-7

(Blank)

## 1.1 GENERAL

The MS8604A Digital Mobile Transmitter Tester, when combined with an external controller, can automate your measurement system. For this purpose the MS8604A is equipped with a GPIB interface bus (IEEE std 488.2 1987) as a standard feature.

## 1.2 Functions of GPIB

The functions of the MS8604A GPIB are as follows:

- (1) Controls all functions except the power switch and some keys including the [ Local ] key
- (2) Reads out all setting conditions
- (3) Sets the GPIB address from the panel
- (4) Executes interrupts and serial polling
- (5) Configures the automatic measurement system when the MS8604A is combined with a personal computer and other measuring instruments
- (6) Configures the GPIB with two ports: GPIB 1 and GPIB 2

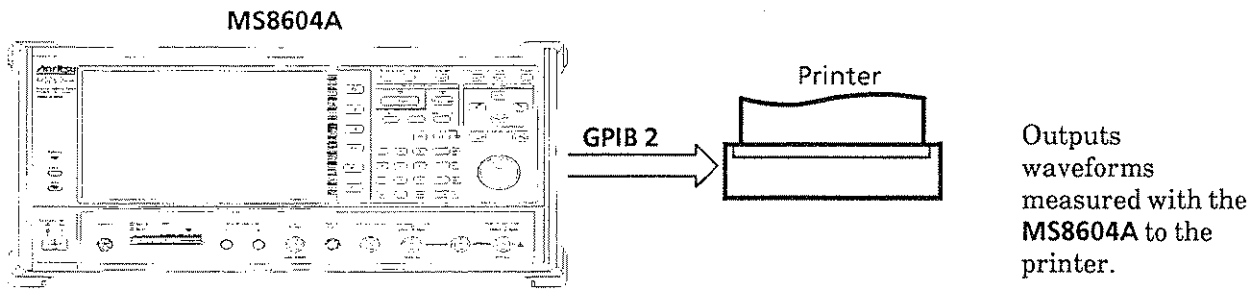
## 1.3 Functions of the GPIB with two ports

The MS8604A Digital Mobile Radio Transmitter Tester provides two GPIB ports, GPIB 1 and GPIB 2. The port on the GPIB 1 side is connected to an external controller to automate measurements by remote control, while the port on the GPIB 2 side is used to control peripherals such as printers, plotters, and other devices.

This Operation Manual describes the GPIB 1 port which is used for remote control. For hard-copying and other processing via the GPIB 2 port, refer to paragraph 4.7 in the Panel Operation Part of the separate Operation Manual.

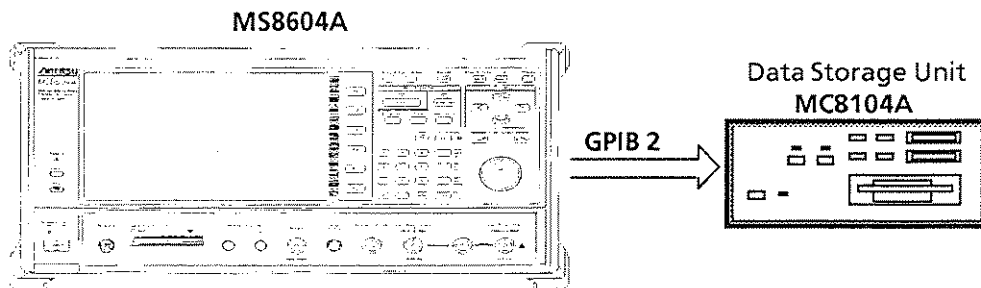
### 1.4 Examples of system configuration using GPIB 1/GPIB 2

(1) Stand-alone type 1.....Panel operation



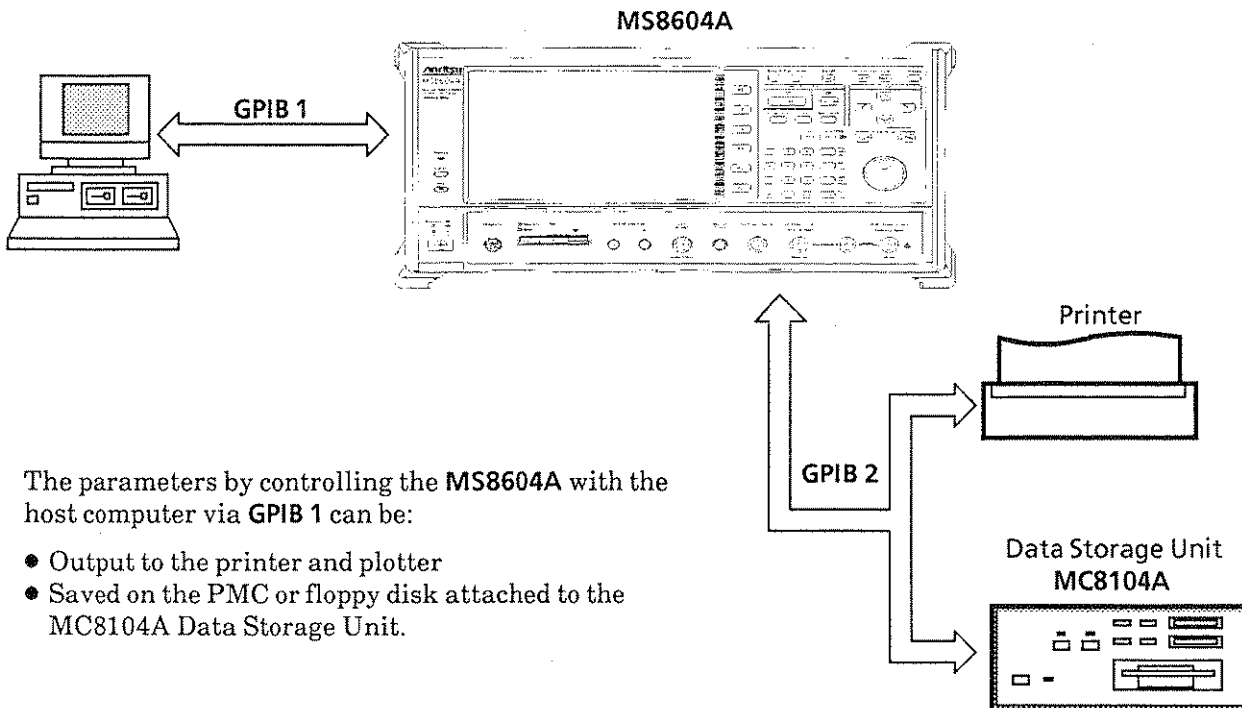
Outputs waveforms measured with the MS8604A to the printer.

(2) Stand-alone type 2.....Panel operation



Saves parameters by the MS8604A on the PMC or floppy disk attached to the MC8104A Data Storage Unit.

(3) Control by the host computer



The parameters by controlling the MS8604A with the host computer via GPIB 1 can be:

- Output to the printer and plotter
- Saved on the PMC or floppy disk attached to the MC8104A Data Storage Unit.

## 1.5 Specifications

### 1.5.1 Interface function

IEEE 488.2 determines the minimum required subsets for a measuring instrument from the GPIB interface functions specified in IEEE 488.1. It enables the subsets to be used for a system.

The MS8604A's GPIB 1 and GPIB 2 provide the IEEE 488.1 subsets listed in the code columns of the table below.

**GPIB 1 Interface Functions**

Code	Interface function
SH1	All source handshake functions are provided. Synchronizes the timing of data transmission.
AH1	All acceptor handshake functions are provided. Synchronizes the timing for receiving data.
T6	Basic talker functions are provided. The serial poll function is provided. The talk-only function is not provided. The talker can be canceled by MLA.
L4	Basic listener functions are provided. The listen-only function is not provided. The listener can be canceled by MTA.
SR1	All Service request and status byte functions are provided.
RL1	All Remote/local functions are provided. The local lockout function is provided.
PP0	Parallel poll functions are not provided.
DC1	All Device clear functions are provided.
DT1	Device trigger functions are provided.
C0	Controller functions are not provided. (When the PTA is not used)
C1 to C3 C24	Controller functions are provided. (When the PTA is used)

## GPIB 2 Interface Functions

Code	Interface Function
SH1	All source handshake functions are provided. Synchronizes the timing of data transmission.
AH1	All acceptor handshake functions are provided. Synchronizes the timing for receiving data.
T6	Basic talker functions are provided. Serial poll functions are provided. The talk-only function is not provided. A talker can be canceled by MLA.
L4	Basic listener functions are provided. The listen-only function is not provided. A listener can be canceled by MTA.
SR0	Service request and status byte functions are not provided.
RL0	Remote/local functions are not provided. Local lockout functions are not provided.
PP0	Parallel poll functions are not provided.
DC0	Device clear functions are not provided.
DT0	Device trigger functions are not provided.
C1 to C4, C28	Controller functions are provided.
E2	Tri-state output



### 1.5.2 Device message list

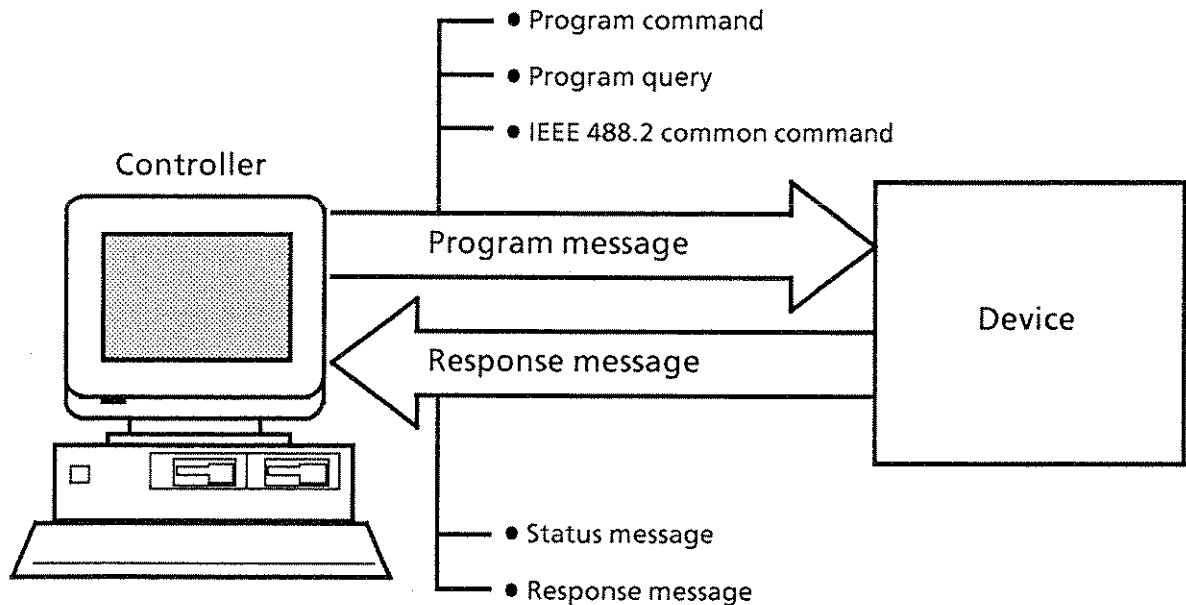
A device message is a data message transmitted between the controller and device via the system interface when the bus mode (ATN line) is false (FALSE, or 0). Device messages are classified into program messages and response messages.

A program message is an ASCII data message transferred from the controller to the device. Program messages are classified into program commands and program queries. These commands and queries are listed on the following pages.

Program commands are classified into device-specific commands used exclusively to control the MS8604A, and IEEE 488.2 common commands. IEEE 488.2 common commands include commands for the MS8604A, and are also used for other measuring instruments conforming to IEEE 488.2 connected to the bus.

A program query is a command used to obtain a response message from the device. It is transferred from the controller to the device in advance, then the controller receives the response message from the device.

A response message is an ASCII data message transferred from the device to the controller. This paragraph lists response messages.



Program messages and response messages may have a suffix (units) at the end of the numeric data.

The messages explained so far are transferred via the input and output buffers of the device. The output buffer is also called the output queue. The input buffer and output buffer (queue) are briefly explained below.

Input buffer	Output queue
The input buffer is a first-in first-out (FIFO) type memory area in which data bytes (DAB [program and query messages] ) to be executed are temporarily stored. A message is executed after its syntax has been analyzed. The MS8604A has a 256-byte input buffer.	The output queue is also a FIFO type memory area. All DABs (response messages) output from the device to the controller are stored in this memory until the controller reads them. The MS8604A has a 256-byte output queue.

## (1) GPIB Suffix Code

The table below shows the suffixes used for the MS8604A.

Table of MS8604A Suffix codes

Classification	Unit	Suffix code
Frequency	GHz	GHZ, GZ
	MHz	MHZ, MZ
	kHz	KHZ, KZ
	Hz	HZ
	Default	HZ
Time	second	S
	m second	MS
	$\mu$ second	US
	Default	MS
Level	dB	DB
	dBm	DBM, DM
	dB $\mu$ V	DBUV
	dBmV	DBMV
	dB $\mu$ V (emf)	DBUVE
	Default	Determined in conformance with the set scale unit
Level	V	V
	mV	MV
	$\mu$ V	UV
	Default	UV

## (2) IEEE 488.2 common commands and supported commands

The table below lists 39 types of common commands specified in the IEEE 488.2 standard. IEEE 488.2 common commands which are supported by the MS8604A are indicated with ☉ symbol in the table.

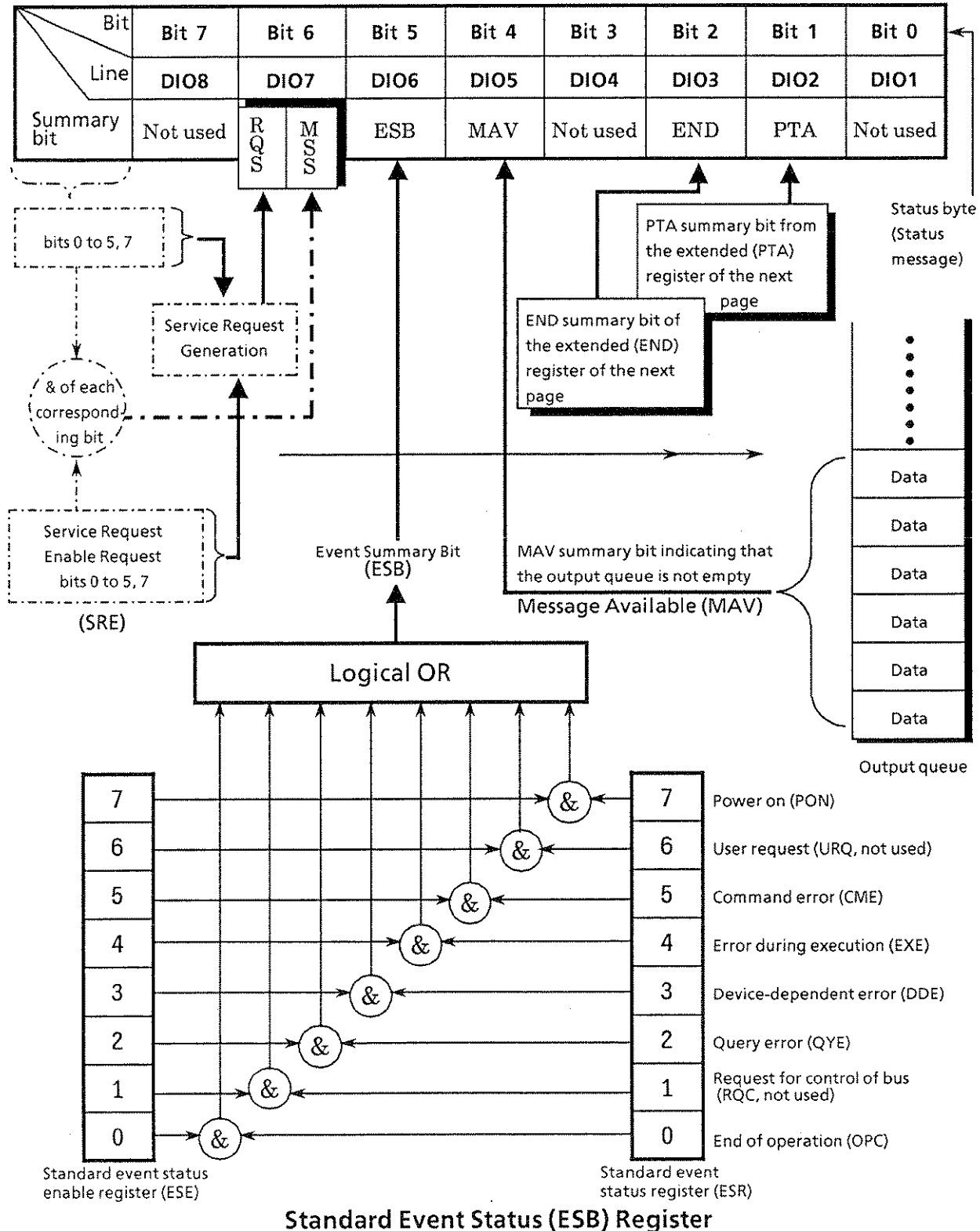
Mnemonic	Command name	IEEE 488.2 Standard	MS8604A supported commands
*AAD	Accept Address Command	Optional	
*CAL?	Calibration Query	Optional	
*CLS	Clear Status Command	Mandatory	☉
*DDT	Define Device Trigger Command	Optional	
*DDT?	Define Device Trigger Query	Optional	
*DLF	Disable Listener Function Command	Optional	
*DMC	Define Macro Command	Optional	
*EMC	Enable Macro Command	Optional	
*EMC?	Enable Macro Query	Optional	
*ESE	Standard Event Status Enable Command	Mandatory	☉
*ESE?	Standard Event Status Enable Query	Mandatory	☉
*ESR?	Standard Event Status Register Query	Mandatory	☉
*GMC?	Get Macro Contents Query	Optional	
*IDN?	Identification Query	Mandatory	☉
*IST?	Individual Status Query	Optional	
*LMC?	Learn Macro Query	Optional	
*LRN?	Learn Device Setup Query	Optional	
*OPC	Operation Complete Command	Mandatory	☉
*OPC?	Operation Complete Query	Mandatory	☉
*OPT?	Option Identification Query	Optional	
*PCB	Pass Control Back Command	Mandatory if other than CO.	☉
*PMC	Purge Macro Command	Optional	
*PRE	Parallel Poll Register Enable Command	Optional	
*PRE?	Parallel Poll Register Enable Query	Optional	
*PSC	Power On Status Clear Command	Optional	☉
*PSC?	Power On Status Clear Query	Optional	☉
*PUD	Protected User Data Command	Optional	
*PUD?	Protected User Data Query	Optional	
*RCL	Recall Command	Optional	
*RDT	Resource Description Transfer Command	Optional	
*RDT?	Resource Description Transfer Query	Optional	
*RST	Reset Command	Mandatory	☉
*SAV	Save Command	Optional	
*SRE	Service Request Enable Command	Mandatory	☉
*SRE?	Service Request Enable Query	Mandatory	☉

Mnemonic	Command name	IEEE 488.2 Standard	MS8604A supported commands
*STB?	Read Status Byte Query	Mandatory	<input type="radio"/>
*TRG	Trigger Command	Mandatory if DT1	<input type="radio"/>
*TST?	Self Test Query	Mandatory	<input type="radio"/>
*WAI	Wait to Continue Command	Mandatory	<input type="radio"/>

Note: The first letter of IEEE 488.2 common commands is always \*.

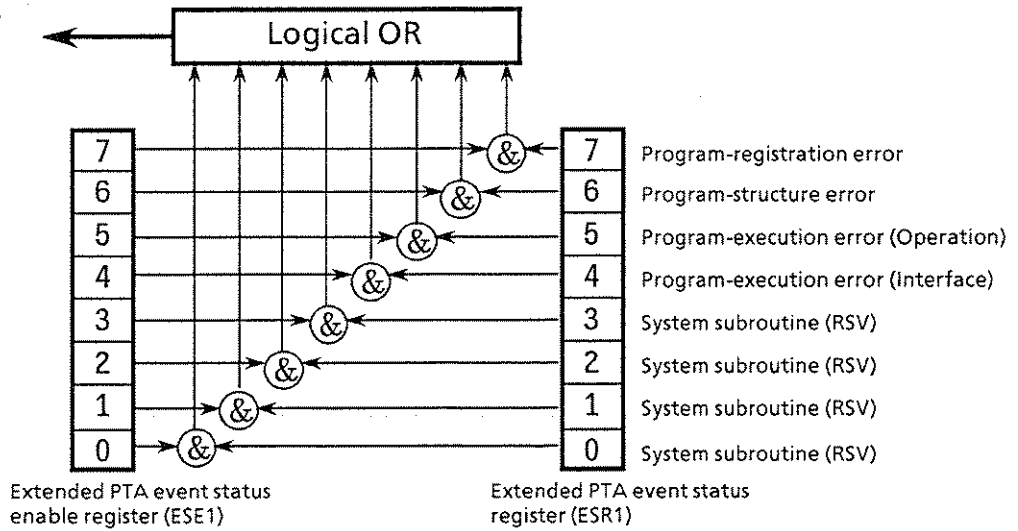
### (3) Status messages

The diagram below shows the structure of service-request summary messages for the status byte register (STB) used with the MS8604A.



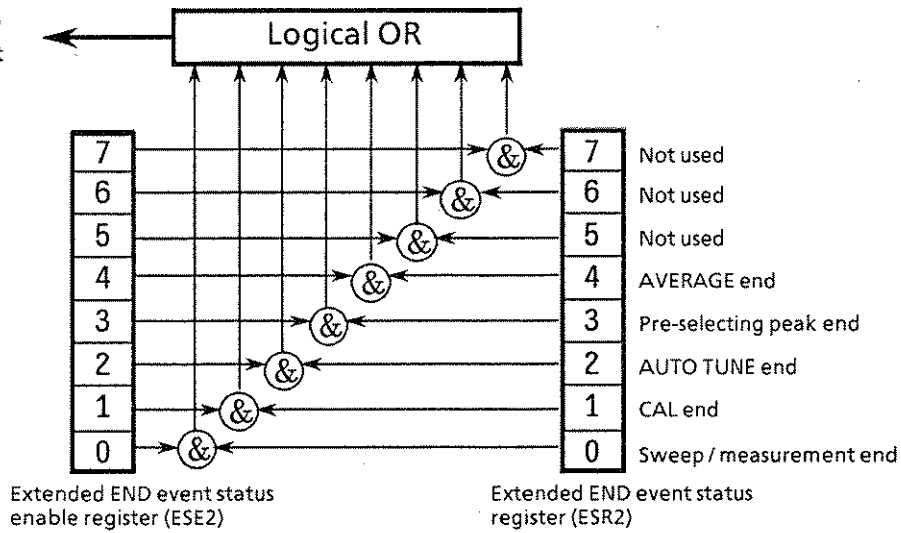
Note: & indicates a conjunction (AND).

To previous page  
PTA summary bit



Extended PTA event status (PTA) register

To previous page  
END summary bit



Extended END event status (END) register

## (4) MS8604A device message list

MS8604A-specific program commands, query messages, and response messages are listed from the next page.

- MS8604A device message table

- (a) Command messages (Command Msg)/query messages (Query Msg)

- (i) Uppercase characters : Reserved words
- (ii) Numeric : Reserved words (numeric code)
- (iii) Lower-case characters in argument
  - f (frequency) : Real number or integer with decimal point
    - Units : GHZ, MHZ, KHZ, HZ, GZ, MZ, KZ, no units = HZ
  - t (time) : Real number or integer with decimal point
    - Units : S, SC, MS, US, no unit = MS
  - ℓ (level) : Real number or integer with decimal point
    - Units : DB, DBM, DM, DBMV, DBUV, DBUVE, V, MV, UV, W, MW, UW, NW, no units = set SCALE units
  - s (symbol) : Real number or integer with decimal point
  - n (no units integer) : Integer
  - r (no units real number) : Real number
  - p (position data) : Integer indicating a trace data position
  - on (On/Off switch) : ON/OFF or 1/0 toggle switching data
  - i (Data Item No.) : Integer indicating the number of a data items
  - d (Data Count) : Integer indicating a number of data items

- (b) Response messages (Response Msg)

- (i) Uppercase characters : Reserved words
- (ii) Numeric : Reserved words (numeric code)
- (iii) Lower-case characters in argument
  - f (frequency) : 12-character fixed integer units = Hz
  - t (time) : Real number or integer with decimal point
  - ℓ (level) : Real number or integer with decimal point
  - u (ratio) : Real number or integer with decimal point
  - k (speed) : Real number or integer with decimal point
  - s (symbol) : Real number or integer with decimal point
  - n (no units integer) : Integer, variable number of digits (Significant digits are output.)
  - r (no units real number) : Real number with decimal point, variable number of digits (Significant digits are output.)
  - on (On/Off switch) : ON/OFF
  - a (flag) : Measurement result classification flag

Notes: ● Integer : NR1 format, real number : NR2 format

- Ø : Zero



## (a) MS8604A common command

## (i) Save/Recall command

Class	Function	Item	Program Msg	Query Msg	Response Msg
Save & Recall					
	Recall from memory	Memory 1	RGRC 1	_____	_____
		Memory 2	RGRC 2	_____	_____
		Memory 3	RGRC 3	_____	_____
		Memory 4	RGRC 4	_____	_____
	Mem-Dir		RGDIR	_____	_____
	Recall File	file-no	RCM n	_____	_____
	Select Media	Int PMC	PMCS IPMC	PMCS?	IPMC
		Ext PMC1	PMCS EPMC1	PMCS?	EPMC1
		Ext PMC2	PMCS EPMC2	PMCS?	EPMC2
		Ext FD	PMCS EFD	PMCS?	EFD
	Recalled Data	Trace & Param	RDATA TP	RDATA?	TP
		Param	RDATA P	RDATA?	P
	Save to memory	Memory 1	RGSV 1	_____	_____
		Memory 2	RGSV 2	_____	_____
		Memory 3	RGSV 3	_____	_____
Memory 4		RGSV 4	_____	_____	
Save File	file-no	SVM n			

## (ii) Hard Copy command

Class	Function	Item	Program Msg	Query Msg	Response Msg
Hard Copy					
	Copy		PLS 0	_____	_____
			PRINT	_____	_____
	<<Copy Control>>				
	Printer Type	HP2225	PMOD 3	PMOD?	PMOD 3
		ESC/P (8DOT)	PMOD 2	PMOD?	PMOD 2
		ESC/P (24DOT)	PMOD 6	PMOD?	PMOD 6
		uA455A	PMOD 5	PMOD?	PMOD 5
		MC8104A	PMOD 4	PMOD?	PMOD 4
Printer-address	address	PRIA n	PRIA?	n	

## (iii) Waveform memory read command

Class	Function	Item	Program Msg	Query Msg	Response Msg
Wave Memory Data					
	Mem A	TRACE-FREQ	XMA p,b	XMA? p,d	b,b,...
	Mem B	ADJ. CH:STADARD, OCC. BW:STANDARD	XMB p,b	XMB? p,d	b,b,...
	Mem Time	TRACE-TIME	XMT p,b	XMT? p,d	b,b,...
	Mem C	Constellation	XMC p0,p1,b	XMC? p0,p1,d	b,b,...
	Mem D	Power measurement	XMD p,b	XMD? p,d	b,b,...
	Mem E	HIGH SPEED OCC. BW	XME p,b	XME? p,d	b,b,...
	Output Format	ASCII	BIN 0	_____	_____
			BIN OFF	_____	_____
		BINARY	BIN 1	_____	_____
			BIN ON	_____	_____

## (iv) Single/Continuous switching command

Class	Function	Item	Program Msg	Query Msg	Response Msg
Single/Continuous					
	Single		SNGLS	_____	_____
		Measurement/sweep end	_____	SWP?	SWP 0
		During measurement/sweep	_____	SWP?	SWP 1
	Measurement /sweep sync	Single	SWP	_____	_____
			TS	_____	_____
	Continuous		CONTS	_____	_____

## (v) Panel mode switching command

Class	Function	Item	Program Msg	Query Msg	Response Msg
Panel Mode					
	TX Tester		PNLMD TESTER	PNLMD?	TESTER
	Spectrum		PNLMD SPECT	PNLMD?	SPECT
	System		PNLMD SYSTEM	PNLMD?	SYSTEM

## (vi) System command

Class	Function	Item	Program Msg	Query Msg	Response Msg	
System setup						
<TITLE>						
	User define		TITLE "....."	TITLE?	.....	
			KSE "....."	_____	_____	
	Display	Date/Time	TTL DATE	TTL?	DATE	
		User Define	TTL USER	TTL?	USER	
	off	TTL OFF	TTL?	OFF		
<SETUP DATE/TIME>						
	Date		DATE yy,mm,dd	DATE?	yy,mm,dd	
	Time		TIME hh,mm,ss	TIME?	hh,mm,ss	
<HARDWARE>						
	RF input	HIGH	RFINPUT HIGH	RFINPUT?	HIGH	
		LOW	RFINPUT LOW	RFINPUT?	LOW	
	Alarm	OFF		ALARM OFF	ALARM?	OFF
				BEP 0	_____	_____
			BEP OFF	_____	_____	
		ON	ALARM ON	ALARM?	ON	
		BEP 1	_____	_____		
		BEP ON	_____	_____		
<INTERFACE>						
	GPIB2 Address		GPIA n	GPIA?	n	
	MC8104A Address		MC8104 n	MC8104?	n	
	RS-232C (Option)					
	Baud rate	9600		BAUD 9600	BAUD?	9600
		4800		BAUD 4800	BAUD?	4800
		2400		BAUD 2400	BAUD?	2400
		1200		BAUD 1200	BAUD?	1200
		600		BAUD 600	BAUD?	600
		300		BAUD 300	BAUD?	300
	Parity	Even		PRTY EVEN	PRTY?	EVEN
		Odd		PRTY ODD	PRTY?	ODD
		Off		PRTY OFF	PRTY?	OFF
	Data	7Bits		DTAB 7	DTAB?	7
		8Bits		DTAB 8	DTAB?	8
	Stop Bit	1bit		STPB 1	STPB?	1
		1.5bits		STPB 1.5	STPB?	1.5
2bits			STPB 2	STPB?	2	
Timeout			TOUT t	TOUT?	t	

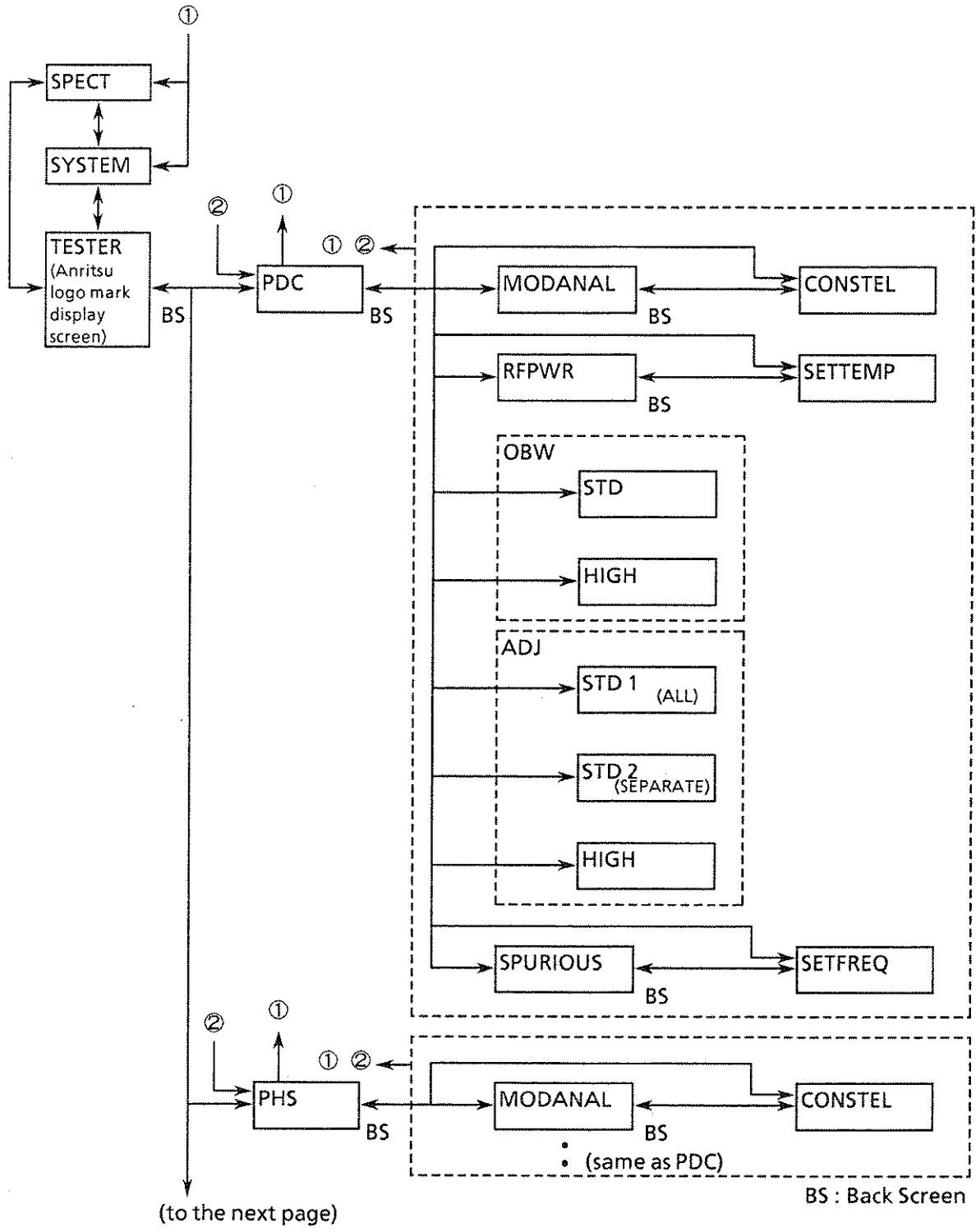
(vii) Other command

Class	Function	Item	Program Msg	Query Msg	Response Msg	
Other						
	Display	Off	DSPL OFF	=====	=====	
		On	DSPL ON	=====	=====	
	Buzzer		BZR	=====	=====	
	Terminater	LF	TRM 0	=====	=====	
		CR/LF	TRM 1	=====	=====	
	Initialize		PRE	=====	=====	
			INI	=====	=====	
			IP	=====	=====	
	GPIB Timeout		GTOUT t	GTOUT?	t	
	Extended status (PTA)	EVENT STATUS ENABLE	ESE1 n	=====	ESE1?	n
		EVENT STATUS REGISTER	=====	=====	ESR1?	n
	Extended status (END)	EVENT STATUS ENABLE	ESE2 n	=====	ESE2?	n
		EVENT STATUS REGISTER	=====	=====	ESR2?	n

## (b) TX Test mode command

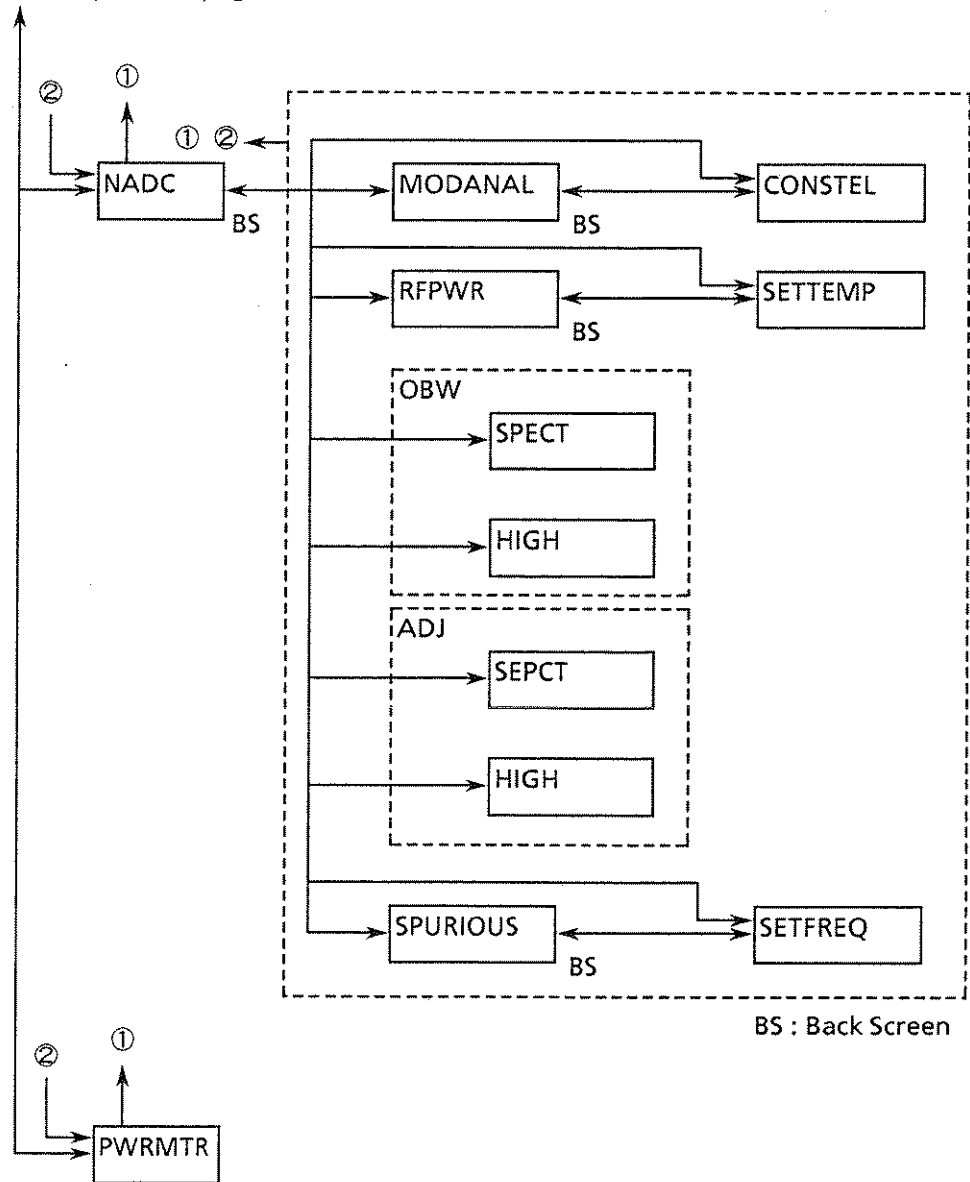
## (i) Screen level switching command and measurement status error information read command

Class	Function	Item	Program Msg	Query Msg	Response Msg
«System screen switching»					
	PDC Setup parameter		SYS PDC	SYS?	PDC
	PHS Setup parameter		SYS PHS	SYS?	PHP
	NADC Setup parameter		SYS NADC	SYS?	NADC
	Power Meter		SYS PWRMTR	SYS?	PWRMTR
«Measurement screen switching»					
	Modulation Analysis		MEAS MODANAL	MEAS?	MODANAL
	Constellation		MEAS CONSTEL	MEAS?	CONSTEL
	RF Power		MEAS RFPWR	MEAS?	RFPWR
	Setup Template		MEAS SETTEMP	MEAS?	SETTEMP
	Occupied Bandwidth	STD, SPECTRUM	MEAS OBW, STD	MEAS?	OBW, STD
		SPECTRUM	MEAS OBW, SPECT	MEAS?	OBW, SPECT
		HIGH SPEED	MEAS OBW, HIGH	MEAS?	OBW, HIGH
	Adjacent Channel Power	STD (ALL)	MEAS ADJ, STD1	MEAS?	ADJ, STD1
		STD (SEPARATE)	MEAS ADJ, STD2	MEAS?	ADJ, STD2
		Spect (ALL)	MEAS ADJ, SPECT1	MEAS?	ADJ, SPECT1
		Spect (SEPARATE)	MEAS ADJ, SPECT2	MEAS?	ADJ, SPECT2
		HIGH SPEED	MEAS ADJ, HIGH	MEAS?	ADJ, HIGH
	Spurious Emissions	Spot	MEAS SPURIOUS, SPOT	MEAS?	SPURIOUS, SPOT
		Search	MEAS SPURIOUS, SEARCH	MEAS?	SPURIOUS, SEARCH
	Setup Frequency Table		MEAS SETTABLE	MEAS?	SETTABLE
	Setup Sensor Cal Factor		MEAS SCALTBL	MEAS?	SCALTBL
	Back screen		BS	_____	_____
«Measurement status error information read»					
			_____	MSTAT?	n



Screen level command transition (1/2)

(from the previous page)



Screen level command transition (2/2)

## (ii) Command at each screen

Class	Function	Item	Program Msg	Query Msg	Response Msg	
Setup Parameter						
INPUT						
Terminal	RF		TERM RF	TERM?	RF	
	I, Q-AC		TERM IQAC	TERM?	IQAC	
	I, Q-DC		TERM IQDC	TERM?	IQDC	
RF level	-12 to 40		RFLVL ℓ	RFLVL?	ℓ	
	Step UP		RFLVL UP	_____	_____	
	Step DOWN		RFLVL DN	_____	_____	
SIGNAL						
Measuring object	MS-CONT (PDC)		MEASOBJ MSCONT	MEASOBJ?	MSCONT	
	MS-COM (PDC)		MEASOBJ MSCOM	MEASOBJ?	MSCOM	
	MS-SYNC (PDC)		MEASOBJ MSSYNC	MEASOBJ?	MSSYNC	
	BS-CONT (PDC)		MEASOBJ BSCONT	MEASOBJ?	BSCONT	
	BS-COM (PDC)		MEASOBJ BSCOM	MEASOBJ?	BSCOM	
	BS-SYNC (PDC)		MEASOBJ BSSYNC	MEASOBJ?	BSSYNC	
	PS-CONT (PHS)		MEASOBJ PSCONT	MEASOBJ?	PSCONT	
	PS-COM (PHS)		MEASOBJ PSCOM	MEASOBJ?	PSCOM	
	CS-CONT (PHS)		MEASOBJ CSCONT	MEASOBJ?	CSCONT	
	CS-COM (PHS)		MEASOBJ CSCOM	MEASOBJ?	CSCOM	
	CONTINUOUS (PHS)		MEASOBJ CONT	MEASOBJ?	CONT	
	MOBILE (NADC)		MEASOBJ MOBILE	MEASOBJ?	MOBILE	
	SHORTENED (NADC)		MEASOBJ SHORT	MEASOBJ?	SHORT	
	BASE (NADC)		MEASOBJ BASE	MEASOBJ?	BASE	
Ramp up symbol point		_____	RAMPASYM?	n		
Number of symbols		_____	NUMSYM?	n		
Channels per carrier	FULL RATE		CHCARR FULL	CHCARR?	FULL	
	HALF RATE		CHCARR HALF	CHCARR?	HALF	
Frequency						
Channel	1 to 9999		CHAN n	CHAN?	n	
	Step UP		CHAN UP	_____	_____	
	Step DOWN		CHAN DN	_____	_____	
Frequency			FREQ f	FREQ?	f	
			CNF f	CNF?	CNF f	
			CF f	CF?	f	
	Step UP			FREQ UP	_____	_____
				FUP	_____	_____
				CF UP	_____	_____
	Step DOWN			FREQ DN	_____	_____
				FDN	_____	_____
			CF DN	_____	_____	



(ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Setup Parameter (contd.)					
	Ch & Freq		CHFREQ n, f	_____	_____
		Channel (NADC)	CHNUM n	_____	_____
	Channel spacing		CHSPC f	CHSPC?	f
SYNC(UNIQUE) WORD					
	Pattern	S1 (PDC)	PATT S1	PATT?	S1
		S2 (PDC)	PATT S2	PATT?	S2
		S3 (PDC)	PATT S3	PATT?	S3
		S4 (PDC)	PATT S4	PATT?	S4
		S5 (PDC)	PATT S5	PATT?	S5
		S6 (PDC)	PATT S6	PATT?	S6
		S7 (PDC)	PATT S7	PATT?	S7
		S8 (PDC)	PATT S8	PATT?	S8
		S9 (PDC)	PATT S9	PATT?	S9
		S10 (PDC)	PATT S10	PATT?	S10
		S11 (PDC)	PATT S11	PATT?	S11
		S12 (PDC)	PATT S12	PATT?	S12
		SS1 (PDC)	PATT SS1	PATT?	SS1
		SS2 (PDC)	PATT SS2	PATT?	SS2
		SS3 (PDC)	PATT SS3	PATT?	SS3
		SS4 (PDC)	PATT SS4	PATT?	SS4
		SS5 (PDC)	PATT SS5	PATT?	SS5
		SS6 (PDC)	PATT SS6	PATT?	SS6
		32 bits (PHS)	PATT B32	PATT?	B32
		16 bits (PHS)	PATT B16	PATT?	B16
		Sync1 (NADC)	PATT SYNC1	PATT?	SYNC1
		Sync2 (NADC)	PATT SYNC2	PATT?	SYNC2
		Sync3 (NADC)	PATT SYNC3	PATT?	SYNC3
		Sync4 (NADC)	PATT SYNC4	PATT?	SYNC4
		Sync5 (NADC)	PATT SYNC5	PATT?	SYNC5
		Sync6 (NADC)	PATT SYNC6	PATT?	SYNC6
		NO	PATT NO	PATT?	NO
USER	PATT USER	PATT?	USER		
	BPATT bit Bit:0...0~1...1	BPATT?	0...0~1...1		
	Start point		STARTPT n	STARTPT?	n
	Transmit timing	OFF	TXTIME OFF	TXTIME?	OFF
		ON	TXTIME ON	TXTIME?	ON

(ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Setup Parameter (contd.)					
	Band	800 MHz	FBAND 800MHz	FBAND?	800MHz
			FBAND 800Mz	_____	_____
		1500 MHz	FBAND 1500MHz	FBAND?	1500MHz
			FBAND 1500Mz	_____	_____
	CS slot number	OFF	CSSL0T n,OFF	CSSL0T? n	OFF
		ON	CSSL0T n,ON	CSSL0T? n	ON
		active	CSNUM n	CSNUM?	n
	PS slot	OFF	PSSL0T OFF	PSSL0T?	OFF
		ON	PSSL0T ON	PSSL0T?	ON
	RCR STD-27 version	B	STD27 B	STD27?	B
		C	STD27 C	STD27?	C
	Root-Nyquist filter	Yes	RTNYQ YES	RTNYQ?	YES
			RTNYQ ON	_____	_____
		No	RTNYQ NO	RTNYQ?	NO
			RTNYQ OFF	_____	_____
	Symbol timing		SYMTIME n	SYMTIME?	n
Modulation Analysis					
	10 Burst average	On	BSTAVG ON	BSTAVG?	ON
		Off	BSTAVG OFF	BSTAVG?	OFF
	Calibration	Auto range	AUTORNG	AUTORNG?	ℓ
		Frequency adjust	FREQADJ	FREQADJ?	a, f
	Measurement result	Carrier frequency	_____	CARRF?	a, f
		Carrier frequency error	_____	CARRFERR?	a, f
		RMS vector error	_____	VECTERR?	a, u
		First 10 symboles RMS vector error	_____	FVECTERR?	a, u
		Peak vector error	_____	PVECTERR?	a, u
		Magnitude error	_____	MAGTDERR?	a, u
			_____	MAGERR?	a, u
		Phase error	_____	PHASEERR?	a, u
		Origin offset	_____	ORGN0FS?	a, ℓ
		Droop factor	_____	DRPFACT?	a, ℓ
		Bit rate	_____	BITR?	a, k
		Bit rate error	_____	BITRERR?	a, u
Peak vector symbol	_____	PVECTSYM?	a, s		

