

**MS2602A**  
**Spectrum Analyzer**  
**Operation Manual Vol.3**  
**(GPIB Remote Control)**

**Ninth Edition**

**Read this manual before using the equipment.**  
**Keep this manual with the equipment.**

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

**ANRITSU CORPORATION**

JUN.  
1999

Document No.: M-W0653AE-9.0

## Safety Symbols

To prevent the risk of personal injury or loss related to equipment malfunction, Anritsu Corporation uses the following safety symbols to indicate safety-related information. Insure that you clearly understand the meanings of the symbols BEFORE using the equipment.

### Symbols used in manual

- DANGER** This indicates a very dangerous procedure that could result in serious injury or death if not performed properly.
- WARNING** This indicates a hazardous procedure that could result in serious injury or death if not performed properly.
- CAUTION** This indicates a hazardous procedure or danger that could result in light-to-severe injury, or loss related to equipment malfunction, if proper precautions are not taken.

### Safety Symbols Used on Equipment and in Manual

(Some or all of the following five symbols may not be used on all Anritsu equipment. In addition, there may be other labels attached to products which are not shown in the diagrams in this manual.) The following safety symbols are used inside or on the equipment near operation locations to provide information about safety items and operation precautions. Insure that you clearly understand the meanings of the symbols and take the necessary precautions BEFORE using the equipment.



This indicates a prohibited operation. The prohibited operation is indicated symbolically in or near the barred circle.



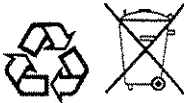
This indicates an obligatory safety precaution. The obligatory operation is indicated symbolically in or near the circle.



This indicates warning or caution. The contents are indicated symbolically in or near the triangle.



This indicates a note. The contents are described in the box.



These indicate that the marked part should be recycled.

MS2602A Spectrum Analyzer  
Operation Manual Vol.3 (GPIB Remote Control)

December 1992 (First Edition)  
April 1999 (Ninth Edition)

Copyright © 1992–1999 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

## Strategies for the Year 2000 problem

This equipment does not suffer from the Year 2000 problem (Note). However, we recommend that you should check whether your application software, in which this equipment is used as a part, has any Year 2000 bugs because this equipment indicates a year using its last two digits.

### **Note: The Year 2000 problem**

The Year 2000 problem is defined as follow:

Systems computing date data has represented the year as a two-digit number (for example, 1997 is represented as "97"). When the Year 2000 comes or data of 2000 or later is to be computed, the year is indicated as "00", "01" and so on, which causes troubles in comparison of dates, sorting using dates and computation of date data.

Another problem is that Year 2000, a leap year, is not recognized as so.



## For Safety



Repair

**WARNING** 

Falling Over

### DANGER

NEVER touch parts where the label shown on the left is attached. Such parts have high voltages of at least 1 kV and there is a risk of receiving a fatal electric shock.

### WARNING

1. Always refer to the operation manual when working near locations at which the alert mark shown on the left is attached. If the operation, etc., is performed without heeding the advice in the operation manual, there is a risk of personal injury. In addition, the equipment performance may be reduced.  
Moreover, this alert mark is sometimes used with other marks and descriptions indicating other dangers.
2. This equipment cannot be repaired by the user. DO NOT attempt to open the cabinet or to disassemble internal parts. Only Anritsu-trained service personnel or staff from your sales representative with a knowledge of electrical fire and shock hazards should service this equipment. There are high-voltage parts in this equipment presenting a risk of severe injury or fatal electric shock to untrained personnel. In addition, there is a risk of damage to precision parts.
3. This equipment should be used in the correct position. If the cabinet is turned on its side, etc., it will be unstable and may be damaged if it falls over as a result of receiving a slight mechanical shock.

## For Safety

### CAUTION

#### Cleaning



#### Check Terminal



1. Keep the power supply and cooling fan free of dust.
  - Clean the power inlet regularly. If dust accumulates around the power pins, there is a risk of fire.
  - Keep the cooling fan clean so that the ventilation holes are not obstructed. If the ventilation is obstructed, the cabinet may overheat and catch fire.
2. Use two or more people to lift and move this equipment, or use a trolley. There is a risk of back injury, if this equipment is lifted by one person.
3. Never input a signal of more than DC 0 V between the measured terminal and ground. Input of an excessive signal may damage the equipment.

Refer to the Section 2 in Basic operating instructions of Operation manual, except the above descriptions.

# Equipment Certificate

Anritsu Corporation certifies that this equipment was tested before shipment using calibrated measuring instruments with direct traceability to public testing organizations recognized by national research laboratories including the Electrotechnical Laboratory, the National Research Laboratory of Metrology and the Communications Research Laboratory, and was found to meet the published specifications.

## Anritsu Warranty

Anritsu Corporation will repair this equipment free-of-charge if a malfunction occurs within 1 year after shipment due to a manufacturing fault, provided that this warranty is rendered void under any or all of the following conditions.

- The fault is outside the scope of the warranty conditions described in the operation manual.
- The fault is due to mishandling, misuse, or unauthorized modification or repair of the equipment by the customer.
- The fault is due to severe usage clearly exceeding normal usage.
- The fault is due to improper or insufficient maintenance by the customer.
- The fault is due to natural disaster including fire, flooding, earthquake, etc.
- The fault is due to use of non-specified peripheral equipment, peripheral parts, consumables, etc.
- The fault is due to use of a non-specified power supply or in a non-specified installation location.

In addition, this warranty is valid only for the original equipment purchaser. It is not transferable if the equipment is resold.

Anritsu Corporation will not accept liability for equipment faults due to unforeseen and unusual circumstances, nor for faults due to mishandling by the customer.

## Anritsu Corporation Contact

If this equipment develops a fault, contact Anritsu Corporation or its representatives at the address in this manual.

---

'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.



## ABOUT DETECTION MODE

This instrument is a spectrum analyzer which uses a digital storage system. The spectrum analyzer makes level measurements in frequency steps obtained by dividing the frequency span by the number of measurement data points (501 or 1002). This method of measurement cannot detect the signal peak level if the spectrum of a received signal is narrower than these frequency steps.

To resolve this problem, this instrument usually operates in positive peak detection mode. In this mode, the highest level within the frequency range between the sample points can be held and traced.

Positive peak detection mode should be used for almost all measurements including normal signal level measurement, pulsed noise analysis, and others. It is impossible to measure the signal level accurately in sample detection mode or in negative peak detection mode.

Use of sample detection mode is restricted to random noise measurement, occupied frequency bandwidth measurement for analog communication systems, and adjacent-channel leakage power measurement, etc.