## (ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Constellation					
	Interpolation	NON-	INTPOL NON	INTPOL?	NON
		LINEAR-	INTPOL LIN	INTPOL?	LIN
		4POINT-	INTPOL POINT4	INTPOL?	POINT4
	Error scale	5%	ERRSC 5	ERRSC?	5
		10%	ERRSC 10	ERRSC?	10
		20%	ERRSC 20	ERRSC?	20
		Off	ERRSC OFF	ERRSC?	OFF
	Error scale offset	0	SCOFS 0	SCOFS?	0
		22.5	SCOFS 22.5	SCOFS?	22.5
	Calibration	Auto range	AUTORNG	AUTORNG?	ℓ
		Frequency adjust	FREQADJ	FREQADJ?	f
	Measurement result	Carrier freq error	———	CARRFERR?	a, f
		RMS vector error	———	VECTERR?	a, u
		Peak vector error	———	PVECTERR?	a, u
Origin offset		———	ORGNOSFS?	a, ℓ	

## (ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
RF Power					
	Window	Slot	WINDOW SLOT	WINDOW?	SLOT
		Frame	WINDOW FRAME	WINDOW?	FRAME
		Leading	WINDOW LEAD	WINDOW?	LEAD
			WINDOW RISE	_____	_____
		Trailing	WINDOW TRAIL	WINDOW?	TRAIL
			WINDOW FALL	_____	_____
	Marker	Normal	MKR NRM	MKR?	NRM
			MKR Ø	_____	_____
			M2	_____	_____
			MKN	_____	_____
		Off	MKR OFF	MKR?	OFF
			MKR 2	_____	_____
			MKOFF	_____	_____
			MKOFF ALL	_____	_____
			M1	_____	_____
			M1	_____	_____
	MKR position		MKP p	MKP?	p
			MKZ P	MKZ?	p
		Symbol	MKRS s	MKRS?	s
			MKZF s	MKZF?	s
			MKN s	MKN?	s
		Step UP	MKRS UP	_____	_____
			MKN UP	_____	_____
		Step DOWN	MKRS DN	_____	_____
			MKN DN	_____	_____
		MKR level		_____	MKL?
	Unit	dBm	UNIT DBM	UNIT?	DBM
			UNT Ø	UNT?	UNT Ø
AUNITS DBM			AUNITS?	DBM	
KSA			_____	_____	
nW/μW/mW/W		UNIT WATT	UNIT?	WATT	
		_____	UNT?	UNT WATT	
		_____	AUNITS?	WATT	
For other screen		_____	UNIT? RFPWR	DBM or WATT	
Level		Absolute	LVLREL OFF	LVLREL?	OFF
			MTEMPREL OFF	MTEMPREL?	OFF
	Relative	LVLREL ON	LVLREL?	ON	
		MTEMPREL ON	MTEMPREL?	ON	

## (ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
RF Power (contd.)						
	Storage mode	Normal	STORAGE NRM	STORAGE?	NRM	
		Max hold	STORAGE MAX	STORAGE?	MAX	
		Min hold	STORAGE MIN	STORAGE?	MIN	
		Average				
		ON	STORAGE AVG	STORAGE?	AVG	
			VAVG ON	_____	_____	
			VAVG 1	_____	_____	
		KSG	_____	_____	_____	
			OFF	VAVG OFF	_____	_____
			VAVG 0	_____	_____	
		KSH	_____	_____	_____	
		Cumulative	STORAGE CUM	STORAGE?	CUM	
		Over write	STORAGE OVER	STORAGE?	OVER	
		Wide dynamic range	STORAGE WIDE	STORAGE?	WIDE	
	For other screen	_____	STORAGE? RFPWR	NRM MAX MIN AVG CUM OVER WIDE		
	Average No.	2 to 99	AVR n	AVR?	n	
			VAVG n	VAVG?	n	
		For other screen	_____	AVR? RFPWR	n	
	Calibration	(See Modulation Analysis)				
	Select template	No.1	SLCTTEMP NO1	SLCTTEMP?	NO1	
		No.2	SLCTTEMP NO2	SLCTTEMP?	NO2	
		No.3	SLCTTEMP NO3	SLCTTEMP?	NO3	
		Standard	SLCTTEMP STD	SLCTTEMP?	STD	
		Off	SLCTTEMP OFF	SLCTTEMP?	OFF	
		Not selected	_____	SLCTTEMP?	NOT	
	Measurement result	TX power	_____	TXPWR?	a, ℓ depends on display units	
			_____	TXPWR? DBM	a, ℓ	
			_____	TXPWR? WATT	a, ℓ	
		Carrier OFF power	_____	OFFPWR?	a, ℓ depends on display units	
			_____	OFFPWR? DBM	a, ℓ	
			_____	OFFPWR? WATT	a, ℓ	
		ON/OFF ratio	_____	RATIO?	a, u	

## (ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
RF Power (contd.)						
	Measurement result (contd.)	Modulation power	_____	HMODPWR?	a, ℓ depends on display units	
			_____	HMODPWR? DBM	a, ℓ	
			_____	HMODPWR? WATT	a, ℓ	
		Timing	_____	TIMING?	a, s	
		Jitter (+)	_____	JITTER? +	a, s	
		Jitter (-)	_____	JITTER? -	a, s	
		Slot MOD power	_____	SLOTPWR? n	a, ℓ	
Setup template						
	Save template	No.1	SAVETEMP NO1	_____	NO1	
		No.2	SAVETEMP NO2	_____	NO2	
		No.3	SAVETEMP NO3	_____	NO3	
	Level Modify	Limit-1	TEMPLVL 1, ℓ	TEMPLVL? 1	ℓ	
		Limit-2	TEMPLVL 2, ℓ	TEMPLVL? 2	ℓ	
		Limit-3	TEMPLVL 3, ℓ	TEMPLVL? 3	ℓ	
		Limit-4	TEMPLVL 4, ℓ	TEMPLVL? 4	ℓ	
		Limit-5	TEMPLVL 5, ℓ	TEMPLVL? 5	ℓ	
	Occupied Bandwidth					
	Method	STD, SPECTRUM	MEAS OBW, STD	MEAS?	OBW, STD	
		SPECTRUM	MEAS OBW, SPECT	MEAS?	OBW, SPECT	
		HIGH SPEED	MEAS OBW, HIGH	MEAS?	OBW, HIGH	
	Storage mode	(See RF power : Normal and Average)				
		For other screen	_____	STORAGE? OBW	NRM AVG	
	Average No.	(See RF power)				
		For other screen	_____	AVR? OBW	n	
	Calibration	(See Modulation Analysis)				
	Measurement result	Occ. BW	_____	OCCBW?	a, f	
			_____	OBW?	a, f	
		CENTER	_____	OBWFREQ? CENTER	a, f	
		LOWER	_____	OBWFREQ? LOWER	a, f	
			_____	OBWFREQ? -	a, f	
		UPPER	_____	OBWFREQ? UPPER	a, f	
	_____	OBWFREQ? +	a, f			

(ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg		
Adjacent Chanel Power							
	Method	STD (ALL)	MEAS ADJ, STD1	MEAS?	ADJ, STD1		
		STD (SEPARATE)	MEAS ADJ, STD2	MEAS?	ADJ, STD2		
		Spect (ALL)	MEAS ADJ, SPECT1	MEAS?	ADJ, SPECT1		
		Spect (SEPARATE)	MEAS ADJ, SPECT2	MEAS?	ADJ, SPECT2		
		HIGH SPEED	MEAS ADJ, HIGH	MEAS?	ADJ, HIGH		
	Unit	dBm	UNIT DBM	UNIT DBM	UNIT?	DBM	
			UNIT Ø	UNIT Ø	UNIT?	UNIT Ø	
			AUNITS DBM	AUNITS DBM	AUNITS?	DBM	
			KSA	_____	_____	_____	
		mW	UNIT MW	UNIT MW	UNIT?	MW	
			_____	_____	UNIT?	UNIT MW	
			_____	_____	AUNITS?	MW	
		µW	UNIT UW	UNIT UW	UNIT?	UW	
			_____	_____	UNIT?	UNIT UW	
			_____	_____	AUNITS?	UW	
		nW	UNIT NW	UNIT NW	UNIT?	NW	
			_____	_____	UNIT?	UNIT NW	
			_____	_____	AUNITS?	NW	
		dB	UNIT DB	UNIT DB	UNIT?	DB	
			_____	_____	UNIT?	UNIT DB	
			_____	_____	AUNITS?	DB	
		For other screen	_____	_____	UNIT? ADJ	DBM MW UW NW DB	
		Storage mode	(See RF power)	_____	_____	_____	_____
			For other screen	_____	_____	STORAGE?ADJ	NRM AVG
		Average No.	(See Occupied Bandwidth)	_____	_____	_____	_____
			For other screen	_____	_____	AVR? ADJ	n
		Calibration	(See Modulation Analysis)	_____	_____	_____	_____
		Marker	OFF	ADJMEAS OFF	ADJMEAS OFF	ADJMEAS?	OFF
			ON	ADJMEAS ON	ADJMEAS ON	ADJMEAS?	ON

(ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Adjacent Chanel Power (contd.)					
	MKR position		MKP p	MKP?	p
			MKZ p	MKP?	p
		Frequency	MKRS f	MKRS?	f
			MKZF f	MKZF?	f
			MKN f	MKN?	f
		step UP	MKRS UP	_____	_____
			MKN UP	_____	_____
		step DOWN	MKRS DN	_____	_____
	MKN DN		_____	_____	
	MKR level		_____	MKL?	ℓ
	Measurement result	Lower-3	_____	ADJCH? LOW3	a, ℓ
		Lower-2	_____	ADJCH? LOW2	a, ℓ
Lower-1		_____	ADJCH? LOW1	a, ℓ	
Upper-1		_____	ADJCH? UP1	a, ℓ	
Upper-2		_____	ADJCH? UP2	a, ℓ	
Upper-3		_____	ADJCH? UP3	a, ℓ	
			_____	ADJCH? ps, un	a, ℓ
Peak Power (HIGH)	Lower-3 to Upper-3	_____	PEAKPWR? ps	a, ℓ depends on display units.	
Peak Power from Switching Transients (HIGH)		_____	PEAKPWR? ps, un	a, ℓ	
Mean Power (HIGH)	Lower-2 to Upper-2	_____	MEANPWR? ps	a, ℓ depends on display units.	
		_____	MEANPWR? ps, un	a, ℓ	
Mean Power due to Modulation (HIGH)	Lower-3 to Upper-3	_____	MODPWR? ps	a, ℓ depends on display units.	
		_____	MODPWR? ps, un	a, ℓ	
Peak Power from Switching Transients (HIGH)	Lower-3	_____	SWPWR? ps	a, ℓ depends on display units.	
	Upper-3	_____	SWPWR? ps, un	a, ℓ	
			ps:LOW3, LOW2, LOW1, UP1, UP2, UP3 un:DBM, WATT, DB		



## (ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
<b>Spurious Emissions</b>						
	Select frequency table	No.1	SLCTTBL NO1	SLCTTBL?	NO1	
		No.2	SLCTTBL NO2	SLCTTBL?	NO2	
		No.3	SLCTTBL NO3	SLCTTBL?	NO3	
		Not selected	_____	SLCTTBL?	NOT	
	Unit	(See Adjacent CH Power)				
		For other screen	_____	UNIT? SPURIOUS	DBM MW UW NW DB	
	Calibration	(See Modulation Analysis)				
	RF Attenuation	Normal	RFATTN NRM	RFATTN?	NRM	
		Noise	RFATTN NOISE	RFATTN?	NOISE	
	Measurement result	f1 to f15	_____	SPULVL? fn	a, ℓ depends on display units.	
			_____	SPULVL? fn, un	a, ℓ	
			fn:F1 to F15,un:DBM,WATT,DB			
	Spurious mode	SPOT	MEAS SPURIOUS,SPOT	MEAS?	SPURIOUS,SPOT	
		SEARCH	MEAS SPURIOUS,SEARCH	MEAS?	SPURIOUS,SEARCH	
	<b>Setup Frequency Table</b>					
	Save frequency table	No.1	SAVETBL NO1	_____	_____	
		No.2	SAVETBL NO2	_____	_____	
		No.3	SAVETBL NO3	_____	_____	
	Frequency Modify	f1	SPUFREQ F1,f	SPUFREQ? F1	f	
		f2	SPUFREQ F2,f	SPUFREQ? F2	f	
		f3	SPUFREQ F3,f	SPUFREQ? F3	f	
		f4	SPUFREQ F4,f	SPUFREQ? F4	f	
		F5	SPUFREQ F5,f	SPUFREQ? F5	f	
		f6	SPUFREQ F6,f	SPUFREQ? F6	f	
		f7	SPUFREQ F7,f	SPUFREQ? F7	f	
		f8	SPUFREQ F8,f	SPUFREQ? F8	f	
		f9	SPUFREQ F9,f	SPUFREQ? F9	f	
		f10	SPUFREQ F10,f	SPUFREQ? F10	f	
		f11	SPUFREQ F11,f	SPUFREQ? F11	f	
		f12	SPUFREQ F12,f	SPUFREQ? F12	f	
		f13	SPUFREQ F13,f	SPUFREQ? F13	f	
		f14	SPUFREQ F14,f	SPUFREQ? F14	f	
		f15	SPUFREQ F15,f	SPUFREQ? F15	f	
		Cancel	SPUFREQ fn,0 (fn:F1 to F15)	_____	_____	
	Harmonics	SPUFREQ HRM	_____	_____		
SPUHARM		_____	_____			

(ii) Command at each screen (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Power Meter					
	Frequency		FREQ f	FREQ?	f
			CNF f	CNF?	CNF f
			CF f	CF?	f
		Step UP	FREQ UP	_____	_____
			FUP	_____	_____
			CF UP	_____	_____
		Step DOWN	FREQ DN	_____	_____
			FDN	_____	_____
			CF DN	_____	_____
		Set relative		SETREL	_____
	DBR			_____	_____
	Sensor connect to	Tester	SCNCT TESTER	SCNCT?	TESTER
		DUT	SCNCT DUT	SCNCT?	DUT
	Cal factor	Sensor	_____	CSCAL?	ℓ
		User	UCAL ℓ	UCAL?	ℓ
OFF ℓ			_____	_____	
Reference		_____	CRCAL?	ℓ	
Cal oscillator	On	CALOSC ON	CALOSC?	ON	
		COS	_____	_____	
	Off	CALOSC OFF	CALOSC?	OFF	
		CST	_____	_____	
Cal adjust		CALADJ	_____	_____	
		CDJ	_____	_____	
Zero set		ZEROSET	_____	_____	
		ZAJ	_____	_____	
Range	Auto	RNG AUTO	RNG?	AUTO	
		RNGA	_____	_____	
	Hold	RNG HOLD	RNG?	HOLD	
	Hold (0.1 mW)	RNG2	_____	_____	
	Hold (1 mW)	RNG3	_____	_____	
	Hold (10 mW)	RNG4	_____	_____	
	Hold (100 mW)	RNG5	_____	_____	
Measurement result	Power	_____	POWER? DBM	ℓ	
		_____	POWER? WATT	ℓ	
		_____	POWER? DB	ℓ	
Setup Sensor Cal Factor					
	Cal Factor		SCALT f, ℓ	SCALT? f	ℓ

## (c) Spectrum mode command

Class	Function	Item	Program Msg	Query Msg	Response Msg
Signal Search					
	Auto Tune		ATUN	_____	_____
	Peak to CF		PCF	_____	_____
	Peak to REF		PRL	_____	_____
Freq/Amplitude					
	<<Frequency>>				
Freq Mode	Center-Span		FRQ 0	FRQ?	FRQ 0
	Start-Span		FRQ 1	FRQ?	FRQ 1
	Start-Stop		FRQ 2	FRQ?	FRQ 2
Center Freq			CNF f	CNF?	CNF f
			CF f	CF?	f
Freq step UP			FUP	_____	_____
			CF UP	_____	_____
Freq step DOWN			FDN	_____	_____
			CF DN	_____	_____
Start Freq			STF f	STF?	STF f
			FA f	FA?	f
Stop Freq			SOF f	SOF?	SOF f
			FB f	FB?	f
Freq Step Size			FSS f	FSS?	FSS f
			SS f	SS?	f
<<Span>>					
Freq Span			SPF f	SPF?	SPF f
			SP f	SP?	f
Freq-Span step UP			SPU	_____	_____
			SP UP	_____	_____
Freq-Span step DOWN			SPD	_____	_____
			SP DN	_____	_____
Full Span			FS	_____	_____
Zero Span			SPF 0	_____	_____
Band Select	Auto: 0 Hz to 8.5 GHz		BNDC AUTO	BNDC?	AUTO
			HNLOCK OFF	HNLOCK?	OFF
			HNUNLK	_____	_____

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
Freq/Amplitude (contd.)						
	Band Select (contd.)	Band 0 : 0 Hz to 2 GHz	BNDC 0	BNDC?	0	
			HNLOCK 0	HNLOCK?	ON	
			HN 0	HN?	0	
		Band 1 - : 1.7 to 8.5 GHz	BNDC 1-	BNDC?	1-	
			HNLOCK 1	HNLOCK ?	ON	
			HN 1	HN?	1	
		Band 1 + : 6.5 to 8.5 GHz	BNDC 1+	BNDC?	1+	
			HNLOCK 2	HNLOCK?	ON	
			HN 2	HN?	2	
	<<Amplitude>>					
	REF Level		RLV <i>ℓ</i>	RLV?	RLV <i>ℓ</i>	
			RL <i>ℓ</i>	RL?	<i>ℓ</i>	
	REF-level step UP		RL UP	_____	_____	
	REF-level step DOWN		RL DN	_____	_____	
	Log Scale Step Size	Manual	LSS <i>ℓ</i>	LSS?	LSS <i>ℓ</i>	
		AUTO: 1 div	LSSA 1	LSSA?	LSSA 1	
		2 div	LSSA 2	LSSA?	LSSA 2	
		5 div	LSSA 5	LSSA?	LSSA 5	
		10 div	LSSA 10	LSSA?	LSSA 10	
	Log Scale Range	1 dB/Div	SCL 0	SCL?	SCL 0	
			LG 10B	LG?	1	
2 dB/Div		SCL 1	SCL?	SCL 1		
		LG 2DB	LG?	2		
5 dB/Div		SCL 2	SCL?	SCL 2		
		LG 5DB	LG?	5		
10 dB/Div		SCL 3	SCL?	SCL 3		
		LG 10DB	LG?	10		
Scale UP		LG UP	_____	_____		
Scale DOWN		LG DN	_____	_____		
Linear Scale Range	Lin scale	LN	_____	_____		
		LG 0	_____	_____		
	1 %/Div	SCL 4	SCL?	SCL 4		
	2 %/Div	SCL 5	SCL?	SCL 5		
	5 %/Div	SCL 6	SCL?	SCL 6		
	10 %/Div	SCL 7	SCL?	SCL 7		

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
	Log Unit	dBm	UNT 0	UNT?	UNT 0
			AUNITS DBM	AUNITS?	DBM
			KSA	_____	_____
		dB $\mu$ V	UNT 1	UNT?	UNT 1
			AUNITS DBUV	AUNITS?	DBUV
			KSC	_____	_____
		dBmV	UNT 2	UNT?	UNT 2
			AUNITS DBMV	AUNITS?	DBMV
			KSB	_____	_____
		V	UNT 3	UNT?	UNT 3
			AUNITS V	AUNITS?	V
			KSD	_____	_____
		dB $\mu$ V (emf)	UNT 4	UNT?	UNT 4
			AUNITS DBUVE	AUNITS?	DBUVE
Sweep Control					
	Zone Sweep	Off	PSW 0	_____	_____
			PSW OFF	PSW?	PSW OFF
		On	PSW 1	_____	_____
			PSW ON	PSW?	PSW?
	Tracking	Off	MKTRACK 0	_____	_____
			MKTRACK OFF	MKTRACK?	OFF
			MT 0	_____	_____
		On	MKTRACK 1	_____	_____
			MKTRACK ON	MKTRACK?	ON
			MT1	_____	_____
	Gate	Off	GATE 0	_____	_____
			GATE OFF	GATE?	OFF
		On	GATE 1	_____	_____
			GATE ON	GATE?	ON
	Gate Delay		GD t	GD?	t
	Gate Length		GL t	GL?	t
	Gate End	Int	GE INT	GE?	INT
		Ext	GE EXT	GE?	EXT
Trigger SW	Freerun	TRGS FREE	TRGS?	FREE	
	Triggered	TRGS TRGD	TRGS?	TRGD	

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
Sweep Control (contd.)						
	Trigger Mode	Freerun	TRG 0	TRG?	TRG 0	
			TM FREE	TM?	FREE	
		Video	TRG 1	TRG?	TRG 1	
			TM VID	TM?	VID	
		Line	TRG 2	TRG?	TRG 2	
			TM LINE	TM?	LINE	
		Ext	TRG 3	TRG?	TRG 3	
			TM EXT	TM?	EXT	
		TV	TRG 6	TRG?	TRG 6	
			TM TV	TM?	TV	
		Trigger Source	Video	TRGSOURCE VID	TRGSOURCE?	VID
			Line	TRGSOURCE LINE	TRGSOURCE?	LINE
	Ext		TRGSOURCE EXT	TRGSOURCE?	EXT	
	TV		TRGSOURCE TV	TRGSOURCE?	TV	
	Ext Trigger Type	Input 1 ( $\pm 10$ V)	EXTTYPE 10V	EXTTYPE?	10V	
		Input 2 (TTL)	EXTTYPE TTL	EXTTYPE?	TTL	
	TV Type	PAL	TVSTND PAL	TVSTND?	PAL	
		NTSC	TVSTND NTSC	TVSTND?	NTSC	
	TV-Sync	V-sync	TVSFRM VERTICAL	TVSFRM?	VERTICAL	
		H-sync (even)	TVSFRM EVEN	TVSFRM?	EVEN	
		H-sync (odd)	TVSFRM ODD	TVSFRM?	ODD	
	TV (H-Sync line)		TVLINE n	TVLINE?	n	
	Trigger Level		TRGLVL r	TRGLVL?	r	
	Trigger Slope	Rise	TRGSLP RISE	TRGSLP?	RISE	
Fall		TRGSLP FALL	TRGSLP?	FALL		
(Read/Write SW)						
	Trace-Freq Write Switch	View	AWR 0	_____	_____	
			AWR OFF	AWR?	AWR OFF	
			VIEW TRA	_____	_____	
		Write	AWR 1	_____	_____	
			AWR ON	AWR?	AWR ON	
			CLRW TRA	_____	_____	
	Trace-Time Write Switch	View	TMWR 0	_____	_____	
			TMWR OFF	TMWR?	TMWR OFF	
			VIEW TRTIME	_____	_____	
		Write	TMWR 1	_____	_____	
			TMWR ON	TMWR?	TMWR ON	
			CLRW TRTIME	_____	_____	

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
Coupled Function (RBW/VBW/SWT/ATTEN)						
	RBW	Manual	ARB 0	ARB?	ARB 0	
		Auto	ARB 1	ARB?	ARB 1	
			RB AUTO	_____	_____	
			CR	_____	_____	
	RBW (Manual)	10 Hz	RB 10HZ	RB?	10	
			RBW 13	RBW?	RBW 13	
		30 Hz	RB 30HZ	RB?	30	
			RBW 0	RBW?	RBW 0	
		100 Hz	RB 100HZ	RB?	100	
			RBW 1	RBW?	RBW 1	
		300 Hz	RB 300HZ	RB?	300	
			RBW 2	RBW?	RBW 2	
		1 kHz	RB 1KHZ	RB?	1000	
			RBW 3	RBW?	RBW 3	
		3 kHz	RB 3KHZ	RB?	3000	
			RBW 4	RBW?	RBW 4	
		10 kHz	RB 10KHZ	RB?	10000	
			RBW 5	RBW?	RBW 5	
		30 kHz	RB 30KHZ	RB?	30000	
			RBW 6	RBW?	RBW 6	
		100 kHz	RB 100KHZ	RB?	100000	
			RBW 7	RBW?	RBW 7	
		300 kHz	RB 300KHZ	RB?	300000	
			RBW 8	RBW?	RBW 8	
		1 MHz	RB 1MHZ	RB?	1000000	
			RBW 9	RBW?	RBW 9	
		3 MHz	RB 3MHZ	RB?	3000000	
			RBW 14	RBW?	RBW 14	
			Step UP	RB UP	_____	_____
			Step DOWN	RB DN	_____	_____
		VBW	Manual	AVB 0	AVB?	AVB 0
			AUTO	AVB 1	AVB?	AVB 1
	VB AUTO			_____	_____	
CV	_____			_____		

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Coupled Function (RBW/VBW/SWT/ATTEN) (contd.)					
	VBW (Manual)	1 Hz	VB 1HZ	VB?	1
			VBW 0	VBW?	VBW 0
		3 Hz	VB 3HZ	VB?	3
			VBW 8	VBW?	VBW 8
		10 Hz	VB 10HZ	VB?	10
			VBW 1	VBW?	VBW 1
		30 Hz	VB 30HZ	VB?	30
			VBW 9	VBW?	VBW 9
		100 Hz	VB 100HZ	VB?	100
			VBW 2	VBW?	VBW 2
		300 Hz	VB 300HZ	VB?	300
			VBW 10	VBW?	VBW 10
		1 kHz	VB 1KHZ	VB?	1000
			VBW 3	VBW?	VBW 3
		3 kHz	VB 3KHZ	VB?	3000
			VBW 11	VBW?	VBW 11
		10 kHz	VB 10KHZ	VB?	10000
			VBW 4	VBW?	VBW 4
		30 kHz	VB 30KHZ	VB?	30000
			VBW 12	VBW?	VBW 12
		100 kHz	VB 100KHZ	VB?	100000
			VBW 5	VBW?	VBW 5
		300 kHz	VB 300KHZ	VB?	300000
			VBW 13	VBW?	VBW 13
		1 MHz	VB 1MHZ	VB?	1000000
			VBW 7	VBW?	VBW 7
		3 MHz	VB 3MHZ	VB?	3000000
			VBW 14	VBW?	VBW 14
		Off	VB OFF	VB?	OFF
			VBW 6	VBW?	VBW 6
	AVB 2		AVB?	AVB 2	
		Step UP	VB UP	_____	_____
		Step DOWN	VB DN	_____	_____
	VBW/RBW ratio	Ratio	VBR r	VBR?	r
SWT	Manual	AST 0	AST 0	AST?	AST 0
		AST 1	AST 1	AST?	AST 1
	AUTO	ST AUTO	_____	_____	
		CT	_____	_____	



## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Coupled Function (RBW/VBW/SWT/ATTEN) (contd.)					
	SWT (Manual)	Time	SWT t	SWT?	SWT t
			ST t	ST?	t
		Step UP	ST UP	_____	_____
		Step DOWN	ST DN	_____	_____
	ATTEN	Manual	AAT 0	AAT?	AAT 0
		Auto	AAT 1	AAT?	AAT 1
			AT AUTO	_____	_____
		CA	_____	_____	
	ATTEN (Manual)	0 to 11	ATT n	ATT?	ATT n
		0 dB	AT 0	AT?	0
		5 dB	AT 5	AT?	5
		10 dB	AT 10	AT?	10
		15 dB	AT 15	AT?	15
		20 dB	AT 20	AT?	20
		25 dB	AT 25	AT?	25
		30 dB	AT 30	AT?	30
		35 dB	AT 35	AT?	35
		40 dB	AT 40	AT?	40
		45 dB	AT 45	AT?	45
		50 dB	AT 50	AT?	50
		55 dB	AT 55	AT?	55
		60 dB	AT 60	AT?	60
		65 dB	AT 65	AT?	65
		70 dB	AT 70	AT?	70
		75 dB	AT 75	AT?	75
		Step UP	AT UP	_____	_____
		Step DOWN	AT DN	_____	_____
BW/SWT Auto		BSAUTO	_____	_____	
ALL Auto		AUTO	_____	_____	

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Marker					
	Marker Mode	Normal	MKR Ø	MKR?	MKR Ø
			M2	_____	_____
			Delta	MKR 1	MKR?
		MKD	_____	_____	
		M3	_____	_____	
		Off	MKR 2	MKR?	MKR 2
		MKOFF	_____	_____	
		MKOFF ALL	_____	_____	
		M1	_____	_____	
		Zone Center	Position	MKZ p	MKZ?
	MKP p			MKP?	p
	Freq set		MKZF f	MKZF?	f
			MKN f	MKN?	f
	Time set		MKZF t	MKZF?	t
			MKN t	MKN?	t
	Step UP		MKN UP	_____	_____
	Step DOWN		MKN DN	_____	_____
	Zone Width	Frequency	MZWF f	MZWF?	f
		Points	MZW p	MZW?	MZW p
		Spot	MKW 1	MKW?	MKW 1
		1 div	MKW 5	MKW?	MKW 5
2 div		MKW 6	MKW?	MKW 6	
5 div		MKW 7	MKW?	MKW 7	
10 div		MKW 2	MKW?	MKW 2	
<<Marker Func>>					
MKR to CF		MKR 3	_____	_____	
		MKCF	_____	_____	
		E2	_____	_____	
MKR to REF		MKR 4	_____	_____	
		MKRL	_____	_____	
		E4	_____	_____	
MKR to CFstep		MKR 5	_____	_____	
		MKSS	_____	_____	
		E3	_____	_____	
Δ MKR to Span		MKR 6	_____	_____	
		MKSP	_____	_____	
		KSO	_____	_____	
Zone to Span		MKR 7	_____	_____	

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Marker (contd.)					
<<Multi Marker>>					
Multi Marker	Off		MKMULTI 0	_____	_____
			MKMULTI OFF	MKMULTI?	OFF
	On		MKMULTI 1	_____	_____
			MKMULTI ON	MKMULTI?	ON
Multi Marker (Highest 10)		MKMHI	_____	_____	
Multi Marker (Harmonics)		MKMHRM	_____	_____	
Multi Marker	Off		MKSLCT n,0	_____	_____
			MKSLCT n,OFF	MKSLCT? n	OFF
	On		MKSLCT n,1	_____	_____
			MKSLCT n,ON	MKSLCT? n	ON
Select Active Marker	1 to 10	MKACT n	MKACT?	n	
Setup Marker Position	1 to 10, f	MKMP n, f	MKMP? n	f	
Clear Multi Marker		MKMCL	_____	_____	
Multi Marker List	Off		MKLIST 0		
			MKLIST OFF	MKLIST?	OFF
	On		MKLIST 1		
			MKLIST ON	MKLIST?	ON
Multi Marker List Query (Level)		_____	MKML? n	ℓ	
<<Peak Search>>					
Peak			MKS 0	_____	_____
			MKPK	_____	_____
			MKPK HI	_____	_____
			E1	_____	_____
Next Peak			MKS 1	_____	_____
			MKPK NH	_____	_____
Min Dip			MKS 2	_____	_____
			MKMIN	_____	_____
Next Right Peak			MKS 9	_____	_____
			MKPK NR	_____	_____
Next Left Peak			MKS 10	_____	_____
			MKPK NL	_____	_____
Next Dip			MKS 11	_____	_____

(c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg
Marker (contd.)					
	<<Input Position>>				
	Reference Marker (query)		_____	RMK?	RMK p
	Current Marker (query)		_____	CMK?	CMK p
	Marker Pos (query)	Freq	_____	MKF?	f
		Time	_____	MKF?	t
		Level	_____	MKL?	ℓ
			_____	MKA?	ℓ
		FM : freq	_____	MKL?	f
			_____	MKA?	f
		EXT TRIG : time	_____	MKL?	t
			_____	MKA?	t
Display					
	<<Display Mode>>				
	A (FREQ)		DFMT A	DFMT?	A
	Time		DFMT TIME	DFMT?	TIME
	<<Storage Mode>>				
	Normal	A (FREQ)	AMD 0	AMD?	AMD 0
		Time	TMMD 0	TMMD?	TMMD 0
	Max hold	A	AMD 1	AMD?	AMD 1
			MXMH TRA	_____	_____
		A2	_____	_____	
		Time	TMMD 1	TMMD?	TMMD 1
	Min hold	A	AMD 3	AMD?	AMD3
		Time	TMMD 3	TMMD?	TMMD 3
	Average	A	AMD 2	AMD?	AMD 2
		Time	TMMD 2	TMMD?	TMMD 2
		Off	VAVG 0	_____	_____
			VAVG OFF	_____	_____
			KSH	_____	_____
		On	VAVG 1	_____	_____
			VAVG ON	_____	_____
	KSG		_____	_____	
	Cumulative	A	AMD 4	AMD?	AMD 4
		Time	TMMD 4	TMMD?	TMMD 4
	Over write	A	AMD 5	AMD?	AMD 5
		Time	TMMD 5	TMMD?	TMMD 5
	View	A	VIEW TRA	_____	_____
		Time	VIEW TRTIME	_____	_____

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
Display (contd.)						
	Average count	2	VAVG 2	VAVG?	2	
		4	AVR 0	AVR?	AVR 0	
			VAVG 4	VAVG?	4	
		8	AVR 1	AVR?	AVR 1	
			VAVG 8	VAVG?	8	
		16	AVR 2	AVR?	AVR 2	
			VAVG 16	VAVG?	16	
		32	AVR 3	AVR?	AVR 3	
			VAVG 32	VAVG?	32	
		64	VAVG 64	VAVG?	64	
	128		AVR 4	AVR?	AVR 4	
		VAVG 128	VAVG?	128		
	256	VAVG 256	VAVG?	256		
		Det mode (A)	Positive Peak	DETM TRA,POS	DETM? TRA	POS
	Sample		DETM TRA,SMP	DETM? TRA	SMP	
	Negative peak		DETM TRA,NEG	DETM? TRA	NEG	
	< <time > >					
	Time Delay	t	TDLY t	TDLY?	t	
	Time Span	t	TSP t	TSP?	t	
	Zone (expand mode)	Off	TZONE 0	_____	_____	
			TZONE OFF	TZONE?	OFF	
		On	TZONE 1	_____	_____	
			TZONE ON	TZONE?	ON	
	Expand mode	Off	TEXPAND 0			
			TEXPAND OFF	TEXPAND?	OFF	
On		TEXPAND 1				
		TEXPAND ON	TEXPAND?	ON		
Zone Start	t	TZSTART t	TZSTART?	t		
Zone Span	t	TZSP t	TZSP?	t		
Special Function (FM/TRIG MONITOR)	FM	SPFUNC FM	SPFUNC?	FM		
	Ext Trigger Monitor	SPFUNC EXT	SPFUNC?	EXT		
	Off	SPFUNC OFF	SPFUNC?	OFF		
FM Range	200 kHz/div	FMRNG 200KHZ	FMRNG?	200000		
	20 kHz/div	FMRNG 20KHZ	FMRNG?	20000		
	2 kHz/div	FMRNG 2KHZ	FMRNG?	2000		
Coupling (FM)	AC	COUPLE AC	COUPLE?	AC		
	DC	COUPLE DC	COUPLE?	DC		

## (c) Spectrum mode command (contd.)

Class	Function	Item	Program Msg	Query Msg	Response Msg	
Sound						
	Monitor	AM	MON AM	MON?	AM	
		FM	MON FM	MON?	FM	
		Off	MON OFF	MON?	OFF	
	Monitor Volume	n	MONVOL n	MONVOL?	n	
Parameter						
	Parameter	Page 1	PARAM 1	PARAM?	1	
		Page 2	PARAM 2	PARAM?	2	
		Off	PARAM OFF	PARAM?	OFF	
System						
	Auto Sweep Time	Fast	ASWT FAST	ASWT?	FAST	
		Slow	ASWT SLOW	ASWT?	SLOW	
	Data Point	Normal	DPOINT NRM	DPOINT?	NRM	
		Double	DPOINT DOUBLE	DPOINT?	DOUBLE	
	Couple Mode	Common	VBCOUPLE COM	VBCOUPLE?	COM	
		Independent	VBCOUPLE IND	VBCOUPLE?	IND	
Calibration						
	All		CAL 0	_____	_____	
	Freq		CAL 1	_____	_____	
	Level		CAL 2	_____	_____	
	Preselector Tune	Auto tune		PRESEL AUTO	PRESEL?	AUTO
				PP	_____	_____
		Manual tune		PRESEL n	PRESEL?	n
	Preset		PRESEL PRESET	PRESEL?	PRESET	
Cal/Uncal						
	Uncal display	Off	UNC 0	_____	_____	
			UNC OFF	UNC?	UNC OFF	
		On	UNC 1	_____	_____	
			UNC ON	UNC?	UNC ON	
	Uncal status	Normal		_____	UCL?	UCL 0
		Uncal		_____	UCL?	UCL 1

## SECTION 2

### CONNECTING THE BUS AND SETTING THE ADDRESS

This section describes how to connect the GPIB cable and set the addresses in order to set-up the system before using the GPIB.

#### TABLE OF CONTENTS

2.1	Connecting Devices with GPIB Cables .....	2-3
2.2	Setting and Checking GPIB addresses .....	2-4

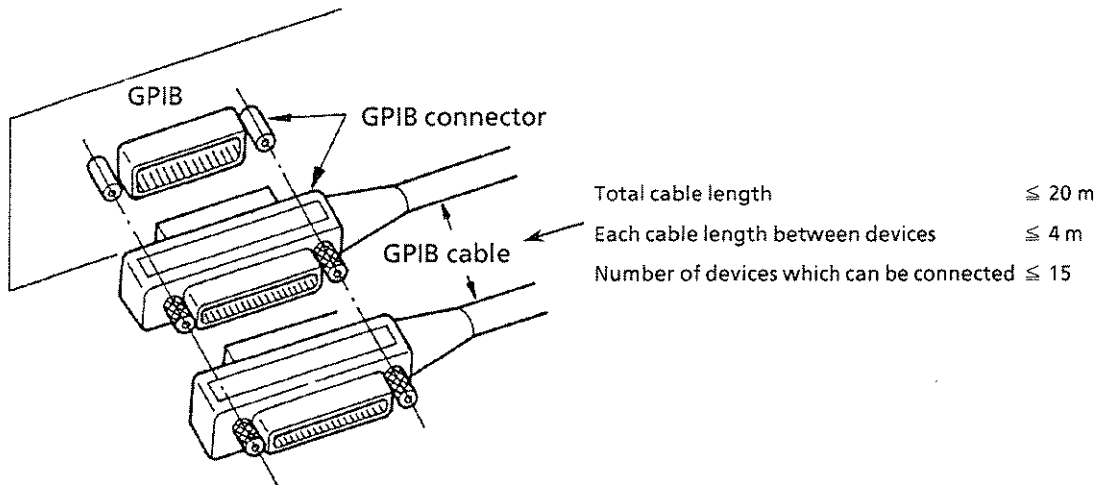
(Blank)



## 2.1 Connecting Devices with GPIB Cables

The rear panel has connectors for connecting GPIB cables.

A maximum of 15 devices, including the controller, can be connected to one system. The restrictions indicated at the right of the diagram below should be observed when connecting many devices to one system.



---

**CAUTION**

---

*The cables must be connected before the power is switched on.*

---

## 2.2 Setting and Checking GPIB addresses

Initial values of GPIB addresses are as follows:

GPIB1 address : 1  
 GPIB2 address : 16  
 MC8104A address : 19  
 Printer address : 17

Change these addresses as in the example below.

Example : Set GPIB addresses as follows:

GPIB1 address : 3  
 GPIB2 address : 15  
 MC8104A address : 23  
 Printer address : 20

Step	Key operation	Explanation
1	<b>System</b>	Display the system screen.
2	<b>INTERFACE</b> <b>F5</b>	Display the INTERFACE function key menu.
3	<b>GPIB1 ADDRESS</b> <b>F1</b> <b>3</b> <b>Enter</b>	Display the current GPIB1 address. Change the GPIB1 address to 3.
4	<b>GPIB2 ADDRESS</b> <b>F2</b> <b>1</b> <b>5</b> <b>Enter</b>	Display the current GPIB2 address. Change the GPIB2 address to 15.
5	<b>MC8104A ADDRESS</b> <b>F3</b> <b>2</b> <b>3</b> <b>Enter</b>	Display the current MC8104A address. Set the MC8104A address to 23.
6	<b>RETURN</b> <b>F6</b>	Return to the MC8104A function key menu.
7	<b>TX Test</b> <b>F6</b>	Switch the screen to the transmitter tester screen.
8	<b>Shift</b> <b>Copy</b> <b>SETUP PRINTER</b> <b>F2</b> <b>PRINTER ADDRESS</b> <b>F5</b> <b>2</b> <b>0</b> <b>Enter</b>	Display the COPY CONTROL function key menu. Display the SETUP PRINTER function key menu. Display the current printer address. Set the printer address to 20.

# SECTION 3

## DEVICE MESSAGE FORMAT

This section describes the format of the device messages transmitted on the bus between a controller and devices via the GPIB system.

### TABLE OF CONTENTS

3.1	General Description .....	3-3
3.2	Program message format .....	3-3
3.3	Response message format .....	3-7

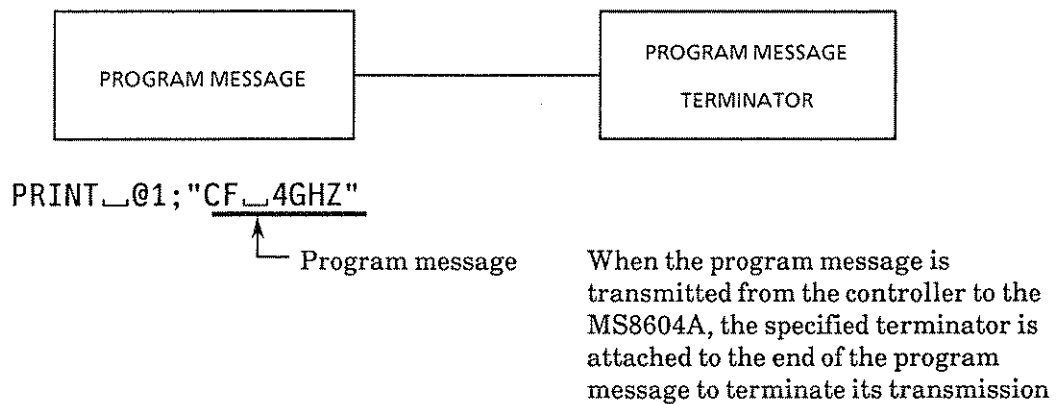
**(Blank)**

### 3.1 General Description

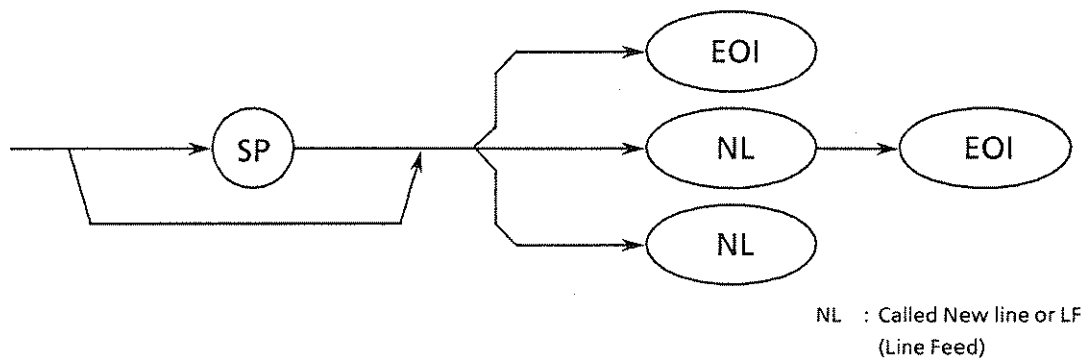
The device messages are data messages that are transmitted between the controller and devices. There are two types of data messages: program messages that are transferred from the controller to MS8604A (device), and response messages that are sent from MS8604A (device) to the controller. There are also two types of program commands and program queries in the program message. The program command is used to set this instrument's parameters and to instruct it to execute processing. The program query is used to query the values of parameters and measured results.

### 3.2 Program message format

To transfer the program messages from the controller program to MS8604A using the PRINT statement, the program message formats are defined as follows.

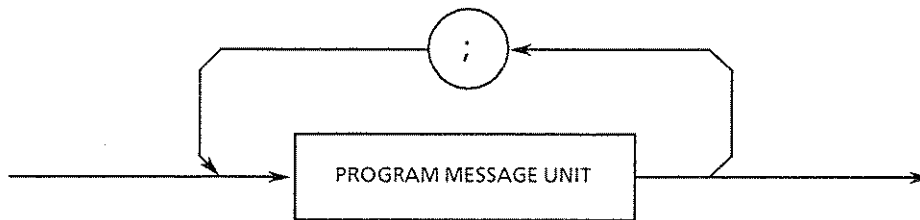


#### (1) PROGRAM MESSAGE TERMINATOR



Carriage Return (CR) is ignored, and is not processed as a terminator.

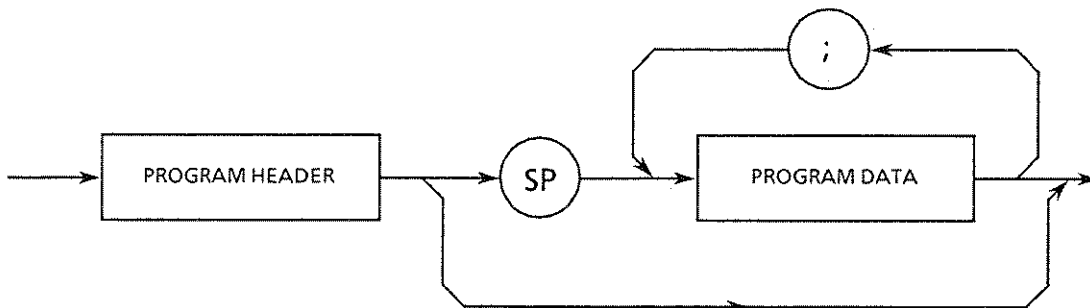
(2) PROGRAM MESSAGE



The program messages consisting of one or more program message units can be output sequentially by concatenating each of them with a semicolon.

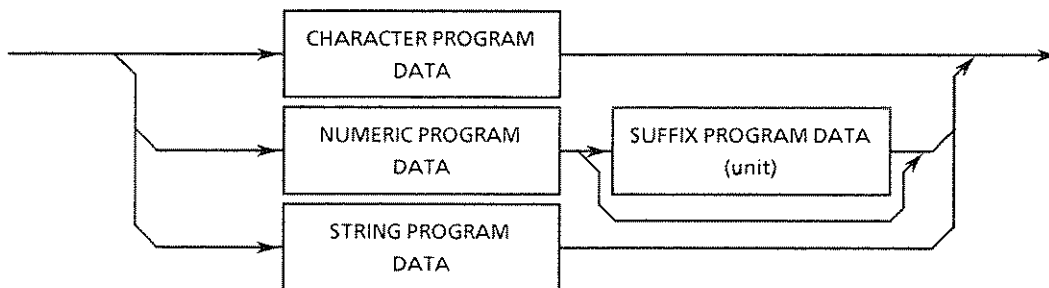
<Example> PRINT\_@1;"CF\_1GHZ;SP\_500KHZ"

(3) PROGRAM MESSAGE UNIT



- The IEEE 488.2 common command has a leading asterisk "\*" that is always placed before the program header.
- The program query has a trailing question mark "?" that is always added at the end of the program header.

(4) PROGRAM DATA



(5) CHARACTER PROGRAM DATA

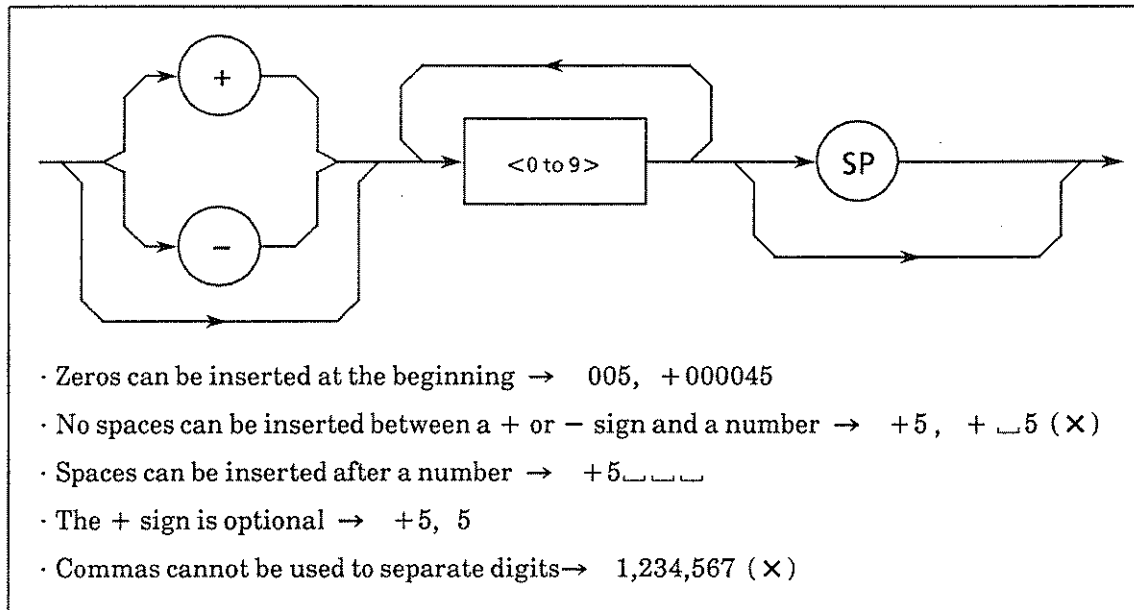
Character program data consists of the upper-case alphabetic characters from A to Z, lower-case alphabetic characters from a to z, the underline of "\_", and the numbers 0 to 9. They can be used in a specified combination.

<Example> PRINT\_@1;\_"ST\_AUTO" ..... Sets Sweep Time to AUTO

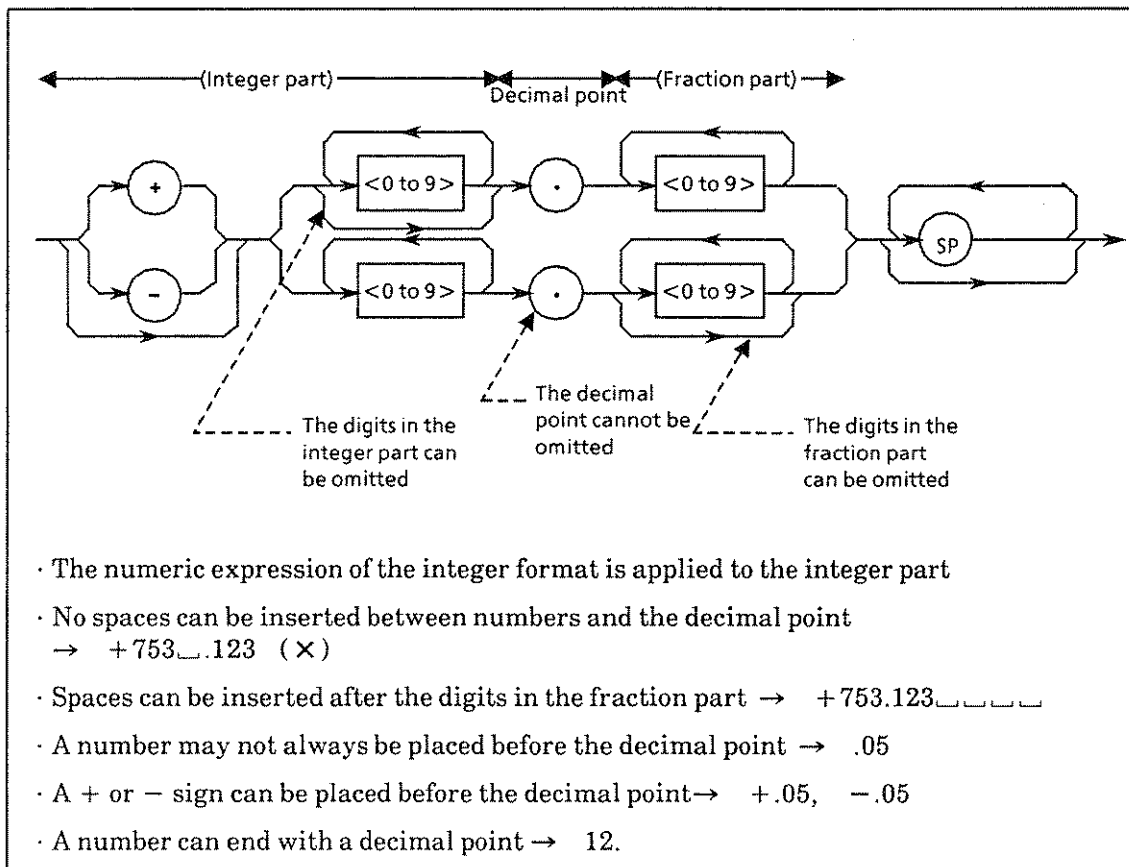
## (6) NUMERIC PROGRAM DATA

Numeric program data has two types of formats: integer format (NR1) and fixed-point real number format (NR2).

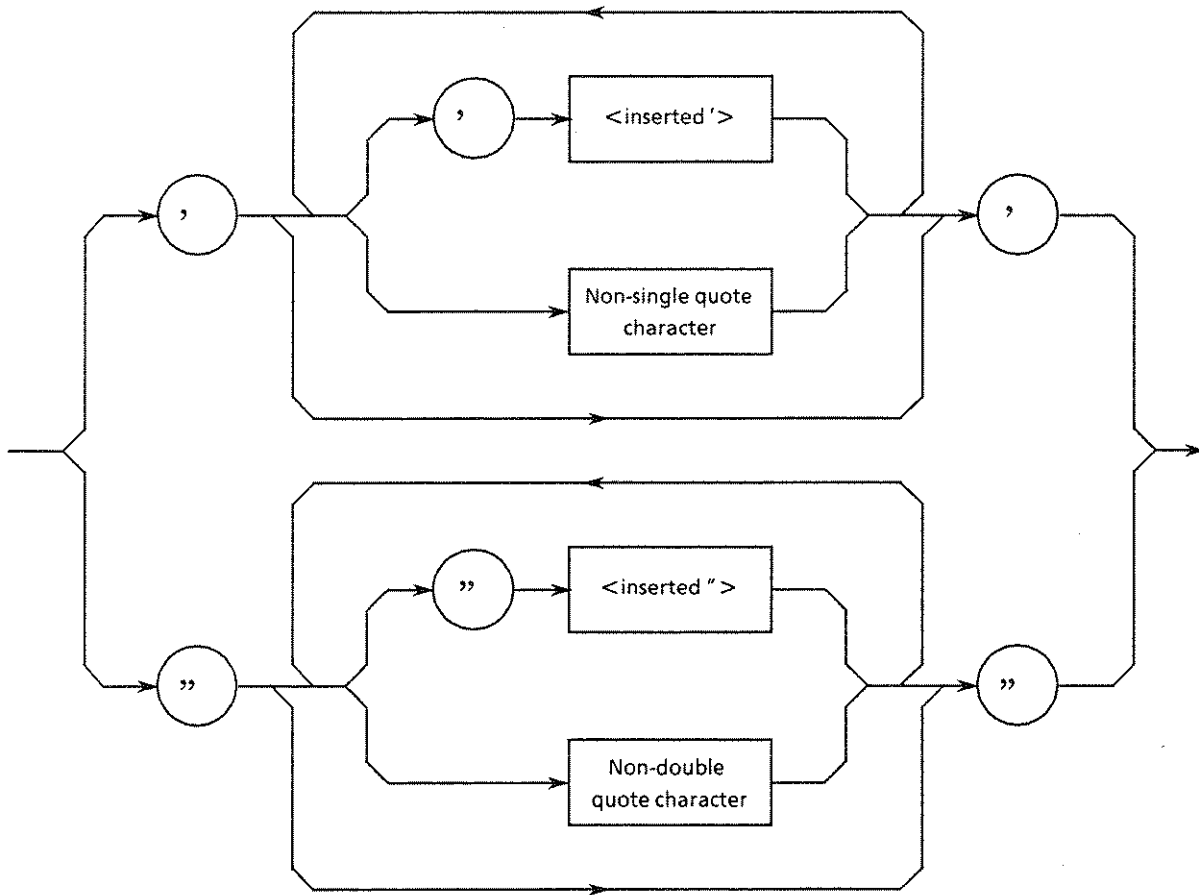
## &lt; Integer Format (NR1) &gt;



## &lt; Fixed-Point (real number) Format (NR2) &gt;



## (7) STRING PROGRAM DATA



- Both ends of string program data must have a pair of single quotation marks '....'

```
PRINT @ 1; "TITLE 'MS8604A'"
```

A single quotation mark used within the character string must be repeated as shown in "

```
PRINT @ 1; "TITLE 'MS8604A' 'NOISE MEAS' ' ' "
```

Executing TITLE results in MS8604A 'NOISE MEAS'

*Note:* To use the double quotation mark " in the PRINT statement, specify CHR\$ (&H22).

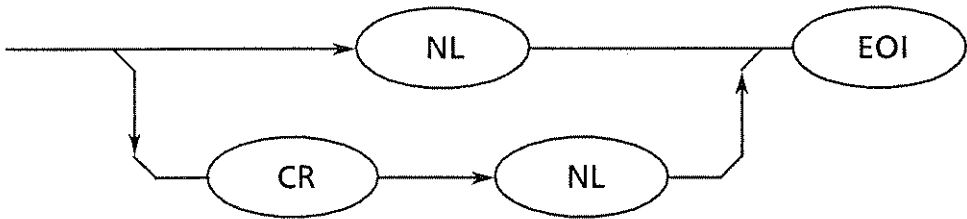


### 3.3 Response message format

To transfer the response messages from the MS8604A to the controller using the INPUT statement, the response message formats are defined as follows.

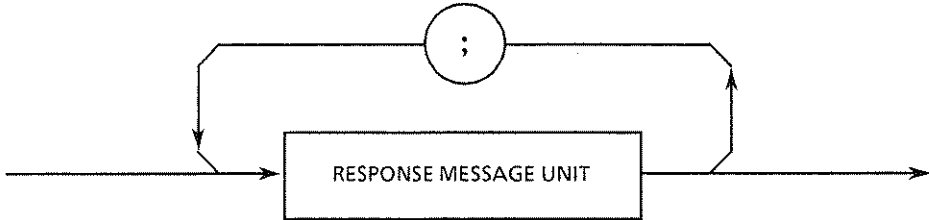


#### (1) RESPONSE MESSAGE TERMINATOR



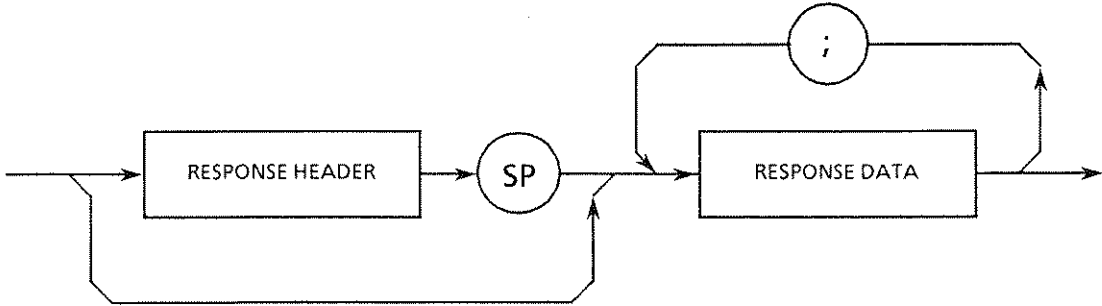
The response message terminator to be used depends on the TRM command.

#### (2) RESPONSE MESSAGE

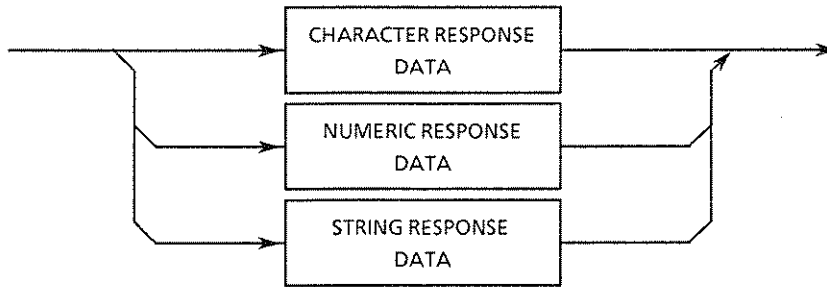


When a query is sent by the PRINT statement with one or more program queries, the response message also consists of one or more response message units.

#### (3) RESPONSE MESSAGE UNIT



(4) RESPONSE DATA

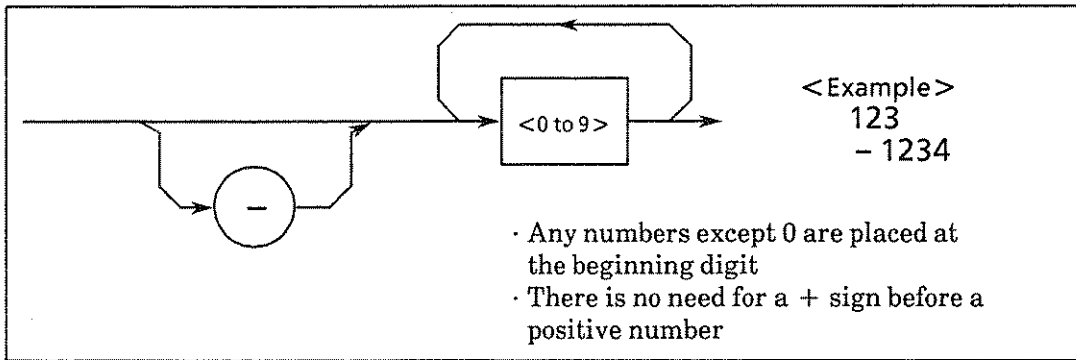


(5) CHARACTER RESPONSE DATA

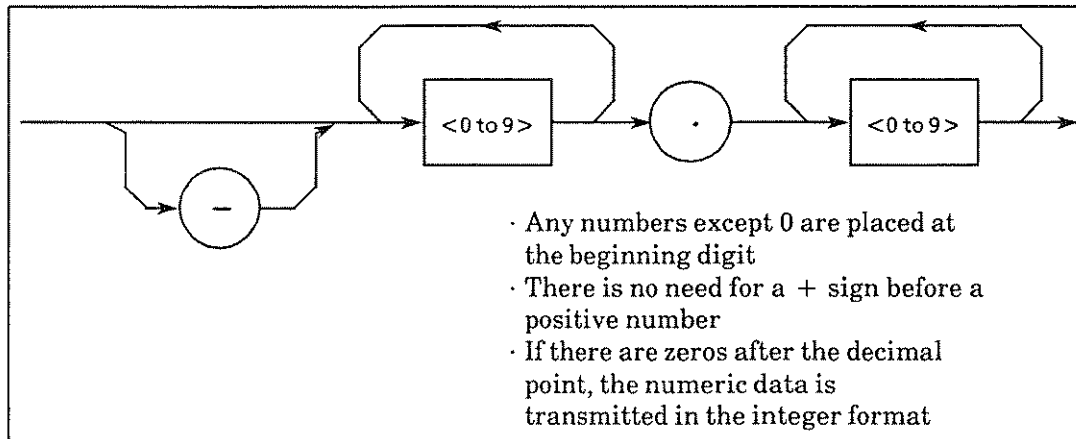
Character response data consists of the upper-case alphabetic characters from A to Z, lower-case alphabetic characters from a to z, the underline “\_”, and the numbers 0 to 9. They can be used in a specified combination.

(6) NUMERIC RESPONSE DATA

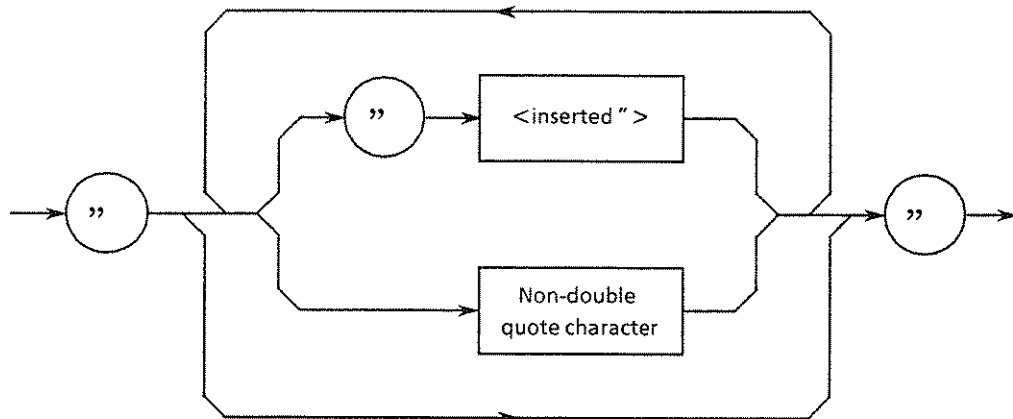
< Integer Format (NR1) >



< Fixed-Point (real number) Format (NR2) >



## (7) STRING RESPONSE DATA



String response data is transmitted as an ASCII character string, which is enclosed with double quotation marks.

## (8) Response message to input the waveform data using binary data

For binary format read and 2-bite binary data transferring, see paragraph 7.2 (4).

**(Blank)**

# SECTION 4

## STATUS MESSAGE

This section describes MS8604A status reports and their data structure and explains the techniques for synchronizing the controller and devices.

In order to obtain more detailed status information, the IEEE 488.2 standard has more common commands and common queries than the IEEE 488.1 standard. Refer to Section 6 for a detailed explanation of these common commands and queries.

### TABLE OF CONTENTS

4.1	IEEE 488.2 Standard Status Model .....	4-4
4.2	Status Byte (STB) Register .....	4-6
4.2.1	ESB and MAV summary messages .....	4-6
4.2.2	Device-dependent summary messages .....	4-7
4.2.3	Reading and clearing the STB register .....	4-8
4.3	Enabling SRQ .....	4-9
4.4	Standard Event Status Register .....	4-10
4.4.1	Bit definition of standard event status register .....	4-10
4.4.2	Query error details .....	4-11
4.4.3	Reading, writing to and clearing the standard event status register .....	4-12
4.4.4	Reading, writing to and clearing the standard event status enable register .....	4-12
4.5	Extended Event Status Register .....	4-13
4.5.1	Bit definition of PTA event status register .....	4-14
4.5.2	Bit definition of END event status register .....	4-15
4.5.3	Reading, writing to and clearing the extended event status register .....	4-16
4.5.4	Reading, writing to and clearing the extended event status enable register .....	4-16
4.6	Techniques for Synchronizing MS8604A with a Controller .....	4-17
4.6.1	Wait for response after *OPC? query is sent .....	4-17
4.6.2	Wait for service request after *OPC is sent .....	4-18

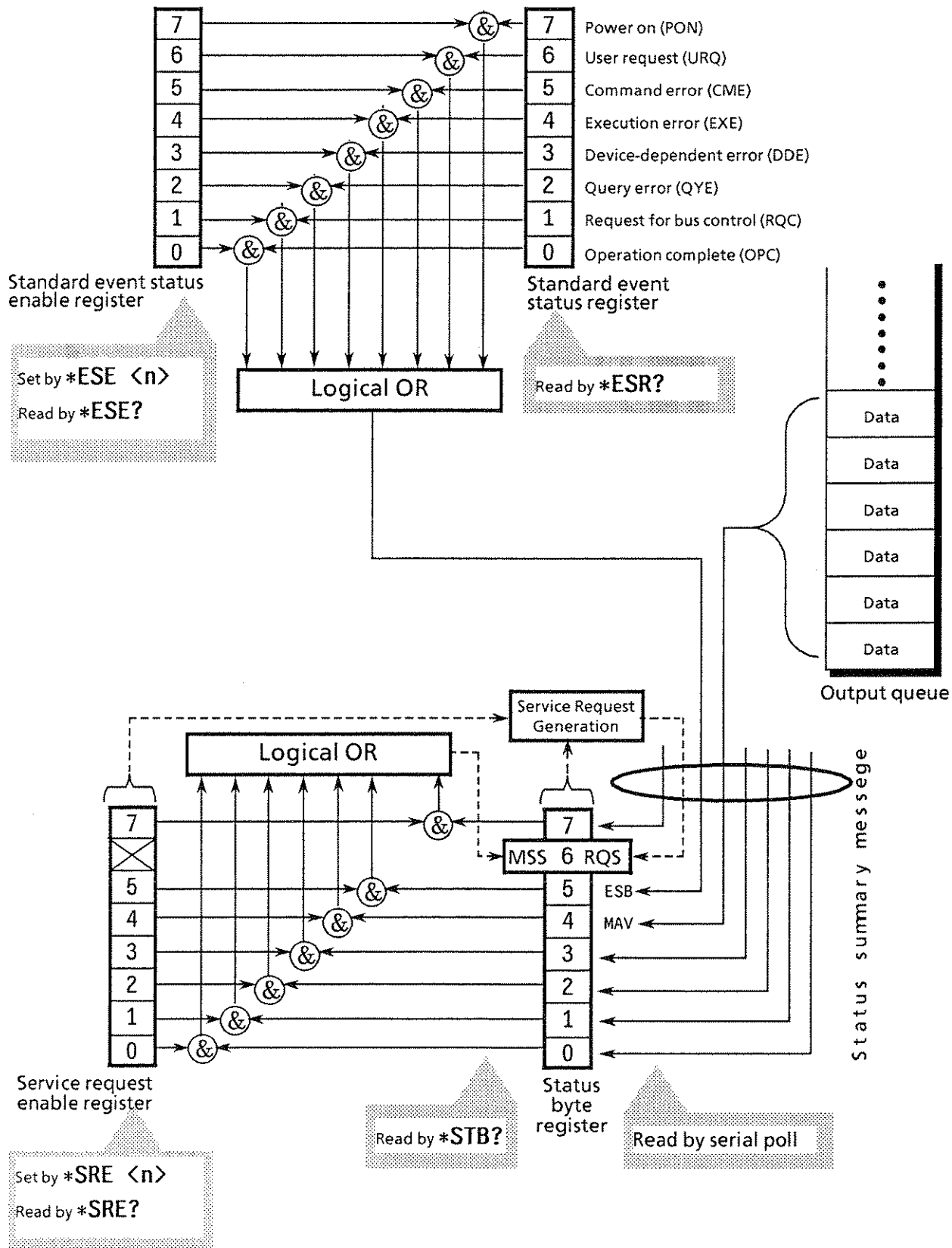
**(Blank)**

The Status Byte (STB) sent by the controller is based on the IEEE 488.1 standard. The bits comprising it are called a status summary message because they represent a summary of the current data contained in registers and queues.

The following pages explain the status summary message and the structure of the status data that constitutes the status summary message bits as well as techniques for synchronizing the MS8604A and controller, which use these status messages.

### 4.1 IEEE 488.2 Standard Status Model

The diagram below shows the standard model for the status data structure stipulated in the IEEE 488.2 standard.



Standard Status Model Diagram



The IEEE 488.1 status byte is used in the status model. This status byte is composed of 7 summary message bits given from the status data structure. For creating the summary message bits, there are 2 models for the data structure – the register model and the queue model.

Register model	Queue model
The register model consists of the two registers used for recording events and conditions encountered by a device. These two registers are the Event Status Register and Event Status Enable Register. When the results of the AND operation of both register contents is not 0, the corresponding bit of the status bit becomes 1. In other cases, it becomes 0. And, when the result of their Logical OR is 1, the summary message bit becomes also 1. If the Logical OR result is 0, the summary message bit becomes 0 too.	The queue in the queue model is for sequentially recording the waiting status values and data. The queue structure is such that the relevant bit is set to 1 when there is data in it and 0 when it is empty.

In IEEE 488.2, there are 3 standard models for status data structure-2 register models and 1 queue model-based on the register model and queue model explained above. They are:

- ① Standard Event Status Register and Standard Event Status Enable Register
- ② Status Byte Register and Service Request Enable Register
- ③ Output queue

Standard Event Status Register	Status Byte Register	Output Queue
The Standard Event Status Register has the structure of the previously described register model. In this register, bits are set for 8 types of standard event encountered by a device, viz. ① Power on, ② User request, ③ Command error, ④ Execution error, ⑤ Device-dependent error, ⑥ Query error, ⑦ Request for bus control and ⑧ Operation complete. The Logical OR output bit is represented by Status Byte Register bit 5 (DIO6) as a summary message for the Event Status Bit (ESB).	The Status Byte Register is a register in which the RQS bit and the 7 summary message bits from the status data structure can be set. It is used together with the Service Request Enable Register. When the results of the OR operation of both register contents is not 0, SRQ becomes ON. To indicate this, bit 6 of the Status Byte Register (DIO7) is reserved by the system as the RQS bit which means that there is a service request for the external controller. The mechanism of SRQ conforms to the IEEE 488.1 standard.	The Output Queue has the structure of the queue model mentioned above. Status Byte Register bit 4 (DIO5) is set as a summary message for Message Available (MAV) to indicate that there is data in the output queue.

## 4.2 Status Byte (STB) Register

The STB register consists of device STB and RQS (or MSS) messages. The IEEE 488.1 standard defines the method of reporting STB and RQS messages but not the setting and clearing protocols or the meaning of STB. The IEEE 488.2 standard defines the device status summary message and the Master Summary Status (MSS) which is sent to bit 6 together with STB in response to an \*STB? common query.

### 4.2.1 ESB and MAV summary messages

The following is a description of the ESB and MAV summary messages.

#### (1) ESB summary messages

The ESB (Event Summary Bit) summary message is a message defined by IEEE 488.2, which is represented by bit 5 of the STB register. This bit indicates whether at least one of the events defined in IEEE 488.2 has occurred or not when the service request enable register is set so that events are enabled after the final reading or clearing of the standard event status register. The ESB summary message bit becomes true when the setting permits events to occur if any one of the events recorded in standard event status register is true. Conversely, it is false if none of the recorded events occurs even if events are set to occur.

When the ESR register was read by the \*ESR? query and when the ESR register was cleared, this bit becomes FALSE (0).

#### (2) MAV summary messages

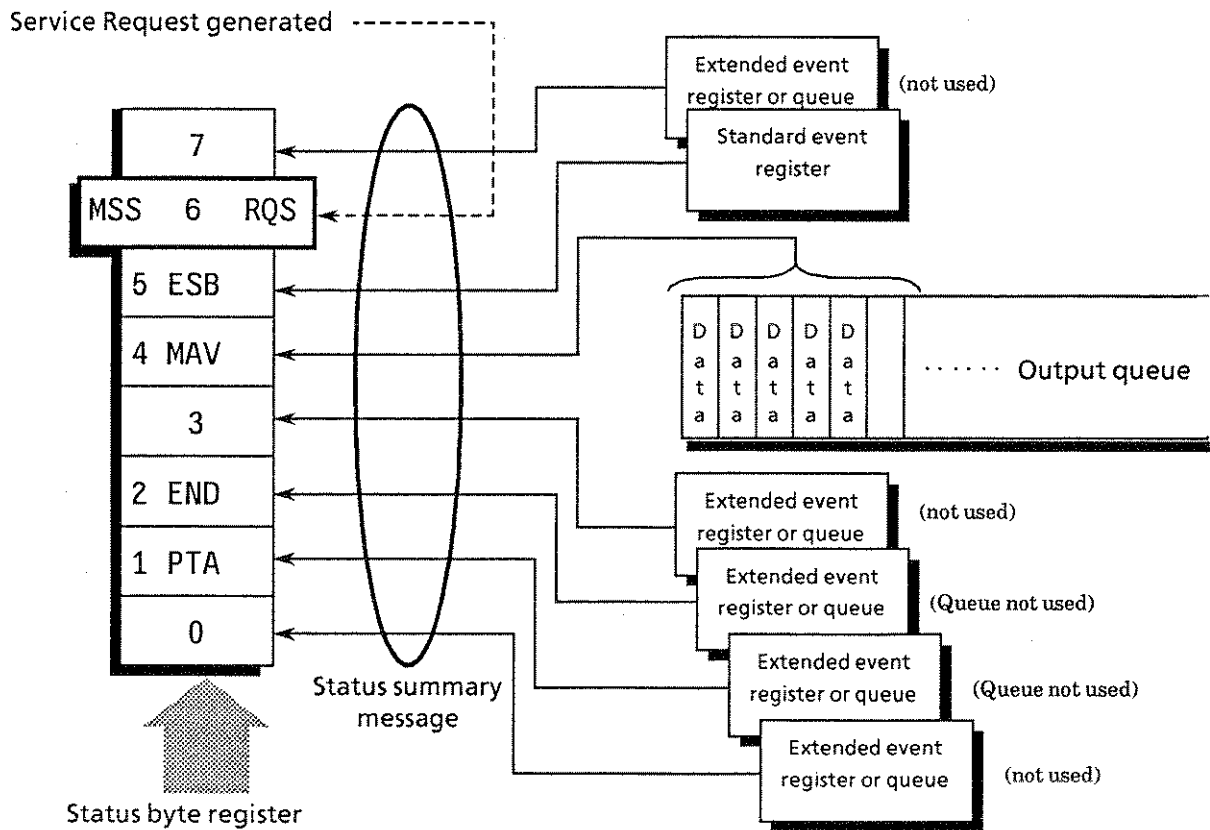
The MAV summary message is a message defined in IEEE 488.2 and represented by bit 4 in the STB register. This bit indicates whether the output queue is empty or not. The MAV summary message bit is set to 1 (true) when a device is ready to receive a request for a response message from the controller and to 0 (false) when the output queue is empty. This message is used to synchronize the exchange of information with the controller. For example, it can be used get the controller to wait till MAV is true after it has sent a query command to a device. While the controller is waiting for a response from the device, it can process other jobs. Reading the output queue without first checking MAV will cause all system bus operations to be delayed until the device responds.

### 4.2.2 Device-dependent summary messages

The IEEE 488.2 standard does not specify whether bits 7 (DIO8) and 3 (DIO4) to 0 (DIO1) of the status byte register are used as status register summary bits, or used to indicate that there is data in a queue. These bits can be used as device-dependent summary messages.

Device-dependent summary messages have the respective status data structures of the register model or the queue model. Thus, the status data structure may be either the register to report events and status in parallel or the queue to report conditions and status in sequence. The summary bit represents a summary of the current status of the corresponding data structure. In the case of the register model, the summary bit is true when there is an event set to permit the occurrence of more than one true; while in the case of the queue model, it is true if the queue is not empty.

As shown below, the MS8604A does not use bits 0, 3 and 7. As it uses bits 1 and 2 as the summary bits of the status register, it has 3 register model types (, where 2 types extended) and one queue model type-an output queue with no extension.



### 4.2.3 Reading and clearing the STB register

Serial poll or the **\*STB?** common query are used to read the contents of STB register. STB messages conforming to IEEE 488.1 can be read by either method, but the value sent to bit 6 is different for each of them.

The STB register can be cleared using the **\*CLS** command.

#### (1) Reading by serial poll

When using the serial poll conforming to IEEE 488.1, the device must return a 7-bit status byte and an RQS message bit which conforms to IEEE 488.1. According to IEEE 488.1, the RQS message indicates whether the device sent SRQ as true or not. The value of the status byte is not changed by serial poll. The device must set the RQS message to false immediately after being polled. As a result, if the device is again polled before there is a new cause for a service request, the RQS message is false.

#### (2) Reading by the **\*STB?** common query

The **\*STB?** common query requires the device to send the contents of the STB register and one integer format response message from the MSS (Master Summary Status) summary message. The response represents the total binary weighted value of the STB register and the MSS summary message. The STB-register bits 0 to 5 and 7 are weighted to 1, 2, 4, 8, 16, 32, and 128; and the MSS to 64, respectively. Thus, excepting the fact that bit 6 represents the MSS summary message instead of the RQS message, the response to **\*STB?** is identical to that for serial poll.

#### (3) Definition of MSS (Master Summary Status)

MSS indicates that there is at least one cause for a service request. The MSS message is represented at bit 6 in a device response to the **\*STB?** query but it is not produced as a response to serial poll. In addition, it is not part of the status byte specified by IEEE 488.1. MSS is produced by the logical OR operation of STB register with SRQ enable (SRE) register. In concrete terms, MSS is defined as follows.

(STB Register bit 0 AND SRE Register bit 0)

OR

(STB Register bit 1 AND SRE Register bit 1)

OR

:

:

(STB Register bit 5 AND SRE Register bit 5)

OR

(STB Register bit 7 AND SRE Register bit 7)

As bit-6 status of the STB and SRQ enable registers are ignored in the definition of MSS, it can be considered that bit-6 status are always being 0 when calculating the value of MSS.

#### (4) Clearing the STB register by the **\*CLS** common command

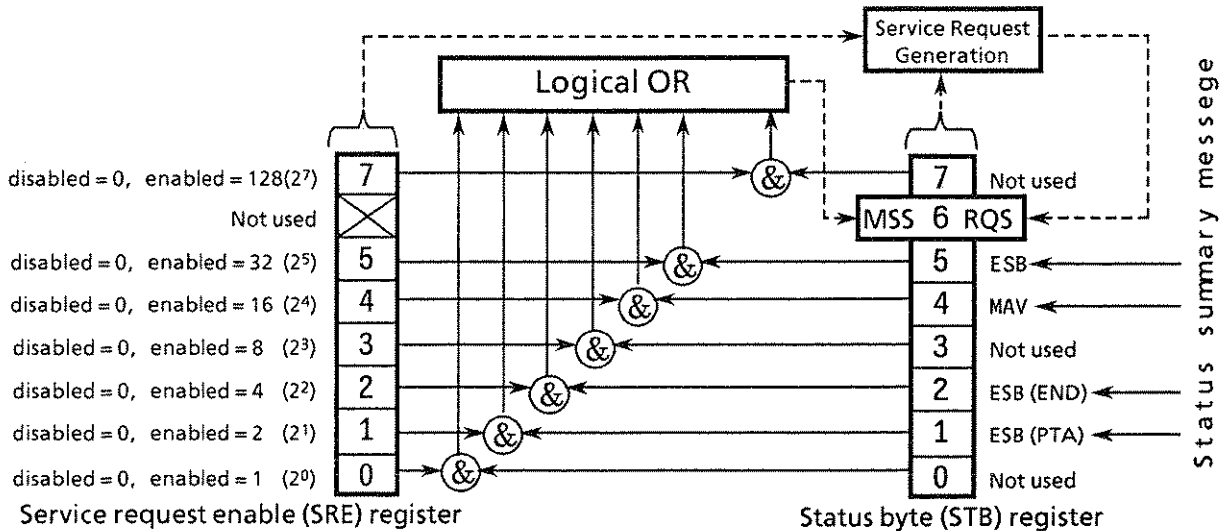
With the exception of the output queue and its MAV summary message, the **\*CLS** common command clears all status data structures (status event registers and queues) as well as the summary messages corresponding to them.

The **\*CLS** command does not affect settings in the enable registers.

### 4.3 Enabling SRQ

All types of summary message in the STB register can be enabled or disabled for service requests by using the SRQ enable function. The service request enable (SRE) register is used for this function to select summary messages as shown in the diagram below.

Bits in the service request enable register correspond to bits in the status byte register. If a bit in the status byte corresponding to an enabled bit in the service request enable register is set to 1, a device makes a service request to the controller with the RQS bit set to 1. For example, if bit 4 (MAV) in the service request enable register is enabled, the device makes a request for service to the controller each time the MAV bit is set to 1 when there is data in the output queue.



#### (1) Reading the SRE register

The contents of the SRE register are read using the `*SRE?` common query. The response message to this query is an integer from 0 to 255 which is the sum of the bit digit weighted values in the SRE register. SRE register bits 0 to 5 and 7 are respectively weighted to 1, 2, 4, 8, 16, 32 and 128. The unused bit 6 must always be set to 0.

#### (2) Updating the SRE register

The `*SRE` common instruction is used to write data in the SRE register. An integer from 0 to 255 is added after the `*SRE` common instruction.

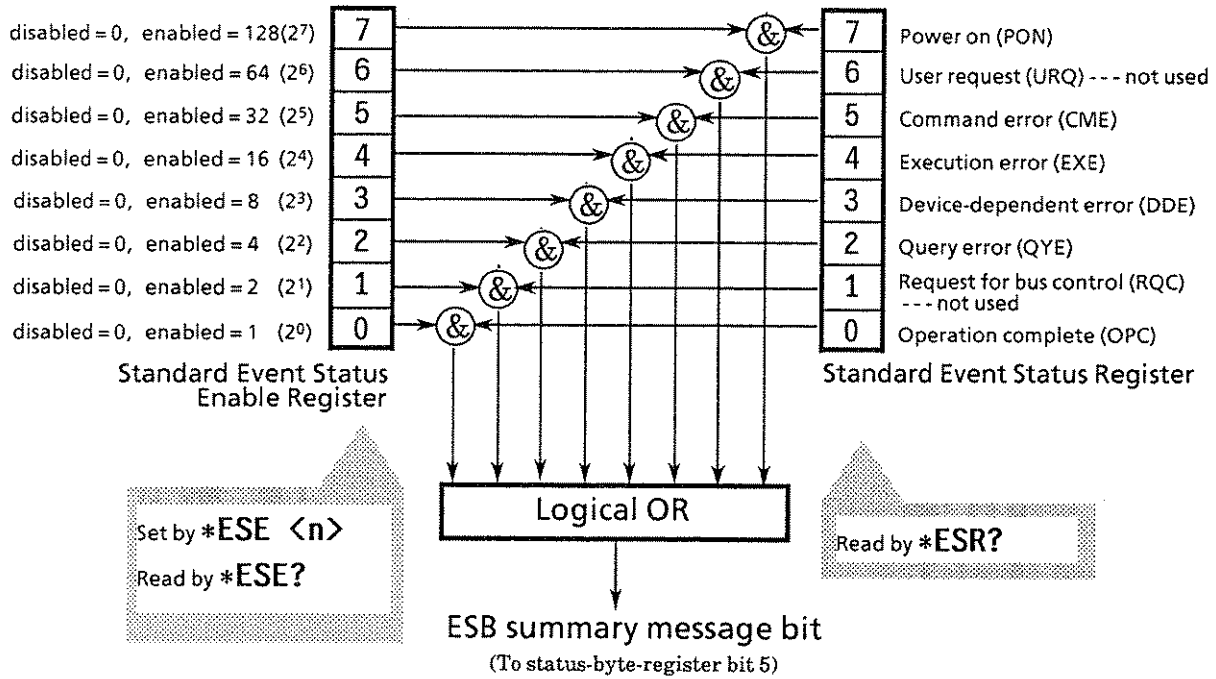
This integer indicates the total number of bits in the SRE register (weighted values: 1, 2, 4, 8, 16, 32, and 128), and sets the corresponding SRE register bits to 0 or 1.

A bit value of 1 indicates an enabled state and 0 indicates a disabled state. Ignore the value of bit 6.

### 4.4 Standard Event Status Register

#### 4.4.1 Bit definition of standard event status register

The standard event status register must be available on all devices conforming to the IEEE 488.2 standard. The diagram below shows the operation of the standard event status register model. Because the operation of the model is the same as that for the other models explained up till now, the following only explains the meaning of each bit in the standard event status register as defined in the IEEE 488.2 standard.



The left side standard event status enable (ESE) register selects whether the register makes the summary message true or not when the corresponding event status register becomes 1.

Bit	Event name	Description
7	PON : Power on	The power is turned to on.
6	URQ : User Request	Request for local control (rtl). This bit is produced regardless of whether a device is in remote or local mode. It is not used for the MS8604A so, it is always set to 0.
5	CME : Command Error	An illegal program message, a misspelt command or a <b>GET</b> command within a program is received.
4	EXE : Execution Error	A legal program message, which cannot be executed, is received.
3	DDE : Device-dependent Error	An error caused by other than CME, EXE or QYE occurred.
2	QYE : Query Error	An attempt is made to read data in the output queue though there is none there, or data is lost from the output queue due to any reason, e.g. overflow etc..
1	RQC : Request Control	A device is requesting control of the bus. This bit is not used on the MS8604A so, it is always set to 0.
0	OPC : Operation Complete	A device has completed operations which were pending and is ready to receive new commands. This bit is only set in response to the <b>*OPC</b> command.

## 4.4.2 Query error details

No.	Item	Description
1	Incomplete program messages	If a device receives an MTA from the controller before it receives the terminator of the program message it is receiving, it aborts the incomplete program message and waits for the next one. In order to abort the incomplete message, the device clears its input buffer and output queue, reports a query error and sets bit 2 in the standard status register to indicate the query error.
2	Interruption of response message	If a device receives an MLA from the controller before it has sent the terminator of the response message it is sending, it automatically interrupts the response message and waits for the next program message. In order to interrupt the response message, the device clears its output queue, reports a query error and sets bit 2 in the standard status register to indicate the query error.
3	Sending the next program message without reading the previous response message	When a device becomes unable to send a response message because the controller has sent another program message immediately following a program or query message, the device aborts the response message and waits for the next program message. It then reports a query error as in No. 2 above.
4	Output queue overflow	When several program and query messages are executed in succession, there may be too many response messages for the output queue (256 bytes). If further query messages are received when the output queue is full, the output queue cannot send responses to them because an overflow situation exists in it. If there is an overflow in the output queue, the device clears it and resets the section where response messages are created. Then it sets bit 2 in the standard event status register to indicate a query error.

#### 4.4.3 Reading, writing to and clearing the standard event status register

Reading	The register is destructively read by the *ESR? common query, i.e. it is cleared after being read. The response message is an integer format data (NR1) value obtained by binary weighting the event bit and converting it to a decimal number.
Writing	With the exception of clearing, writing operations cannot be performed externally.
Clearing	The register is only cleared in the following cases. <ol style="list-style-type: none"> <li>① A *CLS command is received</li> <li>② The power is turned on when the power-on-status-clear flag is true. Devices (for which the power-on sequence is being executed) first clear their standard event status registers but later record events that occurred during the sequence in them. (e.g. the setting of the PON event bit, etc.)</li> <li>③ An event is read for the *ESR? query command</li> </ol>

#### 4.4.4 Reading, writing to and clearing the standard event status enable register

Reading	The register is non-destructively read by the *ESE? common query, i.e. it is not cleared after being read. The response message is returned by an integer format data (NR1) after having been binary weighted and converted to decimal.
Writing	The register is written to by the *ESE common command. As bits 0 to 7 of the register are respectively binary weighted to 1, 2, 4, 8, 16, 32, 64 and 128; data to be written is sent by <DECIMAL NUMERIC PROGRAM DATA> which is the digit total of the bits selected from these bits.
Clearing	The register is cleared in the following cases. <ol style="list-style-type: none"> <li>① A *ESE command with a data value of 0 is received</li> <li>② When the power is turned on when the power-on-status-clear flag is true.</li> </ol> The event status enable register is not affected by the following. <ol style="list-style-type: none"> <li>① Changes of the status of the IEEE 488.1 device clear function</li> <li>② A *RST common command is received</li> <li>③ A *CLS common command is received</li> </ol>

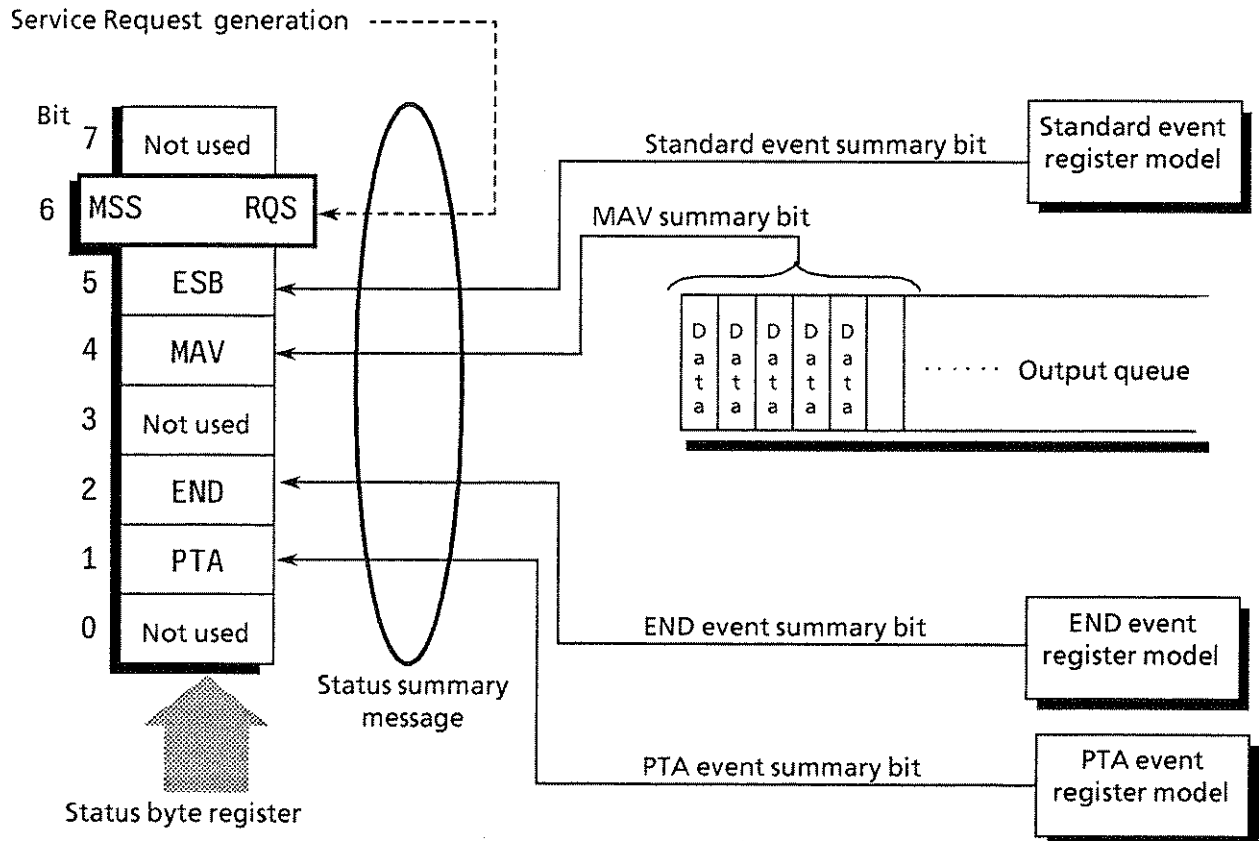


## 4.5 Extended Event Status Register

The register models of the status byte register, standard event status register and enable registers are mandatory for equipment conforming to the IEEE 488.2 standard.

In IEEE 488.2, status-byte-register bits 7 (DIO8), 3 (DIO4) to 0 (DIO1) are assigned to status-summary bits supplied by the extended-register and extended-queue models.