Measurement item	Detection mode
● Normal signal level .....	POS PEAK
● Random noise .....	SAMPLE
● Pulsed noise .....	POS PEAK
● Occupied frequency bandwidth, adjacent-channel leakage power (for analog communication systems) .....	SAMPLE
● Occupied frequency bandwidth, adjacent-channel leakage power (for digital communication systems) .....	POS PEAK or SAMPLE

When a detection mode is specified as one of the measurement methods, make the measurement in the specified detection mode.

---

## 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.

---

## 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.*

# CE Marking

Anritsu affix the CE Conformity Marking on the following product (s) in accordance with the Council Directive 93/68/EEC to indicate that they conform with the EMC directive of the European Union (EU).

## CE Conformity Marking



### 1. Product Name/Model Name

Product Name: Spectrum Analyzer

Model Name: MS2602A

### 2. Applied Directive

EMC: Council Directive 89/336/EEC

Safety: Council Directive 73/23/EEC

### 3. Applied Standards

EMC:

Electromagnetic radiation:

EN55011 (ISM, Group 1, Class A equipment)

Immunity:

EN50082-1

Performance criteria\*

IEC801-2 (ESD) 4 kVCD, 8 kVAD

B

IEC801-3 (Rad.) 3 V/m

A

IEC801-4 (EFT) 1 kV

B

\*: Performance criteria

A: No performance degradation or function loss

B: Self-recovered temporary degradation of performance or temporary loss of function

Harmonic current emissions:

EN61000-3-2 (Class A equipment)

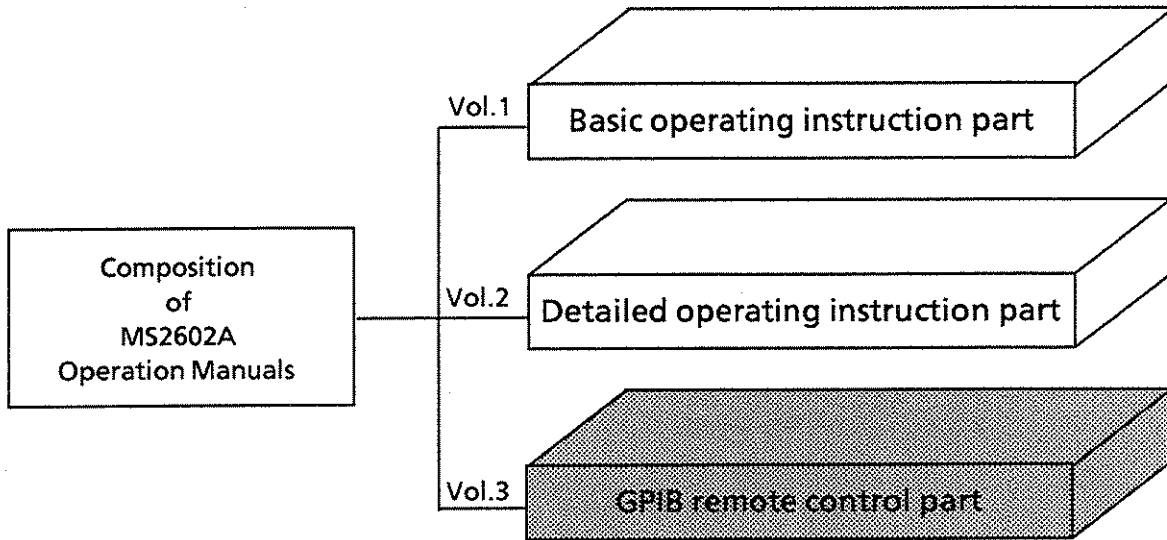
Safety: EN61010-1 (Installation Category II, Pollution Degree 2)

**(Blank)**

# ABOUT THIS MANUAL

## (1) Composition of MS2602A Operation Manuals

The MS2602A Spectrum Analyzer operation manuals of the standard type are composed of the following three documents. Use them properly according to the usage purpose.



Basic operating instruction part:

**Basic Operating Instructions:** Provides information on the MS2602A outline, preparation before use, panel description, basic operation, performance tests, calibration, storage / transportation, and quick reference for functions other than PTA.

Detailed operating instruction part:

**Detailed Operating Instructions:** Provides information on the detailed operating instructions that expand on the panel description and basic operation in the Basic Operating Instruction Part of the separate Operation Manual. An index is available for the function menu to facilitate quick reference.

GPIB remote control part:

Provides information on the MS2602A remote control which conforms with IEEE488.2 standards. To assist creating GPIB programs, this manual gives examples of N<sub>88</sub> Basic language programs that run on the Nippon Electric Co.( NEC ) PC9800 series of personal computers.

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

The GPIB Basic Guide is sold separately in addition to the above GPIB operation manual. It is composed of two parts: GPIB Basic Knowledge, and GPIB Control statements in the ANRITSU PACKET V BASIC.

# TABLE OF CONTENTS

For Safety .....	iii
<b>SECTION 1 GENERAL .....</b>	<b>1-1</b>
1.1 GENERAL .....	1-3
1.1.1 Functions of GPIB .....	1-3
1.1.2 Functions of the GPIB with two ports .....	1-3
1.1.3 Examples of system configuration using GPIB 1 / GPIB 2 .....	1-4
1.1.4 Specifications .....	1-5
<b>SECTION 2 CONNECTING THE BUS AND SETTING THE ADDRESS .....</b>	<b>2-1</b>
2.1 Connecting Devices with GPIB Cables .....	2-3
2.2 Checking and Setting GPIB Address .....	2-4
2.2.1 Panel key layout .....	2-4
2.2.2 Key operation flow .....	2-4
2.2.3 Checking the address .....	2-5
2.2.4 Setting the address for GPIB 1 .....	2-5
<b>SECTION 3 DEVICE MESSAGE FORMAT .....</b>	<b>3-1</b>
3.1 General Description .....	3-3
3.1.1 Program message format .....	3-3
3.1.2 Response message format .....	3-9
<b>SECTION 4 STATUS STRUCTURE .....</b>	<b>4-1</b>
4.1 IEEE488.2 Standard Status Model .....	4-3
4.2 Status Byte ( STB ) Register .....	4-5
4.2.1 ESB and MAV summary messages .....	4-5
4.2.2 Device-dependent summary messages .....	4-6
4.2.3 Reading and clearing STB register .....	4-7
4.3 Service Request ( SRQ ) Enabling Operation .....	4-8
4.4 Standard Event Status Register .....	4-9
4.4.1 Bit definition of Standard Event Status Register .....	4-9
4.4.2 Reading, writing to, and clearing Standard Event Status Register .....	4-10
4.4.3 Reading, writing to, and clearing Standard Event Status Enable Register .....	4-10