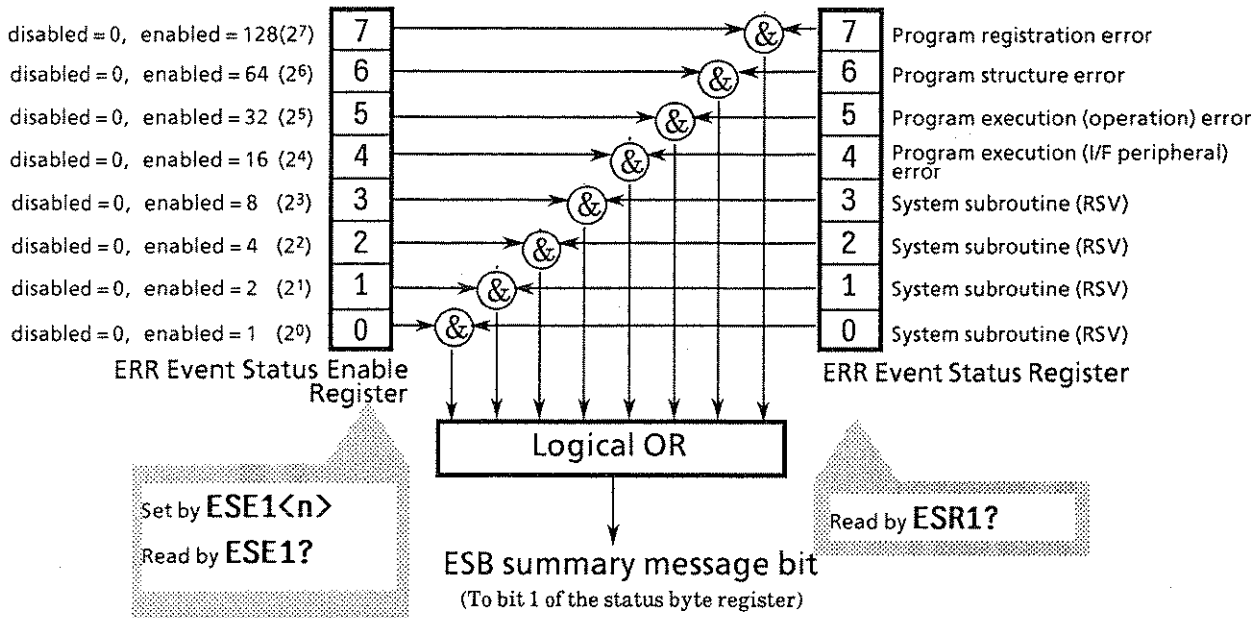
For the MS8604A, as shown in the diagram below, bits 7, 3 and 0 are unused and bits 1 and 2 are assigned to the END and PTA summary bits as the status-summary bits supplied by the extended-register model. As the queue model is not extended, there is only one type of queue—the output queue.



The following pages describe bit definition, the reading, writing to and clearing of registers for the PTA and END extended event register models.

### 4.5.1 Bit definition of PTA event status register

The following describes the operation of the PTA event status register model, the naming of its event bits and what they mean.

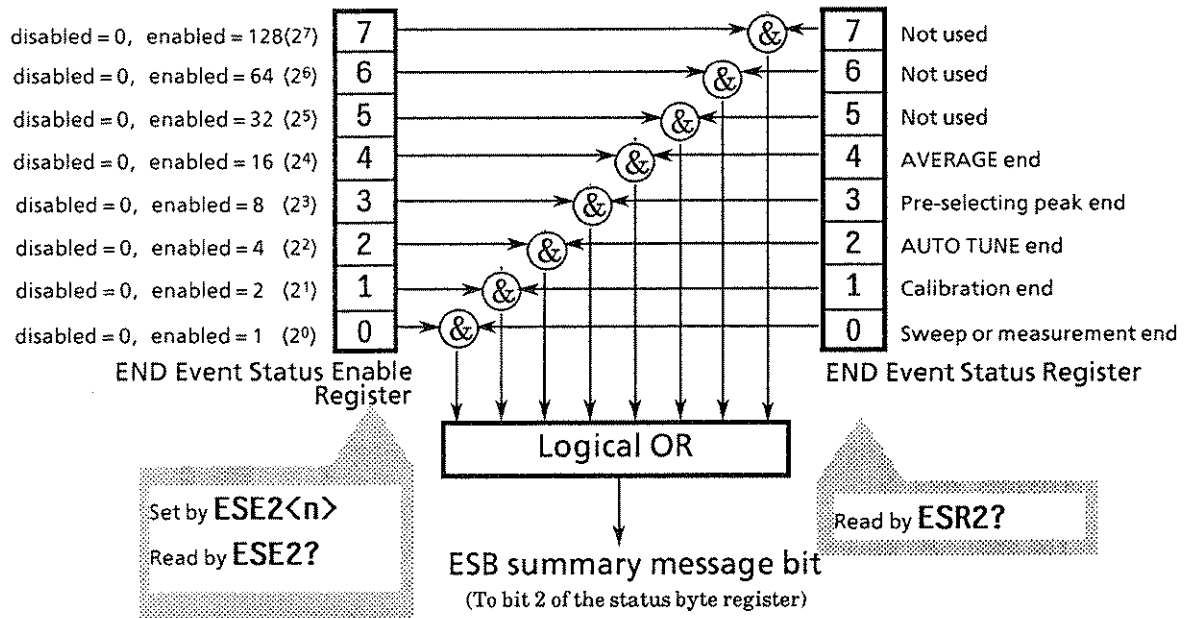


The left side PTA event status enable register selects whether the register makes the summary message true or not when the corresponding event status register becomes 1.

Bit	Event name	Description
7	Registration error	Error at program registration
6	Structure error	Error on program structure
5	Execution (operation) error	Error at operation on program execution
4	Execution (I/F etc.) error	Error at other than program operation
3	SRQ send	Used for sending service request with PTA subroutine CALL RSV (n)
2	SRQ send	
1	SRQ send	
0	SRQ send	

### 4.5.2 Bit definition of END event status register

The following describes the operation of the END event status register model, the naming of its event bits and what they mean.



The left side END event status enable register selects whether the register makes the summary message true or not when the corresponding event status register becomes 1.

Bit	Event name	Description
7	(Not used)	(Not used)
6	(Not used)	(Not used)
5	(Not used)	(Not used)
4	AVERAGE end	This bit is set to 1 when averaging terminates.
3	Pre-selecting peak end	This bit is set to 1 when pre-selecting peak terminates.
2	AUTO TUNE end	This bit is set to 1 when AUTO TUNE processing terminates.
1	CAL end	This bit is set to 1 when calibration terminates.
0	Sweep or measurement end	This bit is set to 1 when sweep or measurement terminates.

### 4.5.3 Reading, writing to and clearing the extended event status register

Reading	The register is destructively read by the a query, i.e. it is cleared after being read. The PTA and END event status registers are read by the ESR1? and ESR2? queries. The read value, an integer format data (NR1), is obtained by binary weighting the event bit and converting it to decimal.
Writing	With the exception of clearing, writing operations cannot be performed externally.
Clearing	The register is cleared in the following cases. ① A *CLS command is received ② The power is turned on when the power-on-status-clear flag is true. ③ An event is read for a query command

### 4.5.4 Reading, writing to and clearing the extended event status enable register

Reading	The register is non-destructively read by a query, i.e. it is not cleared after being read. The PTA and END event status registers are read by the ESE1? and ESE2? queries. The read value, returned by an integer format data (NR1), is obtained by binary weighting the event bit and converting it to decimal.
Writing	The PTA and END event status registers are written to by the ESE1 and ESE2 program commands. As bits 0 to 7 of the registers are respectively binary weighted to 1, 2, 4, 8, 16, 32, 64 and 128, data to be written is sent by < DECIMAL NUMERIC PROGRAM DATA >, the digit total weighted value of the bits selected from among them.
Clearing	The register is cleared in the following cases. ① ESE1 and ESE2 program commands with a data values of 0 are received by the PTA and END event status registers. ② When the power is turned on when the power-on-status-clear flag is true. The extended event status enable register is not affected by the followings: ① Changes of the status of the IEEE 488.1 device clear function ② A *RST common command is received ③ A *CLS common command is received

## 4.6 Techniques for Synchronizing MS8604A with a Controller

The MS8604A usually treats program messages as sequential commands that do not execute the processing of newly-received commands until the previous command has been processed. Thus, special consideration need not be taken for pair-synchronization between MS8604A and the controller.

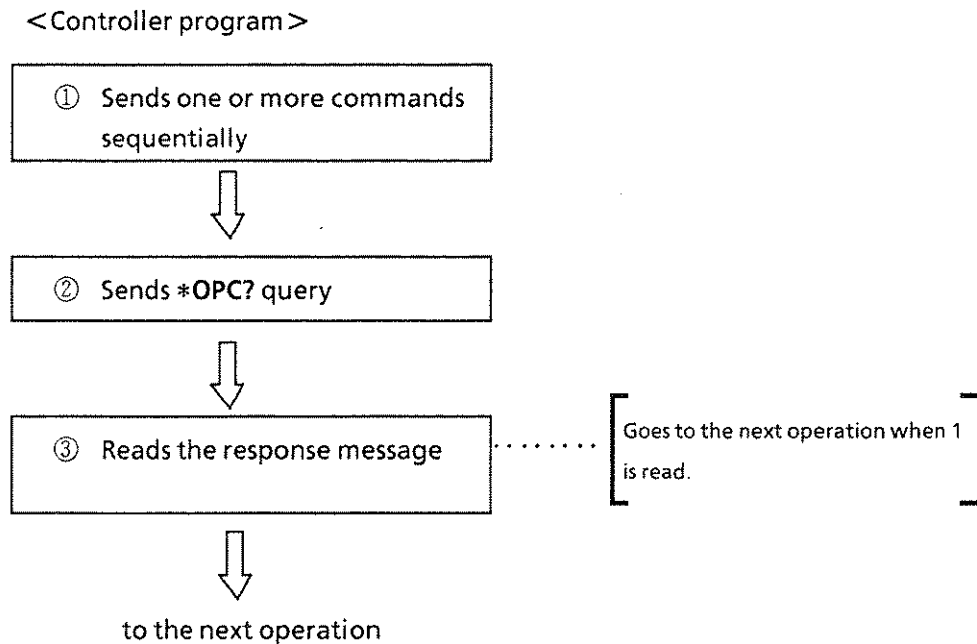
If the controller controls one or more devices and synchronizes with them, after all the commands specified for the MS8604A have been processed, the next commands must be sent to other devices.

There are two ways of synchronizing the MS8604A with the controller.

- ① Wait for a response after \*OPC? query is sent
- ② Wait for SRQ after \*OPC is sent

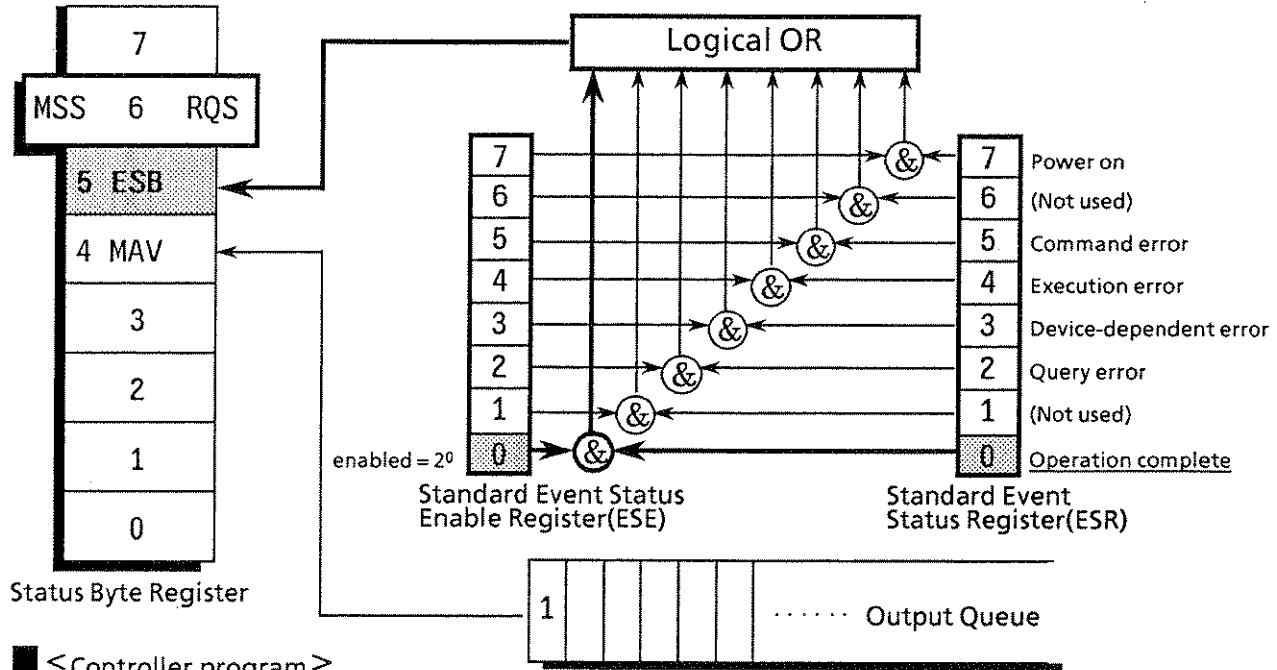
### 4.6.1 Wait for response after \*OPC? query is sent

The MS8604A outputs "1" as the response message when executing the \*OPC? query command. The controller is synchronized with the MS8604A by waiting for the response message to be entered.



### 4.6.2 Wait for service request after \*OPC is sent

The MS8604A sets the operation-complete bit (bit 0) to 1 when executing the \*OPC command. The controller is synchronized with the MS8604A by waiting for SRQ when the operation-complete bit is set for SRQ.



■ <Controller program>

- ① Enables the  $2^0$  bit (1) of the Standard Event Status Enable Register PRINT \_@1; "\*ESE\_1"
- ↓
- ② Enables the  $2^5$  bit (32) of the Service Request Enable Register PRINT \_@1; "\*SRE\_32"
- ↓
- ③ Makes the device execute the specified operation
- ↓
- ④ Sends the \*OPC command PRINT \_@1; "\*OPC"
- ↓
- ⑤ Wait for an SRQ interrupt (ESB summary message) ..... Value of status byte:  $2^6 + 2^5 = 96$

## SECTION 5 INITIAL SETTINGS

There are 3 levels of initialization for the GPIB interface system.

The first level is bus initialization in which the system bus is in the idle state.

The second level is initialization for message exchange in which devices are able to receive program message.

The third level is device initialization in which device functions are initialized.

These levels of initialization prepare a device for operation.

### TABLE OF CONTENTS

5.1	Bus Initialization by the IFC Statement .....	5-4
5.2	Initialization for Message Exchange by DCL and SDC Bus Commands .....	5-5
5.3	Device Initialization by the *RST Command .....	5-6
5.4	Device Initialization by the PRE/INI/IP Command .....	5-7
5.5	Device Status at Power-on .....	5-8

**(Blank)**



In the IEEE 488.2 standard, the initialization levels are stipulated to be divided into three: the first level is “ bus initialization ”, the second level is “ initialization for message exchange ”, and the third level is “ device initialization ”. This standard also stipulates that a device must be set to a known state when the power is turned on.

Level	Initialization type	Description	Level combination and sequence
1	Bus initialization	The IFC message from the controller initializes all interface functions connected to the bus.	Can be combined with other levels, level 1 must be executed before level 2.
2	Initialization for message exchange	The message exchanges of all devices and specified devices on the GPIB are initialized respectively by the DCL (Device Clear) and SDC (Selected Device Clear) GPIB bus commands, which also nullify the function that reports to the controller that operation has completed.	Can be combined with other levels, level 2 must be executed before level 3.
3	Device initialization	The *RST or PRE/INI/IP command returns the specified device to the device-dependent known state, regardless of the conditions under which they were previously being used.	Can be combined with other levels, level 3 must be executed after levels 1 and 2.

The following paragraph describes the commands for executing levels 1, 2, and 3 and the items that are initialized by the execution. It also describes the known state which is set when the power is switched on.

## 5.1 Bus Initialization by the IFC Statement

### ■ Example

```
ISET_IFC
```

### ■ Explanation

The IFC statement initializes the interface functions of all devices connected to the GPIB bus line.

The initialization of interface functions involves erasing the settings made by the controller and resetting them to their initial states. In the table below, ○ indicates the functions which are initialized; △ indicates the functions which are partially initialized.

No	Function	Symbol	Initialization by IFC
1	Source handshake	SH	○
2	Acceptor handshake	AH	○
3	Talker or extended talker	T or TE	○
4	Listener or extended listener	L or LT	○
5	Service request	SR	△
6	Remote/local	RL	
7	Parallel poll	PP	
8	Device clear	DC	
9	Device trigger	DT	
10	Controller	C	○

Bus initialization by the IFC statement does not affect the device-operating state (frequency settings, lamp on/off, etc).

## 5.2 Initialization for Message Exchange by DCL and SDC Bus Commands

### ■ Example

WBYTE\_&H3F, \_&H14; Initializes all devices under the bus for message exchange (sending DCL)

WBYTE\_&H3F,                    Initializes only the device whose address is 3 for message exchange (sending SDC)

### ■ Explanation

This statement executes the initialization for message exchange of all devices or only the specified device on the GPIB of the specified select code.

### ■ Items to be initialized for message exchange

The MS8604A by which the DCL or SDC bus command is accepted executes the following.

- |  |   |
|--|---|
| ① Input buffer and Output Queue .....                        | Cleared, at the same time the MAV bit is also cleared.  |
| ② Parser, Execution Controller, and Response Formatter ..... | Reset   |
| ③ Device commands including *RST .....                       | Clears all commands that prevent these commands from executing.   |
| ④ Processing the *OPC command .....                          | Puts a device in OCIS (Operation Complete Command Idle State). As a result, the operation complete bit cannot be set in the Standard Event Status Register. |
| ⑤ Processing the *OPC? query .....                           | Puts a device in OQIS (Operation Complete Query Idle State). As a result, the operation complete bit 1 cannot be set in the Output Queue.                   |

**Note:** The items listed below are not affected even if processing the DCL and SDC commands is executed.

- ① The current data set or stored in the device.
- ② Front panel settings.
- ③ Other status byte state except MAV bit.
- ④ Device operation in progress.

### 5.3 Device Initialization by the \*RST Command

#### ■ Syntax

---

\*RST

---

#### ■ Example

PRINT\_@1;"\*RST" ..... Initializes the device (MS8604A) whose address is 1 with level 3

#### ■ Explanation

The \*RST (Reset) command is an IEEE 488.2 common command which resets a device with level 3.

The \*RST (Reset) command is used to reset a device (MS8604A) to a specific initial state. Refer to the separate volume " Panel Operation Appendix B " for details of initialization items and initial values.

*Note:* The \*RST command does not affect the items listed below.

- ① IEEE 488.1 interface state
- ② Device address
- ③ Output Queue
- ④ Service Request Enable register
- ⑤ Standard Event Status Enable register
- ⑥ Power-on-status-clear flag setting
- ⑦ Calibration data affecting device specifications
- ⑧ Parameters preset for controlling external devices etc

## 5.4 Device Initialization by the PRE/INI/IP Command

### ■ Syntax

---

PRE  
INI  
IP

---

### ■ Example (program message)

PRINT\_@1;"PRE" ..... Initializes the device (MS8604A) whose address is 1 with level 3

### ■ Explanation

The PRE/INI, and IP commands are MS8604A device-dependent messages which initialize a device with level 3.

Refer to the separate volume " Panel Operation Appendix B " for details of items initialized by the PRE, INI, and IP commands and initial values.

## 5.5 Device Status at Power-on

When the power is switched on:

- ① The device is set to the status before the last power off.
- ② The Input Buffer and Output Queue are cleared.
- ③ The Parser, Execution Controller, and Response Formatter are initialized.
- ④ The device is put into the OCIS (Operation Complete Command Idle State).
- ⑤ The device is put into the OQIS (Operation Complete Query Idle State).
- ⑥ The Standard Event Status and Standard Event Status Enable Registers are cleared. Events can be recorded after the registers have been cleared.

For the special case of ①, when the power supply is first turned on after the device is shipped, the initial values are set to those in the initial setting table (refer to the separate volume "Panel Operation Appendix B).

## SECTION 6

### DETAILS OF DEVICE MESSAGES

This section lists the device messages in alphabetic order and describes their functions. The device messages are classified into four types; IEEE 488.2 common commands, MS8604A common commands, TX Test mode commands, and spectrum mode commands.

#### TABLE OF CONTENTS

6.1	IEEE 488.2 common commands	6-7	GTOUT	6-24
	*CLS	6-8	INI	6-24
	*ESE	6-8	IP	6-25
	*ESR?	6-9	KSE	6-25
	*IDN?	6-9	MC8104	6-26
	*OPC	6-10	PLS	6-26
	*OPC?	6-10	PMCS	6-27
	*PCB	6-11	PMOD	6-27
	*PSC	6-11	PNLMD	6-28
	*PSC?	6-12	PRE	6-28
	*RST	6-12	PRIA	6-29
	*SRE	6-13	PRINT	6-29
	*STB?	6-13	PRTY	6-30
	*TRG	6-14	RC	6-30
	*TST?	6-14	RCM	6-31
	*WAI	6-15	RDATA	6-31
			RFINPUT	6-32
6.2	MS8604A common commands	6-16	RGDIR	6-32
	ALARM	6-17	RGRC	6-33
	BAUD	6-17	RGSV	6-33
	BEP	6-18	SNGLS	6-34
	BIN	6-18	STPB	6-34
	BZR	6-19	SV	6-35
	CONTS	6-19	SVM	6-35
	DATE	6-20	SWP	6-36
	DSPL	6-20	S1	6-36
	DTAB	6-21	S2	6-37
	ESE1	6-21	TIME	6-37
	ESE2	6-22	TITLE	6-38
	ESR1?	6-22	TOUT	6-38
	ESR2?	6-23	TRM	6-39
	GPIA	6-23	TTL	6-39

## TABLE OF CONTENTS (Continued)

XMA .....	6-40	FVECTERR? .....	6-64
XMB .....	6-40	HMODPWR? .....	6-64
XMC .....	6-41	INTPOL .....	6-65
XMD .....	6-41	JITTER? .....	6-65
XME .....	6-42	KSA .....	6-66
XMT .....	6-42	KSG .....	6-66
		KSH .....	6-67
6.3 TX Test commands .....	6-44	LVLREL .....	6-67
ADJCH? .....	6-45	M1 .....	6-68
ADJMEAS .....	6-46	M2 .....	6-68
AUNITS .....	6-46	MAGERR? .....	6-69
AUTORNG .....	6-47	MAGTDERR? .....	6-69
AVR .....	6-47	MEANPWR? .....	6-70
BITR? .....	6-48	MEAS .....	6-71
BITRERR? .....	6-48	MEASOBJ .....	6-72
BPATT .....	6-49	MKL? .....	6-73
BS .....	6-49	MKN .....	6-73
BSTAVG .....	6-50	MKOFF .....	6-74
CALADJ .....	6-50	MKP .....	6-74
CALOSC .....	6-51	MKR .....	6-75
CARRF? .....	6-51	MKRS .....	6-75
CARRFERR? .....	6-52	MKZ .....	6-76
CDJ .....	6-52	MKZF .....	6-76
CF .....	6-53	MODPWR? .....	6-77
CHAN .....	6-53	MSTAT? .....	6-78
CHCARR .....	6-54	MTEMPREL .....	6-78
CHFREQ .....	6-54	NUMSYM? .....	6-79
CHNUM .....	6-55	OBW? .....	6-79
CHSPC .....	6-55	OBWFREQ? .....	6-80
CNF .....	6-56	OCCBW? .....	6-80
COS .....	6-56	OFF .....	6-81
CRCAL? .....	6-57	OFFPWR? .....	6-81
CSCAL? .....	6-57	ORGNOS? .....	6-82
CSNUM .....	6-58	PATT .....	6-82
CSSLLOT .....	6-58	PEAKPWR? .....	6-83
CST .....	6-59	PHASEERR? .....	6-84
DBR .....	6-59	POWER? .....	6-84
DRPFACT? .....	6-60	PSSIOT .....	6-85
ERRSC .....	6-60	PVECTERR? .....	6-85
FBAND .....	6-61	RAMPSYM? .....	6-86
FDN .....	6-61	RATIO? .....	6-86
FREQ .....	6-62	RFATTN .....	6-87
FREQADJ .....	6-63	RFLVL .....	6-87
FUP .....	6-63	RNG .....	6-88



## TABLE OF CONTENTS (Continued)

RTNYQ .....	6-88	AVB .....	6-116
SAVETBL .....	6-89	AVR .....	6-116
SAVETEMP .....	6-89	AWR .....	6-117
SCAL .....	6-90	A1 .....	6-117
SCNCT .....	6-90	A2 .....	6-118
SCOFS .....	6-91	BNDC .....	6-118
SETREL .....	6-91	BSAUTO .....	6-119
SLCTTBL .....	6-92	CA .....	6-119
SLCTTEMP .....	6-92	CAL .....	6-120
SLOTPWR? .....	6-93	CF .....	6-120
SPUFREQ .....	6-94	CLRW .....	6-121
SPUHRM .....	6-95	CMK? .....	6-121
SPULVL? .....	6-95	CNF .....	6-122
STARTPT .....	6-96	COUPLE .....	6-122
STD27 .....	6-96	CR .....	6-123
STORAGE .....	6-97	CT .....	6-123
SWPWR? .....	6-98	CV .....	6-124
SYMTIME .....	6-99	DET .....	6-124
SYS .....	6-99	DETM .....	6-125
TEMPLVL .....	6-100	DFMT .....	6-125
TERM .....	6-101	DPOINT .....	6-126
TIMING? .....	6-101	EXTTYPE .....	6-126
TXPWR? .....	6-102	E1 .....	6-127
TXTIME .....	6-102	E2 .....	6-127
UCAL .....	6-103	E3 .....	6-128
UNIT .....	6-104	E4 .....	6-128
UNT .....	6-105	FA .....	6-129
VAVG .....	6-106	FB .....	6-129
VECTERR? .....	6-106	FDN .....	6-130
WINDOW .....	6-107	FMRNG .....	6-130
ZAJ .....	6-107	FRQ .....	6-131
ZEROSET .....	6-108	FS .....	6-131
		FSS .....	6-132
6.4 Spectrum mode commands .....	6-110	FUP .....	6-132
AAT .....	6-111	GATE .....	6-133
AMD .....	6-111	GD .....	6-133
ARB .....	6-112	GE .....	6-134
AST .....	6-112	GL .....	6-134
ASWT .....	6-113	HN .....	6-135
AT .....	6-113	HNLOCK .....	6-135
ATT .....	6-114	HNUNLK .....	6-136
ATUN .....	6-114	KSA .....	6-136
AUNITS .....	6-115	KSB .....	6-137
AUTO .....	6-115	KSC .....	6-137

## TABLE OF CONTENTS (Continued)

KSD .....	6-138	M2 .....	6-160
KSG .....	6-138	M3 .....	6-160
KSH .....	6-139	PARAM .....	6-161
KSO .....	6-139	PCF .....	6-161
LG .....	6-140	PP .....	6-162
LN .....	6-140	PRESEL .....	6-162
LSS .....	6-141	PRL .....	6-163
LSSA .....	6-141	PSW .....	6-163
MKA? .....	6-142	RB .....	6-164
MKACT .....	6-142	RBW .....	6-164
MKCF .....	6-143	RL .....	6-165
MKD .....	6-143	RLV .....	6-166
MKF? .....	6-144	RMK? .....	6-166
MKL? .....	6-144	SCL .....	6-167
MKLIST .....	6-145	SOF .....	6-167
MKMCL .....	6-145	SP .....	6-168
MKMHI .....	6-146	SPD .....	6-168
MKMHRM .....	6-146	SPF .....	6-169
MKMIN .....	6-147	SPFUNC .....	6-169
MKML? .....	6-147	SPU .....	6-170
MKMP .....	6-148	SS .....	6-170
MKMULTI .....	6-148	ST .....	6-171
MKN .....	6-149	STF .....	6-171
MKOFF .....	6-149	SWT .....	6-172
MKP .....	6-150	TDLY .....	6-172
MKPK .....	6-150	TEXPAND .....	6-173
MKR .....	6-151	TM .....	6-173
MKRL .....	6-151	TMMD .....	6-174
MKS .....	6-152	TMWR .....	6-174
MKSLCT .....	6-152	TRG .....	6-175
MKSP .....	6-153	TRGLVL .....	6-175
MKSS .....	6-153	TRGS .....	6-176
MKTRACK .....	6-154	TRGSLP .....	6-176
MKW .....	6-154	TRGSOURCE .....	6-177
MKZ .....	6-155	TS .....	6-177
MKZF .....	6-155	TSP .....	6-178
MON .....	6-156	TVLINE .....	6-178
MONVOL .....	6-156	TVSFRM .....	6-179
MT0 .....	6-157	TVSTND .....	6-179
MT1 .....	6-157	TZONE .....	6-180
MXMH .....	6-158	TZSP .....	6-180
MZW .....	6-158	TZSTART .....	6-181
MZWF .....	6-159	UCL? .....	6-181
M1 .....	6-159	UNC .....	6-182

## TABLE OF CONTENTS (Continued)

UNT .....	6-182
VAVG .....	6-183
VB .....	6-183
VBCOUPLE .....	6-184
VBR .....	6-184
VBW .....	6-185
VIEW .....	6-185

(Blank)

## 6.1 IEEE 488.2 common commands

## \*CLS

### \*CLS Clear Status Command

■ Function Clears the Status Byte Register.

Header	Program command	Query	Response
*CLS	*CLS	_____	_____

■ Example \*CLS

---

## \*ESE

### \*ESE Standard Event Status Enable

■ Function Sets or clears the Standard Status Enable Register.

Header	Program command	Query	Response
*ESE	*ESE_n	*ESE?	n

■ Value of n 0 to 255

■ Example \*ESE\_20  
\*ESE?

## \* ESR?

### \* ESR?

### Standard Event Status Register Query

■ Function

Returns the current value in the Standard Event Status Register.

Header	Program command	Query	Response
*ESR?	_____	*ESR?	n

■ Value of n

0 to 255

■ Example

\*ESR?

---

## \* IDN?

### \* IDN?

### Identification Query

■ Function

Returns the manufacturer name, model number etc. of the product.

Header	Program command	Query	Response
*IDN?	_____	*IDN?	ANRITSU,MS8604A,0,n

■ Value of n

1 to 99 (firmware version No.)

■ Example

\*IDN?

## \* OPC

### \* OPC      Operation Complete Command

■ Function      Sets the bit 0 digit in the Standard Event Status Register when all pending selected device operations have been completed.

Header	Program command	Query	Response
*OPC	*OPC	_____	_____

■ Example      \*OPC

---

## \* OPC?

### \* OPC?      Operation Complete Query

■ Function      Sets 1 in the Output Queue to generate an MAV summary message when all pending selected device operations have been completed.

Header	Program command	Query	Response
*OPC?	_____	*OPC?	1

■ Example      \*OPC?



## \* PCB

### \* PCB      Pass Control Back

■ Function      Passes the bus control authority.

Header	Program command	Query	Response
*PCB	*PCB_a	_____	_____

■ Value of a      0 to 30

■ Example      \*PCB\_0

---

## \* PSC

### \* PSC      Power on Status Clear

■ Function      Sets or clears the power on status clear flag.

Header	Program command	Query	Response
*PSC	*PSC_n	*PSC?	n

■ Value of n      0,1

■ Example      \*PSC\_0  
\*PSC?

## \*PSC?

### \*PSC? Power on Status Clear Query

■ Function Returns the current value of the power on status clear flag.

Header	Program command	Query	Response
*PSC?	_____	*PSC?	n

■ Value of n 0, 1

■ Example \*PSC?

---

## \*RST

### \*RST Reset Command

■ Function Resets the device in the third level.

Header	Program command	Query	Response
*RST	*RST	_____	_____

■ Example \*RST

**\* SRE                      Service Request Enable Command**

■ **Function**                      Sets the bits in the Service Request Enable Register.

Header	Program command	Query	Response
*SRE	*SRE	*SRE?	n

■ **Value of n**                      0 to 63, 128 to 191 (current value of Service Request Enable Register)

■ **Example**                      \*SRE

**\*STB?                      Read Status Byte Command**

■ **Function**                      Returns the current values of the status bytes including the MSS bit.

Header	Program command	Query	Response
*STB?	————	*STB?	n

■ **Value of n**

Bit	Bit weight	Bit name	Conditions of Status Byte Register
7	128	————	0 = Not used
6	64	MSS	0 = Service not requested, 1 = Service requested
5	32	ESB	0 = Event status not generated, 1 = Event status generated
4	16	MAV	0 = No data in Output Queue, 1 = Data in Output Queue
3	8	————	0 = Not used
2	4	ESB( END )	0 = Event status not generated, 1 = Event status generated
1	2	ESB( PTA )	0 = Event status not generated, 1 = Event status generated
0	1	————	0 = Not used

■ **Example**                      \*STB?

## \* TRG

### \* TRG      Trigger Command

- Function      Same function as that of IEEE488.1 GET-Group Execute Trigger-bus command. For this command, the MS8604A executes a single sweep (same function as SWP.)

Header	Program command	Query	Response
*TRG	*TRG	_____	

- Example      \*TRG

---

## \* TST?

### \* TST?      Self Test Query

- Function      Executes a self-test and returns the results of any errors.

Header	Program command	Query	Response
*TST?	_____	*TST?	n

- Value of n      0:      Indicates that the self-test completed without errors.  
                         -32767 to -1,  
                         1 to 32767:      Indicates that the self-test did not complete, or completed with errors.

- Example      \*TST?

**\*WAI                      Wait-to-Continue Command**

■ **Function**                      Keeps the next command on stand-by while the device is executing a command.

Header	Program command	Query	Response
*WAI	*WAI	_____	

■ **Example**                      \*WAI

## 6.2 MS8604A common commands

---

**WARNING**

---

- *The protective earth terminal of this instrument must be connected to ground. The three-core power cord supplied with the instrument can be plugged into a grounded two pole AC outlet. If no grounded two pole AC outlet is available, the ground pin of the power cord or the earth terminal on the rear panel must be connected to ground before supplying the power to the instrument. Failure to do so could cause dangerous or possibly fatal electric shocks.*
- *Replacing fuses with the power cord still plugged into an AC outlet could also cause electric shocks.*
- *Supplemental explanation about WARNING on the rear panel*

**WARNING** 

NO OPERATOR SERVICE-  
ABLE PARTS INSIDE.  
REFER SERVICING TO  
QUALIFIED PERSONNEL.

} *A supplemental explanation about the WARNING labeled on the rear panel is given in the following:*

*Disassembly, adjustment, maintenance, or other access inside this instrument by unqualified personnel should be avoided. Maintenance of this instrument should be performed only by Anritsu trained service personnel who are familiar with the risks involved of fire and electric shock. Potentially lethal voltages existing inside this instrument, if contacted accidentally, may result in personal injury or death, or in the possibility of damage to precision components.*

---

## ■ SAFETY CONSIDERATIONS:

Anritsu uses the following labels to identify safety precautions which should be followed to prevent personal injury or product damage. Please familiarize yourself with them before operating this product.

### Labels used in this manual:

**WARNING** : Indicates that the procedure could result in personal injury if not correctly performed. Do not proceed before you fully understand the explanation given with this symbol and meet the required conditions.

**CAUTION** : Indicates that the operating procedure could result in damage to the product if not correctly performed. Do not proceed before you fully understand the explanation given with this symbol and meet the required conditions.

**Note** : Indicates that information helpful in understanding the operation of the product is about to be presented.

### Labels or symbols used on/in the product:



: This international caution symbol indicates that the operator should refer to the operation manual before beginning a procedure.



: This symbol indicates an earth (ground) terminal. The product should be grounded via the earth terminal if a three prong power cord is not used.



## CERTIFICATION

ANRITSU CORPORATION certifies that this instrument has been thoroughly tested and inspected, and found to meet published specifications prior to shipping.

Anritsu further certifies that its calibration measurements are based on the Japanese Electrotechnical Laboratory and Radio Research Laboratory standards.

## WARRANTY

All parts of this product are warranted by Anritsu Corporation of Japan against defects in material or workmanship for a period of one year from the date of delivery.

In the event of a defect occurring during the warranty period, Anritsu Corporation will repair or replace this product within a reasonable period of time after notification, free-of-charge, provided that: it is returned to Anritsu; has not been misused; has not been damaged by an act of God; and that the user has followed the instructions in the operation manual.

Any unauthorized modification, repair, or attempt to repair, will render this warranty void.

This warranty is effective only for the original purchaser of this product and is not transferable if it is resold.

***ALL OTHER EXPRESSED WARRANTIES ARE DISCLAIMED AND ALL IMPLIED WARRANTIES FOR THIS PRODUCT, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO A PERIOD OF ONE YEAR FROM THE DATE OF DELIVERY. IN NO EVENT SHALL ANRITSU CORPORATION BE LIABLE TO THE CUSTOMER FOR ANY DAMAGES, INCLUDING LOST PROFITS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE THIS PRODUCT.***

All requests for repair or replacement under this warranty must be made as soon as possible after the defect has been noticed and must be directed to Anritsu Corporation or its representative in your area.

---

'IBM' is a registered trademark of the IBM Corporation.  
'HP' is a registered trademark of the Hewlett-Packard Company.  
'MS-DOS' is a registered trademark of the Microsoft Corporation.  
'NEC' is a registered trademark of the NEC Corporation.

---

## MEMORY BACK-UP BATTERY REPLACEMENT

---

The power for memory back-up is supplied by a Poly-carbomonofluoride Lithium Battery. This battery should only be replaced by a battery of the same type; since replacement can only be made by Anritsu, contact the nearest Anritsu representative when replacement is required.

At the end of it's life, the battery should be recycled or disposed properly.

---

## STORAGE MEDIUM

---

This equipment stores data and programs using Plug-in Memory cards (PMC) and backed-up memories. Data and programs may be lost due to improper use or failure. ANRITSU therefore recommends that you back-up the memory.

### ANRITSU CANNOT COMPENSATE FOR ANY MEMORY LOSS.

Please pay careful attention to the following points. Do not remove the IC card and backed-up memory from equipment being accessed.

(PMC)

- Isolate the card from static electricity.
- The back-up battery in the SRAM card has a limited life; replace the battery periodically.

(Backed-up memory)

- Isolate the memory from static electricity.

**Note:** The battery life is about 7 years. Early battery replacement is recommended.

---

## DISPOSING OF THE PRODUCT

---

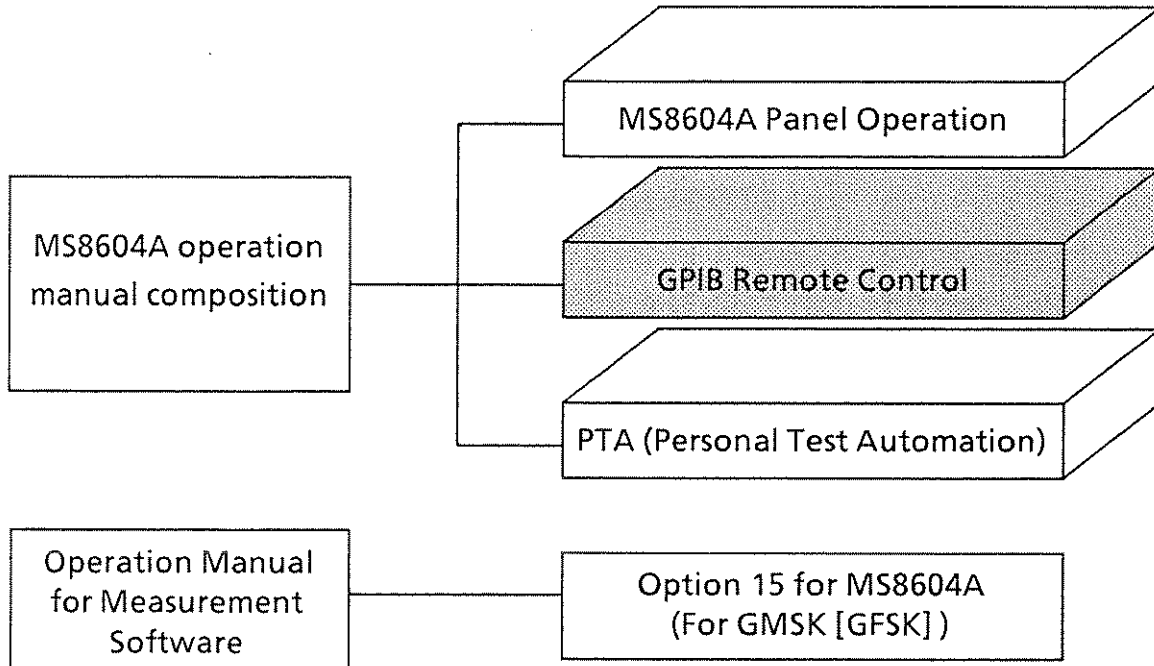
The MS8604A uses chemical compound semiconductor including arsenic.

At the end it's life, the MS8604A should be recycled or disposed properly.

## ABOUT THIS MANUAL

### (1) Operation manual composition

The MS8604A Operation manuals are made up of the following three manuals. Use the manuals matching the usage objective.



**Panel Operation:** Outlines the MS8604A and describes its specifications, preparations, panels, manual (local) operation method, storage, transportation, function keys transition diagram, initial values table, and error messages.

**GPIB Remote Control:** Since the MS8604A is compatible with IEEE488.2, this manual describes GPIB remote control based on IEEE488.2. The descriptions in this manual are based on N<sub>88</sub>-BASIC programs using an NEC PC9800 Series personal computer for program generation reference.

**PTA Personal Test Automation:** Describes how to program high-speed control and high-speed operation processing directly connected to the measurement system by high level language PTL. The program is executed by a personal computer built into the MS8604A and is called PTA (Personal Test Automation). Together with GPIB mentioned above, it promotes measurement automation.

## (2) Basic Guide to GPIB (sold separately)

A "Basic Guide to GPIB" is sold separately from the operation manuals above.

This guide provides a basic knowledge of GPIB and describes the GPIB control statements written in Anritsu PACKET V computer language.

**(Blank)**

# TABLE OF CONTENTS

<b>SECTION 1</b>	<b>GENERAL</b> .....	1-1
1.1	GENERAL .....	1-3
1.2	Functions of GPIB .....	1-3
1.3	Functions of the GPIB with two ports .....	1-3
1.4	Examples of system configuration using GPIB 1/GPIB 2 .....	1-4
1.5	Specifications .....	1-5
1.5.1	Interface function .....	1-5
1.5.2	Device message list .....	1-7
<b>SECTION 2</b>	<b>CONNECTING THE BUS AND SETTING THE ADDRESS</b> .....	2-1
2.1	Connecting Devices with GPIB Cables .....	2-3
2.2	Setting and Checking GPIB addresses .....	2-4
<b>SECTION 3</b>	<b>DEVICE MESSAGE FORMAT</b> .....	3-1
3.1	General Description .....	3-3
3.2	Program message format .....	3-3
3.3	Response message format .....	3-7
<b>SECTION 4</b>	<b>STATUS MESSAGE</b> .....	4-1
4.1	IEEE 488.2 Standard Status Model .....	4-4
4.2	Status Byte (STB) Register .....	4-6
4.2.1	ESB and MAV summary messages .....	4-6
4.2.2	Device-dependent summary messages .....	4-7
4.2.3	Reading and clearing the STB register .....	4-8
4.3	Enabling SRQ .....	4-9
4.4	Standard Event Status Register .....	4-10
4.4.1	Bit definition of standard event status register .....	4-10
4.4.2	Query error details .....	4-11
4.4.3	Reading, writing to and clearing the standard event status register .....	4-12
4.4.4	Reading, writing to and clearing the standard event status enable register .....	4-12

4.5	Extended Event Status Register .....	4-13
4.5.1	Bit definition of PTA event status register .....	4-14
4.5.2	Bit definition of END event status register .....	4-15
4.5.3	Reading, writing to and clearing the extended event status register .....	4-16
4.5.4	Reading, writing to and clearing the extended event status enable register .....	4-16
4.6	Techniques for Synchronizing MS8604A with a Controller .....	4-17
4.6.1	Wait for response after *OPC? query is sent .....	4-17
4.6.2	Wait for service request after *OPC is sent .....	4-18
<b>SECTION 5</b>	<b>INITIAL SETTINGS .....</b>	<b>5-1</b>
5.1	Bus Initialization by the IFC Statement .....	5-4
5.2	Initialization for Message Exchange by DCL and SDC Bus Commands .....	5-5
5.3	Device Initialization by the *RST Command .....	5-6
5.4	Device Initialization by the PRE/INI/IP Command .....	5-7
5.5	Device Status at Power-on .....	5-8
<b>SECTION 6</b>	<b>DETAILS OF DEVICE MESSAGES .....</b>	<b>6-1</b>
6.1	IEEE 488.2 common commands .....	6-7
6.2	MS8604A common commands .....	6-16
6.3	TX Test commands .....	6-44
6.4	Spectrum mode commands .....	6-110
<b>SECTION 7</b>	<b>SAMPLE PROGRAMS .....</b>	<b>7-1</b>
7.1	Precautions on Creating the GPIB Program .....	7-3
7.2	Sample Programs .....	7-4
<b>SECTION 8</b>	<b>WAVEFORM DATA STORAGE FORMAT .....</b>	<b>8-1</b>
8.1	Notes on Fetching the Waveform Data .....	8-3
8.2	Waveform Data Storage Format .....	8-4
<b>APPENDIX A</b>	<b>ASCII*CODE TABLE .....</b>	<b>A-1</b>
<b>APPENDIX B</b>	<b>COMPARISON TABLE OF CONTROLLERS' GPIB INSTRUCTIONS .....</b>	<b>B-1</b>



# ALARM

## ALARM Alarm

- **Function** Turns the alarm on and off.

Header	Program command	Query	Response
ALARM	ALARM_a	ALARM?	a

- **Value of a** ON: Turns the alarm ON and buzzes the alarm once.  
OFF: Turns the alarm OFF.
- **Suffix code** None
- **Initial setting** OFF
- **Example** ALARM\_ON

---

# BAUD

## BAUD Baud Rate (valid only when option 02 RS-232C is used)

- **Function** Sets the Baud rate of the RS-232C.

Header	Program command	Query	Response
BAUD	BAUD_a	BAUD?	a

- **Value of a** 9600, 4800, 2400, 1200, 600, 300,
- **Suffix code** None
- **Initial setting** 2400
- **Example** BAUD\_9800

# BEP

## BEP Buzzer Switch

■ Function Turns the buzzer switch ON or OFF (whether to buzz when an error occurs).

Header	Program command	Query	Response
BEP	BEP_sw sw=ON, 1, OFF, 0	_____	_____

■ Value of sw ON: ON  
1: ON  
OFF: OFF  
Ø: OFF

■ Suffix code None

■ Initial setting ON: ON

■ Example BEP\_ON

---

# BIN

## BIN ASCII/Binary Data Out

■ Function Outputs the trace data in ASCII or BINARY format.

Header	Program command	Query	Response
BIN	BIN_a a=0, 1, ON, OFF	_____	_____

■ Value of a Ø: ASCII  
1: BINARY  
OFF: ASCII  
ON: BINARY

■ Suffix code None

■ Initial setting Ø: ASCII

■ Example BIN\_Ø  
BIN\_ON

# BZR

## BZR Sounds Buzzer

■ Function Sounds the buzzer.

Header	Program command	Query	Response
BZR	BZR	_____	_____

■ Example BZR

---

# CONTS

## CONTS Continuous Sweep Mode

■ Function Sets the sweep mode to continuous mode (same function as S1).

Header	Program command	Query	Response
CONTS	CONTS	_____	_____

■ Example CONTS

# DATE

## DATE

### Date

- **Function** Sets the clock built in MS8604A instrument to the specified date.

Header	Program command	Query	Response
DATE	DATE_ yy, mm, dd	DATE?	yy, mm, dd

- **Value of yy** 00 to 99 (year)
- **Value of mm** 01 to 12 (month)
- **Value of dd** 01 to 31 (day)
- **Suffix code** None
- **Example** DATE\_92,08,31

---

# DSPL

## DSPL

### Display

- **Function** Turns the screen display on and off.

Header	Program command	Query	Response
DSPL	DSPL_ a	DSPL?	a

- **Value of a** ON: On Displays the screen.  
OFF: Off Does not display the screen.
- **Suffix code** None
- **Initial setting** ON
- **Example** DSPL\_OFF

# DTAB

## DTAB Data Bit (valid only when option 02 RS-232C is used)

- Function Sets the data bit of the RS-232C.

Header	Program command	Query	Response
DTAB	DTAB_a	DTAB?	a

- Value of a 7: 7 bit  
8: 8 bit
- Suffix code None
- Initial setting 8
- Example DTAB\_7

---

# ESE1

## ESE1 Event Status Enable (PTA)

- Function Allows the PTA Event Status Enable Register to select which bit in the corresponding Event Register causes a TRUE ESB summary message bit 1 when set.