4.5	Extended Event Status Register .....	4-11
4.5.1	Bit definition of Extended Event Status Register .....	4-12
4.5.2	Reading, writing to, and clearing Extended Event Status Register .....	4-13
4.5.3	Reading, writing to, and clearing Extended Event Status Enable Register .....	4-13
4.6	Techniques for Synchronizing MS2602A with a Controller .....	4-14
4.6.1	Wait for response after *OPC? query is sent .....	4-14
4.6.2	Wait for service request after *OPC is sent .....	4-15
<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 INI / IP Command .....	5-7
5.5	Device Status at Power-on .....	5-7
<b>SECTION 6</b>	<b>SAMPLE PROGRAMS .....</b>	<b>6-1</b>
6.1	Precautions on Creating the GPIB Program .....	6-3
6.2	Sample Programs .....	6-4
6.2.1	Initializing MS2602A .....	6-4
6.2.2	Reading the frequency and level at marker point .....	6-6
6.2.3	Reading trace data .....	6-8
6.2.4	Transferring 2-byte binary data .....	6-11
6.2.5	Delta marker .....	6-13
6.2.6	Multimarker function .....	6-15
6.2.7	Gate functions .....	6-17
6.2.8	Saving and recalling data .....	6-20
6.2.9	Adjacent-channel leakage power measurement .....	6-22
6.2.10	Occupied frequency bandwidth measurement .....	6-23
6.2.11	Setting template data .....	6-24
6.2.12	Measuring template .....	6-26
6.2.13	Burst wave average power measurement .....	6-27
<b>SECTION 7</b>	<b>TABLE OF DEVICE MESSAGES .....</b>	<b>7-1</b>
<b>SECTION 8</b>	<b>DETAILED DESCRIPTION OF COMMANDS .....</b>	<b>8-1</b>

<b>APPENDIX A</b>	<b>TABLE OF MS2602A DEVICE-DEPENDENT INITIAL SETTINGS .....</b>	<b>A-1</b>
<b>APPENDIX B</b>	<b>ASCII*CODE TABLE .....</b>	<b>B-1</b>
<b>APPENDIX C</b>	<b>COMPARISON TABLE OF CONTROLLER'S GPIB INSTRUCTIONS .....</b>	<b>C-1</b>



# SECTION 1

## GENERAL

This section outlines the GPIB functions of the MS2602A Spectrum Analyzer.

### TABLE OF CONTENTS

1.1	GENERAL .....	1-3
1.1.1	Functions of GPIB .....	1-3
1.1.2	Functions of the GPIB with two ports .....	1-3
1.1.3	Examples of system configuration using GPIB 1 / GPIB 2 .....	1-4
1.1.4	Specifications .....	1-5

**( Blank )**

## SECTION 1 GENERAL

### 1.1 GENERAL

The MS2602A Spectrum Analyzer, when combined with an external controller, can automate your measurement system. For this purpose the MS2602A is equipped with a GPIB interface bus ( IEEE std 488.2 1987 ) as a standard feature.

#### 1.1.1 Functions of GPIB

The functions of the MS2602A GPIB are as follows:

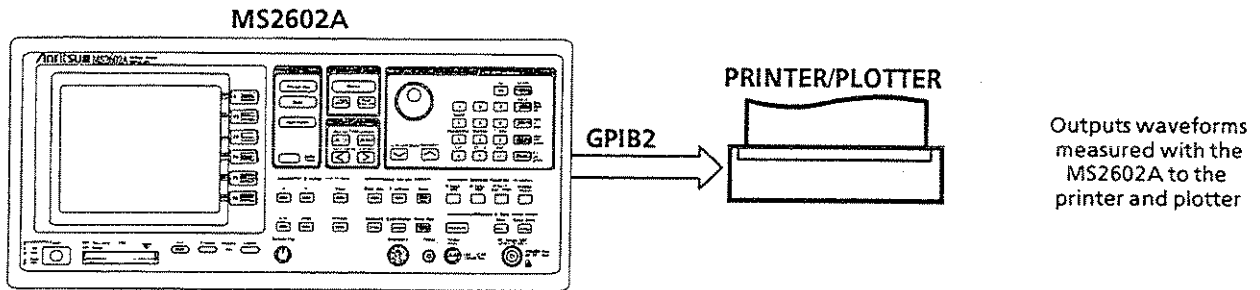
- (1) Controls all functions except the power switch and some keys including the [ LOCAL ] key
- (2) Reads out all setting conditions
- (3) Sets the GPIB address from the panel
- (4) Executes interrupts and serial polling
- (5) Configures the automatic measurement system when the MS2602A is combined with a personal computer and other measuring instruments
- (6) Configures the GPIB with two ports: GPIB 1 and GPIB 2

#### 1.1.2 Functions of the GPIB with two ports

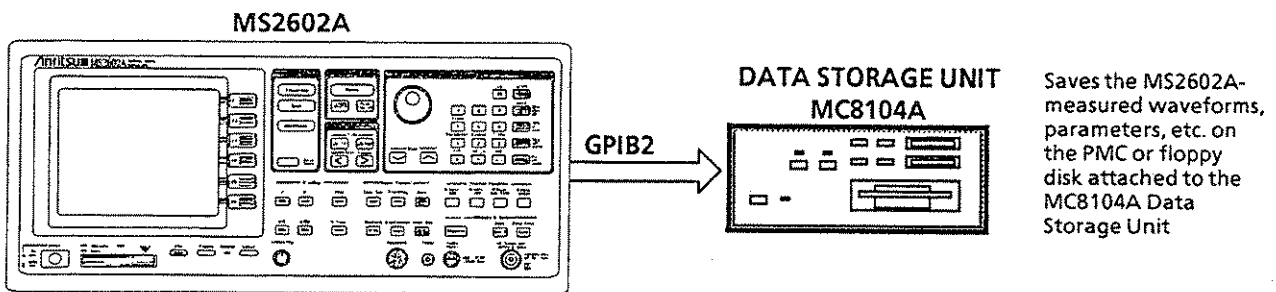
The MS2602A Spectrum Analyzer provides two GPIB ports, GPIB 1 and GPIB 2. The port on the GPIB 1 side is connected to an external controller to automate measurements by remote control, while the port on the GPIB 2 side is used to control peripherals such as printers, plotters, and other devices. This 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 the SECTIONS 11 and 12 in the Detailed Operating Instruction Part of the separate Operation Manual.

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

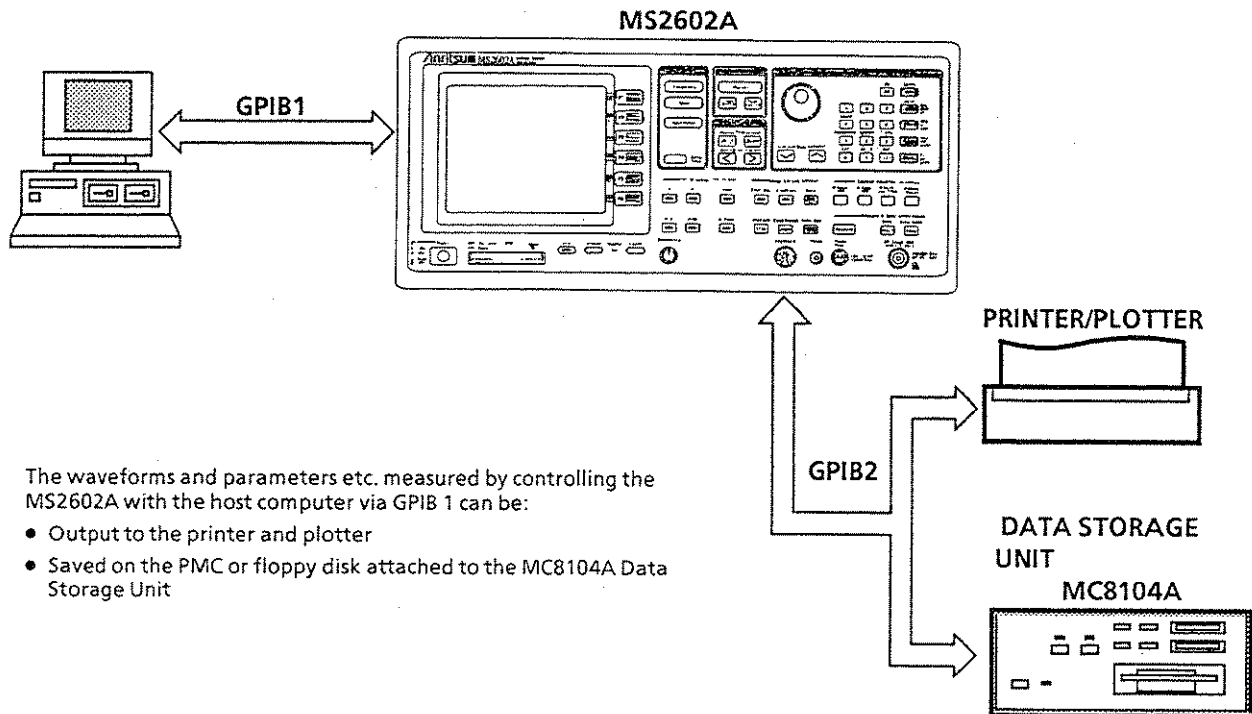
#### (1) Stand-alone type 1.....Panel operation



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



#### (3) Control by the host computer



### 1.1.4 Specifications

The MS2602A's GPIB 1 and GPIB 2 provide the IEEE488.1 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.

### 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.
RLO	Remote / local functions are not provided. Local lockout functions are not provided.
PPO	Parallel poll functions are not provided.
DC0	Device clear functions are not provided.
DT0	Device trigger functions are not provided.
C1,C2,C3,C4	Controller functions are provided.

## 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	Checking and Setting GPIB Address .....	2-4
2.2.1	Panel key layout .....	2-4
2.2.2	Key operation flow .....	2-4
2.2.3	Checking the address .....	2-5
2.2.4	Setting the address for GPIB 1 .....	2-5

( Blank )



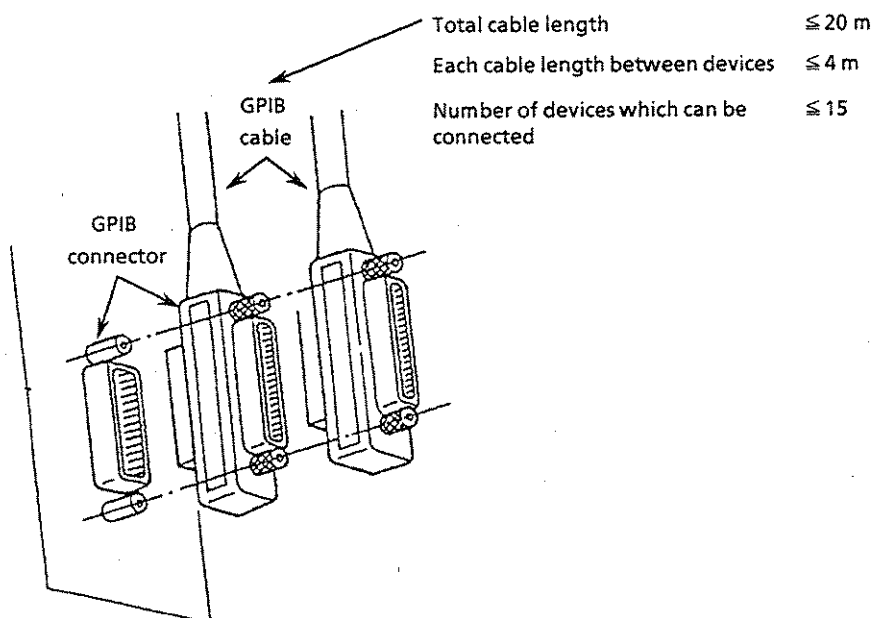
## SECTION 2

### CONNECTING THE BUS AND SETTING THE ADDRESS

#### 2.1 Connecting Devices with GPIB Cables

The rear panel has connectors for connecting GPIB cables. The cable must be connected before the power is switched on.