Header	Program command	Query	Response
ESE1	ESE1_a a = 0 to 255	ESE1?	a a = 0 to 255

- Value of a 0 to 255:  
Represents the sum of the bit weighted values enabled by the square of the bit number corresponding to bits 0, 1, 2, 3, 4, 5, 6, 7 of END Event Status Register.
- Suffix code None
- Example ESE1\_1

## ESE2

### ESE2 Event Status Enable (END)

- **Function** Allows the END Event Status Enable Register to select which bit in the corresponding Event Register causes a TRUE ESB summary message bit 2 when set.

Header	Program command	Query	Response
ESE2	ESE2┐a a = 0 to 255	ESE2?	a a = 0 to 255

- **Value of a** 0 to 255:  
Represents the sum of the bit weighted values enabled by the square of the bit number corresponding to bits 0, 1, 2, 3, 4, 5, 6, 7 of END Event Status Register.
- **Suffix code** None
- **Example** ESE2┐1

---

## ESR1?

### ESR1? Event Status Register (PTA)

- **Function** Allows the sum of the binary-weighted event bit values of the PTA Event Status Register to be read out by converting them to decimals. After readout, the PTA Event Status Register is reset to 0.

Header	Program command	Query	Response
ESR1?	—————	ESR1?	a a = 0 to 255

- **Value of a** 0 to 255
- **Suffix code** None
- **Example** ESR1?

## ESR2?

### ESR2? Event Status Register (END)

■ Function

Allows the sum of the binary-weighted event bit values of the END Event Status Register to be read out by converting them to decimals. After readout, the END Event Status Register is reset to 0.

Header	Program command	Query	Response
ESR2?	_____	ESR2?	a a = 0 to 255

■ Value of a 0 to 255

■ Suffix code None

■ Example ESR2?

---

## GPIA

### GPIA GPIB 2 Self Address

■ Function

Sets GPIB 2 self address.

Header	Program command	Query	Response
GPIA	GPIA_a a=0 to 30	GPIA?	a a=0 to 30

■ Value of a 0 to 30

■ Suffix code None

■ Initial setting Value of a = 16 (provided the address already allocated is not initialized)

■ Example GPIA\_0  
GPIA\_30

# GTOUT

## GTOUT

## GPIB Time Out

- Function Sets the time-out time of the GPIB.

Header	Program command	Query	Response
GTOUT	GTOUT_a	GTOUT?	a

- Value of a 0 to 255 (seconds, 1-second steps)
- Suffix code None
- Initial setting  $\emptyset$
- Example GTOUT\_100

---

# INI

## INI

## Initialize

- Function Initializes all measurement control parameters to be initialized (same function as PRE and IP).

Header	Program command	Query	Response
INI	INI	_____	_____

- Example INI



**IP Initialize**

- **Function** Initializes all measurement control parameters to be initialized (same function as PRE and INI).

Header	Program command	Query	Response
IP	IP	_____	_____

- **Example** IP

**KSE Title Entry**

- **Function** Registers the title character string (same function as TITLE).

Header	Program command	Query	Response
KSE	KSE_text	_____	_____

- **Value of text** Character string within 32 characters enclosed by double quotation marks or single quotation marks
- **Example** KSE\_"MS8604A"  
KSE\_'TX\_TESTER'

# MC8104

## MC8104

## MC8104A Address

■ Function Set the GPIB address of the MC8104A.

Header	Program command	Query	Response
MC8104	MC8104_a a=0 to 30	MC8104?	a a=0 to 30

- Value of a 0 to 30
- Suffix code None
- Initial setting Value of a = 19 (provided the address already allocated is not initialized)
- Example MC8104\_20

---

# PLS

## PLS

## Direct Plot Start

■ Function Starts direct plotting.

Header	Program command	Query	Response
PLS	PLS_a a=0	_____	_____

- Value of a 0: PLOT START
- Suffix code None
- Example PLS\_0

# PMCS

## PMCS          PMC Select

- **Function**          Selects one from the PMC built in the MS8604A, external PMC in the MC8104A, or external floppy disk in the MC8104A.

Header	Program command	Query	Response
PMCS	PMCS_a	PMCS?	a a=IPMC,EPMC1 EPMC2,EFD

- **Value of a**          IPMC:    PMC built in this instrument  
                          EPMC1: External PMC1  
                          EPMC2: External PMC2  
                          EFD:     External FD
- **Suffix code**        None
- **Initial setting**    IPMC: PMC built in this instrument
- **Example**            PMCS\_IPMC  
                          PMCS\_EFD

---

# PMOD

## PMOD          Printer Type

- **Function**          Selects the type of printer for direct plotting.

Header	Program command	Query	Response
PMOD	PMOD_a a=2 to 5	PMOD?	PMOD_a

- **Value of a**          2: Printer ..... ESC/P (8 dots)  
                          3: Printer ..... HP2225 (HP)  
                          4: Printer ..... MC8104A (Anritsu)  
                          5: Video Plotter ..... UA455A  
                          6: Printer ..... ESC/P (24 dots)
- **Suffix code**        None
- **Initial setting**    3: Printer ... HP2225
- **Example**            PMOD\_2  
                          PMOD\_4

# PNLMD

## PNLMD Panel Mode

■ **Function** Sets the operation mode of the MS8604A.

Header	Program command	Query	Response
PNLMD	PNLMD_a	PNLMD?	a

■ **Value of a** TESTER: TX Test mode  
SPECT: Spectrum mode  
SYSTEM: System mode

■ **Suffix code** None

■ **Initial setting** TESTER

■ **Example** PNLMD\_SPECT

---

# PRE

## PRE Preset

■ **Function** Initializes all measurement control parameters which need to be initialized (same function as INI and IP).

Header	Program command	Query	Response
PRE	PRE	_____	_____

■ **Example** PRE

**PRIA Direct Plot Printer Address**

■ **Function** Sets the GPIB address of the printer for direct plotting.

Header	Program command	Query	Response
PRIA	PRIA_a a=0 to 30	PRIA?	a a=0 to 30

- Value of a 0 to 30
- Suffix code None
- Initial setting Value of a = 17 (provided the address already allocated is not initialized)
- Example PRIA\_17

**PRINT**

**PRINT Direct Plot Start**

■ **Function** Starts direct plotting.

Header	Program command	Query	Response
PRINT	PRINT	_____	_____

■ **Example** PRINT

# PRTY

**PRTY**                      **Parity** (valid only when option 02 RS-232C is used)

■ **Function**                      Sets the parity of the RS-232C.

Header	Program command	Query	Response
PRTY	PRTY_a	PRTY?	a

■ **Value of a**                      **EVEN:** Even number  
   **ODD:** Odd number  
   **OFF:** None

■ **Suffix code**                      None

■ **Initial setting**                      EVEN

■ **Example**                              PRTY\_ODD

---

# RC

**RC**                              **Recall Data from Internal Memory**

■ **Function**                              Recalls trace data/parameter data from the built-in memory (same function as RGRC).

Header	Program command	Query	Response
RC	RC_a a=1 to 4	_____	_____

■ **Value of a**                              1 to 4 (memory No.)

■ **Suffix code**                              None

■ **Example**                                      RC\_1

## RCM Recall Data from PMC

- **Function** Recalls measurement conditions (parameter) and trace (spectrum mode only) from PMC or floppy disk.

Header	Program command	Query	Response
RCM	RCM_a a = 1 to 99 101 to 199	_____	_____

- **Value of a** 1 to 99 (File No. of spectrum mode)  
101 to 199 (File No. of TX-Test mode)
- **Suffix code** None
- **Example** RCM\_17  
RCM\_2

---

# RDATA

## RDATA Recalled Data

- **Function** Specifies the contents of data recalled.

Header	Program command	Query	Response
RDATA	RDATA_a	RDATA?	a a = TP, P

- **Value of a** TP: Trace & Parameter (spectrum mode only)  
P: Parameter Only
- **Suffix code** None
- **Initial setting** TP: Trace & Parameter (provided the address already allocated is not initialized)
- **Example** RDATA\_TP

# RFINPUT

## RFINPUT

### RF Input

■ Function Sets the RF input connector.

Header	Program command	Query	Response
RFINPUT	RFINPUT_a	RFINPUT?	a

■ Value of a HIGH: Input HIGH Power  
LOW: Input LOW Power

■ Example RFINPUT\_HIGH

---

# RGDIR

## RGDIR

### Memory Directory

■ Function Displays the directory of recall memory.

Header	Program command	Query	Response
RGDIR	RGDIR	_____	_____

■ Example RGDIR



## RGRC Recall Data from Internal Memory

■ **Function** Recalls trace data/parameter data from built-in memory (same function as RC).

Header	Program command	Query	Response
RGRC	RGRC_ a a=1 to 4	_____	_____

■ **Value of a** 1 to 4 (Memory No.)

■ **Suffix code** None

■ **Example** RGRC\_1

## RGSV Save Data into Internal Memory

■ **Function** Saves trace data (spectrum mode only)/parameter data to built-in memory (same function as SV).

Header	Program command	Query	Response
RGSV	RGSV_ a a=1 to 4	_____	_____

■ **Value of a** 1 to 4 (Memory No.)

■ **Suffix code** None

■ **Example** RGSV\_1

# SNGLS

## SNGLS Single Sweep Mode

■ Function Sets the sweep to single (single sweep) mode (same function as S2).

Header	Program command	Query	Response
SNGLS	SNGLS	_____	_____

■ Example SNGLS

---

# STPB

## STPB Stop Bit (valid only when option 02 RS-232C is used)

■ Function Sets the stop bit of the RS-232C.

Header	Program command	Query	Response
STPB	STPB_ a	STPB?	a

■ Value of a  
1: 1 bit  
1.5: 1.5 bit  
2: 2 bit

■ Suffix code None

■ Initial setting 1

■ Example STPB\_2

## SV Save Data into Internal Memory

- **Function** Saves trace data/parameter data to built-in memory (same function as RGSV).

Header	Program command	Query	Response
SV	SV_a a=1 to 4	_____	_____

- **Value of a** 1 to 4 (Memory No.)
- **Suffix code** None
- **Example** SV\_1

---

## SVM

### SVM Save Data into PMC

- **Function** Saves the measurement conditions (parameter) and trace (spectrum mode only) to PMC or floppy disk.

Header	Program command	Query	Response
SVM	SVM_s s=1 to 99 101 to 199	_____	_____

- **Value of s** 1 to 99 (File No. of spectrum mode)  
101 to 199 (File No. of TX test mode)
- **Suffix code** None
- **Example** SVM\_17  
SVM\_2

# SWP

## SWP Single Sweep/Sweep Status

- **Function** Executes single sweep/Responds to sweep status (sweep completed/sweep in progress) When accepted by the MS8604A device, the SWP command causes a single sweep to be executed by setting the sweep mode to ' SINGLE '. The next command waits without being processed until its single sweep is completed (same function as TS). The SWP? Query command is used to Query the current sweep status (sweep completed/sweep in progress).

Header	Program command	Query	Response
SWP	SWP	SWP?	SWP_a a=0,1

- **Value of a** 0: Sweep completed  
1: Sweep in progress

- **Example** SWP  
SWP?

---

# S1

## S1 Sweep Mode (Continuous)

- **Function** Sets the sweep mode to CONTINUOUS (same function as CONTS).

Header	Program command	Query	Response
S1	S1	_____	_____

- **Example** S1

**S2 Sweep Mode (Single)**

- **Function** Sets the sweep mode to SINGLE (same function as SNGLS).

Header	Program command	Query	Response
S2	S2	_____	_____

- **Example** S2

---

**TIME****TIME Time**

- **Function** Sets the clock built in MS8604A instrument to the specified time.

Header	Program command	Query	Response
TIME	TIME_ hh , mm , ss	TIME?	hh , mm , ss

- **Value of hh** 00 to 23 (hour)
- **Value of mm** 00 to 59 (minute)
- **Value of ss** 00 to 59 (second)
- **Suffix code** None
- **Example** TIME\_08 , 30 , 00

# TITLE

## TITLE Title Entry

■ Function Registers the title character string (same function as KSE).

Header	Program command	Query	Response
TITLE	TITLE_ <u>text</u>	TITLE?	text

■ text Character string within 32 characters enclosed by double quotation marks or single quotation marks

■ Example  
TITLE\_ "MS8604A"  
TITLE\_ 'ANRITSU'  
TITLE\_ " " (Title is not displayed.)

---

# TOUT

## TOUT Time Out (valid only when option 02 RS-232C is valid)

■ Function Sets the time-out time.

Header	Program command	Query	Response
TOUT	TOUT_ <u>a</u>	TOUT?	a

■ Value of a 1 to 255 (seconds, 1-second steps)

■ Suffix code None

■ Initial setting 5

■ Example TOUT\_1

## TRM Terminator

- **Function** Sets the terminator of the Response data transferred on the GPIB.

Header	Program command	Query	Response
TRM	TRM_a a=0,1	_____	_____

- **Value of a** 0: LF  
1: CR/LF
- **Suffix code** None
- **Initial setting** 0: LF (Provided the terminator already registered is not initialized)
- **Example** TRM\_0  
TRM\_1

## TTL Title Display Switch

- **Function** Sets the title display mode to ON or OFF.

Header	Program command	Query	Response
TTL	TTL_sw	TTL?	TTL_sw

- **Value of sw** DATE: Date/Time  
USER: User Define  
ON: User Define  
OFF: OFF
- **Suffix code** None
- **Initial setting** OFF: DATE
- **Example** TTL\_ON

# XMA

## XMA Trace-FREQ Data

- Function Writes/reads the spectrum trace data.

Header	Program command	Query	Response
XMA	XMA <p>, b</p>	XMA <p>, d</p>	b <sub>1</sub> , b <sub>2</sub> , b <sub>3</sub> ..... (When ASCII is specified) b <sub>1</sub> b <sub>2</sub> b <sub>3</sub> ..... (When BINARY is specified)

- Value of p 0 to 1001 (point No. position)
- Value of b LOG scale used:  
Integers having 0.01 dBm unit (independent of display unit system)  
  
LIN scale used:  $b = \frac{\text{Voltage value (V)}}{\text{Reference level (V)}} \times 10000$
- Value of d 1 to 1002 (number of points)
- Initial setting XMA

1, -2000  
XMA?

1, 2 (Reads out two data from point 1)

Note: When the binary format is specified for Response data, each point data is composed of two bytes, and each byte is sent out in sequence from the high-order byte to the low-order byte.

---

# XMB

## XMB OCC. BW/Adj. Ch. Data

- Function Writes/reads the waveform data of the occupied frequency bandwidth (STANDARD) and adjacent channel leakage power (STANDARD).

Header	Program command	Query	Response
XMB	XMB <p>, b</p>	XMB? <p>, d</p>	b <sub>1</sub> , b <sub>2</sub> , b <sub>3</sub> ..... (When ASCII is specified) b <sub>1</sub> b <sub>2</sub> b <sub>3</sub> ..... (When BINARY is specified)

- Value of p 0 to 1001 (point No. position)
- Value of b Integers having 0.01 dBm unit (independent of display unit system)
- Value of d 1 to 1002 (number of points)
- Example XMB

1, -2000  
XMB? 

1, 2 (Reads out two data from point 1)

Note: When the binary format is specified for Response data, each point data is composed of two bytes, and each byte is sent out in sequence from the high-order byte to the low-order byte.



## XMC Constellation Data

- **Function** Writes/reads the constellation display data.

Header	Program command	Query	Response
XMC	XMC <p><math>\varnothing</math>, p1, b</p>	XMC <p><math>\varnothing</math>, p1, d</p>	b <sub>1</sub> , b <sub>2</sub> , b <sub>3</sub> ..... (When ASCII is specified) b <sub>1</sub> b <sub>2</sub> b <sub>3</sub> ..... (When BINARY is specified)

- **Value of p $\varnothing$**   $\varnothing$ : I Data  
1: Q Data
- **Value of p1** 0 to 648
- **Value of b** Integers having 0.0001 dBm unit  
-30000 to 30000
- **Value of d** 1 to 649 (number of points)
- **Example** XMC

$\varnothing$ , 1, 2, -20000  
XMC?

$\varnothing$ , 1, 2 (Reads out two data from point 1 of 1 data)

*Note:* When the binary format is specified for Response data, each point data is composed of two bytes, and each byte is sent out in sequence from the high-order byte to the low-order byte.

## XMD

### XMD RF Power Data

- **Function** Writes/reads the power measurement data.

Header	Program command	Query	Response
XMD	XMD <p><math>\varnothing</math>, b</p>	XMD? <p><math>\varnothing</math>, d</p>	b <sub>1</sub> , b <sub>2</sub> , b <sub>3</sub> ..... (When ASCII is specified) b <sub>1</sub> b <sub>2</sub> b <sub>3</sub> ..... (When BINARY is specified)

- **Value of p** 0 to 4560
- **Value of b** Integers having 0.01 dBm unit (independent of display unit system)
- **Value of d** 1 to 4561 (number of points)
- **Example** XMD

$\varnothing$ , -20000  
XMD?

$\varnothing$ , 1, 2 (Reads out two data from point 1)

*Note:* When the binary format is specified for Response data, each point data is composed of two bytes, and each byte is sent out in sequence from the high-order byte to the low-order byte.

# XME

## XME Occ. BW. Data

■ Function Writes/reads the waveform data of the occupied frequency bandwidth (HIGH SPEED).

Header	Program command	Query	Response
XME	XME <p>␣</p> p, b	XME <p>␣</p> p, d	b <sub>1</sub> , b <sub>2</sub> , b <sub>3</sub> ..... (When ASCII is specified) b <sub>1</sub> b <sub>2</sub> b <sub>3</sub> ..... (When BINARY is specified)

- Value of p 0 to 250 (point No. position)
- Value of b Integers having 0.01 dBm unit ( independent of display unit system )
- Value of d 1 to 251 (number of points)
- Example XME

␣

1, -2000  
XME?

␣

1, 2 (Reads out two data from point 1)

Note: When the binary format is specified for Response data, each point data is composed of two bytes, and each byte is sent out in sequence from the high-order byte to the low-order byte.

# XMT

## XMT Trace-TIME Data

■ Function Writes/reads the spectrum data to/from trace TIME memory.

Header	Program command	Query	Response
XMT	XMT <p>␣</p> p, b	XMT? <p>␣</p> p, d	b <sub>1</sub> , b <sub>2</sub> , b <sub>3</sub> ..... (When ASCII is specified) b <sub>1</sub> b <sub>2</sub> b <sub>3</sub> ..... (When BINARY is specified)

- Value of p 0 to 1001 (point No. position)
- Value of b LOG scale used:  
Integers having 0.01 dBm unit (independent of display unit system)
- LIN scale used:  $b = \frac{\text{Voltage value (V)}}{\text{Reference level (V)}} \times 10000$
- Value of d 1 to 1002 (number of points)
- Example XMT

␣

1, -2000  
XMT?

␣

1, 2 (Reads out two data from point 1)

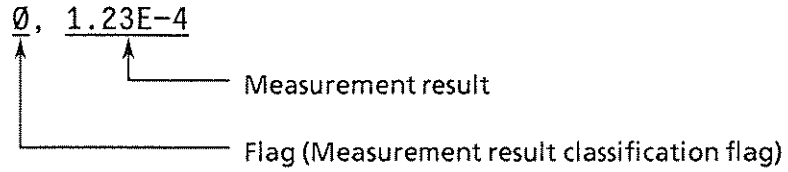
Note: When the binary format is specified for Response data, each point data is composed of two bytes, and each byte is sent out in sequence from the high-order byte to the low-order byte.

**(Blank)**

### 6.3 TX Test Commands

- Query command output format

Measurement result responses other than the waveform memory are output in the following format:



Flags have the following values depending on the result of the measurement:

- Ø: Normal termination
- 1: Level over
- 2: Level under
- 8: No measurement result
- 9: Measurement abnormal termination

# ADJCH?

## ADJCH? Adjacent Channel Power

- **Function** Outputs a measurement result on the adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
ADJCH	_____	ADJCH?_a	c, d (Depends on display units.)
	_____	ADJCH?_a, b	c, d

- **Value of a**

LOW3: Lower-3 (PDC: Invalid, PHS: Invalid, NADC: -90 kHz)  
 LOW2: Lower-2 (PDC: -100 kHz, PHS: -900 kHz, NADC: -60 kHz)  
 LOW1: Lower-1 (PDC: -50 kHz, PHS: -600 kHz, NADC: -30 kHz)  
 UP1 : Upper-1 (PDC: 50 kHz, PHS: 600 kHz, NADC: 30 kHz)  
 UP2 : Upper-2 (PDC: 100 kHz, PHS: 900 kHz, NADC: 60 kHz)  
 UP3 : Upper-3 (PDC: Invalid, PHS: Invalid, NADC: 90 kHz)
- **Value of b**

DBM: dBm  
 WATT: mW,  $\mu$ W, nW  
 DB: dB
- **Value of c** Measurement result classification flag
- **Value of d**

(dBm or dB value) -90 to 40.42  
 (output format: 0, \*\*\*\*\*)  
 (Watt value) 0.001 nW to 1000 mW  
 (output format: 0, \*.\*E+\*\*)
- **Suffix code** None
- **Example** ADJCH?\_UP2, WATT

# ADJMEAS

## ADJMEAS Adjacent Channel Meas

- **Function** Sets the Meas function on the adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
ADJMEAS	ADJMEAS_a	ADJMEAS?	a

- **Value of a** OFF, ON
- **Example** ADJMEAS\_ON

# AUNITS

## AUNITS Unit for Log Scale

- **Function** Sets one of the display unit systems when the LOG scale is selected.

Header	Program command	Query	Response
AUNITS	AUNITS_a	AUNITS?	a
	_____	AUNITS?	b

- **Value of a**
  - DBM: dBm  
Used for the power measurement, adjacent channel leakage power measurement or spurious measurement screen.
  - WATT: nW/ $\mu$ W/mW/W  
Used for the power measurement screen.
- **Value of b**
  - MW: mW  
Used for the adjacent channel leakage power measurement or spurious measurement screen.
  - UW:  $\mu$ W  
Used for the adjacent channel leakage power measurement or spurious measurement screen.
  - NW: nW  
Used for the adjacent channel leakage power measurement or spurious measurement screen.
  - DB: dB  
Used for the adjacent channel leakage power measurement or spurious measurement screen.
- **Suffix code** None
- **Initial setting** DBM: dBm

# AUTORNG

## AUTORNG Auto Range

- **Function** Executes Auto Range of Calibration on modulation analysis, power measurement, occupied bandwidth measurement, adjacent channel leakage power measurement, or spurious measurement screen.

Header	Program command	Query	Response
AUTORNG	AUTORNG	AUTORNG?	a

- **Value of a** Before execution of Auto range: RF level value on the parameter setting screen  
After execution of Auto range: RF level value obtained by the measuring instrument
- **Suffix code** None
- **Initial setting** None
- **Example** AUTORNG

# AVR

## AVR Average Number

- **Function** Sets the mean processing count on the power measurement, occupied bandwidth measurement, or adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
AVR	AVR_ a	AVR?	a
	_____	AVR?_ RFPWR (Use this query when reading Average No. of the power measurement screen from a screen other than the power measurement screen.)	a
	_____	AVR?_ OBW (Use this query to read Average No. of the occupied bandwidth measurement screen from a screen other than the occupied bandwidth measurement screen.)	a
	_____	AVR?_ ADJ (Use this query to read Average No. of the adjacent channel leakage power measurement screen from a screen other than the adjacent channel leakage power measurement screen.)	a

- **Value of a** 2 to 99 (integer steps)
- **Suffix code** None
- **Initial setting** 4
- **Example** AVR\_ 32

# BITR?

## BITR?

### Bit Rate

- **Function** Outputs the bit rate of the measurement result on the modulation analysis screen.

Header	Program command	Query	Response
BITR	_____	BITR?	a , b

- **Value of a** Measurement result classification flag
- **Value of b** 0 to 9 999.999 999 9 kbps (0.000 000 1 steps)
- **Suffix code** None
- **Example** BITR?

---

# BITRERR?

## BITRERR?

### Bit Rate Error

- **Function** Outputs the bit rate error of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
BITRERR	_____	BITRERR?	a , b

- **Value of a** Measurement result classification flag
- **Value of b** - 500.0 to 500.0 ppm (0.1 steps)
- **Suffix code** None
- **Example** BITRERR?



## BPATT

### Bit Pattern

■ **Function**

Sets the USER bit pattern of SYNC (UNIQUE) WORD on the parameter setting screen.

Header	Program command	Query	Response
BPATT	BPATT_a	BPATT?	a

■ **Value of a**

0--0 to 1--1

■ **Suffix code**

None

■ **Initial setting**

----

■ **Example**

BPATT\_10010011

## BS

## BS

### Back Screen

■ **Function**

Returns to the upper level of the screen.

Header	Program command	Query	Response
BS	BS	_____	_____

■ **Example**

BS

# BSTAVG

## BSTAVG 10 Burst Average

■ Function Sets 10 Burst Average on the modulation analysis screen.

Header	Program command	Query	Response
BSTAVG	BSTAVG_a	BSTAVG?	a

- Value of a ON: On  
OFF: Off
- Suffix code None
- Initial setting OFF
- Example BSTAVG\_ON

---

# CALADJ

## CALADJ Cal. Adjust

■ Function Executes Cal.Adjust on the parameter screen.

Header	Program command	Query	Response
CALADJ	CALADJ	_____	_____

- Suffix code None
- Initial setting None
- Example CALADJ

**CALOSC**      **Cal. Oscillator**

■ **Function**      Sets Cal.Oscillator on the parameter screen.

Header	Program command	Query	Response
CALOSC	CALOSC┐ a	CALOSC?	a

- **Value of a**      ON: On  
                         OFF: Off
- **Suffix code**      None
- **Initial setting** OFF
- **Example**          CALOSC┐ ON

**CARRF?**

**CARRF?**      **Carrier Frequency**

■ **Function**      Outputs the carrier frequency of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
CARRF	_____	CARRF?	a, f f= 100 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of a**      Measurement result classification flag
- **Value of f**      100 Hz to 8.5 GHz
- **Suffix code**      None
- **Initial setting** None
- **Example**          CARRF?

# CARRFERR?

## CARRFERR? Carrier Frequency Error

- **Function** Outputs the carrier frequency error of a measurement result on the modulation analysis, or constellation screen.

Header	Program command	Query	Response
CARRFERR	_____	CARRFERR?	a, f f= -999999.9 to 0 to 999999.9 Transfers the data without suffix code as if represented in units of Hz

- **Value of a** Measurement result classification flag
- **Value of f** -999.999 9 kHz to 999.999 9 kHz (0.1 Hz steps)
- **Suffix code** None
- **Example** CARRFERR?

---

# CDJ

## CDJ Cal. Adjust

- **Function** Executes Cal.Adjust on the power meter screen (same function as CALADJ).

Header	Program command	Query	Response
CDJ	CDJ	_____	_____

- **Suffix code** None
- **Initial setting** None
- **Example** CDJ

**CF Center Frequency**

■ **Function** Sets the FREQUENCY frequency on the parameter setting screen and the frequency on the power meter screen (same function as FREQ).

Header	Program command	Query	Response
CF	CF_f CF_a	CF?	f f=100 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

■ **Value of f** 100 Hz to 8.5 GHz

■ **Value of a** UP: CENTER FREQ STEP UP (Same function as FUP)  
DN: CENTER FREQ STEP DOWN (Same function as FDN)

■ **Suffix code** f: None: Hz  
HZ: Hz  
KHZ, KZ: kHz  
MHZ, MZ: MHz  
GHZ, GZ: GHz

a: None

■ **Initial setting** Value of f = 940.025 MHz (PDC), 1895.15 MHz (PHS), 825.03 MHz (NADC), 1 GHz (power meter screen)

■ **Example** CF\_1235456  
CF\_50MHz  
CF\_UP

**CHAN**

**CHAN Channel**

■ **Function** Sets the channel of FREQUENCY on the parameter setting screen.

Header	Program command	Query	Response
CHAN	CHAN_a CHAN_b	CHAN? _____	a _____

■ **Value of a** 0 to 9999

■ **Value of b** UP: step up  
DN: step down

■ **Suffix code** None

■ **Initial setting** 1001 (PDC, PHS), 1 (NADC)

■ **Example** CHAN\_1005

# CHCARR

## CHCARR Channels per Carrier

■ **Function** Sets the channels per carrier of SIGNAL on the parameter setting screen.

Header	Program command	Query	Response
CHCARR	CHCARR_a	CHCARR?	a

■ **Value of a** FULL: 3 (FULL RATE)  
HALF: 6 (HALF RATE)

■ **Suffix code** None

■ **Initial setting** FULL (PDC, NADC)

■ **Example** CHCARR\_HALF

---

# CHFREQ

## CHFREQ Ch & Freq

■ **Function** Sets the Ch & Freq of FREQUENCY on the parameter setting screen.

Header	Program command	Query	Response
CHFREQ	CHFREQ_a, f	_____	_____

■ **Value of a** 0 to 9999

■ **Value of f** 100 Hz to 8.5 GHz

■ **Suffix code** None: Hz  
HZ: Hz  
KHZ, KZ: kHz  
MHZ, MZ: MHz  
GHZ, GZ: GHz

■ **Initial setting** 1001 channels, 940.025 MHz (PDC), 1895.15 MHz (PHS)  
1 channel, 825.03 MHz (NADC)

■ **Example** CHFREQ\_1005, 800MHZ

# CHNUM

## CHNUM Ch & Freq

■ **Function** Sets the Ch & Freq of FREQUENCY on the parameter setting screen of NADC.

Header	Program command	Query	Response
CHNUM	CHNUM_a	_____	_____

■ **Value of a** 1 to 799, 990 to 1023

■ **Suffix code** None: Hz  
HZ: Hz  
KHZ, KZ: kHz  
MHZ, MZ: MHz  
GHZ, GZ: GHz

■ **Initial setting** 1 channel, 825.03 MHz (NADC)

■ **Example** CHNUM\_1005

---

# CHSPC

## CHSPC Channel Spacing

■ **Function** Sets the Channel Spacing of SIGNAL on the parameter setting screen.

Header	Program command	Query	Response
CHSPC	CHSPC_f	CHSPC?	f f= 1 to 1000000 Transfer the data without suffix code as if represented in units of Hz

■ **Value of f** 1 Hz to 1 MHz

■ **Suffix code** None: Hz  
HZ: Hz  
KHZ, KZ: kHz  
MHZ, MZ: MHz  
GHZ, GZ: GHz

■ **Initial setting** 25 kHz (PDC), 300 kHz (PHS), 30 kHz (NADC)

■ **Example** CHSPC\_12.5KHZ

# CNF

## CNF Center Frequency

- **Function** Sets the FREQUENCY frequency on the parameter setting screen and the frequency on the power meter screen.

Header	Program command	Query	Response
CNF	CNF_f	CNF?	CNF_f f = 100 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f** 100 Hz to 8.5 GHz
- **Suffix code** None: Hz  
HZ: Hz  
KHZ, KZ: kHz  
MHZ, MZ: MHz  
GHZ, GZ: GHz
- **Initial setting** 940.025 MHz (PDC), 1895.15 MHz (PHS), 825.03 MHz (NADC),  
1 GHz (power meter screen)
- **Example** CNF\_123456  
CNF\_50MHZ  
CNF?

---

# COS

## COS Cal. Oscillator Start

- **Function** Turns on Cal.Oscillator on the power meter screen (same function as CALOSC ON).

Header	Program command	Query	Response
COS	COS	_____	_____

- **Suffix code** None
- **Initial setting** None
- **Example** COS



## CRCAL?

### CRCAL? Reference Cal Factor

- **Function** Outputs the Reference Cal Factor on the power meter screen.

Header	Program command	Query	Response
CRCAL	_____	CRCAL?	a

- **Value of a** -2.00 dB to 2.00 dB
- **Example** CRCAL?

---

## CSCAL?

### CSCAL? Sensor Cal Factor

- **Function** Outputs the Sensor Cal factor on the power meter.

Header	Program command	Query	Response
CSCAL	_____	CSCAL?	a

- **Value of a** -2.00 dB to 2.00 dB
- **Example** CSCAL?

# CSNUM

## CSNUM CS Slot Number Active

■ Function Sets the CS Slot Number Active on the parameter setting screen.

Header	Program command	Query	Response
CSNUM	CSNUM_ a	CSNUM?	a

■ Value of a 1 to 4

■ Example CSNUM\_2

---

# CSSLLOT

## CSSLLOT CS Slot Number

■ Function Sets the CS Slot Number on the parameter setting screen.

Header	Program command	Query	Response
CSSLLOT	CSSLLOT_ a, b	CSSLLOT?_ a	b

■ Value of a 1 to 4

■ Value of b OFF, ON

■ Example CSSLLOT\_2, ON

## CST Cal. oscillator Stop

- **Function** Turns off Cal.Oscillator on the parameter screen (same function as CALOSC OFF).

Header	Program command	Query	Response
CST	CST	_____	_____

- **Suffix code** None
- **Initial setting** None
- **Example** CST

# DBR

## DBR dB Relative

- **Function** Sets the relative mode on the power meter screen (same function as SETREL).

Header	Program command	Query	Response
DBR	DBR	_____	_____

- **Suffix code** None
- **Initial setting** None
- **Example** DBR

# DRPFACT?

## DRPFACT? Droop Factor

- **Function** Outputs the droop factor of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
DRPFACT	_____	DRPFACT?	a, b

- **Value of a** Measurement result classification flag
- **Value of b** -3.2768 to 3.2767 dB/symbol (0.0001 steps)
- **Suffix code** None
- **Example** DRPFACT?

---

# ERRSC

## ERRSC Error Scale

- **Function** Sets the error scale of the constellation display screen.

Header	Program command	Query	Response
ERRSC	ERRSC_a	ERRSC?	a

- **Value of a** 5: 5 %  
10: 10 %  
20: 20 %  
OFF: Off
- **Suffix code** None
- **Initial setting** OFF
- **Example** ERRSC\_20

## FBAND Frequency Band

■ **Function** Sets the Frequency Band on the parameter setting screen.

Header	Program command	Query	Response
FBAND	FBAND_f	FBAND?	f

- **Value of f** 800, 1500
- **Suffix code** MHz, MZ: MHz
- **Example** FBAND\_800MHz

# FDN

## FDN Frequency Step Down

■ **Function** Decreases the FREQUENCY frequency on the parameter setting screen and frequency on the power meter screen by the steps set by the frequency step size. (same function as FREQ\_DN).

Header	Program command	Query	Response
FDN	FDN	_____	_____

■ **Example** FDN

# FREQ

## FREQ Frequency

- **Function** Sets the FREQUENCY frequency on the parameter setting screen and the frequency on the power meter screen.

Header	Program command	Query	Response
FREQ	FREQ_f  FREQ_a  _____	FREQ? (Use this query on the current screen.)  _____  FREQ?_PWRMTR (Use this query when reading the frequency set on the power meter screen from a screen other than the power meter screen.)	f  f= 100 to 850000000 Transfers the data without suffix code sd if represented in units of Hz f

- **Value of f** 100 Hz to 8.5 GHz
- **Value of a** UP: step up  
DN: step down
- **Suffix code** None: Hz  
HZ: Hz  
KHZ, KZ: kHz  
MHZ, MZ: MHz  
GHZ, GZ: GHz
- **Initial setting** 940.025 MHz (PDC), 1895.15 MHz (PHS), 825.03 MHz (NADC),  
1 GHz (power meter screen)
- **Example** FREQ\_800MHZ

# FREQADJ

## FREQADJ

### Frequency Adjust

■ **Function**

Executes Frequency Adjust of Calibration on the modulation analysis, power measurement, occupied bandwidth measurement, adjacent channel leakage power measurement, or spurious measurement screen.

Header	Program command	Query	Response
FREQADJ	FREQADJ	FREQADJ?	f f= 100 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

■ **Value of f**

Before execution of Frequency adjust: Frequency value on parameter screen  
After execution of Frequency adjust: Frequency value obtained by the measuring instrument

■ **Suffix code**

None

■ **Initial setting**

None

■ **Example**

FREQADJ

# FUP

## FUP

### Frequency Step Up

■ **Function**

Increases the center frequency by the frequency step size, if it has been set (same function as FREQ\_UP).

Header	Program command	Query	Response
FUP	FUP	_____	_____

■ **Example**

FUP

# FVECTERR?

## FVECTERR? First 10 Symbol RMS Vector Error

- **Function** Outputs the first 10 symbol RMS vector error of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
FVECTERR	_____	FVECTERR?	a , b

- **Value of a** Measurement result classification flag
- **Value of b** - 327.68 % to 327.67 % (rms) (0.01 % steps)
- **Suffix code** None
- **Example** FVECTERR?

# HMODPWR?

## HMODPWR? Modulation Power

- **Function** Outputs the measurement result Modulation Power on the power measurement screen.

HMODPWR	_____	HMODPWR?_a	b , c
---------	-------	------------	-------

- **Value of a** Omission: Units displayed on the screen  
 DBM: Units in dB (during REL display) or dBm (during ABS display)  
 WATT: Units in W
- **Value of b** Measurement result classification flag
- **Value of c** Outputs in units specified by value of a.
- **Example** HMODPWR?



# INTPOL

## INTPOL Interpolation

■ Function Sets interpolation of the constellation display screen.

Header	Program command	Query	Response
INTPOL	INTPOL┐a	INTPOL?	a

■ Value of a   NON:    Non Interpolation  
                  LIN:    Linear Interpolation  
                  POINT4: 4-Points Interpolation

■ Suffix code   None

■ Initial setting   NON

■ Example       INTPOL┐POINT4

---

# JITTER?

## JITTER? Jitter

■ Function Outputs the measurement result Jitter of the power measurement screen.

Header	Program command	Query	Response
JITTER	┐	JITTER?┐a	b, c

■ Value of a   +: Jitter (+)  
                  -: Jitter (-)

■ Value of b   Measurement result classification flag

■ Value of c   Outputs the measurement result by the bit units.

■ Example       JITTER?┐+

# KSA

## KSA

### Unit for Log Scale

■ Function

Sets the unit system of LOG scale on the power measurement screen, adjacent channel leakage power measurement screen, and spurious measurement screen to dBm (same function as UNT\_DBM).

Header	Program command	Query	Response
KSA	KSA	_____	_____

■ Example

KSA

---

# KSG

## KSG

### Average ON

■ Function

Allows the averaging to be used when set to ON.

Header	Program command	Query	Response
KSG	KSG	_____	_____

■ Example

KSG

## KSH Average OFF

- **Function** Turns off the averaging processing on the power measurement screen, occupied bandwidth measurement screen, and adjacent channel leakage power measurement screen to set the mode for processing waveform to NORMAL.

Header	Program command	Query	Response
KSH	KSH	_____	_____

- **Example** KSH

# LVLREL

## LVLREL Level Relative

- **Function** Selects absolute or relative power on the power measurement screen.

Header	Program command	Query	Response
LVLREL	LVLREL_a	LVLREL?	a

- **Value of a** OFF: Absolute power  
ON: Relative power
- **Suffix code** None
- **Initial setting** ON
- **Example** LVLREL\_OFF

# M1

## M1 Marker Mode

■ Function Turns OFF the Marker Mode (same function as MKR\_OFF).

Header	Program command	Query	Response
M1	M1	_____	_____

■ Example M1

---

# M2

## M2 Marker Mode

■ Function Sets the marker Mode to Normal (ON) mode (same function as MKR\_NRM).

Header	Program command	Query	Response
M2	M2	_____	_____

■ Example M2

# MAGERR?

## MAGERR? Magnitude Error

- **Function** Outputs the magnitude error of a measurement result on the modulation analysis screen (same function as MAGTDERR).

Header	Program command	Query	Response
MAGERR	_____	MAGERR?	a, b

- **Value of a** Measurement result classification flag
- **Value of b** - 327.68 % to 327.67 % (rms) (0.01 % steps)
- **Suffix code** None
- **Example** MAGERR?

---

# MAGTDERR?

## MAGTDERR? Magnitude Error

- **Function** Outputs the magnitude error of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
MAGTDERR	_____	MAGTDERR?	a, b

- **Value of a** Measurement result classification flag
- **Value of b** - 327.68 % to 327.67 % (rms) (0.01 % steps)
- **Suffix code** None
- **Initial setting** None
- **Example** MAGTDERR?

# MEANPWR?

## MEANPWR? Mean Power

- **Function** Outputs the mean power (PDC or PHS) of a measurement result on the adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
MEANPWR?	_____	MEANPWR?_a	c, d (Depends on display units.)
	_____	MEANPWR?_a, b	c, d

- **Value of a**
  - LOW2: Lower-2 (PDC: -100 kHz, PHS: -900 kHz)
  - LOW1: Lower-1 (PDC: -50 kHz, PHS: -600 kHz)
  - UP1: Upper-1 (PDC: 50 kHz, PHS: 600 kHz)
  - UP2: Upper-2 (PDC: 100 kHz, PHS: 900 kHz)

- **Value of b**
  - DBM: dBm
  - WATT: mW,  $\mu$ W, nW
  - DB: dB

- **Value of c** Measurement result classification flag

- **Value of d**
  - (dBm or dB value) -90 to 40.42  
(output format: 0,\*\*\*\*\*)
  - (Watt value) 0.001 nW to 1000 mW  
(output format: 0,\*\*\*E+\*\*)

- **Suffix code** None

- **Example** MEANPWR?\_UP2, WATT

## MEAS

## Measurement

- **Function** Sets the measurement screen.

Header	Program command	Query	Response
MEAS	MEAS_a	MEAS?	a

- **Value of a**
  - MODANAL: Modulation analysis screen
  - CONSTEL: Constellation screen
  - RFPWR: Power measurement screen
  - SETTEMP: Template setting screen
  - OBW, STD: Occupied frequency bandwidth measurement screen (STANDARD)
  - OBW, SPECT: Occupied frequency bandwidth measurement screen (SPECTRUM)
  - OBW, HIGH: Occupied frequency bandwidth measurement screen (HIGH SPEED)
  - ADJ, STD1: Adjacent channel leakage power measurement screen (STANDARD [ALL])
  - ADJ, STD2: Adjacent channel leakage power measurement screen (STANDARD [SEPARATE])
  - ADJ, SPECT1: Adjacent channel leakage power measurement screen (SPECTRUM [ALL])
  - ADJ, SPECT2: Adjacent channel leakage power measurement screen (SPECTRUM [SEPARATE])
  - ADJ, HIGH: Adjacent channel leakage power measurement screen (HIGH SPEED)
  - SPURIOUS, SPOT: Spurious measurement screen (SPOT measurement)
  - SPURIOUS, SEARCH: Spurious measurement screen (SEARCH measurement)
  - SETTABLE: Spurious frequency setting screen
  - SCALTBL: Sensor cal. factor setting screen
- **Example** MEAS\_RFPWR

# MEASOBJ

## MEASOBJ Measuring Object

■ **Function** Sets the measuring object of SIGNAL on the parameter setting screen.

Header	Program command	Query	Response
MEASOBJ	MEASOBJ_a	MEASOBJ?	a

- **Value of a**
  - MSCONT: MS-CONT
  - MSCOM: MS-COM
  - MSSYNC: MS-SYNC
  - BSCONT: BS-CONT
  - BSCOM: BS-COM
  - BSSYNC: BS-SYNC

} PDC only

  - PSCONT: PS-CONT
  - PSCOM: PS-COM
  - CSCONT: CS-CONT
  - CSCOM: CS-COM
  - CONT: CONTINUOUS

} PHS only

  - MOBILE: MOBILE
  - SHORT: SHORTENED BURST
  - BASE: BASE

} NADC only

■ **Suffix code** None

■ **Initial setting** MS-COM (PDC), PS-COM (PHS), MOBILE (NADC)

■ **Example** MEASOBJ\_BSCONT



## MKL? Marker Value

- **Function** Outputs the value of Marker on the modulation analysis screen and power measurement screen.

Header	Program command	Query	Response
MKL	_____	MKL?	a

- **Value of a**
  - Relative value (REL) display: Outputs the level of the marker value in dB.
  - Absolute value (ABS): Outputs the level of the marker value in dBm.  
    - \_\_\_\_\_ power measurement screen
  - Outputs the level of the marker value in dBm.  
    - \_\_\_\_\_ Adjacent channel leakage power measurement screen

- **Example** MKL?

## MKN Marker or Marker Position

- **Function** Sets the Marker on the power measurement screen and adjacent channel leakage power measurement screen to NORMAL (same function as MKR \_NRM) or sets the Marker position (same function as MKRS).

Header	Program command	Query	Response
MKN	MKN MKN_a MKN_b	_____ MKN?	_____ (Sets to NORMAL.) a (Sets the position.)

- **Value of a** Sets the marker position by the symbol or frequency.
- **Value of b**
  - UP: Step UP
  - DN: Step DOWN
- **Example**
  - MKN\_80.25
  - MKN\_UP

# MKOFF

## MKOFF      Marker Mode

■ Function      Turns off the marker mode.

Header	Program command	Query	Response
MKOFF	MKOFF_a	_____	_____

■ Value of a      ALL: Marker off  
                    None: Marker off

■ Suffix code      None

■ Example      MKOFF\_ALL  
                    MKOFF

---

# MKP

## MKP      Marker Position

■ Function      Sets the marker position on the power measurement screen and adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
MKP	MKP_a	MKP?	a

■ Value of a      0 to 360: power measurement screen  
                    0 to 250: adjacent channel leakage power measurement screen

■ Example      MKP\_200

## MKR Marker

■ **Function** Sets the marker mode of the power measurement screen.

Header	Program command	Query	Response
MKR	MKR_ a	MKR?	b

■ **Value of a** NRM: Normal  
 Ø: Normal  
 OFF: Off  
 2: Off

■ **Value of b** NRM: Normal  
 OFF: Off

■ **Suffix code** None

■ **Initial setting** OFF

■ **Example** MKR\_ NRM

## MKRS Marker Position

■ **Function** Sets the Marker position on the power measurement screen and adjacent channel leakage power measurement screen (same function as MKN).

Header	Program command	Query	Response
MKRS	MKRS_ a MKRS_ b	MKRS? _____	a (Sets the position.)

■ **Value of a** Sets the marker position by the symbol or frequency.

■ **Value of b** UP: Step UP  
 DN: Step DOWN

■ **Example** MKRS\_ UP

# MKZ

## MKZ Marker Position

- **Function** Sets the Marker position on the power measurement screen and adjacent channel leakage power measurement screen (same function as MKP).

Header	Program command	Query	Response
MKZ	MKZ┘a	MKZ?	MKZ┘a

- **Value of a** 0 to 360: power measurement screen  
0 ~ 250: adjacent channel leakage power measurement screen
- **Example** MKZ┘200

---

# MKZF

## MKZF Marker Position

- **Function** Sets the Marker position on the power measurement screen and adjacent channel leakage power measurement screen

Header	Program command	Query	Response
MKZF	MKZF┘a	MKZF?	a

- **Value of a** Sets the marker position by the symbol or frequency.
- **Example** MKZF┘80.25

# MODPWR?

## MODPWR? Mean Power due to Modulation

- **Function** Outputs the mean power due to modulation of a measurement result on the adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
MODPWR	_____	MODPWR?_a	c, d (Depends on display units.)
	_____	MODPWR?_a, b	c, d

- **Value of a**

LOW3: Lower-3 (PDC: Invalid, PHS: Invalid, NADC: -90 kHz)  
 LOW2: Lower-2 (PDC: -100 kHz, PHS: -900 kHz, NADC: -60 kHz)  
 LOW1: Lower-1 (PDC: -50 kHz, PHS: -600 kHz, NADC: -30 kHz)  
 UP1: Upper-1 (PDC: 50 kHz, PHS: 600 kHz, NADC: 30 kHz)  
 UP2: Upper-2 (PDC: 100 kHz, PHS: 900 kHz, NADC: 60 kHz)  
 UP3: Upper-3 (PDC: Invalid, PHS: Invalid, NADC: 90 kHz)
- **Value of b**

DBM: dBm  
 WATT: mW,  $\mu$ W, nW  
 DB: dB
- **Value of c** Measurement result classification flag
- **Value of d**

(dBm or dB value) -90 to 40.42  
 (output format: 0, \*\*\*\*\*)  
 (Watt value) 0.001 nW to 1000 mW  
 (output format: 0, \*.\*E+\*\*)
- **Suffix code** None
- **Initial setting** None
- **Example** MODPWR?\_UP2, WATT

# MSTAT?

## MSTAT? Measurement Status

■ Function Reads measurement status error information.

Header	Program command	Query	Response
MSTAT	_____	MSTAT?	a

■ Value of a

- Ø: Normal termination
- 1: Level over (LEVEL OVER)
- 2: Level under (LEVEL UNDER)
- 3: Measurement unavailable (SIGNAL ABNORMAL)
- 4: Synchronous word detection unavailable (SYNC WORD NOT FOUND)
- 5: Power sensor abnormal (POWER SENSOR ABNORMAL)
- 6: An RF input exceeded the hardware limit (INPUT LEVEL OVER).
- 7: A spurious measurement frequency is not entered (SET FREQUENCY TABLE).

■ Suffix code None

■ Example MSTAT?

---

# MTEMPREL

## MTEMPREL Magnitude Template Relative

■ Function Selects absolute or relative power on the power measurement screen.

Header	Program command	Query	Response
MTEMPREL	MTEMPREL_ a	MTEMPREL?	a

■ Value of a

- OFF: Absolute power
- ON: Relative power

■ Suffix code None

■ Initial setting ON

■ Example MTEMPREL\_OFF

# NUMSYM?

## NUMSYM? Number of symbols

- **Function** Outputs the number of analytical symbols on the parameter setting screen.

Header	Program command	Query	Response
NUMSYM?	_____	NUMSYM?	a

- **Value of a** Refer to "Panel Operation Part".
- **Suffix code** None
- **Example** NUMSYM?

---

# OBW?

## OBW? Occupied Bandwidth

- **Function** Outputs a measurement result on the occupied bandwidth measurement screen (same function as OCCBW?).

Header	Program command	Query	Response
OBW	_____	OBW?	a, f Transfers the data without suffix code as if represented in units of Hz

- **Value of a** Measurement result classification flag
- **Value of f** 0.0 to 99.9 kHz (PDC), 0.0 to 999.9 kHz (PHS) ; 0.1 kHz steps
- **Suffix code** None
- **Example** OBW?

# OBWFREQ?

## OBWFREQ? Occupied Bandwidth Center Frequency

■ **Function** Outputs a measurement result on the occupied bandwidth measurement screen.

Header	Program command	Query	Response
OBWFREQ?	_____	OBWFREQ? a	b, f Transfers the data without suffix code as if represented in units of Hz

- **Value of a**
  - CENTER: Center frequency
  - LOWER: Lower frequency
  - Minus(-): Lower frequency
  - UPPER: Upper frequency
  - Plus(+): Upper frequency
- **Value of b** Measurement result classification flag
- **Value of f** - 999.9 to 999.9 kHz (0.1 kHz steps)
- **Suffix code** None
- **Example**
  - OBWFREQ? \_CENTER
  - OBWFREQ? \_+

# OCCBW?

## OCCBW? Occupied Bandwidth

■ **Function** Outputs a measurement result on the occupied bandwidth measurement screen.

Header	Program command	Query	Response
OCCBW	_____	OCCBW?	a, f Transfers the data without suffix code as if represented in units of Hz

- **Value of a** Measurement result classification flag
- **Value of f** 0.0 to 99.9 kHz (PDC), 0.0 to 999.9 kHz (PHS) ; 0.1 kHz steps
- **Suffix code** None
- **Example** OCCBW?



**OFF**                      **Offset**

■ **Function**                      Sets User Cal.Factor on the power meter screen (same function as that of UCAL).

Header	Program command	Query	Response
OFF	OFF┐a	_____	_____

■ **Value of a**                      - 30.00 to 30.00 dB (0.01 steps)

■ **Suffix code**                      dB

■ **Initial setting**                      ∅

■ **Example**                      OFF┐3.∅

**OFFPWR?**

**OFFPWR?**                      **Carrier Off Power**

■ **Function**                      Outputs the carrier off power of a measurement result on the power measurement screen.

Header	Program command	Query	Response
OFFPWR	_____	OFFPWR?	a, b
	_____	OFFPWR?┐DBM	a, c
	_____	OFFPWR?┐WATT	a, d

■ **Value of a**                      Measurement result classification flag

■ **Value of b**                      Depends on the displayed units.

■ **Value of c**                      (dBm value) - 70.46 to 40.42  
(output format: 0, \*\*\*\*\*)

■ **Value of d**                      (Watt value) 0.1 nW to 10 W  
(output format: 0, \*.\*E+\*\*)

■ **Suffix code**                      None

■ **Example**                      OFFPWR?

# ORGNOFS?

## ORGNOFS? Origin Offset

- **Function** Outputs the origin offset of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
ORGNOFS	_____	ORGNOFS?	a, b

- **Value of a** Measurement result classification flag
- **Value of b** - 327.68 to 0 dB (0.01 steps)
- **Suffix code** None
- **Initial setting** None
- **Example** ORGNOFS?

# PATT

## PATT Pattern

- **Function** Sets a SYNC (UNIQUE) WORD pattern on the parameter setting screen.

Header	Program command	Query	Response
PATT	PATT_a	PATT?	a

- **Value of a**

S1 to S12:	S1 to S12	} PDC only
SS1 to SS6:	SS1 to SS6	
B32:	32 bit	} PHS only
B16:	16 bit	
SYNC1 to SYNC6:	Sync1 to Sync6	NADC only
NO:	NO	
USER:	USER	
- **Suffix code** None
- **Initial setting** S1 (PDC), 16 bit (PHS), Sync 1 (NADC)
- **Example** PATT\_USER

# PEAKPWR?

## PEAKPWR? Peak Power

- **Function** Outputs the peak power (PDC or PHS) or mean power due to switching transients (NADC) of a measurement result on the adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
PEAKPWR	_____	PEAKPWR?_a	c, d (Depends on display units.)
	_____	PEAKPWR?_a, b	c, d

- **Value of a**

LOW3: Lower-3 (PDC: Invalid, PHS: Invalid, NADC: -90 kHz)  
 LOW2: Lower-2 (PDC: -100 kHz, PHS: -900 kHz, NADC: -60 kHz)  
 LOW1: Lower-1 (PDC: -50 kHz, PHS: -600 kHz, NADC: -30 kHz)  
 UP1: Upper-1 (PDC: 50 kHz, PHS: 600 kHz, NADC: 30 kHz)  
 UP2: Upper-2 (PDC: 100 kHz, PHS: 900 kHz, NADC: 60 kHz)  
 UP3: Upper-3 (PDC: Invalid, PHS: Invalid, NADC: 90 kHz)

- **Value of b**

DBM: dBm  
 WATT: mW,  $\mu$ W, nW  
 DB: dB

- **Value of c** Measurement result classification flag

- **Value of d**

(dBm or dB value) -90 to 40.42  
 (output format: 0, \*\*\*\*\*)  
 (Watt value) 0.001 nW to 1000 mW  
 (output format: 0, \*.\*E+\*\*)

- **Suffix code** None

- **Example** PEAKPWR?\_UP2, WATT

## PHASEERR?

### PHASEERR? Phase Error

- **Function** Outputs the phase error of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
PHASEERR	_____	PHASEERR?	a, b

- **Value of a** Measurement result classification flag
- **Value of b** - 327.68 to 327.67 deg (rms) (0.01 steps)
- **Suffix code** None
- **Initial setting** None
- **Example** PHASEERR?

---

## POWER?

### POWER? Power

- **Function** Outputs a measurement result on the power meter screen.

Header	Program command	Query	Response
POWER	_____	POWER?_a	b, c

- **Value of a** DBM: Outputs a value in dBm.  
WATT: Outputs a value in Watt.  
DB: Outputs a value in dB.
- **Value of b** Measurement result classification flag
- **Value of c** (dBm or dB value) - 30.00 to 40.42  
(output format: 0, \*\*\*\*\*)  
(Watt value) 1  $\mu$ W to 10 W  
(output format: 0, \*.\*E+\*\*)
- **Suffix code** None
- **Example** POWER?\_DB

## PSSLOT

### PS Slot

■ **Function**

Sets the PS Slot on the parameter setting screen.

Header	Program command	Query	Response
PSSLOT	PSSLOT_a	PSSLOT?	a

■ **Value of a**

OFF, ON

■ **Example**

PSSLOT\_a

## PVECTERR?

### PVECTERR?

#### Peak Vector Error

■ **Function**

Outputs the peak vector error of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
PVECTERR	_____	PVECTERR?	a, b

■ **Value of a**

Measurement result classification flag

■ **Value of b**

- 327.68 % to 327.67 % (0.01 % steps)

■ **Suffix code**

None

■ **Example**

PVECTERR?

# RAMPSYM?

## RAMPSYM? Ramp up symbol point

- Function Outputs a ramp up symbol point value on the parameter setting screen.

Header	Program command	Query	Response
RAMPSYM?	_____	RAMPSYM?	a

- Value of a Refer to "Panel Operation Part".
- Suffix code None
- Example RAMPSYM?

---

# RATIO?

## RATIO? On/Off Ratio

- Function Outputs the on/off ratio of a measurement result on the power measurement screen.

Header	Program command	Query	Response
RATIO	_____	RATIO?	a, b

- Value of a Measurement result classification flag
- Value of b - 999.99 to 999.99 (0.1 steps)
- Suffix code None
- Example RATIO?

# RFATTN

## RFATTN RF Attenuater

- **Function** Sets the RF Atten on the spurious measurement screen.

Header	Program command	Query	Response
RFATTN	RFATTN┐a	RFATTN?	a

- **Value of a** NRM: Normal  
NOISE: Noise
- **Suffix code** None
- **Initial setting** NRM
- **Example** RFATTN┐NOISE

---

# RFLVL

## RFLVL RF Level

- **Function** Sets the RF level of INPUT on the parameter setting screen.

Header	Program command	Query	Response
RFLVL	RFLVL┐a RFLVL┐b	RFLVL? _____	a _____

- **Value of a** -32 to 40
- **Value of b** UP: step up  
DN: step down
- **Suffix code** None
- **Initial setting** 30
- **Example** RFLVL┐20

# RNG

## RNG Range

■ **Function** Sets a measurement range on the power meter screen.

Header	Program command	Query	Response
RNG	RNG_a RNGA (Automatic switching)	RNG?	a
	RNG2 (0.1 mW hold)	_____	_____
	RNG3 (1 mW hold)	_____	_____
	RNG4 (10 mW hold)	_____	_____
	RNG5 (100 mW hold)	_____	_____

■ **Value of a** AUTO: Automatically switches the value.  
HOLD: Holds the current status.

■ **Suffix code** None

■ **Initial setting** AUTO

■ **Example** RNG\_HOLD

# RTNYQ

## RTNYQ Root-Nyquist Filter

■ **Function** Sets the Root-Nyquist filter on the parameter setting screen.

Header	Program command	Query	Response
RTNYQ	RTNYQ_a	RTNYQ?	a

■ **Value of a** YES: Yes  
ON: Yes \_\_\_\_\_ Program commands only  
NO: No  
OFF: No \_\_\_\_\_ Program commands only

■ **Suffix code** None

■ **Initial setting** YES

■ **Example** RTNYQ\_NO



# SAVETBL

## SAVETBL Save Frequency Table

- **Function** Sets the memory number in which the frequency table on the spurious frequency creation screen is saved.

Header	Program command	Query	Response
SAVETBL	SAVETBL_a	_____	_____

- **Value of a** N01: Memory No.1  
N02: Memory No.2  
N03: Memory No.3
- **Suffix code** None
- **Initial setting** N01
- **Example** SAVETBL\_N03

---

# SAVETEMP

## SAVETEMP Save Template

- **Function** Saves the template on the template creation screen.

Header	Program command	Query	Response
SAVETEMP	SAVETEMP_a	_____	_____

- **Value of a** N01: Saves the template in internal memory No.1.  
N02: Saves the template in internal memory No.2.  
N03: Saves the template in internal memory No.3.
- **Suffix code** None
- **Initial setting** None
- **Example** SAVETEMP\_N03

# SCALT

## SCALT Sensor Cal Table

■ **Function** Sets the Sensor Cal Table on the sensor cal. factor setting screen.

Header	Program command	Query	Response
SCALT	SCALT_ a, b	SCALT?_ a	b

■ **Value of a** Sets the table frequency.

■ **Value of b** Sets the sensor cal. factor value.

■ **Suffix code** None: Hz  
Hz: Hz  
KHz, KZ: MHz  
GHz, GZ: GHz

■ **Example** SCALT\_ 50MHz, 0.01

---

# SCNCT

## SCNCT Sensor Connect To

■ **Function** Sets the Sensor Connect To on the power meter screen.

Header	Program command	Query	Response
SCNCT	SCNCT_ a	SCNCT?	a

■ **Value of a** TESTER: Connects the power sensor to the transmitter tester.  
DUT: Connects the power sensor to a device to be measured.

■ **Suffix code** None

■ **Initial setting** TESTER

■ **Example** SCNCT\_ DUT

## SCOFS Error Scale Offset

- **Function** Sets an error scale offset on the constellation screen.

Header	Program command	Query	Response
SCOFS	SCOFS_a	SCOFS?	a

- **Value of a**     $\emptyset$ : 0 degree  
                          22.5: 22.5 degrees
- **Suffix code**    None
- **Initial setting**  $\emptyset$
- **Example**        SCOFS\_22.5

# SETREL

## SETREL Set Relative

- **Function** Sets the Relative mode of the power meter screen.

Header	Program command	Query	Response
SETREL	SETREL	_____	_____

- **Suffix code**    None
- **Initial setting** None
- **Example**        SETREL

# SLCTTBL

## SLCTTBL Select Frequency Table

- Function Sets the frequency table on the spurious measurement screen.

Header	Program command	Query	Response
SLCTTBL	SLCTTBL_a	SLCTTBL?	a

- Value of a NO1: No.1 Frequency Table  
NO2: No.2 Frequency Table  
NO3: No.3 Frequency Table  
NOT: Not selected ——— Responses only
- Suffix code None
- Initial setting NO1
- Example SLCTTBL\_N03

---

# SLCTTEMP

## SLCTTEMP Select Template

- Function Sets the template of the power measurement screen.

Header	Program command	Query	Response
SLCTTEMP	SLCTTEMP_a	SLCTTEMP?	a

- Value of a NO1: No.1  
NO2: No.2  
NO3: No.3  
STD: Standard  
OFF: Off  
NOT: Not selected ——— Responses only
- Suffix code None
- Initial setting STD
- Example SLCTTEMP\_N03

# SLOTPWR?

## SLOTPWR? Slot Mean Power

- **Function** Outputs the Slot Mean Power on the power measurement screen.

Header	Program command	Query	Response
SLOTPWR	_____	SLOTPWR?_a	b, c (No CRT display)

- **Value of a** 1 to 8
- **Value of b** Measurement result classification flag
- **Value of c** Outputs the measurement result in dBm units.
- **Example** SLOTPWR?\_3

# SPUFREQ

## SPUFREQ Spurious Frequency

■ **Function** Sets the spurious frequency on the spurious frequency creation screen.

Header	Program command	Query	Response
SPUFREQ	SPUFREQ_ a, f  SPUFREQ_ b	SPUFREQ?_ a  _____	f f= 100 to 850000000 Transfers the data without suffix code as if represented in units of Hz  _____

■ **Value of a** F1 to F15

■ **Value of f** 100 Hz to 8.5 GHz  
Ø: Cancels a setting.

■ **Value of b** HRM: Sets Harmonics.

■ **Suffix code** None

■ **Initial setting** (Units: MHz)

	PDC	PHS	NADC
f1	1 880.050 000	3 790.300 000	1 650.060 000
f2	2 820.075 000	5 685.450 000	2 475.090 000
f3	3 760.100 000	7 580.600 000	3 300.120 000
f4	4 700.125 000	_____	4 125.150 000
f5			5 950.180 000
f6			6 775.210 000
f7			7 600.240 000
f8			8 425.270 000
f9			9 250.300 000

■ **Example** SPUFREQ\_ F4, 1.5GZ

SPUFREQ\_ F5, Ø

# SPUHRM

## SPUHRM Spurious Harmonics

- **Function** Sets a higher harmonic frequency on the spurious frequency creation screen (same function as SPUFREQHRM).

Header	Program command	Query	Response
SPUHRM	SPUHRM	_____	_____

- **Suffix code** None
- **Initial setting** None
- **Example** SPUHRM

# SPULVL?

## SPULVL? Spurious Level

- **Function** Outputs a measurement result on the spurious harmonics measurement screen.

Header	Program command	Query	Response
SPULVL	_____	SPULVL?_a	c, d (Depends on display units.)
	_____	SPULVL?_a, b	c, d

- **Value of a** F1 to F15
- **Value of b** DBM : dBm  
WATT: mW,  $\mu$ W, nW  
DB : dB
- **Value of c** Measurement result classification flag
- **Value of d** (dBm or dB value) -90 to 40.42  
(output format: 0, \*\*\*\*\*)  
(Watt value) 0.001 nW to 1000 mW  
(output format: 0, \*.\*E+\*\*)
- **Suffix code** None
- **Example** SPULVL?\_F3

# STARTPT

## STARTPT      Start Point

■ Function      Sets the start point of SYNC (UNIQUE) WORD on the parameter setting screen.

Header	Program command	Query	Response
STARTPT	STARTPT_a	STARTPT?	a

- Value of a      2 to 140
- Suffix code      None
- Initial setting      59 (PDC), 6 (PHS), 6 (NADC)
- Example      STARTPT\_34

---

# STD27

## STD27      RCR STD-27 Version

■ Function      Sets the PDC RCR STD-27 version.

Header	Program command	Query	Response
STD27	STD27_a	STD27?	a

- Value of a      B,C
- Example      STD27\_C



# STORAGE

## STORAGE

### Storage Mode

■ Function

Sets the storage mode of the power measurement, occupied bandwidth measurement, or adjacent channel leakage power measurement screen.

Header	Program command	Query	Response
STORAGE	STORAGE_ a	STORAGE?	a
	_____	STORAGE?_ RFPWR (Use this query when reading the storage mode of the power measurement screen from a screen other than the power measurement screen.)	a
	_____	STORAGE?_ OBW (Use this query when reading the storage mode of the occupied bandwidth measurement screen from a screen other than the occupied bandwidth measurement screen.)	a
	_____	STORAGE?_ ADJ (Use this query when reading the storage mode of the adjacent channel leakage power measurement screen from a screen other than the adjacent channel leakage power measurement screen.)	a

■ Value of a

- NRM: Normal
- MAX: Max Hold (RF POWER only)
- MIN: Min Hold (RF POWER only)
- AVG: Average
- CUM: Cumulative (RF POWER only)
- OVER: Over-write (RF POWER only)
- WIDE: Wide Dynamic Range (RF POWER only)

■ Suffix code

None

■ Initial setting

NRM

■ Example

STORAGE\_ MAX

# SWPWR?

## SWPWR?

### Mean Power due to Switching Transients

#### ■ Function

Outputs the mean power due to switching transients of a measurement result on the adjacent channel leakage power measurement screen of NADC.

Header	Program command	Query	Response
SWPWR	_____	SWPWR?_a	c, d (Depends on display units.)
	_____	SWPWR?_a, b	c, d

- Value of a
  - LOW3: Lower-3 (PDC: Invalid, PHS: Invalid, NADC: -90 kHz)
  - LOW2: Lower-2 (PDC: -100 kHz, PHS: -900 kHz, NADC: -60 kHz)
  - LOW1: Lower-1 (PDC: -50 kHz, PHS: -600 kHz, NADC: -30 kHz)
  - UP1: Upper-1 (PDC: 50 kHz, PHS: 600 kHz, NADC: 30 kHz)
  - UP2: Upper-2 (PDC: 100 kHz, PHS: 900 kHz, NADC: 60 kHz)
  - UP3: Upper-3 (PDC: Invalid, PHS: Invalid, NADC: 90 kHz)

- Value of b
  - DBM: dBm
  - WATT: mW,  $\mu$ W, nW
  - DB: dB

- Value of c Measurement result classification flag

- Value of d
  - (dBm or dB value) -90 to 40.42  
(output format: 0, \*\*\*\*\*)
  - (Watt value) 0.001 nW to 1000 mW  
(output format: 0, \*.\*E+\*\*)

- Suffix code None

- Initial setting None

- Example SWPWR?\_UP2, WATT

# SYMTIME

## SYMTIME Symbol Timing

■ Function Sets a symbol timing on the parameter setting screen.

Header	Program command	Query	Response
SYMTIME	SYMTIME_a	SYMTIME?	a

■ Value of a -0.20 to 0.20 symbol (0.01 symbol steps)

■ Suffix code None

■ Initial setting 0.00

■ Example SYMTIME\_0.02

---

# SYS

## SYS System

■ Function Sets a communication system.

Header	Program command	Query	Response
SYS	SYS_a	SYS?	a

■ Value of a PDC (JDC), PHS (JDCT), NADC, PWRMTR

■ Suffix code None

■ Example SYS\_PHS

# TEMPLVL

## TEMPLVL Template-Level Modify

■ **Function** Creates a template on the template creation screen.

Header	Program command	Query	Response
TEMPLVL	TEMPLVL_ a, b	TEMPLVL?_ a	b

■ **Value of a**  
 1: Limit 1  
 2: Limit 2  
 3: Limit 3  
 4: Limit 4  
 5: Limit 5

■ **Value of b** -110.0 to 10.0 dB (0.1 steps)

■ **Suffix code** None

■ **Initial setting** (Units: dB)

	PDC	PHS	NADC
1: Limit 1	-70	-56	-60
2: Limit 2	-60	-56	-60
3: Limit 3	4	4	3
4: Limit 4	-60	-56	-60
5: Limit 5	-14	-14	-20

■ **Example** TEMPLVL?\_3

# TERM

## TERM Terminal

- **Function** Sets the INPUT terminal on the parameter setting screen.

Header	Program command	Query	Response
TERM	TERM_a	TERM?	a

- **Value of a** RF: RF  
IQAC: I, Q - AC (option 03)  
IQDC: I, Q - DC (option 03)
- **Suffix code** None
- **Initial setting** RF
- **Example** TERM\_IQAC

---

# TIMING?

## TIMING? Timing

- **Function** Outputs the measurement result Timing on the power measurement screen.

Header	Program command	Query	Response
TIMING	_____	TIMING?	a, b

- **Value of a** Measurement result classification flag
- **Value of b** Outputs the measurement result in bit units.
- **Example** TIMING?

## TXPWR?

### TXPWR?

### TX Power

- **Function** Outputs the TX power of a measurement result on the power measurement screen.

Header	Program command	Query	Response
TXPWR	_____	TXPWR?	a, b
	_____	TXPWR?_DBM	a, c
	_____	TXPWR?_WATT	a, d

- **Value of a** Measurement result classification flag
- **Value of b** Depends on displayed units
- **Value of c** (dBm value) -70.46 to 40.42  
(output format: 0, \*\*\*\*\*)
- **Value of d** (Watt value) 0.1 nW to 10 W  
(output format: 0, \*.\*E+\*\*)
- **Suffix code** None
- **Example** TXPWR?

## TXTIME

### TXTIME

### Transmit Timing

- **Function** Sets the Transmit Timing on the parameter setting screen.

Header	Program command	Query	Response
TXTIME	TXTIME_a	TXTIME?	a

- **Value of a** OFF, ON
- **Example** TXTIME\_ON

## UCAL User Cal. Factor

■ **Function** Sets the User Cal.Factor of the power meter screen.

Header	Program command	Query	Response
UCAL	UCAL┘a	UCAL?	a

■ **Value of a** -30.00 to 30.00 dB (0.01 steps)

■ **Suffix code** dB

■ **Initial setting** 0

■ **Example** UCAL┘3.0

# UNIT

## UNIT

## Unit

### ■ Function

Sets the units of the power measurement, adjacent channel leakage power measurement, or spurious measurement screen.

Header	Program command	Query	Response
UNIT	UNIT_a	UNIT? (Use the current screen.)	a
	_____	UNIT?_RFPWR (Use this query when reading the units of the power measurement screen from a screen other than the power measurement screen.)	a
	_____	UNIT?_ADJ (Use this query when reading the units of the adjacent channel leakage power measurement screen from a screen other than the adjacent channel leakage power measurement screen.)	a
	_____	UNIT?_SPURIOUS (Use this query when reading the units of the spurious measurement screen from a screen other than the spurious measurement screen.)	a

### ■ Value of a

DBM: dBm

Use this value on the power measurement, adjacent channel leakage power measurement, or spurious measurement screen.

WATT: nW/ $\mu$ W/mW/W

Use this value on the power measurement screen.

MW: mW

Use this value on the adjacent channel leakage power measurement or spurious measurement screen.

UW:  $\mu$ W

Use this value on the adjacent channel leakage power measurement or spurious measurement screen.

NW: nW

Use this value on the adjacent channel leakage power measurement or spurious measurement screen.

DB: dB

Use this value on the adjacent channel leakage power measurement or spurious measurement screen.

### ■ Suffix code

None

### ■ Initial setting

DBM

### ■ Example

UNIT\_WATT



## UNT Unit for Log Scale

■ **Function** Sets the display unit systems in LOG scale mode.

Header	Program command	Query	Response
UNT	UNT_a _____ a=0	UNT? UNT?	UNT_a UNT_b a=0

■ **Value of a** ∅: dBm  
Used for the power measurement, adjacent channel leakage power measurement or suprious measurement screen.

■ **Value of b**

WATT: nW/μW/mW/W  
Used for the power measurement screen.

MW: mW  
Used for the adjacent channel leakage power measurement or suprious measurement screen.

UW: μW  
Used for the adjacent channel leakage power measurement or suprious measurement screen.

NW: nW  
Used for the adjacent channel leakage power measurement or suprious measurement screen.

DB: dB  
Used for the adjacent channel leakage power measurement or suprious measurement screen.

■ **Suffix code** None

■ **Initial setting** ∅: dBm

■ **Example** UNT\_∅

# VAVG

## VAVG Average

- **Function** Sets averaging processing of the power and occupied bandwidth and adjacent channel leakage power measurement screens.

Header	Program command	Query	Response
VAVG	VAVG sw VAVG a	VAVG?	a a = 2 to 99

- **Value of sw** ON: ON  
1: ON  
OFF: OFF  
Ø: OFF
- **Value of a** 2 to 99 (1step): Averaging rate to process
- **Suffix code** None
- **Initial setting** 4: 4 times
- **Example** VAVG ON  
VAVG 16

# VECTERR?

## VECTERR? RMS Vector Error

- **Function** Outputs the RMS vector error of a measurement result on the modulation analysis screen.

Header	Program command	Query	Response
VECTERR	—	VECTERR?	a, b

- **Value of a** Measurement result classification flag
- **Value of b** - 327.68 % to 327.67 % (rms) (0.01 % steps)
- **Suffix code** None
- **Example** VECTERR?

# WINDOW

## WINDOW

## Winwow

- **Function** Sets a window on the power measurement screen.

Header	Program command	Query	Response
WINDOW	WINDOW_a	WINDOW?	a

- **Value of a**
  - SLOT: Slot
  - FRAME: Frame
  - LEAD: Leading
  - RISE: Leading ——— Program commands only
  - TRAIL: Trailing
  - FALL: Trailing ——— Program commands only
- **Suffix code** None
- **Initial setting** SLOT
- **Example** WINDOW\_FRAME

# ZAJ

## ZAJ

## Zero Adjust

- **Function** Sets the power meter screen to zero (same function as ZEROSET).

Header	Program command	Query	Response
ZAJ	ZAJ	—————	—————

- **Suffix code** None
- **Initial setting** None
- **Example** ZAJ

# ZEROSET

## ZEROSET      Zero Set

■ **Function**      Executes the zero set on the power meter screen.

Header	Program command	Query	Response
ZEROSET	ZEROSET	_____	_____

■ **Suffix code**      None

■ **Initial setting**      None

■ **Example**      ZEROSET

**(Blank)**

## 6.4 Spectrum mode commands

**AAT RF Attenuator**

- **Function** Switches the RF attenuator setting mode to AUTO or MANUAL.

Header	Program command	Query	Response
AAT	AAT $\_a$ <span style="float: right;">a=0, 1</span>	AAT?	AAT $\_a$ <span style="float: right;">a=0, 1</span>

- **Value of a**  $\emptyset$ : MANUAL  
1: AUTO
- **Suffix code** None
- **Initial setting** 1: AUTO
- **Example** AAT $\_1$

**AMD Trace A Storage Mode**

- **Function** Selects the mode for processing trace A waveform.

Header	Program command	Query	Response
AMD	AMD $\_a$ <span style="float: right;">a=0 to 5</span>	AMD?	AMD $\_a$ <span style="float: right;">a=0 to 5</span>

- **Value of a**  $\emptyset$ : NORMAL  
1: MAX HOLD  
2: AVERAGE  
3: MIN HOLD  
4: CUMULATIVE  
5: OVER WRITE
- **Suffix code** None
- **Initial setting**  $\emptyset$ : NORMAL
- **Example** AMD $\_\emptyset$

# ARB

## ARB Resolution Band Width

■ Function Switches the mode for setting resolution bandwidth to AUTO or MANUAL.

Header	Program command	Query	Response
ARB	ARB_a a=0, 1	ARB?	ARB_a a=0, 1

■ Value of a Ø: MANUAL  
1: AUTO

■ Suffix code None

■ Initial setting 1: AUTO

■ Example ARB\_Ø  
ARB\_1

---

# AST

## AST Sweep Time

■ Function Switches the mode for setting frequency sweep time to AUTO or MANUAL.

Header	Program command	Query	Response
AST	AST_a a=0, 1	AST?	AST_a a=0, 1

■ Value of a Ø: MANUAL  
1: AUTO

■ Suffix code None

■ Initial setting 1: AUTO

■ Example AST\_Ø  
AST\_1



## ASWT      Auto Sweep Time

■ Function      Sets the auto sweep time.

Header	Program command	Query	Response
ASWT	ASWT_ <u>sw</u> sw=FAST, SLOW	ASWT?	SW      sw=FAST, SLOW

■ Value of sw      FAST: FAST  
                          SLOW: SLOW

■ Suffix code      None

■ Initial setting      SLOW (provided the address already allocated is not initialized)

■ Example      ASWT\_FAST  
                          ASWT\_SLOW

# AT

## AT      RF Attenuator

■ Function      Sets the RF attenuator.

Header	Program command	Query	Response
AT	AT_ <u>a</u>	AT?	a

■ Value of a      AUTO: AUTO  
                          UP: UP  
                          DN: DOWN  
                          0 to 75: 20 to 75 dB (5 dB step)

■ Suffix code      None: dB  
                          DB: dB

■ Initial setting      Calculated value when AUTO is selected for ATT

■ Example      AT\_10  
                          AT\_55

# ATT

## ATT RF Attenuator

- **Function** Sets the RF attenuator.

Header	Program command	Query	Response
ATT	ATT_a a=0 to 11	ATT?	ATT_a a=0 to 11

- **Value of a**

LOW/HIGH	LOW/HIGH
0: 0/20 dB	6: 5/25 dB
1: 10/30 dB	7: 15/35 dB
2: 20/40 dB	8: 25/45 dB
3: 30/50 dB	9: 35/55 dB
4: 40/60 dB	10: 45/65 dB
5: 50/70 dB	11: 55/75 dB

- **Suffix code** None
- **Initial setting** Calculated value when AUTO is selected for ATT
- **Example** ATT\_1

---

# ATUN

## ATUN Auto Tune

- **Function** Executes automatic tuning.

Header	Program command	Query	Response
ATUN	ATUN	_____	_____

- **Example** ATUN
- **Notes** Automatic tuning terminates when the parameters for the next sweep were set. After executing the ATUN command, wait for the sweep to terminate before executing the next command.

# AUNITS

## AUNITS Unit for Log Scale

- **Function** Sets one of the display unit systems when the LOG scale is selected.

Header	Program command	Query	Response
AUNITS	AUNITS_ a a=DBM, DBUV, DBMV, DBUVE, V	AUNITS?	a a=DBM, DBUV, DBMV, DBUVE, V

- **Value of a**
  - DBM: dBm
  - DBUV: dB $\mu$ V
  - DBMV: dBmV
  - DBUVE: dBmV (emf)
  - V: V
- **Suffix code** None
- **Initial setting** DBM: dBm (provided the address already allocated is not initialized)

---

# AUTO

## AUTO Coupled Function All Auto

- **Function** Executes all coupled functions (RBW, VBW, SWT, ATT) setting in AUTO mode.

Header	Program command	Query	Response
AUTO	AUTO	_____	_____

- **Example** AUTO

# AVB

## AVB Video Band Width

■ **Function** Switches the mode for setting the video bandwidth to AUTO or MANUAL.

Header	Program command	Query	Response
AVB	AVB_a a=0, 1, 2	AVB?	AVB_a a=0, 1, 2

■ **Value of a**    ∅: MANUAL  
                  1: AUTO  
                  2: OFF

■ **Suffix code**   None

■ **Initial setting** 1: AUTO

■ **Example**        AVB\_∅  
                  AVB\_1

---

# AVR

## AVR Number of Trace Average

■ **Function** Sets the averaging rate (number of sweep repetitions) to average the trace display.

Header	Program command	Query	Response
AVR	AVR_a a=0 to 4	AVR?	AVR_a a=0 to 4

■ **Value of a**    ∅: 4 times  
                  1: 8 times  
                  2: 16 times  
                  3: 32 times  
                  4: 128 times

■ **Suffix code**   None

■ **Initial setting** 1: 8 times

■ **Example**        AVR\_∅  
                  AVR\_3

**AWR Trace A Write Switch**

■ **Function** Controls writing the waveform data to trace A.

Header	Program command	Query	Response
AWR	AWR_ <u>a</u> a=ON, 1, OFF, 0	AWR?	AWR_ <u>a</u> a=ON, OFF

■ **Value of a**      ON: TRACE A WRITE ON (Same function as CLRW\_TRA)  
                           1: TRACE A WRITE ON (Same function as CLRW\_TRA)  
                           OFF: TRACE A WRITE OFF (Same function as VIEW\_TRA)  
                           ∅: TRACE A WRITE OFF (Same function as VIEW\_TRA)

■ **Suffix code**    None

■ **Initial setting** 1: TRACE A WRITE ON

■ **Example**        AWR\_∅

**A1**

**A1 Trace A Write ON**

■ **Function** Clears trace A (FREQ) waveform data to set the write mode to ON.  
 (same function as AWR\_1/CLRW\_TRA)

Header	Program command	Query	Response
A1	A1	_____	_____

■ **Example**        A1

# A2

## A2 Trace A Max Hold

- **Function** Allows trace A (FREQ) waveform to be processed in MAX HOLD mode (same function as AMD\_1).

Header	Program command	Query	Response
A2	A2	_____	_____

- **Example** A2

---

# BNDC

## BNDC Band Select

- **Function** Selects one of the bands in the frequency range of 0 to 8.5 GHz.

Header	Program command	Query	Response
BNDC	BNDC_a a=AUTO, 0, 1 <sup>-</sup> , 1 <sup>+</sup>	BNDC?	BNDC_a a=AUTO, 0, 1 <sup>-</sup> , 1 <sup>+</sup>

- **Value of a**
  - AUTO: BAND AUTO = 0 Hz to 8.5 GHz
  - Ø: BAND 0 = 0 Hz to 2 GHz
  - 1<sup>-</sup>: BAND 1<sup>-</sup> = 1.7 GHz to 7.5 GHz
  - 1<sup>+</sup>: BAND 1<sup>+</sup> = 6.5 GHz to 8.5 GHz

- **Suffix code** None

- **Initial setting** AUTO: BAND AUTO = 0 Hz to 8.5 GHz

- **Example**
  - BNDC\_AUTO
  - BNDC\_1<sup>+</sup>

# BSAUTO

## BSAUTO BW/SWT Auto

■ Function Allows RBW, VBW, and sweep time to be set in AUTO mode.

Header	Program command	Query	Response
BSAUTO	BSAUTO	_____	_____

■ Example BSAUTO

---

# CA

## CA RF Attenuator Auto

■ Function Sets the attenuator to AUTO mode (same function as AAT\_1, AT\_AUTO).

Header	Program command	Query	Response
CA	CA	_____	_____

■ Example CA

# CAL

## CAL Calibration

■ **Function** Selects the way of calibrating this instrument using internal CAL signal.

Header	Program command	Query	Response
CAL	CAL_a a = 0 to 2	_____	_____

■ **Value of a**    0: All  
                       1: Frequency  
                       2: Level

■ **Suffix code**    None

■ **Example**        CAL\_0

# CF

## CF Center Frequency

■ **Function** Sets the center frequency (same function as CNF).

Header	Program command	Query	Response
CF	CF_f CF_a	CF?	f f = -100000000 to 0 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

■ **Value of f**        - 100 MHz to 8.5 GHz

■ **Value of a**        UP: CENTER FREQ STEP UP (Same function as FUP)  
                       DN: CENTER FREQ STEP DOWN (Same function as FDN)

■ **Suffix code**     f:    None:    Hz (10<sup>0</sup>)  
                           HZ:    Hz (10<sup>0</sup>)  
                           KHZ, KZ: kHz (10<sup>3</sup>)  
                           MHZ, MZ: MHz (10<sup>6</sup>)  
                           GHZ, GZ: GHz (10<sup>9</sup>)

a: None

■ **Initial setting**    Value of f = 4.25 GHz

■ **Example**        CF\_1235456  
                       CF\_50MHz  
                       CF\_UP



# CLRW

## CLRW Clear & Write

- Function Clears the trace waveform data to set the write mode to ON.

Header	Program command	Query	Response
CLRW	CLRW <tr< td=""><td>_____</td><td>_____</td></tr<>	_____	_____

- Value of tr TRA: Trace A (same function as AWR\_1)  
TRTIME: Trace TIME (same function as TMWR\_1)
- Example CLRW\_TRA

---

# CMK?

## CMK? Current Marker Position

- Function Reads the current marker position.

Header	Program command	Query	Response
CMK?	_____	CMK?	CMK_a a=0 to 500

- Value of a 0 to 500
- Example CMK?

# CNF

## CNF Center Frequency

■ Function Sets the center frequency (same function as CF).

Header	Program command	Query	Response
CNF	CNF_ <u>f</u>	CNF?	CNF_ <u>f</u> f = -100000000 to 0 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

■ Value of f - 100 MHz to 8.5 GHz

■ Suffix code  
None: Hz (10<sup>0</sup>)  
HZ: Hz (10<sup>0</sup>)  
KHZ, KZ: kHz (10<sup>3</sup>)  
MHZ, MZ: MHz (10<sup>6</sup>)  
GHZ, GZ: GHz (10<sup>9</sup>)

■ Initial setting Value of f = 4.25 GHz

■ Example  
CNF\_123456  
CNF\_50MHZ  
CNF?

---

# COUPLE

## COUPLE Coupling Mode

■ Function Switches the coupling to AC or DC to monitor FM waveform.

Header	Program command	Query	Response
COUPLE	COUPLE_ <u>sw</u>	COUPLE?	SW sw = AC, DC

■ Value of sw AC: AC COUPLING  
DC: DC COUPLING

■ Suffix code None

■ Initial setting AC: AC COUPLING

■ Example  
COUPLE\_AC  
COUPLE\_DC

**CR Resolution Band Width Auto**

■ **Function** Sets the resolution bandwidth selection to the AUTO mode (same function as ARB\_1, RB\_AUTO).

Header	Program command	Query	Response
CR	CR	_____	_____

■ **Example** CR

**CT Sweep Time Auto**

■ **Function** Sets the frequency sweep time to AUTO mode (same function as AST\_1, ST\_AUTO).

Header	Program command	Query	Query
CT	CT	_____	_____

■ **Example** CT

# CV

## CV Video Band Width Auto

- Function Sets the video bandwidth to AUTO mode (same function as AVB\_1, VB\_AUTO).

Header	Program command	Query	Response
CV	CV	_____	_____

- Example CV

---

# DET

## DET Detection Mode

- Function Selects the detection mode of the waveform data being displayed.

Header	Program command	Query	Response
DET	DET_d d=0 to 2 POS, SMP, NEG	DET?	d d=POS, SMP, NEG

- Value of d
  - Ø: POSITIVE PEAK
  - 1: SAMPLE
  - 2: NEGATIVE PEAK
  - POS: POSITIVE PEAK
  - SMP: SAMPLE
  - NEG: NEGATIVE PEAK

- Suffix code None

- Initial setting Ø: POSITIVE PEAK

- Example DET\_Ø  
DET\_SMP

## DETM Detection Mode

- **Function** Selects the detection mode of the trace specified to A or B.

Header	Program command	Query	Response
DETM	DETM <tr,sw< td=""> <td>DETM?tr</td> <td>SW sw = POS, SMP, NEG</td> </tr,sw<>	DETM?tr	SW sw = POS, SMP, NEG

- **Value of tr** TRA: Trace A
- **Value of sw** POS: POSITIVE PEAK  
SMP: SAMPLE  
NEG: NEGATIVE PEAK
- **Suffix code** None
- **Initial setting** POS: POSITIVE PEAK
- **Example** DETM\_TRA, POS

# DFMT

## DFMT Display Format

- **Function** Specifies the format used in the display mode / display.

Header	Program command	Query	Response
DFMT	DFMTsw	DFMT?	SW sw = A, TIME

- **Value of sw** A: Trace A  
TIME: Trace TIME
- **Suffix code** None
- **Initial setting** A: Trace A
- **Example** DFMT\_TIME

# DPOINT

## DPOINT Data Point

■ Function Sets the data point

Header	Program command	Query	Response
DPOINT	DPOINT <sub>sw</sub> sw=NRM, DOUBLE	DPOINT?	SW sw=NRM, DOUBLE

- Value of sw NRM: NORMAL  
DOUBLE: DOUBLE
- Suffix code None
- Initial setting NRM: NORMAL (provided the address already allocated is not initialized)
- Example DPOINT<sub>NRM</sub>

---

# EXTTYPE

## EXTTYPE Ext Trigger Input Type

■ Function Chooses the level input from the external trigger when EXT is selected for trigger source.

Header	Header	Query	Response
EXTTYPE	EXTTYPE <sub>sw</sub>	EXTTYPE?	SW sw=10 V, TTL

- Value of sw 10V: INPUT 1 (±10V)  
TTL: INPUT 2 (TTL)
- Suffix code None
- Initial setting 10V: INPUT 1 (±10V)
- Example EXTTYPE<sub>10V</sub>  
EXTTYPE<sub>TTL</sub>

**E1 Peak Search**

■ **Function** Executes the function for peak search (same function as MKS\_0, MKMP).

Header	Program command	Query	Response
E1	E1	_____	_____

■ **Example** E1

**E2 Marker to CF**

■ **Function** Sets the marker frequency to the center frequency (same function as MKR\_3, MKCF).

Header	Program command	Query	Response
E2	E2	_____	_____

■ **Example** E2

## E3

### E3 Marker to CF Step Size

- **Function** Sets the marker frequency to the frequency step size (same function as MKR\_5, MKSS).

Header	Program command	Query	Response
E3	E3	_____	_____

- **Example** E3

---

## E4

### E4 Marker to REF

- **Function** Sets the marker level to the reference level (same function as MKR\_4, MKRL).

Header	Program command	Query	Response
E4	E4	_____	_____

- **Example** E4



## FA Start Frequency

- **Function** Sets the start frequency (same function as STF).

Header	Program command	Query	Response
FA	FA_f	FA?	f f= - 100000000 to 0 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f** - 100 MHz to 8.5 GHz
- **Suffix code**
  - None: Hz ( $10^0$ )
  - HZ: Hz ( $10^0$ )
  - KHZ, KZ: kHz ( $10^3$ )
  - MHZ, MZ: MHz ( $10^6$ )
  - GHZ, GZ: GHz ( $10^9$ )
- **Initial setting** Value of f = 0 Hz
- **Example** FA\_1GZ

## FB Stop Frequency

- **Function** Sets the stop frequency (same function as SOF).

Header	Program command	Query	Response
FB	FB_f	FB?	f f= - 100000000 to 0 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f** - 100 MHz to 8.5 GHz
- **Suffix code**
  - None: Hz ( $10^0$ )
  - HZ: Hz ( $10^0$ )
  - KHZ, KZ: kHz ( $10^3$ )
  - MHZ, MZ: MHz ( $10^6$ )
  - GHZ, GZ: GHz ( $10^9$ )
- **Initial setting** Value of f = 8.5 GHz
- **Example** FB\_5GHZ

# FDN

## FDN Center Frequency Step Down

- **Function** Decreases the center frequency by the frequency step size, if it has been set (same function as CF\_DN).

Header	Program command	Query	Response
FDN	FDN	_____	_____

- **Example** FDN

---

# FMRNG

## FMRNG FM Range

- **Function** Sets the bandwidth for demodulating FM when trace TIME is selected for FM monitoring

Header	Program command	Query	Response
FMRNG	FMRNG_ f	FMRNG?	f f=2000 to 200000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f**  
2 kHz/div  
20 kHz/div  
200 kHz/div
- **Suffix code**  
None: Hz/div  
HZ: Hz/div  
KHZ, KZ: kHz/div  
MHZ, MZ: MHz/div  
GHZ, GZ: GHz/div
- **Initial setting** 200 kHz/div
- **Example** FMRNG\_20KHZ

## FRQ Frequency Mode

- **Function** Selects the mode for setting the frequency band.

Header	Program command	Query	Response
FRQ	FRQ┐a a=0 to 2	FRQ?	FRQ┐a a=0 to 2

- **Value of a**
  - ∅: CENTER-SPAN
  - 1: START-SPAN
  - 2: START-STOP
- **Suffix code** None
- **Initial setting** 2: START-STOP
- **Example**
  - FRQ┐∅
  - FRQ┐1

---

## FS

### FS Full Span

- **Function** Sets the frequency span to the full span that is the maximum value settable in the frequency band being set.

Header	Program command	Query	Response
FS	FS	_____	_____

- **Example** FS

# FSS

## FSS Frequency Step Size

- **Function** Sets the frequency step size for stepping up/down the frequency (same function as SS).

Header	Program command	Query	Response
FSS	FSS_f	FSS?	FSS_f f=1 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f** 1 Hz to 8.5 GHz
- **Suffix code**  
None: Hz ( $10^0$ )  
HZ: Hz ( $10^0$ )  
KHZ ,KZ: kHz ( $10^3$ )  
MHZ ,MZ: MHz ( $10^6$ )  
GHZ ,GZ: GHz ( $10^9$ )
- **Initial setting** 1 GHz
- **Example** FSS\_1GHZ  
FSS\_1000

---

# FUP

## FUP Frequency Step Up

- **Function** Increases the center frequency by the frequency step size, if it has been set (same function as FREQ\_UP).

Header	Program command	Query	Response
FUP	FUP	_____	_____

- **Example** FUP

## GATE Gate Sweep ON/OFF

- Function Allows the gate function to be set to ON or OFF

Header	Program command	Query	Response
GATE	GATE└SW <i>sw=ON, 1, OFF, 0</i>	GATE?	SW <i>sw=ON, OFF</i>

- Value of sw ON: ON  
1: ON  
OFF: OFF  
Ø: OFF
- Suffix code None
- Initial setting OFF: OFF
- Example GATE└ON

# GD

## GD Gate Delay

- Function Sets the delay time of gate.