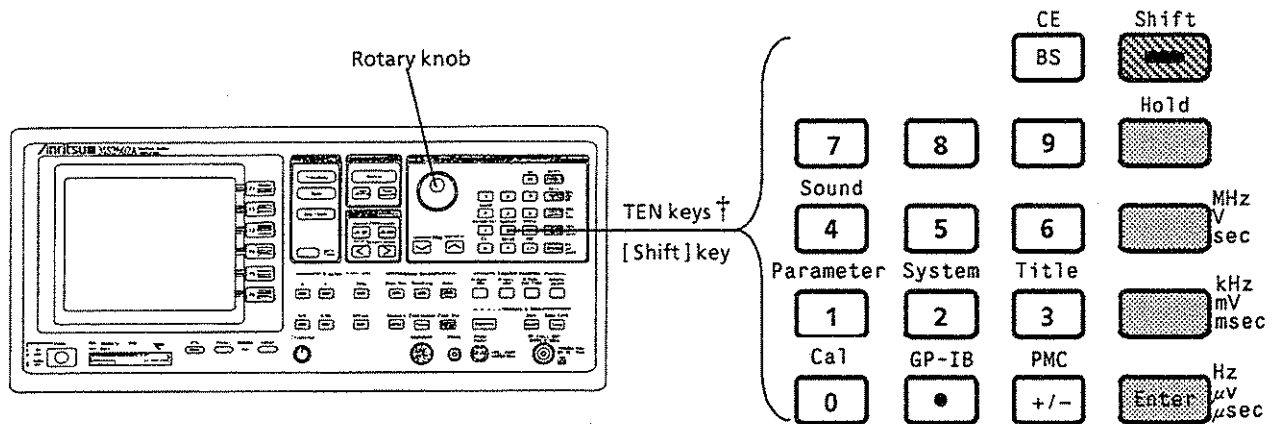
A maximum of 15 devices, including the controller, can be connected to one system. The restrictions indicated at the right of the diagram below should be observed when connecting many devices to one system.



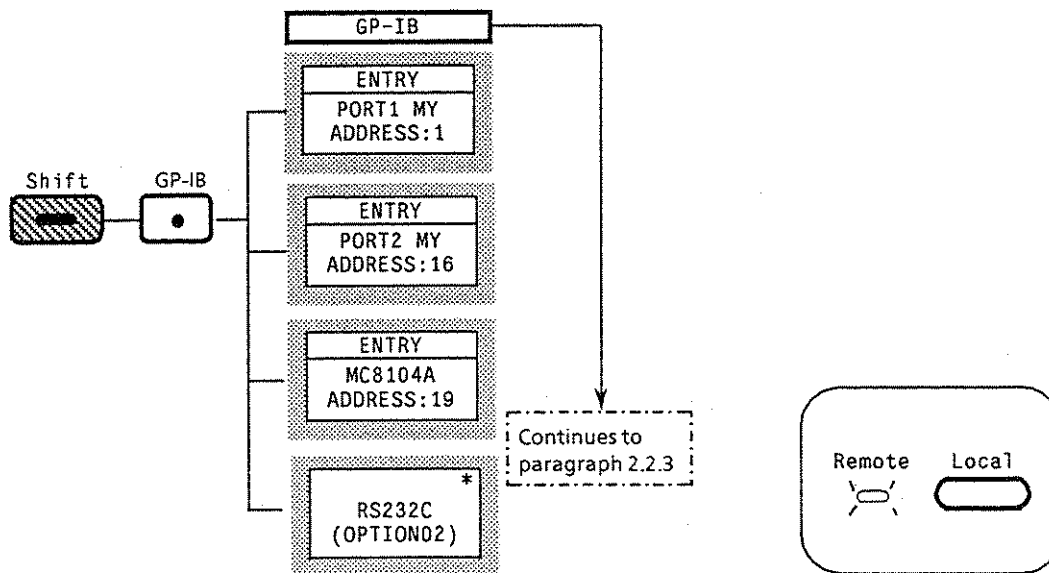
## 2.2 Checking and Setting GPIB address

The GPIB address of the MS2602A should be set after switching on the power. The address of GPIB 1 has been set to 01 and held by the battery backup at shipment from the factory. If this same number is used for the MS2602A address, it is not necessary to set the address again. To set a new address, the numeric keys and other keys on the front panel are used while the MS2602A is in local mode. The devices on the GPIB are set to local mode at power-on.

### 2.2.1 Panel key layout



### 2.2.2 Key operation flow



It is impossible to operate the keys on the panel in remote mode while the Remote LED is lit as shown in the upper-right diagram. In this case, press the [ Local ] key, and the Remote LED goes off to signify local mode. The panel keys can then be used.

† These 18 keys composed of the numeric keys, unit keys, backspace ( BS ) key, and Shift key are called TEN keys for convenience.

### 2.2.3 Checking the address

Press the panel keys in the order of [ Shift ] [ ● ] ( GP-IB ), and the GPIB menu below will be displayed on the screen. The current address can be checked on this screen.

GP-IB ENTRY PORT1 MY ADDRESS: 1	The current value indicates the GPIB 1's address to control the MS2602A with an external controller. To modify this address, press the F1 soft key, then use the TEN keys to set a new address number.
ENTRY PORT2 MY ADDRESS: 16	The current value indicates the MS2602A address on the GPIB 2 side. To modify this address, press the F2 soft key, then use the TEN keys to set a new address number.
ENTRY MC8104A ADDRESS: 19	The current value indicates the MC8104A address to control the MC8104A Data Storage Unit via GPIB 2. To modify this address, press the F3 soft key, then use the TEN keys to set a new address number.
RS232C (OPTION02) *	Options menu

### 2.2.4 Setting the address for GPIB 1

- Address setting range :  $00 \leq \text{ADDRESS} \leq 30$

---

Example : Checking current address 1 and changing it to address 6

---

1. Press the panel keys in the order of [ Shift ] [ ● ] ( GP-IB ). Check the current address 1 from the menu corresponding to the [ F1 ] soft key.
2. Press the [ F1 ] soft key, and the response shown in the figure below appears in the communication field.