Header	Program command	Query	Response
GD	GD└t	GD?	t <i>t=0 to 65500</i> Transfers the data without suffix code as if represented in units of $\mu$ sec.

- Value of t 0 to 65.5 msec
- Suffix code None: msec  
US:  $\mu$ sec  
MS: msec  
S: sec
- Initial setting Value of t = 0 sec
- Example GD└2ØMS

## GE

### GE Gate End

- Function Allows the interval of gate to be ended internally or externally.

Header	Program command	Query	Response
GE	GE┘sw sw=INT, EXT	GE?	SW sw=INT, EXT

- Value of sw INT: INTERNAL (inside)  
EXT: EXTERNAL (outside)
- Suffix code None
- Initial setting INT: INTERNAL (inside)
- Example GE┘INT

---

## GL

### GL Gate Length

- Function Sets the length of gate.

Header	Program command	Query	Response
GL	GL┘t	GL?	t t=20 to 65500 Transfers the data without suffix code as if represented in units of $\mu$ sec

- Value of t 20  $\mu$ sec to 65.5 msec
- Suffix code None: msec  
US:  $\mu$ sec  
MS: msec  
S: sec
- Initial setting Value of t = 1 msec
- Example GL┘20MS

## HN Band Select

- **Function** Selects one of the bands in the frequency range from 0 to 8.5 GHz.

Header	Program command	Query	Response
HN	HN_ <u>a</u> a=0 to 2	HN?	a                      a=0 to 2 ***

- **Value of a**    Ø: BAND0  
                    1: BAND1<sup>-</sup>  
                    2: BAND1<sup>+</sup>
- **Suffix code**    None
- **Initial setting** (BAND\_AUTO)
- **Example**        HN\_Ø

**Note:** The response when the band is selected for BAND AUTO becomes “\*\*\*”.

## HNLOCK

### HNLOCK Band Select

- **Function** Selects one of the bands in the frequency range from 0 to 8.5 GHz.

Header	Program command	Query	Response
HNLOCK	HNLOCK_ <u>a</u> a=0 to 2, OFF	HNLOCK?	HNLOCK_ <u>b</u> b=ON, OFF

- **Value of a**    Ø: BAND0                      (Same function as BNDC\_Ø)  
                    1: BAND1<sup>-</sup>                      (Same function as BNDC\_1<sup>-</sup>)  
                    2: BAND1<sup>+</sup>                      (Same function as BNDC\_1<sup>+</sup>)  
                    OFF: BAND AUTO              (Same function as BNDC\_AUTO)
- **Value of b**    ON: BAND 0, 1<sup>-</sup>, 1<sup>+</sup>  
                    OFF: BAND AUTO
- **Suffix code**    None
- **Initial setting** OFF: BAND AUTO
- **Example**        HNLOCK\_2

# HNUNLK

## HNUNLK Band Select

- **Function** Allows the bands to be selected in the AUTO mode ( Same function as BNDC\_AUTO, HNLOCK\_OFF ).

Header	Program command	Query	Response
HNUNLK	HNUNLK	_____	_____

- **Example** HNUNLK

---

# KSA

## KSA Unit for Log Scale

- **Function** Sets the unit system of LOG scale to dBm (same function as UNT\_0).

Header	Program command	Query	Response
KSA	KSA	_____	_____

- **Example** KSA



**KSB Unit for Log Scale**

■ **Function** Sets the unit system of LOG scale to dBmV (same function as UNT\_2).

Header	Program command	Query	Response
KSB	KSB	_____	_____

■ **Example** KSB

**KSC Unit for Log Scale**

■ **Function** Sets the unit system of LOG scale to dB $\mu$ V (same function as UNT\_1).

Header	Program command	Query	Response
KSC	KSC	_____	_____

■ **Example** KSC

# KSD

## KSD Unit for Log Scale

■ Function Sets the unit system of LOG scale to V (same function as UNT\_3).

Header	Program command	Query	Response
KSD	KSD	_____	_____

■ Example KSD

---

# KSG

## KSG Average ON

■ Function Allows the averaging to be used when set to ON.

Header	Program command	Query	Response
KSG	KSG	_____	_____

■ Example KSG

## KSH Average OFF

- **Function** Turns off the averaging operation to set the mode for processing waveform to NORMAL.

Header	Program command	Query	Response
KSH	KSH	_____	_____

- **Example** KSH

## KSO Delta Marker to Span

- **Function** Sets the delta marker frequency to frequency span (same function as MKR\_6, MKSP).

Header	Program command	Query	Response
KSO	KSO	_____	_____

- **Example** KSO

# LG

## LG

### Scale

- Function Sets the magnification of Y axis and scale.

Header	Program command	Query	Response
LG	LG_ℓ LG_a	LG?	ℓ ℓ=0, 1, 2, 5, 10

- Value of ℓ
  - ∅: Sets the scaling function to linear mode.
  - 1: 1 dB/div (Sets the scaling function to logarithmic mode)
  - 2: 2 dB/div (Sets the scaling function to logarithmic mode)
  - 5: 5 dB/div (Sets the scaling function to logarithmic mode)
  - 1∅: 10 dB/div (Sets the scaling function to logarithmic mode)
- Value of a
  - UP: SCALE UP
  - DN: SCALE DOWN
- Suffix code
  - None: dB/div
  - DB, DBM, DM: dB/div
- Initial setting 1∅: 10 dB/div
- Example
  - LG\_UP
  - LG\_5DB

---

# LN

## LN

### Linear Scale

- Function Sets the magnification of Y axis and scale.

Header	Program command	Query	Response
LN	LN	_____	_____

- Example LN

## LSS Reference Level Step Size (Manual)

- **Function** Sets the step size (manual values) of reference level increasing or decreasing in the specified step level.

Header	Program command	Query	Response
LSS	LSS_ℓ	LSS?	LSS_ℓ ℓ=0.1 to 100.0 Transfers the data without suffix code as if represented in units of dB

- **Value of ℓ** 0.1 to 100.0 dB (0.1dB step)
- **Suffix code** None: dB  
DB, DBM, DM: dB
- **Initial setting** Value of ℓ = 1 dB
- **Example** LSS\_6  
LSS\_10

## LSSA Reference Level Step Size (Auto)

- **Function** Sets the step size (auto values) of reference level increasing or decreasing in the specified step level during LOG SCALE.

Header	Program command	Query	Response
LSSA	LSSA_a	LSSA?	LSSA_a      a=1, 2, 5, 10

- **Value of a** 1: 1 div  
2: 2 div  
5: 5 div  
10: 10 div
- **Suffix code** None
- **Initial setting** 1: 1 div
- **Example** LSSA\_10

# MKA?

## MKA? Marker Level Read

- **Function** Reads out the level data at the marker point. At the delta marker point, the level differences are read out (same function as MKL?).

Header	Program command	Query	Response
MKA?	_____	MKA?	ℓ v f

- **Value of ℓ** No unit, Level data with 1 dB unit (when display unit system for marker level is dB), Resolution 0.01 dB
- **Value of v** No unit, Level data with 1 nV unit (when display unit system for marker level is V), Resolution 0.1 nV  
No unit, Level data with 1 V unit (for EXT TRIG MONITOR), Resolution 0.001 V
- **Value of f** No unit, Frequency data with 1 Hz unit (for FM MONITOR), Resolution 1 Hz
- **Example** MKA?

# MKACT

## MKACT Marker Active

- **Function** Selects the active marker among the multimarkers.

Header	Program command	Query	Response
MKACT	MKACT_a a=1 to 10	MKACT?	a a=1 to 10

- **Value of a** 1 to 10 (multimarker No.)
- **Suffix code** None
- **Initial setting** 1: 1
- **Example** MKACT\_1

# MKCF

## MKCF      Marker to CF

- **Function**      Sets the marker frequency to the center frequency (same function as MKR  $\square$  3, E2).

Header	Program command	Query	Response
MKCF	MKCF	_____	_____

- **Example**      MKCF

---

# MKD

## MKD      Delta Marker Mode

- **Function**      Sets the marker mode to delta marker mode.

Header	Program command	Query	Response
MKD	MKD	_____	_____

- **Example**      MKD

## MKF?

### MKF?

### Marker Frequency Read

■ Function

Reads out the frequency or time data at the marker point. At the delta marker mode, the frequency or time differences are read out.

Header	Program command	Query	Response
MKF?	_____	MKF?	f t

■ Value of f

No unit, frequency data with 1 Hz unit, Resolution 0.1 Hz

■ Value of t

No unit, time data with 1  $\mu$ s unit, Resolution 0.1  $\mu$ sec

■ Example

MKF?

---

## MKL?

### MKL?

### Marker Level Read

■ Function

Reads out the level data at the marker point. At the delta marker mode, the level differences are read out.

Header	Program command	Query	Response
MKL?	_____	MKL?	$\ell$ v f

■ Value of  $\ell$

No unit, Level data with 1 dB unit (when display unit system for marker level is dB), Resolution 0.01 dB

■ Value of v

No unit, Level data with 1 nV unit (when display unit system for marker level is V), Resolution 0.1 nV

No unit, Level data with 1 V unit (for EXT TRIG MONITOR), Resolution 0.001 V

■ Value of f

No unit, Frequency data with 1 Hz unit (for FM MONITOR), Resolution 1 Hz

■ Example

MKL?



# MKLIST

## MKLIST Multi Marker List

■ Function Turns ON or OFF the multimarker list.

Header	Program command	Query	Response
MKLIST	MKLIST_SW sw=ON, 1, OFF, 0	MKLIST?	SW sw=ON,OFF

■ Value of sw ON: ON  
1: ON  
OFF: OFF  
Ø: OFF

■ Suffix code None

■ Initial setting OFF: OFF

■ Example MKLIST\_ON

---

# MKMCL

## MKMCL Clear Multi Marker

■ Function Clears all the registered multimarkers.

Header	Program command	Query	Response
MKMCL	MKMCL	_____	_____

■ Example MKMCL

# MKMHI

## MKMHI

### ■ Function

## Multi Marker

Registers multimarkers on the peak point from the maximum level down to the tenth in descending order.

Header	Program command	Query	Response
MKMHI	MKMHI	_____	_____

### ■ Example

MKMHI

---

# MKMHRM

## MKMHRM

### ■ Function

## Multi Marker

Registers multimarkers on the harmonic frequency ranging from the reference active marker frequency up to the tenth.

Header	Program command	Query	Response
MKMHRM	MKMHRM	_____	_____

### ■ Example

MKMHRM

## MKMIN Minimum Search

- **Function** Searches the spectrum being displayed for the minimum point and moves the marker to that point.

Header	Program command	Query	Response
MKMIN	MKMIN	_____	_____

- **Example** MKMIN

## MKML? Multi Marker List Query (Level)

- **Function** Reads out the level data at multimarkers.

Header	Program command	Query	Response
MKML	_____	MKML?_a	ℓ v f

- **Value of a** 1 to 10 (multimarker No.)
- **Value of ℓ** No unit, Level data with 1 dB unit (when display unit system for marker level is dB), Resolution 0.01 dB
- **Value of v** No unit, Level data with 1 nV unit (when display unit system for marker level is V), Resolution 0.1 nV  
No unit, Level data with 1 V unit (for EXT TRIG MONITOR) Resolution 0.001 V
- **Value of f** No unit, Frequency data with 1 Hz unit (for FM MONITOR) Resolution 1 Hz
- **Suffix code** None

# MKMP

## MKMP Marker Position

■ Function Specifies the frequency of the designated multimarker number.

Header	Program command	Query	Response
MKMP	MKMP┐a, f	MKMP?┐a a=1 to 10	f f= -100000000 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

■ Value of a 1 to 10 (Multimarker No.)

■ Value of f -100 MHz to 8.5 GHz

■ Suffix code  
None: Hz (10<sup>0</sup>)  
HZ: Hz (10<sup>0</sup>)  
KHZ, KZ: kHz (10<sup>3</sup>)  
MHZ, MZ: MHz (10<sup>6</sup>)  
GHZ, GZ: GHz (10<sup>9</sup>)

■ Example MKMP┐5, 2400MHZ

# MKMULTI

## MKMULTI Multi Marker

■ Function Turns ON or OFF the multimarker.

Header	Program command	Query	Response
MKMULTI	MKMULTI┐SW sw=ON, 1, OFF, 0	MKMULTI?	SW sw=ON, OFF

■ Value of sw  
ON: ON  
1: ON  
OFF: OFF  
Ø: OFF

■ Suffix code None

■ Initial setting OFF: OFF

■ Example MKMULTI┐ON

## MKN Marker Position

- **Function** Specifies the zone marker center position on the X axis in the frequency or time unit.

Header	Program command	Query	Response
MKN	MKN_f MKN_t MKN_a	MKN?	f, t f = -100000000 to 0 to 8500000000 Transfers the data without suffix code as if represented in units of Hz t = -1000000000 to 1000000000 Transfers the data without suffix code as if represented in units of $\mu$ sec

- **Value of f** - 100 MHz to 8.5 GHz  
(0 to 8.5 GHz (specified when the valid trace is A, B, or BG))
- **Value of t** - 1000 sec to 1000 sec (specified when the valid trace is TIME)
- **Value of a** UP: UP  
DN: DOWN
- **Suffix code**

f: None:	Hz (10 <sup>0</sup> )	t: None:	msec
HZ:	Hz (10 <sup>0</sup> )	US:	$\mu$ sec
KHZ, KZ:	kHz (10 <sup>3</sup> )	MS:	msec
MHZ, MZ:	MHz (10 <sup>6</sup> )	S:	sec
GHZ, GZ:	GHz (10 <sup>9</sup> )		
- **Example** MKN\_100MHZ  
MKN\_UP

# MKOFF

## MKOFF Marker Mode

- **Function** Turns off the marker mode.

Header	Program command	Query	Response
MKOFF	MKOFF_a	_____	_____

- **Value of a** ALL: Marker off  
None: Marker off
- **Suffix code** None
- **Example** MKOFF\_ALL  
MKOFF

# MKP

## MKP

### Marker Position

- **Function** Specifies the zone marker center position on the X axis in the point unit (same function as MKZ).

Header	Program command	Query	Response
MKP	MKP_ p	MKP?	p p=0 to 500

- **Value of p** 0 to 500
- **Suffix code** None
- **Initial setting** Value of p = 250
- **Example** MKP\_250  
MKP\_500

---

# MKPK

## MKPK

### Peak Search

- **Function** Searches the spectrum being displayed for one of the special points and moves the marker to that point.

Header	Program command	Query	Response
MKPK	MKPK_ a	_____	_____

- **Value of a** None: SEARCH PEAK (MAX)  
HI: SEARCH PEAK (MAX)  
NH: SEARCH NEXT PEAK  
NR: SEARCH NEXT RIGHT PEAK  
NL: SEARCH NEXT LEFT PEAK
- **Suffix code** None
- **Example** MKPK\_HI  
MKPK\_NL

## MKR Marker Mode

- **Function** Switches the marker mode and executes the 'MKR to ' functions.

Header	Program command	Query	Response
MKR	MKR_a a=0 to 7	MKR?	MKR_a a=0 to 2

- **Value of a**
  - Ø: NORMAL
  - 1: DELTA
  - 2: OFF
  - 3: MKR to CF
  - 4: MKR to REF
  - 5: MKR to CFstep size
  - 6: \_MKR to SPAN
  - 7: ZONE to SPAN

- **Suffix code** None

- **Initial setting** Ø: NORMAL

- **Example** MKR\_Ø

# MKRL

## MKRL Marker to REF

- **Function** Sets the marker level to the reference level (same function as MKR\_4, E4)

Header	Program command	Query	Response
MKRL	MKRL	_____	_____

- **Example** MKRL

# MKS

## MKS Peak Search

- **Function** Searches the spectrum being displayed for one of the special points and moves the marker to that point.

Header	Program command	Query	Response
MKS	MKS_ a a=0 to 2, 9 to 11	_____	_____

- **Value of a**
  - ∅: SEARCH PEAK (MAX)
  - 1: SEARCH NEXT PEAK
  - 2: SEARCH DIP (MIN)
  - 9: SEARCH NEXT RIGHT PEAK
  - 1∅: SEARCH NEXT LEFT PEAK
  - 11: SEARCH NEXT DIP

- **Suffix code** None

- **Example**
  - MKS\_∅
  - MKS\_9

# MKSLCT

## MKSLCT Select Multi Marker

- **Function** Specifies one of the 1 to 10 multimarkers and sets the specified marker to ON or OFF.

Header	Program command	Query	Response
MKSLCT	MKSLCT_ a , sw a=1 to 10 sw=ON, 1, OFF, 0	MKSLCT?_ a a=1 to 10	SW sw=ON, OFF

- **Value of a** 1 to 10 (multimarker No.)
- **Value of sw**
  - ON: ON
  - 1: ON
  - OFF: OFF
  - ∅: OFF
- **Suffix code** None
- **Initial setting** OFF: OFF
- **Example** MKSLCT\_3,ON



## MKSP Delta Marker to Span

■ **Function** Sets the delta marker frequency to the span (same function as MKR\_6, KSO).

Header	Program command	Query	Response
MKSP	MKSP	_____	_____

■ **Example** MKSP

## MKSS Marker to CF Step Size

■ **Function** Sets the marker frequency as a frequency step size (same function as MKR\_5, E3).

Header	Program command	Query	Response
MKSS	MKSS	_____	_____

■ **Example** MKSS

# MKTRACK

## MKTRACK Tracking ON/OFF

■ Function Sets the signal tracking function to ON or OFF.

Header	Program command	Query	Response
MKTRACK	MKTRACK <sub>□</sub> SW sw=ON, 1, OFF, 0	MKTRACK?	SW sw=ON, OFF

■ Value of sw  
ON: ON  
1: ON  
OFF: OFF  
∅: OFF

■ Suffix code None

■ Initial setting OFF: OFF

■ Example MKTRACK<sub>□</sub>ON

---

# MKW

## MKW Zone Marker Width

■ Function Specifies the zone marker width in the div unit.

Header	Program command	Query	Response
MKW	MKW <sub>□</sub> a a=1 to 2, 5 to 7	MKW?	MKW <sub>□</sub> a a=1 to 2, 5 to 7

■ Value of a  
1: Spot  
2: 10 div  
5: 1 div  
6: 2 div  
7: 5 div

■ Suffix code None

■ Initial setting 5: 1 div

■ Example  
MKW<sub>□</sub>1  
MKW<sub>□</sub>5

**MKZ Zone Marker Position**

■ **Function** Specifies the zone marker center position on the X axis in the point unit (same function as MKP).

Header	Program command	Query	Response
MKZ	MKZ <p>p</p> p=0 to 500	MKZ?	MKZ <p>p</p> p=0 to 500

- Value of p 0 to 500
- Suffix code None
- Initial setting Value of p = 250
- Example MKZ

250

  
MKZ

500

**MKZF Zone Marker Position**

■ **Function** Specifies the zone marker center position on the X axis in frequency or time units.

Header	Program command	Query	Response
MKZF	MKZF <p>f</p> MKZF <p>t</p>	MKZF?	f t f= -100000000 to 0 to 8500000000 Transfers the data without suffix code as if represented in units of Hz t= -100000000 to 8500000000 Transfers the data without suffix code as if represented in units of $\mu$ sec

- Value of f -100 MHz to 8.5 GHz (specified when the valid trace is A, B, or BG)
- Value of t -1000 sec to 1000 sec (specified when the valid trace is TIME)
- Suffix code f: None: Hz ( $10^0$ )      t: None: msec  
HZ: Hz ( $10^0$ )      US:  $\mu$ sec  
KHZ, KZ: kHz ( $10^3$ )      MS: msec  
MHZ, MZ: MHz ( $10^6$ )      S: sec  
GHZ, GZ: GHz ( $10^9$ )
- Example MKZF

100MHZ

  
MKZF

1200000000

# MON

## MON Monitor Mode

- Function Selects the function for monitoring the sound from the detector output.

Header	Program command	Query	Response
MON	MON_m	MON?	m m=AM, FM, OFF

- Value of m AM: Amplitude Modulation  
FM: Frequency Modulation  
OFF: OFF
- Suffix code None
- Initial setting OFF: OFF
- Example MON\_AM

---

# MONVOL

## MONVOL Monitor Volume

- Function Adjusts the volume of sound monitor.

Header	Program command	Query	Response
MONVOL	MONVOL_v v=0 to 20	MONVOL?	v v=0 to 20

- Value of v 0 to 20 (1 step)
- Suffix code None
- Initial setting Value of V = 10
- Example MONVOL\_10

**MTØ Tracking OFF**

■ **Function** Sets the signal tracking function to OFF.

Header	Program command	Query	Response
MTØ	MTØ	_____	_____

■ **Example** MTØ

**MT1 Tracking ON**

■ **Function** Sets the signal tracking function to ON.

Header	Program command	Query	Response
MT1	MT1	_____	_____

■ **Example** MT1

# MXMH

## MXMH Max Hold

- **Function** Sets the mode for processing the trace waveform to MAX HOLD.

Header	Program command	Query	Response
MXMH	MXMH <tr< td=""><td>_____</td><td>_____</td></tr<>	_____	_____

- **Value of tr** TRA: Trace A (FREQ)
- **Suffix code** None
- **Example** MXMH TRA

---

# MZW

## MZW Zone Marker Width

- **Function** Specifies the zone marker width on the X axis in the point unit.

Header	Program command	Query	Response
MZW	MZW_w w = 1 to 501	MZW?	MZW_w w = 1 to 501

- **Value of w** 1 to 501 (odd number)
- **Suffix code** None
- **Initial setting** Value of w = 51
- **Example** MZW\_1  
MZW\_51  
MZW\_501

## MZWF Zone Marker Width

- **Function** Specifies the zone marker width on the X axis in the frequency.

Header	Program command	Query	Response
MZWF	MZWF_ f	MZWF?	f f=1 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f** 1 Hz to 8.5 GHz
- **Suffix code**
  - None: Hz (10<sup>0</sup>)
  - HZ: Hz (10<sup>0</sup>)
  - KHZ, KZ: kHz (10<sup>3</sup>)
  - MHZ, MZ: MHz (10<sup>6</sup>)
  - GHZ, GZ: GHz (10<sup>9</sup>)
- **Initial setting** frequency width equivalent to 1 div (850 MHz)
- **Example**
  - MZWF\_100
  - MZWF\_1MHZ

# M1

## M1 Marker Mode

- **Function** Turns off the marker mode (same function as MKR\_2).

Header	Program command	Query	Response
M1	M1	_____	_____

- **Example** M1

## M2

### M2 Marker Mode

■ Function Sets the marker mode to NORMAL mode (same function as MKR\_0).

Header	Program command	Query	Response
M2	M2	_____	_____

■ Example M2

---

## M3

### M3 Marker Mode

■ Function Sets the marker mode to delta marker mode (same function as MKR\_1).

Header	Program command	Query	Response
M3	M3	_____	_____

■ Example M3



# PARAM

## PARAM      Parameter

- **Function**      Displays the currently parameter list on the screen.

Header	Program command	Query	Response
PARAM	PARAM _a	PARAM?	a a=1,2,OFF

- **Value of a**      OFF: OFF  
                         1:    Parameter list 1  
                         2:    Parameter list 2
- **Suffix code**      None
- **Initial setting** OFF: OFF
- **Example**          PARAM\_1

---

# PCF

## PCF      Peak to Center Frequency

- **Function**      Searches the spectrum being displayed for the maximum point and sets the frequency at the point to the center frequency.

Header	Program command	Query	Response
PCF	PCF	_____	_____

- **Example**          PCF

## PP

### PP

### Presel Auto

- **Function** Adjusts the bias of preselector automatically (same function as PRESEL\_AUTO).

Header	Program command	Query	Response
PP	PP	_____	_____

- **Example** PP

---

## PRESEL

### PRESEL

### Presel Tune

- **Function** Adjusts the bias of preselector.

Header	Program command	Query	Response
PRESEL	PRESEL_a	PRESEL?	a a = -128 to 127

- **Value of a** AUTO: Automatic adjustment  
-128 to 127: Setting values manually  
PRESET: Sets 0
- **Suffix code** None
- **Initial setting** 0 (MANUAL) (provided the address already allocated is not initialized)
- **Example** PRESEL\_AUTO

**PRL Peak to Reference Level**

- **Function** Searches the spectrum being displayed for the maximum point and sets its level to the reference level.

Header	Program command	Query	Response
PRL	PRL	_____	_____

- **Example** PRL

**PSW Zone Sweep**

- **Function** Sets the zone sweep to ON or OFF.

Header	Program command	Query	Response
PSW	PSW┘SW <small>sw = ON, 1, OFF, 0</small>	PSW?	PSW┘SW <small>sw = ON, OFF</small>

- **Value of sw** ON: ON  
1: ON  
OFF: OFF  
Ø: OFF
- **Suffix code** None
- **Initial setting** OFF: OFF
- **Example** PSW┘ON

# RB

## RB Resolution Band Width

■ Function Sets the resolution bandwidth (same function as RBW).

Header	Program command	Query	Response
RB	RB_f RB_a	RB?	f f= 10 to 3000000 Transfers the data without suffix code as if represented in units of Hz

■ Value of f 10 Hz to 3 MHz (1-3 sequence)

■ Value of a  
UP: RBW UP  
DN: RBW DOWN  
AUTO: RBW AUTO

■ Suffix code  
f: None: Hz (10<sup>0</sup>)  
HZ: Hz (10<sup>0</sup>)  
KHZ, KZ: kHz (10<sup>3</sup>)  
MHZ, MZ: MHz (10<sup>6</sup>)  
GHZ, GZ: GHz (10<sup>9</sup>)

a: None

■ Initial setting RBW = Calculated value when AUTO is selected for RBW.

■ Example RB\_3KHZ

# RBW

## RBW Resolution Band Width

■ Function Sets the resolution bandwidth.

Header	Program command	Query	Response
RBW	RBW_a a=0 to 9, 13, 14	RBW?	RBW_a a=0 to 9, 13, 14

■ Value of a

0: 30 Hz	6: 30 kHz
1: 100 Hz	7: 100 kHz
2: 300 Hz	8: 300 kHz
3: 1 kHz	9: 1 MHz
4: 3 kHz	13: 10 Hz
5: 10 kHz	14: 3 MHz

■ Suffix code None

■ Initial setting RBW = Calculated value when AUTO is selected for RBW

■ Example RBW\_5

**RL Reference Level**

- **Function** Sets the reference level (same function as RLV).

Header	Program command	Query	Response
RL	RL_ℓ RL_a	RL?	ℓ ℓ: Available for the current scale unit, provided μV unit is always selected in LIN mode.

- **Value of ℓ** Value equivalent to -100 dBm to +50 dBm (0.1 dB step)
- **Value of a** UP: LEVEL STEP UP  
DN: LEVEL STEP DOWN
- **Suffix code** None: Available for the current scale unit, provided μV unit is always selected in LIN mode
  - DB, DBM, DM: dBm
  - DBMV: dBmV
  - DBUV: dBμV
  - DBUVE: dBμV (emf)
  - V: V
  - MV: mV
  - UV: μV
  - W: W
  - MW: mW
  - UW: μW
  - NW: nW
  - PW: pW
  - FW: fW
- **Initial setting** Value of ℓ = 10 dBm
- **Example** RL\_ -100DBM  
RL\_5V  
RL\_ -10  
RL\_UP

# RLV

## RLV Reference Level

■ Function Sets the reference level (same function as RL).

Header	Program command	Query	Response
RLV	RLV_ℓ	RLV?	RLV_ℓ ℓ: Transfers the data without suffix code as if represented in the current scale units (always μV unit in LIN mode).

■ Value of ℓ Value equivalent to -100 to +50 dBm (0.1 dB step)

■ Suffix code None: Available for the current scale unit, provided μV unit is always selected in LIN mode

DB, DBM, DM: dBm  
 DBMV: dBmV  
 DBUV: dBμV  
 DBUVE: dBμV (emf)  
 V: V  
 MV: mV  
 UV: μV

■ Initial setting Value of ℓ = 10 dBm

■ Example RLV\_ -70DBM  
 RLV\_5V  
 RLV\_ -10

# RMK?

## RMK? Reference Marker Position

■ Function Reads out the position of reference marker.

Header	Program command	Query	Response
RMK?	_____	RMK?	RMK_a

■ Value of a 0 to 500

■ Example RMK?

**SCL**                      **Log/Linear Scale**

- **Function**                      Sets the magnification of LOG/LIN scale (Y axis).

Header	Program command	Query	Response
SCL	SCL_a                                      a=0 to 7	SCL_?	SCL_a                                      a=0 to 7

- **Value of a**                      0: 1 dB/div (LOG SCALE)  
     1: 2 dB/div (LOG SCALE)  
     2: 5 dB/div (LOG SCALE)  
     3: 10 dB/div (LOG SCALE)  
     4: 1 %/div (LIN SCALE)  
     5: 2 %/div (LIN SCALE)  
     6: 5 %/div (LIN SCALE)  
     7: 10 %/div (LIN SCALE)

- **Suffix code**                      None

- **Initial setting**                      3: 10 dB/div (LOG SCALE)

- **Example**                              SCL\_0  
     SCL\_5

**SOF****SOF**                      **Stop Frequency**

- **Function**                      Sets the stop frequency (same function as FB).

Header	Program command	Query	Response
SOF	SOF_f	SOF?	SOF_f f = -100000000 to 0 to 850000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f**                              - 100 MHz to 8.5 GHz

- **Suffix code**                              None:        Hz (10<sup>0</sup>)  
     HZ:         Hz (10<sup>0</sup>)  
     KHZ, KZ:   kHz (10<sup>3</sup>)  
     MHZ, MZ:   MHz (10<sup>6</sup>)  
     GHZ, GZ:   GHz (10<sup>9</sup>)

- **Initial setting**                              Value of f = 8.5 GHz

- **Example**                                      SOF\_123MHZ  
     SOF\_45.6KHZ

# SP

## SP Frequency Span

■ Function Sets the frequency span (same function as SPF).

Header	Program command	Query	Response
SP	SP_f SP_a	SP?	f f=0 to 8600000000 Transfers the data without suffix code as if represented in units of Hz

■ Value of f 0 Hz to 8.6 GHz

■ Value of a UP: FREQ SPAN STEP UP (Same function as SPU)  
DN: FREQ SPAN STEP DOWN (Same function as SPD)

■ Suffix code None: Hz ( $10^0$ )  
HZ: Hz ( $10^0$ )  
KHZ, KZ: kHz ( $10^3$ )  
MHZ, MZ: MHz ( $10^6$ )  
GHZ, GZ: GHz ( $10^9$ )

■ Initial setting Value of f = 8.6 GHz

■ Example SP\_6GHZ

---

# SPD

## SPD Frequency Span Step Down

■ Function Decreases the frequency span in the 5-2-1 steps (same function as SP\_DN).

Header	Program command	Query	Response
SPD	SPD	_____	_____

■ Example SPD



## SPF Frequency Span

- **Function** Sets the frequency span (same function as SP).

Header	Program command	Query	Response
SPF	SPF_f	SPF?	SPF_f f=0 to 8600000000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f** 0 Hz to 8.6 GHz
- **Suffix code**
  - None: Hz (10<sup>0</sup>)
  - HZ: Hz (10<sup>0</sup>)
  - KHZ, KZ: kHz (10<sup>3</sup>)
  - MHZ, MZ: MHz (10<sup>6</sup>)
  - GHZ, GZ: GHz (10<sup>9</sup>)
- **Initial setting** Value of f=8.6 GHz
- **Example**
  - SPF\_101MHZ
  - SPF\_3.5GHZ

# SPFUNC

## SPFUNC Time Trace Special Function

- **Function** Sets the function for monitoring the trace time waveform (FM/TRIG MONITOR).

Header	Program command	Query	Response
SPFUNC	SPFUNC_sw	SPFUNC?	SW sw=OFF, FM, EXT

- **Value of sw**
  - OFF: OFF
  - FM: FM MONITOR
  - EXT: EXT TRIGGER MONITOR
- **Suffix code** None
- **Initial setting** OFF: OFF
- **Example** SPFUNC\_FM

# SPU

## SPU

### Frequency Span Step Up

■ Function      Increases the frequency span in the 1-2-5 steps (same function as SP\_UP).

Header	Program command	Query	Response
SPU	SPU	_____	_____

■ Example      SPU

---

# SS

## SS

### Frequency Step Size

■ Function      Sets the frequency step size for stepping up/down the frequency (same function as FSS).

Header	Program command	Query	Response
SS	SS_f	SS?	f f=1 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

■ Value of f      1 Hz to 8.5 GHz

■ Suffix code      None:      Hz (10<sup>0</sup>)  
HZ:      Hz (10<sup>0</sup>)  
KHZ, KZ:      kHz (10<sup>3</sup>)  
MHZ, MZ:      MHz (10<sup>6</sup>)  
GHZ, GZ:      GHz (10<sup>9</sup>)

■ Example      SS\_1MHZ

## ST Sweep Time

- Function Sets the frequency sweep time/time span.

Header	Program command	Query	Response
ST	ST_t ST_a	ST?	t t=50 to 1000000000 Transfers the data without suffix code as if represented in units of $\mu$ sec

- Value of t 50  $\mu$ sec to 1000 sec (20 msec to 1000 sec for frequency axis)
- Value of a UP: SWT UP  
DN: SWT DOWN  
AUTO: SWT AUTO
- Suffix code t: None: msec S: sec  
US:  $\mu$  sec a: None  
MS: msec
- Initial setting Calculated value when AUTO is selected for SWT
- Example ST\_AUTO  
ST\_20MS

## STF Start Frequency

- Function Sets the start frequency (same function as FA).

Header	Program command	Query	Response
STF	STF_f	STF?	STF_f f= -100000000 to 0 to 8500000000 Transfers the data without suffix code as if represented in units of Hz

- Value of f - 100 MHz to 8.5 GHz
- Suffix code None: Hz ( $10^0$ )  
HZ: Hz ( $10^0$ )  
KHZ, KZ: kHz ( $10^3$ )  
MHZ, MZ: MHz ( $10^6$ )  
GHZ, GZ: GHz ( $10^9$ )
- Initial setting Value of f = 0 Hz
- Example STF\_123MHZ  
STF\_45.6KHZ

# SWT

## SWT Sweep Time

■ **Function** Sets the frequency sweep time/time span (same function as ST).

Header	Program command	Query	Response
SWT	SWT_t	SWT?	SWT_t t=50 to 1000000000 Transfers the data without suffix code as if represented in units of $\mu$ sec

■ **Value of t** 50  $\mu$ sec to 1000 sec (20 msec to 1000 sec for frequency axis)

■ **Suffix code** None: msec  
US:  $\mu$ sec  
MS: msec  
S: sec

■ **Initial setting** Calculated value when AUTO is selected for SWT

■ **Example** SWT\_1S  
SWT\_20MS

---

# TDLY

## TDLY Delay Time

■ **Function** Sets the delay time from the point where trace time trigger occurs.

Header	Program command	Query	Response
TDLY	TDLY_t	TDLY?	t t= -1000000000 to 65500 Transfers the data without suffix code as if represented in units of $\mu$ sec

■ **Value of t** -1000 sec to 65.5 msec

■ **Suffix code** None: msec  
US:  $\mu$ sec  
MS: msec  
S: sec

■ **Initial setting**  $\emptyset$ : 0 sec

■ **Example** TDLY\_20MS

# TEXPAND

## TEXPAND

### Time Expand

- **Function** Turns ON or OFF the expand functions of trace time.

Header	Program command	Query	Response
TEXPAND	TEXPAND_ <u>sw</u> sw=ON, 1, OFF, 0	TEXPAND?	SW sw=ON, OFF

- **Value of sw** ON: ON  
1: ON  
OFF: OFF  
Ø: OFF
- **Suffix code** None
- **Initial setting** OFF
- **Example** TEXPAND\_ON

---

## TM

## TM

### Trigger

- **Function** Sets the trigger switch/trigger source (same function as TRG).  
This command allows both the trigger switch and the trigger source to be set.

Header	Program command	Query	Response
TM	TM_ <u>a</u>	TM?	a a=FREE, VID, LINE, EXT, TV

- **Value of a** FREE: FREERUN  
VID: VIDEO  
LINE: LINE  
EXT: EXT  
TV: TV
- **Suffix code** None
- **Initial setting** FREE: FREERUN
- **Example** TM\_FREE

# TMMD

## TMMD Trace Time Storage Mode

■ **Function** Selects the mode for processing the trace TIME waveform.

Header	Program command	Query	Response
TMMD	TMMD_a a=0 to 5	TMMD?	TMMD_a a=0 to 5

■ **Value of a**    ∅: NORMAL  
                  1: MAX HOLD  
                  2: AVERAGE  
                  3: MIN HOLD  
                  4: CUMULATIVE  
                  5: OVER WRITE

■ **Suffix code**   None

■ **Initial setting** ∅: NORMAL

■ **Example**        TMMD\_∅

---

# TMWR

## TMWR Trace Time Write Switch

■ **Function** Controls writing the waveform to the trace TIME.

Header	Program command	Query	Response
TMWR	TMWR_sw sw=ON, 1, OFF, 0	TMWR?	TMWR_sw sw=ON, OFF

■ **Value of sw**   ON: ON  
                  1: ON  
                  OFF: OFF  
                  ∅: OFF

■ **Suffix code**   None

■ **Initial setting** ON: ON

■ **Example**        TMWR\_ON

**TRG Trigger**

- **Function** Sets the trigger switch/trigger source (same function as TM). This command allows both the trigger switch and the trigger source to be set.

Header	Program command	Query	Response
TRG	TRG_a a=0 to 3,6	TRG?	TRG_a a=0 to 3,6

- **Value of a**
  - ∅: FREERUN
  - 1: VIDEO
  - 2: LINE
  - 3: EXT
  - 6: TV
- **Suffix code** None
- **Initial setting** ∅: FREERUN
- **Example** TRG\_∅

**TRGLVL**

**TRGLVL Trigger Level**

- **Function** Sets the threshold level of the trigger for starting the sweep when trigger source = VIDEO, EXT (Input ) are selected

Header	Program command	Query	Response
TRGLVL	TRGLVL_ℓ	TRGLVL?	ℓ

- **Value of ℓ**
  - 10.0 to +10.0 (0.1 Step) .... when the trigger source is EXT (Input1) (V unit)
  - 100 to +100 (1 Step) ..... when trigger source is VIDEO (% unit)
- **Suffix code**
  - when trigger source is VIDEO None
  - when the trigger source is EXT None: V
  - V: V
- **Initial setting** Value of ℓ = 0.0
- **Example** TRGLVL\_-10.0  
TRGLVL\_9.9

# TRGS

## TRGS

### Trigger Switch

- **Function** Switches the trigger switch to Free run or Triggered.

Header	Program command	Query	Response
TRGS	TRGS_ <u>sw</u> <i>sw = FREE, TRGD</i>	TRGS?	SW <i>sw = FREE, TRGD</i>

- **Value of sw** FREE: FREERUN  
TRGD: TRIGGERED
- **Suffix code** None
- **Initial setting** FREE: FREERUN
- **Example** TRGS\_FREE

---

# TRGSLP

## TRGSLP

### Trigger Slope

- **Function** Selects the positive or negative leading edge of the trigger when trigger source is VIDEO or EXT mode.

Header	Program command	Query	Response
TRGSLP	TRGSLP_ <u>sw</u>	TRGSLP?	SW <i>sw = RISE, FALL</i>

- **Value of sw** RISE: Positive leading edge  
FALL: Negative leading edge
- **Suffix code** None
- **Initial setting** RISE: Positive leading edge
- **Example** TRGSLP\_RISE



# TRGSOURCE

## TRGSOURCE Trigger Source

- **Function** Selects the trigger source. Trigger switch setting is not changed by this command.

Header	Program command	Query	Response
TRGSOURCE	TRGSOURCE_ <u>sw</u>	TRGSOURCE?	SW sw = VID, LINE, EXT, TV

- **Value of sw** VID: VIDEO  
LINE: LINE  
EXT: EXT  
TV: TV
- **Suffix code** None
- **Initial setting** VID: VIDEO
- **Example** TRGSOURCE\_VID

---

## TS

## TS Take Sweep

- **Function** Executes single sweep synchronously (same function as SWP).

Header	Program command	Query	Response
TS	TS	_____	_____

- **Example** TS

# TSP

## TSP Time Span

■ Function Sets the span of trace time.

Header	Program command	Query	Response
TSP	TSP_t	TSP?	t t=50 to 1000000000 Transfers the data without suffix code as if represented in units of $\mu$ sec

■ Value of t 50  $\mu$ sec to 1000 sec

■ Suffix code  
None: msec  
US:  $\mu$ sec  
MS: msec  
S: sec

■ Initial setting 200 msec

■ Example TSP\_100  
TSP\_100S

# TVLINE

## TVLINE TV (H-Sync: line)

■ Function Sets the number of lines used in the TV horizontal synchronous signal.

Header	Program command	Query	Response
TVLINE	TVLINE_a	TVLINE?	a a=5 to 310

■ Value of a 9 to 262 (NTSC EVEN)  
10 to 263 (NTSC ODD)  
5 to 310 (PAL EVEN)  
6 to 310 (PAL ODD)

■ Suffix code None

■ Initial setting a=10

■ Example TVLINE\_10

# TVSFRM

## TVSFRM TV Synchronizing Signal

- **Function** Sets the TV synchronous signal when TV is selected for trigger source.

Header	Program command	Query	Response
TVSFRM	TVSFRM_SW	TVSFRM?	SW sw = VERTICAL, EVEN, ODD

- **Value of sw** VERTICAL: VERTICAL  
EVEN: H-EVEN  
ODD: H-ODD
- **Suffix code** None
- **Initial setting** VERTICAL: VERTICAL
- **Example** TVSFRM\_VERTICAL

---

# TVSTND

## TVSTND TV Type

- **Function** Sets the TV broadcasting system when TV is selected for trigger source.

Header	Program command	Query	Response
TVSTND	TVSTND_SW	TVSTND?	SW sw = PAL, NTSC

- **Value of sw** PAL: PAL  
NTSC: NTSC
- **Suffix code** None
- **Initial setting** NTSC: NTSC
- **Example** TVSTND\_NTSC

# TZONE

## TZONE Expand Zone

- Function Switches the time expand (magnified display) mode to ON or OFF.

Header	Program command	Query	Response
TZONE	TZONE_ <u>sw</u> sw = ON, 1, OFF, 0	TZONE?	SW                      sw = ON, OFF

- Value of sw    ON: ON  
                  1: ON  
                  OFF: OFF  
                  ∅: OFF
- Suffix code    None
- Initial setting OFF: OFF
- Example        TZONE\_ON

---

# TZSP

## TZSP Expand Zone Span

- Function Sets the zone for expanding the time expand (magnified display).

Header	Program command	Query	Response
TZSP	TZSP_ <u>t</u>	TZSP?	t t = 50 to 1000000000 Transfers the data without suffix code as if represented in units of $\mu$ sec

- Value of t     50  $\mu$ sec to 1000 sec
- Suffix code    None: msec  
                  US:  $\mu$ sec  
                  MS: msec  
                  S: sec
- Initial setting 200 msec
- Example        TZSP\_10MS

# TZSTART

## TZSTART Expand Zone Start

- Function Sets the start time of time expand (magnified display).

Header	Program command	Query	Response
TZSTART	TZSTART_t	TZSTART?	t t = -1000000000 to 65500 Transfers the data without suffix code as if represented in units of $\mu$ sec

- Value of t -1000 sec to 65.5 msec
- Suffix code  
None: msec  
US:  $\mu$ sec  
MS: msec  
S: sec
- Initial setting 0 sec
- Example TZSTART\_10MS

---

# UCL?

## UCL? Query Uncal Status

- Function Reads out the UNCAL status.

Header	Program command	Query	Response
UCL?	_____	UCL?	UCL_a a=0,1

- Value of a  
0: NORMAL  
1: During UNCAL
- Example UCL?

# UNC

## UNC                      Uncal Display ON/OFF

■ Function                      Sets whether 'UNCAL' is displayed or not when UNCAL has occurred.

Header	Program command	Query	Response
UNC	UNC <sub>sw</sub> sw=ON, 1, OFF, 0	UNC?	UNC <sub>sw</sub> sw=ON, OFF

■ Value of sw                      ON: ON  
  1: ON  
  OFF: OFF  
  ∅: OFF

■ Suffix code                      None

■ Initial setting                      ON: ON

■ Example                              UNC<sub>ON</sub>

---

# UNT

## UNT                      Unit for Log Scale

■ Function                      Sets one of the display unit systems in LOG scale mode.

Header	Program command	Query	Response
UNT	UNT <sub>a</sub> a=0 to 4	UNT?	UNT <sub>a</sub> a=0 to 4

■ Value of a                      ∅: dBm  
  1: dB $\mu$ V  
  2: dBmV  
  3: V  
  4: dB $\mu$ V (emf)  
  5: W

■ Suffix code                      None

■ Initial setting                      ∅: dBm

■ Example                              UNT<sub>∅</sub>

## VAVG Average

- **Function** Sets averaging to ON, OFF, or number of processing.

Header	Program command	Query	Response
VAVG	VAVG sw VAVG a	VAVG?	a a = 2 to 256

- **Value of a**
  - ∅: dBm
  - 1: dB $\mu$ V
  - 2: dBmV
  - 3: V
  - 4: dB $\mu$ V (emf)
  - 5: W
- **Suffix code** None
- **Initial setting** ∅: dBm
- **Example** VAVG ON  
VAVG 128

# VB

## VB Video Band Width

- **Function** Sets the video bandwidth (same function as VBW).

Header	Program command	Query	Response
VB	VB f VB a	VB?	f f = 1 to 3000000 or OFF Transfers the data without suffix code as if represented in units of Hz

- **Value of f** 1 Hz to 3 MHz (1/3 sequence)  
OFF: OFF
- **Value of a** AUTO: AUTO  
UP: VBW UP  
DN: VBW DOWN
- **Suffix code**
  - f: None: Hz (10<sup>0</sup>) MHz, MZ: MHz (10<sup>6</sup>)
  - HZ: Hz (10<sup>0</sup>) GHZ, GZ: GHz (10<sup>9</sup>)
  - KHZ, KZ: kHz (10<sup>3</sup>)
  - a: None
- **Initial setting** Calculated value when VBW is selected for AUTO
- **Example** VB 300HZ

# VBCOUPLE

## VBCOUPLE Couple Mode

- **Function** Sets the coupled functions (RBW/VBW/SWT) commonly or independently between frequency domain or time domain.

Header	Program command	Query	Response
VBCOUPLE	VBCOUPLE_ sw	VBCOUPLE?	SW sw = COM, IND

- **Value of sw** COM: Common setting  
IND: Independent setting
- **Suffix code** None
- **Initial setting** COM: Common setting (provided the unit already registered is not initialized)
- **Example** VBCOUPLE\_COM

---

# VBR

## VBR VBW/RBW ratio

- **Function** Sets the ratio of video bandwidth and resolution bandwidth when VBW is selected for AUTO.

Header	Program command	Query	Response
VBR	VBR_r	VBR?	r r = 0.0001 to 100

- **Value of r** 0.0001 to 100 (1/3 sequence)
- **Suffix code** None
- **Initial setting** Trace FREQ VBW/RBW RATIO = 1  
Trace TIME VBW/RBW RATIO = 1
- **Example** VBR\_1



## VBW Video Band Width

■ Function Sets the video bandwidth.

Header	Program command	Query	Response
VBW	VBW_a a=0 to 14	VBW?	VBW_a a=0 to 14

■ Value of a

0:	1 Hz	8:	3 Hz
1:	10 Hz	9:	30 Hz
2:	100 Hz	10:	300 Hz
3:	1 kHz	11:	3 kHz
4:	10 kHz	12:	30 kHz
5:	100 kHz	13:	300 kHz
6:	OFF	14:	3 MHz
7:	1 MHz		

■ Suffix code None

■ Initial setting Calculated value when VBW is selected for AUTO

■ Example VBW\_3

## VIEW View

■ Function Stops writing the waveform data.