GP-IB PORT1

3. Press the panel keys in the order of [ 6 ] [ Enter ] to change the address to 6.
-

(Blank)

## 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
3.1.1	Program message format .....	3-3
3.1.2	Response message format .....	3-9

(Blank)

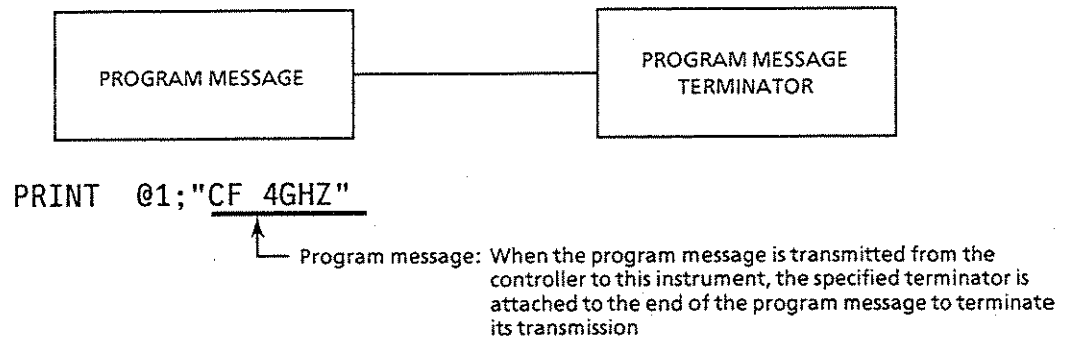
## SECTION 3 DEVICE MESSAGE FORMAT

### 3.1 General Description

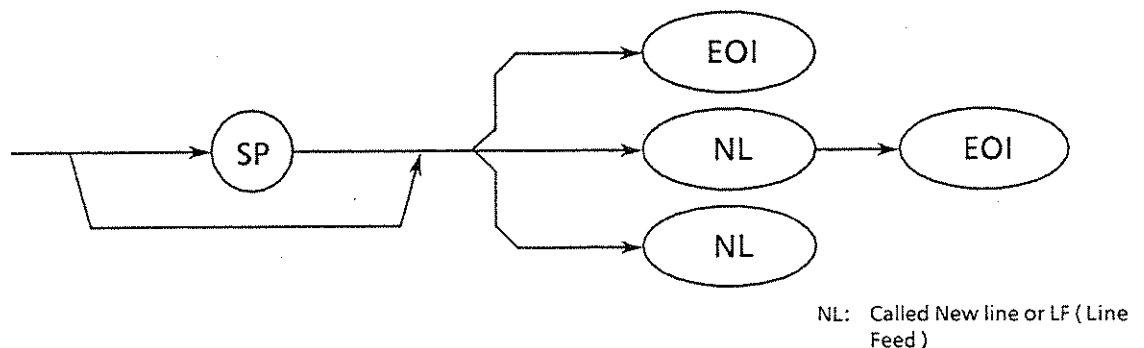
The device messages are data messages that are transmitted between the controller and devices. There are two types of data messages: program messages that are transferred from the controller to this instrument ( device ), and response messages that are sent from this instrument ( device ) to the controller. 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.

#### 3.1.1 Program message format

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

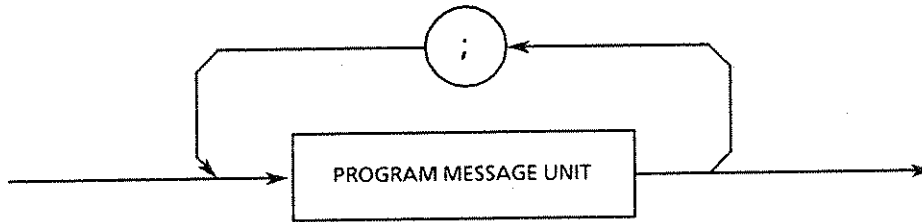


#### (1) PROGRAM MESSAGE TERMINATOR



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

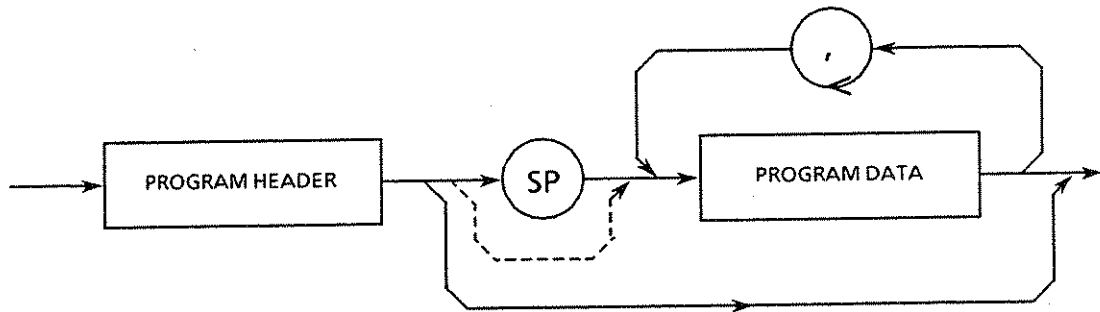
(2) PROGRAM MESSAGE



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

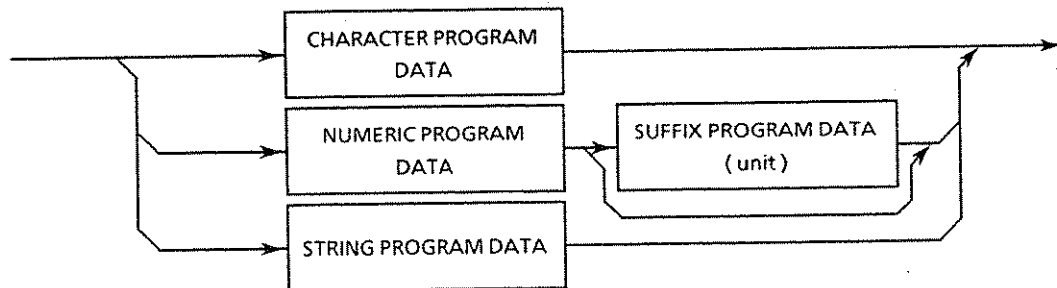
<Example> PRINT @1;"CF 1GHZ;SP 500KHZ"

(3) PROGRAM MESSAGE UNIT



- The IEEE488.2 common command has a leading asterisk "\*" that is always placed before the program header.
- A numeric program data allows the (SP) between header and data to be omitted.
- 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

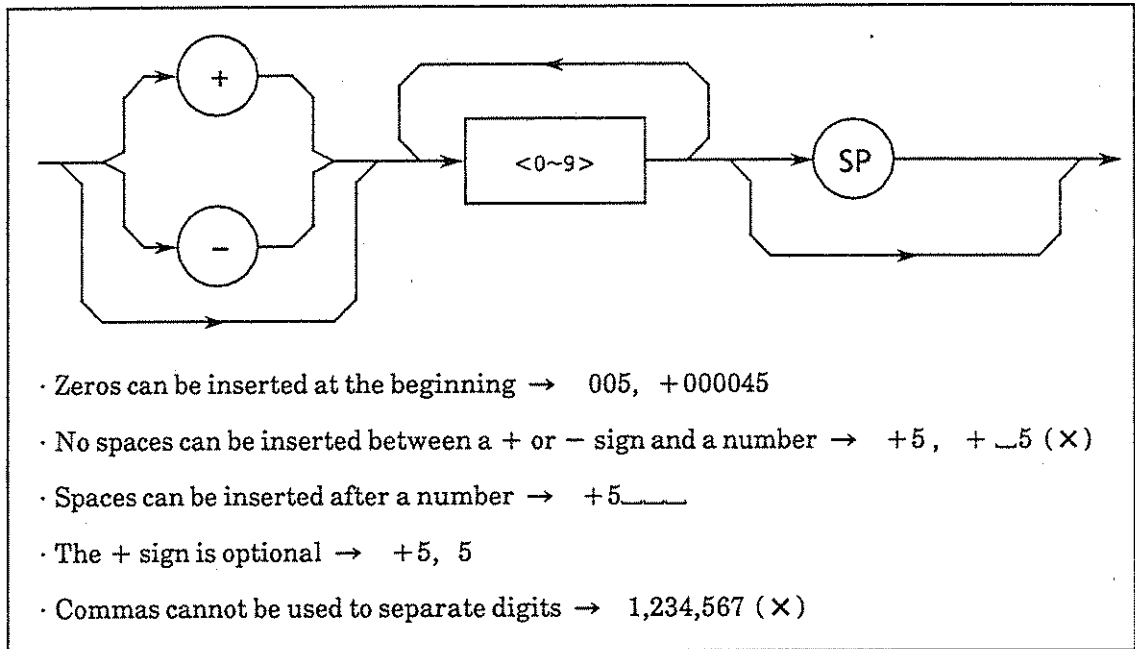
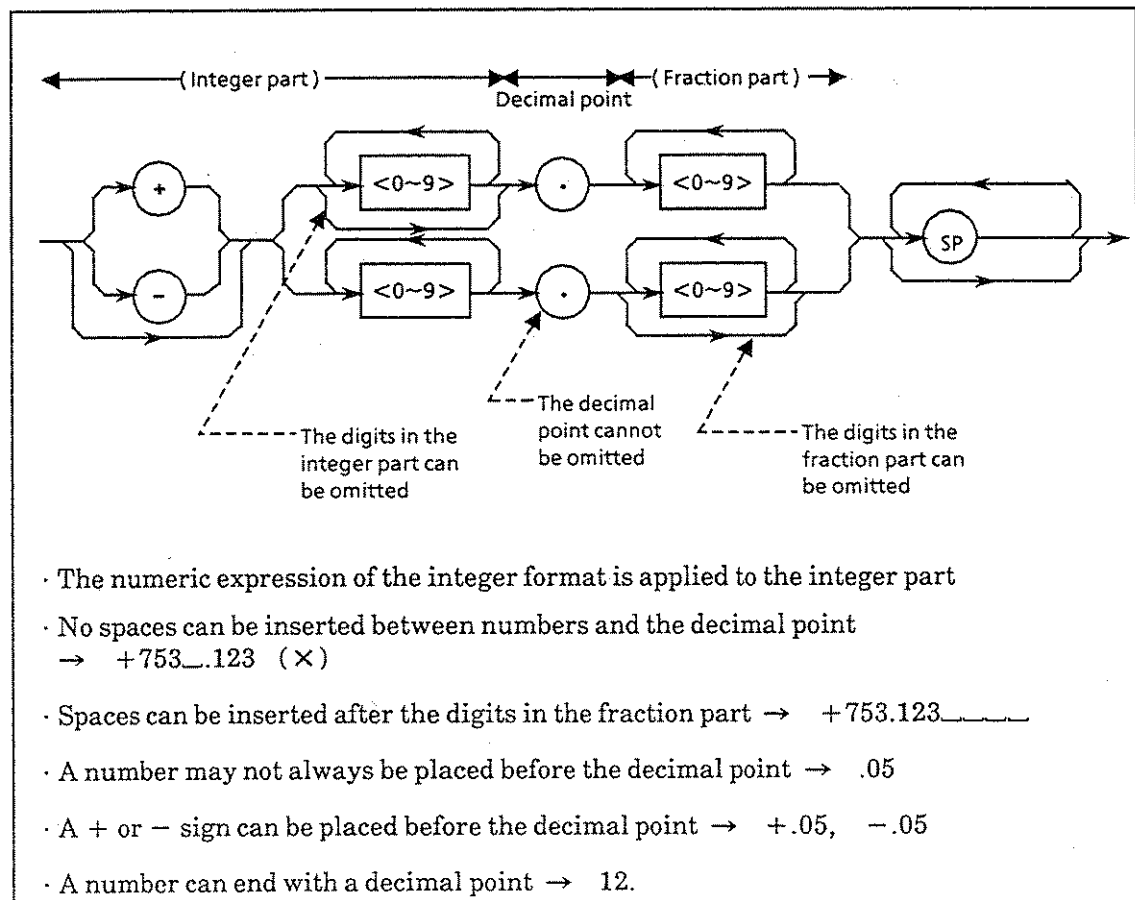
Character program data consists of the upper-case alphabetic characters from A to Z, lower-case alphabetic characters from a to z, the underline of "\_", and the numbers 0 to 9. They can be used in a specified combination.

<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 format (NR2).

**< Integer Format (NR1) >****< Fixed-Point Format (NR2) >**

**(7) SUFFIX PROGRAM DATA ( unit )**

The table below shows the suffixes used for the MS2602A.