Header	Program command	Query	Response
VIEW	VIEW_tr	_____	_____

■ Value of tr

TRA:	Trace A (FREQ)
TRTIME:	Trace TIME

■ Suffix code None

■ Example VIEW\_TRB

(Blank)

# SECTION 7

## SAMPLE PROGRAMS

This section gives some examples of the N<sub>88</sub>-Basic program that controls the MS8604A from the NEC-PC9800 personal computer which is used as a controller.

### TABLE OF CONTENTS

7.1	Precautions on Creating the GPIB Program .....	7-3
7.2	Sample Programs .....	7-4
(1)	Initializing MS8604A .....	7-4
(2)	Reading the frequency and level at marker point .....	7-5
(3)	Reading the modulation precision .....	7-7
(4)	Reading the I-Q data .....	7-8
(5)	Reading marker point levels .....	7-13
(6)	Measuring the occupied frequency bandwidth .....	7-14
(7)	Measuring the adjacent channel leakage power .....	7-15
(8)	Measuring spurious harmonics .....	7-16

(Blank)

## 7.1 Precautions on Creating the GPIB Program

Note the following points when writing GPIB control programs.

No.	Precaution	Description
1	Be sure to initialize each device.	<p>There may be a number of the state in which each device is not proper to be actually used due to operation on its own panel or execution of other programs. It is necessary to using individual devices with a prescribed condition resulting from initializing them.</p> <p>Execute the following.</p> <ul style="list-style-type: none"> <li>① Initializing the interface functions (ISET IFC)</li> <li>② Initializing message exchange functions of each device (WBYTE &amp;H3F, &amp;H14)</li> <li>③ Initializing the functions proper to each device (PRE, INI, or *RST)</li> </ul>
2	Turn the device to the remote state of RWLS (Remote With Lockout State).	<p>In a simple remote state, pressing [ Local ] turns the device to the local state. Pressing a panel key in this moment causes device's automatic measurement to function improperly, thus measurement data are likely to turn out unreliable. Set the device to the locally locked out state with WBYTE &amp;H3F, &amp;H11 to prevent it from returning to the local state. (Turn every device to the locally controlled state with WBYTE &amp;H3F, &lt;listener address&gt;, [secondary address], &amp;H01.)</p>
3	Do not send any command (related to the device) other than the INPUT @ statement immediately after sending a query.	<p>If MLA is received when a command other than the INPUT @ statement is sent to the controller before the response to a query is read, the output buffer is cleared, and the response message disappears. For this reason, write the INPUT @ statement in immediate succession to a query.</p>
4	Create a program that avoids an exception processing of the protocol	<p>Avoid stoppage of execution (caused by an error) by means of providing a program with exception-processing section against exceptions that can be foreseen.</p>
5	Confirm the interface functions of each device (subset).	<p>Execution of program does not advance if necessary subset (s) has (have) not been prepared in the device. Be sure to confirm the subset (s) of each device. Also confirm that each device complies with IEEE 488.2.</p>

## 7.2 Sample Programs

### (1) Initializing MS8604A

<Example 1 > Initializes MS8604A

```

10 '-----
20 ' MS8604A GPIB SAMPLE PROGRAM
30 ' INITIALIZE
40 '-----
50 ISET IFC ..... Initializes the interface function
60 ISET REN ..... Sets remote enable to true
70 CMD DELIM=0 ..... Selects CR + LF as the delimiter
80 LET SPA=1 ..... Assigns MS8604A address to variable SPA
90 PRINT @SPA;"INI" ..... Initializes MS8604A
100 END

```

Separate volume " Panel Operation Appendix B " describes the parameters initialized by the above program.

There is a '\*RST' command in another GPIB command for executing initialization. The '\*RST' command is used to execute initialization over a wider range. For the range of initialization level, see SECTION 5. The usage of the 'IP' and 'PRE' commands are identical to the 'INI' command.

For general usage of INI and \*RST, first initialize the MS8604A device functions with the IP or INI command, then use the program commands to set only the functions to be changed. This prevents the MS8604A from being controlled while unnecessary functions are set.

## (2) Reading the frequency and level at marker point

<Example 2> Sets the center frequency to 500 MHz and span to 10 MHz, then displays the frequency and level reading at the peak point on the controller screen when a signal to be measured is received.

```

10 '-----
20 ' MS8604A GPIB SAMPLE PROGRAM
30 ' OUTPUT MKR FREQ&LEVEL
40 '-----
50 ISET IFC
60 ISET REN
70 CMD DELIM=0
80 LET SPA=1
90 PRINT @SPA;"INI"
95 PRINT @SPA;"PNLMD SPECT" ..... Sets to spectrum mode
100 PRINT @SPA;"CNF 500MHZ" ..... Sets the center frequency
110 PRINT @SPA;"SPF 10MHZ" ..... Sets the frequency span
120 PRINT @SPA;"SWP" ..... Executes single sweep
130 PRINT @SPA;"PCF" ..... PEAK→CF
140 PRINT @SPA;"PRL" ..... PEAK→REF
150 PRINT @SPA;"MKS 0" ..... Peak search
160 PRINT @SPA;"MKF?" ..... Queries reading frequency at the marker point
170 INPUT @SPA;FREQ ..... Reads the frequency at the marker point
180 PRINT @SPA;"MKL?" ..... Queries reading level at the marker point
190 INPUT @SPA;LEVEL ..... Reads the level at the marker point
200 PRINT USING "MARKER FREQ=####.###MHz";FREQ/1E+06
210 PRINT USING "          LEVEL=####.## dBm";LEVEL
220 END

```

The center frequency and frequency span are set at line 100 and line 110 respectively. The SWP sweep command at line 120 does not execute the next message unless the sweep is completed. This command thus prevents the peak search and other program lines from being executed before the sweep is completed.

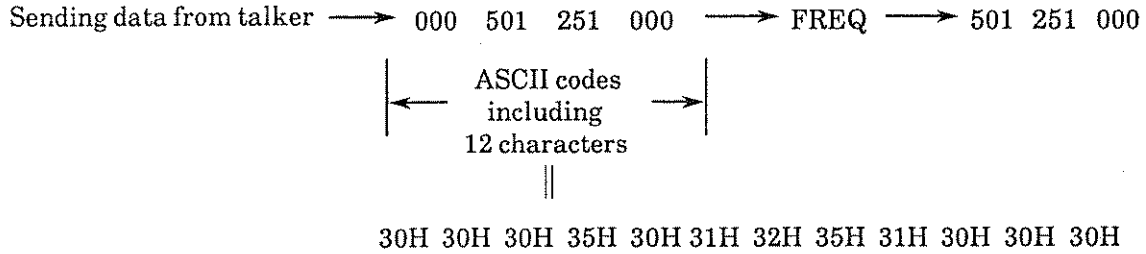
The PCF and PRL commands at lines 130 and 140 operate as follows: the former sets the peak point on the screen to the center frequency, and the latter sets its peak level center frequency to the reference level.

The "MKF?" and "MKL?" at lines 160 and 180 query the frequency and level at the marker point respectively, and the data is read with the INPUT@ statement on the next line. If MLA (My Listen Address) is received when a command other than the INPUT@ statement is sent before the response to a query is read, the output buffer is cleared, and the response message is deleted. For this reason, write the INPUT@ statement immediately after a query.

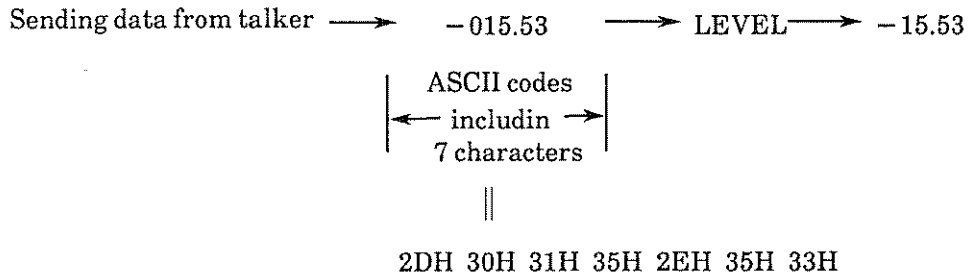
[Sending data]

If the frequency and level at the peak point is 501.251 MHz and -15.53 dBm respectively,

■ **FREQ**



■ **LEVEL**



Program execution result of <Example 2>

MARKER FREQ=501.251MHz  
LEVEL=-15.53 dBm



## (3) Reading the modulation precision

<Example 3> Reads vector and peak vector errors, then display the read errors on the controller screen.

```

100 '-----
110 ' MS8604A GPIB SAMPLE PROGRAM
120 ' OUTPUT VECTOR ERROR
130 '-----
140 ISET IFC
150 ISET REN
160 CMD DELIM=0
170 SPA=1
180 '
190 PRINT @SPA;"INI"
200 PRINT @SPA;"PNLMD TESTER"
210 PRINT @SPA;"SYS PHS"
220 PRINT @SPA;"MEAS MODANAL"
230 PRINT @SPA;"SWP" ..... Executes measurement once.
240 '
250 PRINT @SPA;"VECTERR?" ..... Vector error
260 INPUT @SPA;VSTS,VERR
270 PRINT @SPA;"PVECTERR?" ..... Peak vector error
280 INPUT @SPA;PVSTS,PVERR
290 '
300 PRINT USING "RMS vector error = ###.## % (rms)";VERR
310 PRINT USING "Peak vector error = ###.## % ";PVERR
320 END

```

Lines 200 and 210 switch the system to the PHS system in transmitter tester mode. (If always using the device as a transmitter tester, this switching is not necessary). Line 220 switches the screen to the modulation analysis screen.

Lines 250 and 270 read the vector and peak vector errors. Two data items separated by a comma (,) are output as the response data to the query command. The first data item is the measurement result status. If this value is 0, it indicates normal termination (see to the "GPIB Command Reference"). The second data item is the measurement result.

## (4) Reading the I-Q data

<Example 4-1> Reads all data items of the I-Q constellation.

```

100 '-----
110 ' MS8604A GPIB SAMPLE PROGRAM
120 ' OUTPUT I/Q DATA (ASCII)
130 '-----
140 ISET IFC
150 ISET REN
160 CMD DELIM=0
170 SPA=1
180 NUM=438 ..... Number of data items to be read
190 DIM TRACE(1,NUM) ..... Declaration of array variable TRACE ( )
200 '
210 PRINT @SPA;"INI"
220 PRINT @SPA;"PNLMD TESTER"
230 PRINT @SPA;"SYS PHS"
240 PRINT @SPA;"MEAS CONSTEL"
250 PRINT @SPA;"BIN 0" ..... Sets to the data to be read to ASCII.
260 PRINT @SPA;"SWP" ..... Executes measurement once.
270 '
280 FOR I=0 TO 1
290 FOR J=0 TO NUM-1
300     PRINT @SPA;"XMC? "+STR$(I)+", "+STR$(J)+", 1"
                                     ..... Data query
310     INPUT @SPA;TRC$ ..... Data reading
320     TRACE(I,J)=VAL(TRC$) ..... ASCII to numeric data conversion
330 NEXT J
340 NEXT I
350 '
360 FOR J=0 TO NUM-1
370     PRINT USING "##.####,##.####";TRACE(0,J)/10000,TRACE(1
,J)/10000
380 NEXT J
390 END

```

"BIN\_0" in line 250 is the command that converts read data into ASCII format.

"XMA?", "XMB?", "XMC?", "XMD?", "XME?", and "XMT?" output data in ASCII or binary format. Example 4-2 shows an example of reading binary format data.

<Example 4-2> Reads the I/Q data in binary format.

```

100 '-----
110 ' MS8604A GPIB SAMPLE PROGRAM
120 ' OUTPUT I/Q DATA (BINARY)
130 '-----
140 ISET IFC
150 ISET REN
160 CMD DELIM=0
170 SPA=1
180 NUM=438 ..... Number of read data items
190 DIM TRACE(1,NUM)
200 '
210 PRINT @SPA;"INI"
220 PRINT @SPA;"PNLMD TESTER"
230 PRINT @SPA;"SYS PHS"
240 PRINT @SPA;"MEAS CONSTEL"
250 PRINT @SPA;"BIN 1" ..... Sets the read data to binary format.
260 PRINT @SPA;"TRM 1" ..... Sets the terminator to CR + LF.
270 PRINT @SPA;"SWP" ..... Executes measurement once.
280 '
290 FOR I=0 TO 1
300 PRINT @SPA;"XMC? "+STR$(I)+",0,"+STR$(NUM)
..... Data query
310 WBYTE &H3F,&H5F,&H20,&H41;
..... Specifies the PC9801 to the talker and MS8604A to the listener.
320 FOR J=0 TO NUM-1
330   RBYTE ;UPRBYTE,LWRBYTE ..... Receives the data every two-byte units.
340   TRACE(I,J)=UPRBYTE*256+LWRBYTE ... Converts the data to decimal data.
350   IF UPRBYTE>=128 THEN TRACE(I,J)=TRACE(I,J)-65536!
..... Converts negative data.
360 NEXT J
370 RBYTE ;DMY1,DMY2 ..... Terminator receiving
380 WBYTE &H3F,&H5F; ..... Listener and talker release
390 NEXT I
400 '
410 FOR J=0 TO NUM-1
420   PRINT USING "##.####,##.####";TRACE(0,J)/10000,TRACE(1
,J)/10000
430 NEXT J
440 END

```

The "BIN\_1" at line 250 specifies the data format as the binary format. The terminator indicating that the last data byte transmission is completed is set to CR + LF at line 260. The data query in line 300 is performed after measurement is executed once in line 270. At line 310, the PC9801 (address 0) is specified as a listener, and the MS8604A (address 1) as a talker. At line 330, the 2-byte binary data is read by the PC9801 in sequence from the high-order byte to the low-order byte. At line 340, the 2-byte binary data is converted into decimals to assign them to variable TRACE (1). Line 350 allows data to be correctly read even if the data is negative. Statements from the FOR statement to the NEXT statement in lines 320 to 360 are executed repeatedly for each read data item.

At line 370, the terminators transferred immediately after the last data byte transmission is completed are assigned to dummy variables DMY1 and DMY2. This prevents the CR+LF (LF for TRM\_0) terminators from being assigned elsewhere. Although two variables DMY1 and DMY2 are used here, one variable is used if the terminator is LF. Line 380 is used to release the talker and listener.

Statements from the FOR statement to the NEXT statement in lines 290 to 390 are executed twice to read the I/Q data.

[Interface message at line 310]

```

WBYTE   &H3F, &H5F, &H20, &H41 ;
         [ ]   [ ]   [ ]   [ ]
         UNL   UNT   MLA   MTA

```

UNL=Unlisted : Specifies all devices so that the listener status can be released.

UNT=Untalk : Specifies all devices so that the talker status can be released.

MLA=My Listen Address : If a device receives the listener address code (listener address command -MLA) corresponding to the address (address number) specified for the device (device and controller), it becomes a listener.

The &H20 listener address code specifies a device with address number 0 as listener.

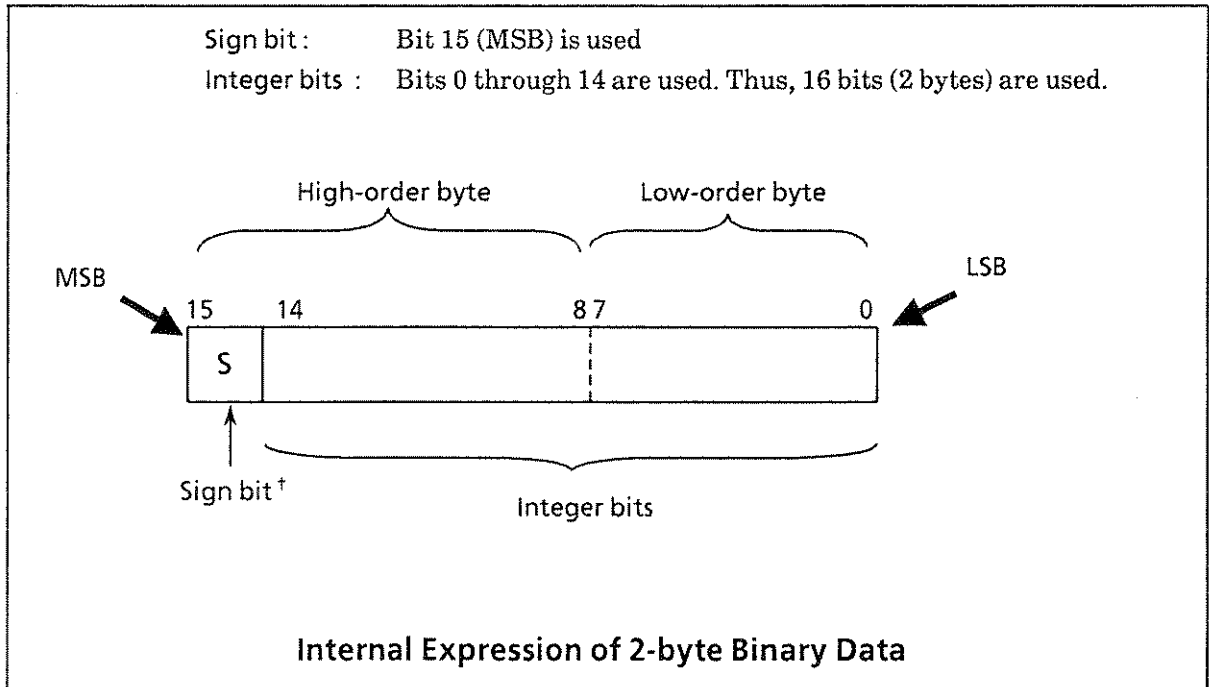
MTA=My Talk Address : If a device receives the talker address code (talker address command -MTA) corresponding to the address (address number) specified for the device (device and controller), it becomes a talker.

The &H41 talker address code specifies a device with address number 1 as talker.

[Transferring 2-byte binary data]

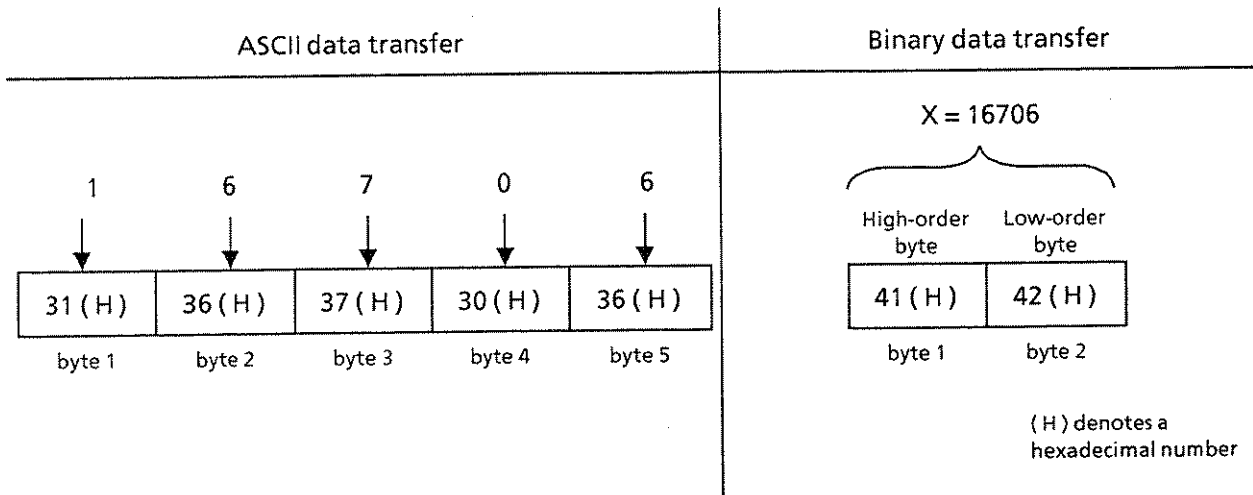
Two-byte binary data can lie in the range of the 65536 integers from  $-32768$  to  $32767$  as shown below, and each is sent out in sequence from the high-order byte to the low-order byte.

16-Bit Binary	With Sign	No Sign
1000000000000000	-32768	32768
1000000000000001	-32767	32769
1000000000000010	-32766	32770
1111111111111101	-3	65533
1111111111111110	-2	65534
1111111111111111	-1	65535
0000000000000000	0	0
0000000000000001	1	1
0000000000000010	2	2
0000000000000011	3	3
0111111111111101	32765	32765
0111111111111110	32766	32766
0111111111111111	32767	32767

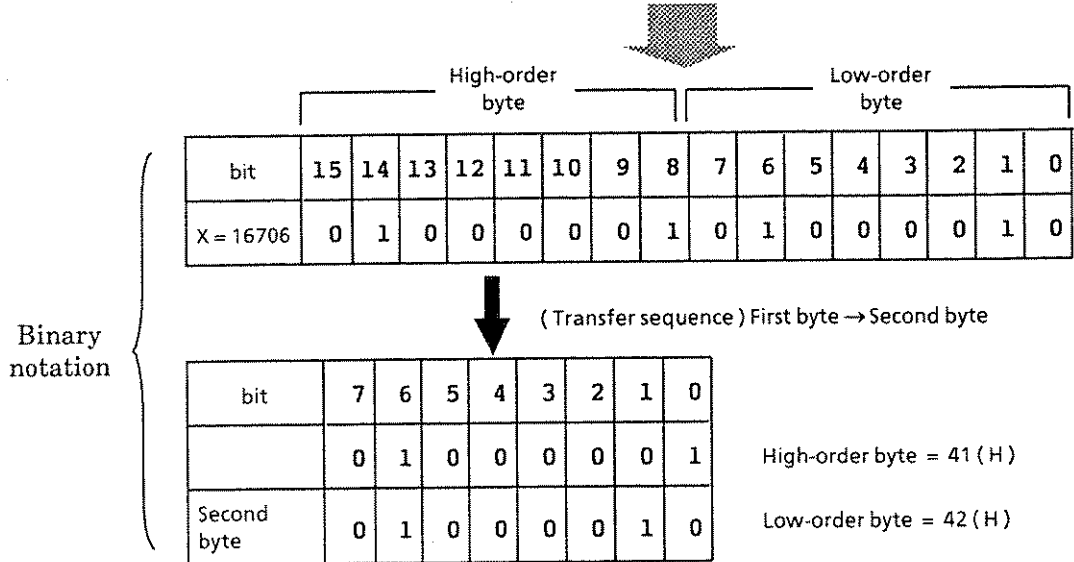


† When a negative integer is stored in a numeric variable, its MSB is set to 1 to indicate that it is negative. The negative value is also stored in a numeric variable as a two's-complement.

The transfer of integer 16706 as ASCII data and as binary data is compared below. 5 bytes are necessary for ASCII, whereas only 2 bytes are necessary for binary and no data format conversion is required. Thus, the latter is often used for high speed transfer.



$$16706 (D) = 4 \times 16^3 + 1 \times 16^2 + 4 \times 16^1 + 2 \times 16^0$$



## (5) Reading marker point levels

<Example 5> Reads levels of the first and second symbols.

```

100 '-----
110 ' MS8604A GPIB SAMPLE PROGRAM
120 ' OUTPUT RF POWER(MARKER)
130 '-----
140 ISET IFC
150 ISET REN
160 CMD DELIM=Ø
170 SPA=1
180 '
190 PRINT @SPA;"INI"
200 PRINT @SPA;"PNLMD TESTER"
210 PRINT @SPA;"SYS PHS"
220 PRINT @SPA;"MEAS RFPWR"
230 PRINT @SPA;"LVLREL ON" ..... Sets this item to a relative value.
240 PRINT @SPA;"SWP" ..... Executes measurement once.
250 '
260 PRINT @SPA;"MKRS 1" ..... Sets the marker to the first symbol.
270 PRINT @SPA;"MKL?" ..... Marker level query
280 INPUT @SPA;MKR1
290 PRINT @SPA;"MKRS 2" ..... Sets the marker to the second symbol.
300 PRINT @SPA;"MKL?" ..... Marker level query
310 INPUT @SPA;MKR2
320 '
330 PRINT USING "1 symbol = ###.## dB";MKR1
340 PRINT USING "2 symbol = ###.## dB";MKR2
350 END

```

Line 230 sets the relative value display. Lines 260 and 290 move the markers to the specified symbol locations.

Use this program to read the level change between symbols at the rising and falling edges of a burst.

## (6) Measuring the occupied frequency bandwidth

<Example 6> Reads the mean value of 16 measurements in HIGH SPEED mode.

```

100 '-----
110 ' MS8604A GPIB SAMPLE PROGRAM
120 ' OUTPUT OCC.BW
130 '-----
140 ISET IFC
150 ISET REN
160 CMD DELIM=0
170 SPA=1
180 '
190 PRINT @SPA;"INI"
200 PRINT @SPA;"PNLMD TESTER"
210 PRINT @SPA;"SYS PHS"
220 PRINT @SPA;"MEAS OBW,HIGH"
230 PRINT @SPA;"*CLS" ..... Clears the status register.
240 PRINT @SPA;"AVR 16" ..... Sets the average count to 16.
250 PRINT @SPA;"STORAGE AVG" ..... Sets the average mode.
260 PRINT @SPA;"SWP" ..... Executes measurement once.
270 '
280 WHILE ENDSTS<>17 ..... Repeats processing until averaging terminates.
290 PRINT @SPA;"ESR2?" ..... END-STATUS query
300 INPUT @SPA;ENDSTS
310 WEND
320 '
330 PRINT @SPA;"OCCBW?"
340 INPUT @SPA;OSTS,OCCBW
350 '
360 PRINT USING "Occupied bandwidth = ###.# kHz";OCCBW/1000
370 END

```

Line 230 clears the status registers. Line 240 sets the average count to 16 and line 250 switches the mode to the average mode. After SWP in line 260 executes measurement once, the program executes measurement for the average count, then terminates.

Lines 280 to 310 check the average count.



## (7) Measuring the adjacent channel leakage power

<Example 7> Reads the mean power in HIGH SPEED mode and displays the read power on the controller screen.

```

100 '-----
110 ' MS8604A GPIB SAMPLE PROGRAM
120 ' OUTPUT ADJ.CH
130 '-----
140 ISET IFC
150 ISET REN
160 CMD DELIM=0
170 SPA=1
180 '
190 PRINT @SPA;"INI"
200 PRINT @SPA;"PNLMD TESTER"
210 PRINT @SPA;"SYS PHS"
220 PRINT @SPA;"MEAS ADJ,HIGH"
                ..... Sets the adjacent channel leakage power to HIGH SPEED.
230 PRINT @SPA;"UNIT DBM" ..... Sets the units to dBm.
240 PRINT @SPA;"SWP" ..... Executes measurement once.
250 '
260 PRINT @SPA;"MEANPWR? LOW2,DBM"
                ..... Lower next adjacent channel leakage power query
270 INPUT @SPA;L2STS,LOW2
280 PRINT @SPA;"MEANPWR? LOW1,DBM"
                ..... Lower adjacent channel leakage power query
290 INPUT @SPA;L1STS,LOW1
300 PRINT @SPA;"MEANPWR? UP1,DBM"
                ..... Upper adjacent channel leakage power query
310 INPUT @SPA;U1STS,UP1
320 PRINT @SPA;"MEANPWR? UP2,DBM"
                ..... Upper next adjacent channel leakage power query
330 INPUT @SPA;U2STS,UP2
340 '
350 PRINT "Adjacent channel power"
360 PRINT USING "      Mean power (-900kHz) = ###.## dBm";LOW2
370 PRINT USING "      (-600kHz) = ###.## dBm";LOW1
380 PRINT USING "      ( 600kHz) = ###.## dBm";UP1
390 PRINT USING "      ( 900kHz) = ###.## dBm";UP2
400 END

```

## (8) Measuring spurious harmonics

<Example 8> Reads the spurious harmonics in SPOT mode and displays them on the controller screen.

```

100 '-----
110 ' MS8604A GPIB SAMPLE PROGRAM
120 ' OUTPUT SPURIOUS
130 '-----
140 ISET IFC
150 ISET REN
160 CMD DELIM=0
170 SPA=1
180 '
190 PRINT @SPA;"INI"
200 PRINT @SPA;"PNLMD TESTER"
210 PRINT @SPA;"SYS PHS"
220 PRINT @SPA;"MEAS SETTABLE" ..... Sets the frequency setting screen.
230 PRINT @SPA;"SPUFREQ HRM" ..... Set up to the fifth harmonic.
240 PRINT @SPA;"MEAS SPURIOUS, SPOT"
      ..... Sets spot measurement of the spurious harmonics.
250 PRINT @SPA;"UNIT DBM"
260 PRINT @SPA;"SWP" ..... Executes measurement once.
270 '
280 PRINT @SPA;"SPULVL? F1" ..... Spurious harmonics query
290 INPUT @SPA;F1STS,F1LVL
300 PRINT @SPA;"SPULVL? F2" ..... Spurious harmonics query
310 INPUT @SPA;F2STS,F2LVL
320 PRINT @SPA;"SPULVL? F3" ..... Spurious harmonics query
330 INPUT @SPA;F3STS,F3LVL
340 PRINT @SPA;"SPULVL? F4" ..... Spurious harmonics query
350 INPUT @SPA;F4STS,F4LVL
360 '
370 PRINT "Spurious emissions"
380 PRINT USING "      F1 = ###.## dBm";F1LVL
390 PRINT USING "      F2 = ###.## dBm";F2LVL
400 PRINT USING "      F3 = ###.## dBm";F3LVL
410 PRINT USING "      F4 = ###.## dBm";F4LVL
420 END

```

Line 220 switches to the frequency setting screen and line 230 sets the frequency for the spurious harmonics two to five times higher than the frequency on the parameter setting screen.

Line 240 switches to the spurious harmonics measurement screen. Lines 280 to 350 examine the spurious harmonics.

# SECTION 8

## WAVEFORM DATA STORAGE FORMAT

This section describes the storage format of waveform data fetched by the PTA or external computer. The use examples are described in NEC 9800 series N<sub>88</sub>-BASIC.

### TABLE OF CONTENTS

8.1	Notes on Fetching the Waveform Data .....	8-3
8.2	Waveform Data Storage Format .....	8-4
	(1) XMA, XMT (Spectrum screen) .....	8-4
	(2) XMB (STANDARD method of the occupied frequency bandwidth and adjacent channel leakage power measurement screen) .....	8-5
	(3) XMC (Constellation screen) .....	8-6
	(4) XMD (Power measurement screen) .....	8-7
	(5) XME (HIGH SPEED method of the occupied frequency bandwidth screen) .....	8-8

**(Blank)**

## 8.1 Notes on Fetching the Waveform Data

### (1) Fetching screen

- The waveform data to be fetched needs to be displayed on the screen.
- Switch to SINGLE measurement mode before fetching the waveform data, and check that the measurement terminates. If CONTINUOUS measurement mode is set or measurement does not terminate, the correct data cannot be fetched.

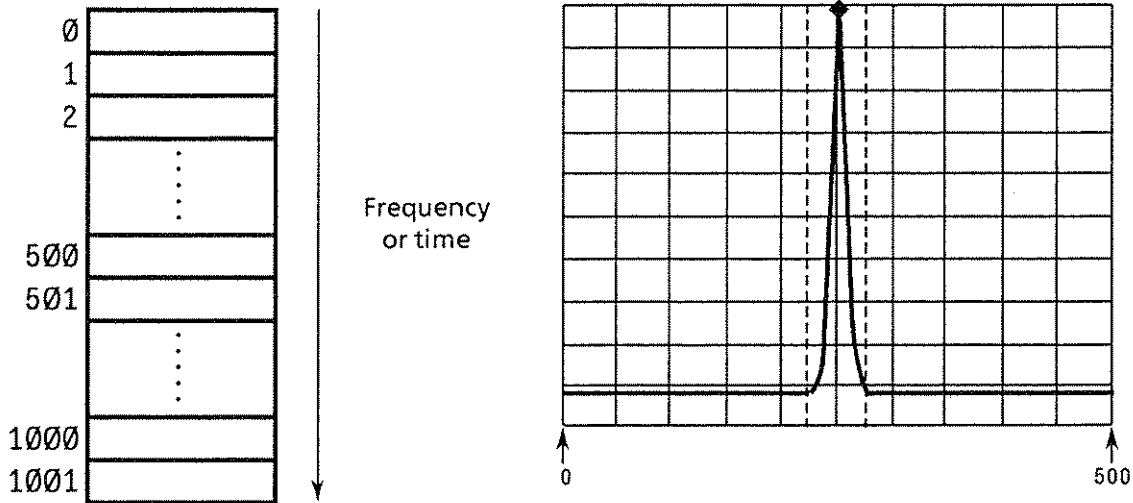
### (2) Response data

- When fetching two or more data items together, commas (,) are output as data separators.
- The query command format is designed so that it can fetch all data items together. However, the number of data items actually fetched depends on the restrictions of the PTA or external computer.

## 8.2 Waveform Data Storage Format

### (1) XMA, XMT

#### a) Format



- The number of data items depends on the set data point value.  
 In NORMAL mode: 501 points (0 to 500)  
 In DOUBLE mode: 1002 points (0 to 1001)
- Correspondences to screen displays are as follows:  
 In NORMAL mode: Waveform data  $Y_n$  is displayed as the  $n$ th data item on the screen.  
 In DOUBLE mode: Waveform data  $\frac{Y_{2n} + Y_{2n+1}}{2}$  is displayed as the  $n$ th data item on the screen.

#### b) Range

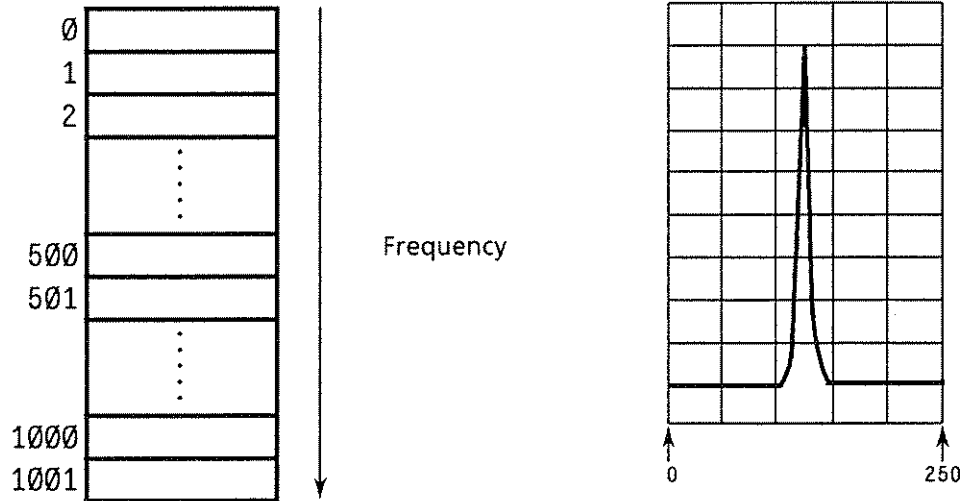
- In LOG scale mode: Displays an integer value in 0.01 dBm units (1 dBm = 100).
- In LIN scale mode: Displays  $\frac{\text{Voltage value (V)}}{\text{Reference level (V)}} \times 10000$ .

#### c) Use examples

```
PRINT _@1; "XMA? _@0, 1"
INPUT _@1; LVL
```

## (2) XMB

## a) Format



- The number of data items depends on the set data point value.

In NORMAL mode: 501 points (0 to 500)

In DOUBLE mode: 1002 points (0 to 1001)

- Correspondences to screen displays are as follows:

In NORMAL mode: Waveform data  $\frac{Y_{2n} + Y_{2n+1}}{2}$  is displayed as the nth data item on the screen.

In DOUBLE mode: Waveform data  $\frac{Y_{4n} + Y_{4n+1} + Y_{4n+2} + Y_{4n+3}}{4}$  is displayed as the nth data item on the screen.

## b) Range

- Displays an integer value in 0.01 dBm units (1 dBm = 100).

## c) Use examples

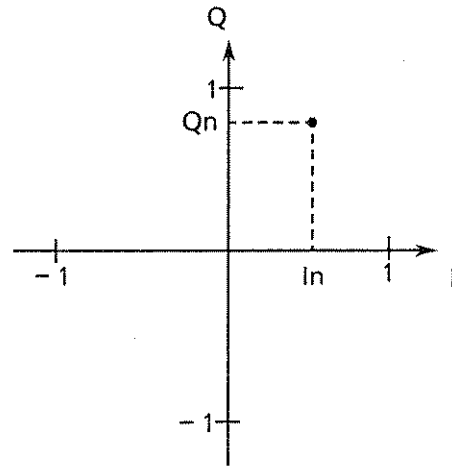
```
PRINT_@1;"XMB_1,2"
```

```
INPUT_@1;LVL1,LVL2
```

(3) XMC

a) Format

	I	Q	
0			Origin offset symbol point
1			
2			Interpolation data
3			
4			
5			Symbol point
	⋮	⋮	
n-3			Interpolation data
n-2			
n-1			
n			Symbol point



- Stored with the two-dimensional I-Q data.
- n depends on the measurement signal.
- Correspondences to screen displays are as follows (INTERPOLATION mode):  
 NON: Dots represent symbol points (1, 5, ..., n).  
 LINEAR: A symbol point is enclosed in a linear line frame (1-5-...-n).  
 4POINT: Dotted lines connect a symbol point and interpolation data (1-2-...-n).

b) Range

- Displays an integer value in 0.0001 units (ideal signal 1 = 100000).

c) Use examples

- Fetch origin offsets.

```
PRINT_@1;"XMC?_0,0,1"
INPUT_@1;I0
PRINT_@1;"XMC?_1,0,1"
INPUT_@1;Q0
```



## (4) XMD

## a) Format

$\emptyset$		- 120.00 symbol
1		- 119.75 symbol
	⋮	
480		0.00 symbol
481		0.25 symbol
	⋮	
$4(n-\emptyset.25)+480$		n-0.25 symbol
$4n+480$		n symbol

- n depends on the measuring system (PDC, PHS, 960, NADC, or 1020).
- The data corresponds to the horizontal axis scale of the screen display (RF POWER screen) in a one-to-one ratio.

## b) Range

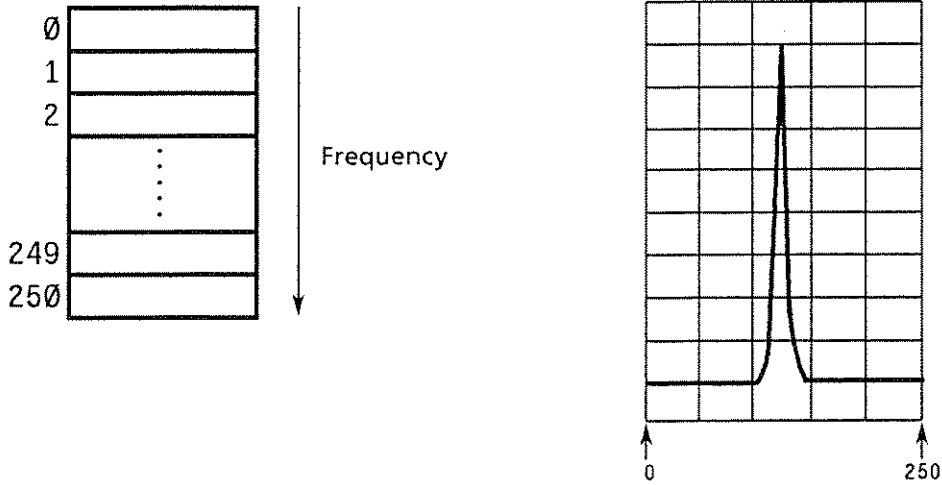
- Displays an integer value in 0.01 dBm units (1 dBm = 100).

## c) Use examples

```
PRINT_@1;"XMD?_480,1"
INPUT_@1;LVL
```

(5) XME

a) Format



- The number of data items is fixed to 251 points.
- The data corresponds to the horizontal axis scale of the screen display (HIGH SPEED method of the occupied frequency bandwidth screen) in one-to-one ratio.

b) Range

- An integer in a 0.01 dBm unit (1 dBm = 100) represents the range.

c) Use examples

```
PRINT _@1; "XME? _0, 0"
```

```
INPUT _@1; LVL
```

APPENDIX A ASCII\*CODE TABLE

BITS B4 B3 B2 B1	0 0 0 0		0 0 0 1		0 1 0 1		1 0 0 1		1 1 0 1		1 1 1 1				
	CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE		
0 0 0 0	0 NUL	10 DLE	20 SP	30 0	40 @	50 P	60 ,	70 p	80	90	100	110			
0 0 0 1	1 SOH	11 DC1	21 !	31 1	41 A	51 Q	61 a	71 q	81	91	101	111			
0 0 1 0	2 STX	12 DC2	22 "	32 2	42 B	52 R	62 b	72 r	82	92	102	112			
0 0 1 1	3 ETX	13 DC3	23 #	33 3	43 C	53 S	63 c	73 s	83	93	103	113			
0 1 0 0	4 EOT	14 DC4	24 \$	34 4	44 D	54 T	64 d	74 t	84	94	104	114			
0 1 0 1	5 ENO	15 NAK	25 %	35 5	45 E	55 U	65 e	75 u	85	95	105	115			
0 1 1 0	6 ACK	16 SYN	26 &	36 6	46 F	56 V	66 f	76 v	86	96	106	116			
0 1 1 1	7 BEL	17 ETB	27 ' (	37 7	47 G	57 W	67 g	77 w	87	97	107	117			
1 0 0 0	8 BS	18 CAN	28 (	38 8	48 H	58 X	68 h	78 x	88	98	108	118			
1 0 0 1	9 HT	19 EM	29 )	39 9	49 I	59 Y	69 i	79 y	89	99	109	119			
1 0 1 0	10 LF	20 SUB	30 *	40 :	50 J	60 Z	70 j	80 z	90	100	110	120			
1 0 1 1	11 VT	21 ESC	31 ÷	41 ;	51 K	61 [	71 k	81 {	91	101	111	121			
1 1 0 0	12 FF	22 FS	32 ' <	42 L	52 \	62	72	82 i	92	102	112	122			
1 1 0 1	13 CR	23 GS	33 - =	43 M	53 ]	63 m	73 }	83 }	93	103	113	123			
1 1 1 0	14 SO	24 RS	34 . >	44 N	54 ^	64 n	74 ~	84	94	104	114	124			
1 1 1 1	15 SI	25 US	35 / ?	45 O	55 _	65 o	75 (RUBOUT (DEL))	85	95	105	115	125			
	Address command	Universal command	Listen address	Talk address	Secondary address or command										

KEY octal 25 PPU GPIB code  
 hex 15 NAK ASCII character  
 21 decimal

\* USA Standard Code for Information Interchange



Notes :  
 ① MSG = INTERFACE MESSAGE (Sent by ATN of True: Low level.)  
 ② b<sub>1</sub>=DIO1 ..... b<sub>7</sub>=DIO7 (b<sub>1</sub> through b<sub>7</sub> correspond to DIO1 to DIO7 sequence.)

Table of Address Assignments

Address character		Address switch setting										Primary address	Factory address set device
Talk	Listen	5	4	3	2	1					10 Decimal		
b <sub>7</sub> b <sub>6</sub>	b <sub>7</sub> b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>							
1 0	0 1	↓	↓	↓	↓	↓							
@	SP	0	0	0	0	0					0		
A	!	0	0	0	0	1					1		
B	"	0	0	0	0	1					2		
C	#	0	0	0	0	1					3		
D	\$	0	0	0	1	0					4		
E	%	0	0	0	1	0					5		
F	&	0	0	0	1	1					6		
G	'	0	0	0	1	1					7		
H	(	0	1	0	0	0					8		
I	)	0	1	0	0	1					9		
J	*	0	1	0	1	0					10		
K	+	0	1	0	1	1					11		
L	,	0	1	1	0	0					12		
M	-	0	1	1	0	1					13		
N	.	0	1	1	1	0					14		
O	/	0	1	1	1	1					15		
P	0	1	0	0	0	0					16		
Q	1	1	0	0	0	1					17		
R	2	1	0	0	1	0					18		
S	3	1	0	0	1	1					19		
T	4	1	0	1	0	0					20		
U	5	1	0	1	0	1					21		
V	6	1	0	1	1	0					22		
W	7	1	0	1	1	1					23		
X	8	1	1	0	0	0					24		
Y	9	1	1	0	0	1					25		
Z	:	1	1	0	1	0					26		
[	;	1	1	0	1	1					27		
\	<	1	1	1	0	0					28		
]	=	1	1	1	1	0					29		
^	>	1	1	1	1	1					30		
?	--	1	1	1	1	1					31	UNL, UNT	

Table of Interface Message Group

D	D	D	D	D	D	D	D	D	D	D	D	Interface message group (G)
0	0	0	0	0	0	0	0	0	0	0	0	Addressed command G
1	0	0	0	0	0	0	0	0	0	0	1	Universal command G
1	0	0	0	0	0	0	0	0	0	0	1	Listen address G
1	0	0	0	0	0	0	0	0	0	0	1	Unlisten (UNL)
1	0	0	0	0	0	0	0	0	0	0	1	Talker address G
1	0	0	0	0	0	0	0	0	0	0	1	Untalk (UNT)
1	0	0	0	0	0	0	0	0	0	0	1	Secondary command G

(Blank)

## APPENDIX B COMPARISON TABLE OF CONTROLLERS' GPIB INSTRUCTIONS

Function	Controller			
	PACKET V (Anritsu)	PC9800 (NEC)	IBM-PC	HP9000 series
Outputs data to a device	WRITE @ device number:data	PRINT @ listener address; data	CALL IBWRT( )	OUTPUT device selector;data
Output binary data to a device	BIN WRITE @ device number:data	WBYTE command; data		
Assigns data entered from a device to a variable	READ @ device number:variable	INPUT @ talker address,listener address;variable LINE INPUT @ talker address,listener address;variable	CALL IBRD( )	ENTER device selector; variable
Assigns binary data entered from a device to a variable	BIN READ @ device number:variable	RBYTE command;variable		
Initializes an interface	IFC @ select code	ISET IFC	CALL IBSIC( )	ABORT select code
Turns REN line on	REN @ select code	ISET REN	CALL IBSRE( )	REMOTE device selector (select code)
Turns REN line off	LCL @ select code (sets all devices local) LCL @ device number (sets only specified devices to listeners, and sends out GTL command)	IRESET REN  WBYTE &H3F,listener address, secondary address, &H01;	CALL IBSRE( )  CALL IBLOC( )	LOCAL device selector (select code) LOCAL device selector (select code + primary address)

Function	Controller			
	PACKET V (Anritsu)	PC9800 (NEC)	IBM-PC	HP9000 series
Outputs interface message(s) and data	COMMAND @ select code: character string for message [;data]		CALL IBCMD( ) CALL IBCMDA( ) (asynchronous)	SEND select code;message string
Triggers a specified device	TRG @ device number	WBYTE &H3F, listener address, secondary address, &H08;	CALL IBTRG( )	TRIGGER device selector
Initializes devices	DCL @ select code (all devices bearing a specified select code) DCL @ device number (specified devices only)	WBYTE &H3F,&H14;  WBYTE &H3F, listener address, secondary address,&H04;	  CALL IBCLR( )	CLEAR device selector (select code)  CLEAR device selector (select code + primary address)
Disables a device from being switched over from remote to local	LLO @ select code	WBYTE &H3F,&H11;		LOCAL LOCKOUT
Transfers control to a specified device	PCT @ device number	WBYTE talker address,&H09;	CALL IBPCT ( )	PASS CONTROL
Sends out a service request	SRQ @ select code	ISSET SRQ	CALL IBRSV( )	REQUEST select code
Performs serial polling	STATUS @ device number	POLL	CALL IBRSP( )	SPOLL (device selector) (function)
Sets a terminator code	TERM IS	CMD DELIM	CALL IBEOS( ) CALL IBEOT( )	
Sets a limit value for checking a timeout		CMD TIMEOUT	CALL IBTOM( )	