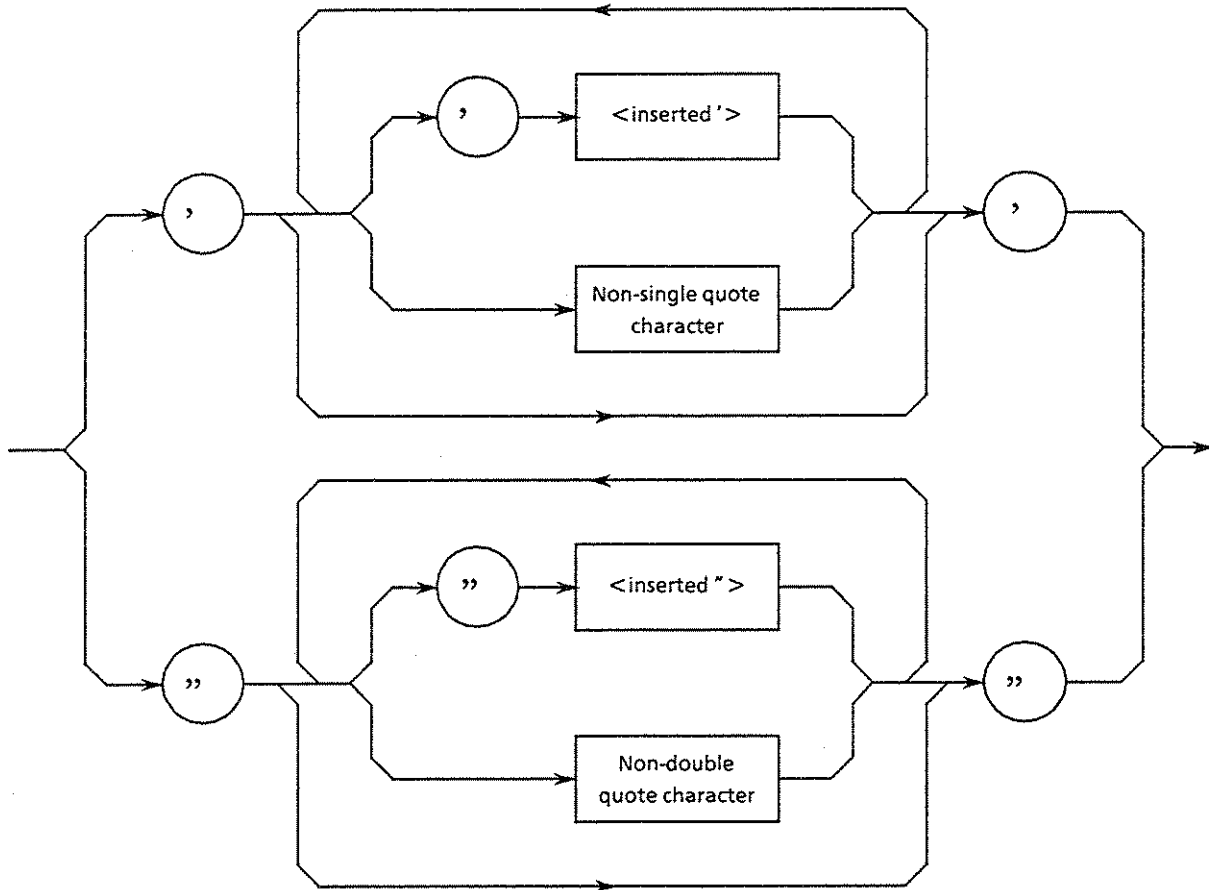
**Table of MS2602A Suffix codes (1/2)**

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 system)	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 system)	V	V
	mV	MV
	$\mu$ V	UV
	Default	UV

Table of MS2602A Suffix codes (2/2)

Classification	Unit	Suffix code
Level (W system)	W	W
	mW	MW
	$\mu$ W	UW
	nW	NW
	pW	PW
	fW	FW
	Default	UW

## (8) STRING PROGRAM DATA



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

```
PRINT @ 1; "TITLE 'MS2602A'"
```

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

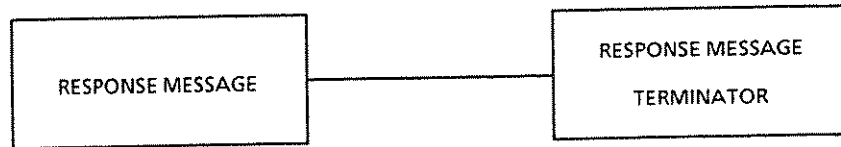
```
PRINT @ 1; "TITLE 'MS2602A' 'NOISE MEAS' ' ' "
```

Executing TITLE results in MS2602A 'NOISE MEAS'

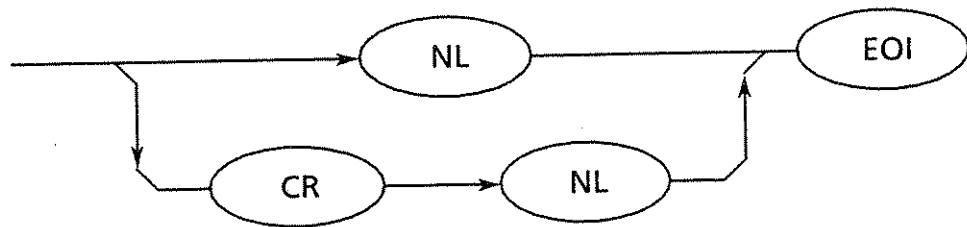
**Note:** To use the double quotation mark " in the PRINT statement, specify CHR\$( &H22 ).

### 3.1.2 Response message format

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

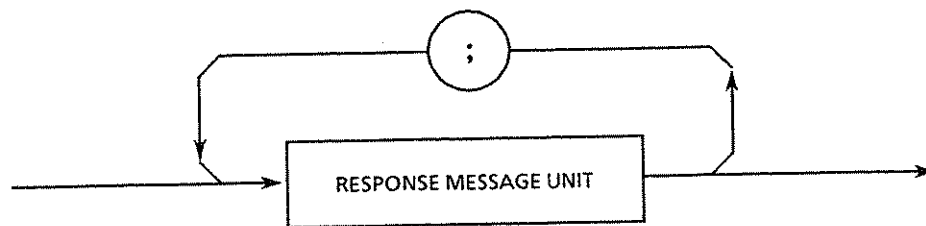


#### (1) RESPONSE MESSAGE TERMINATOR



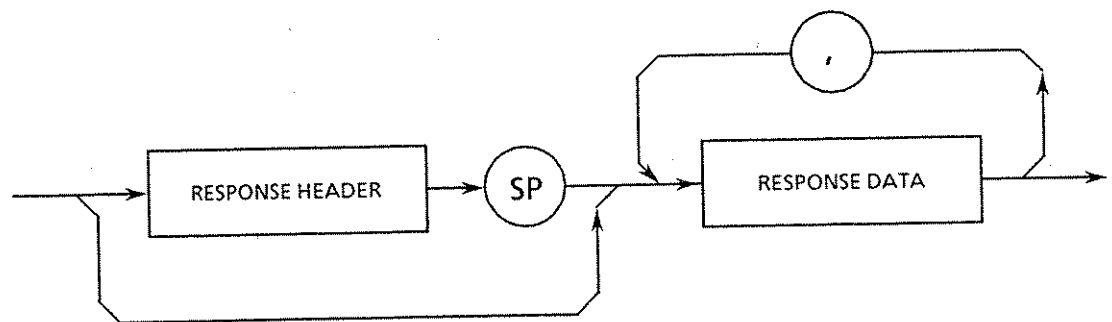
The response message terminator to be used depends on the TRM command.

#### (2) RESPONSE MESSAGE

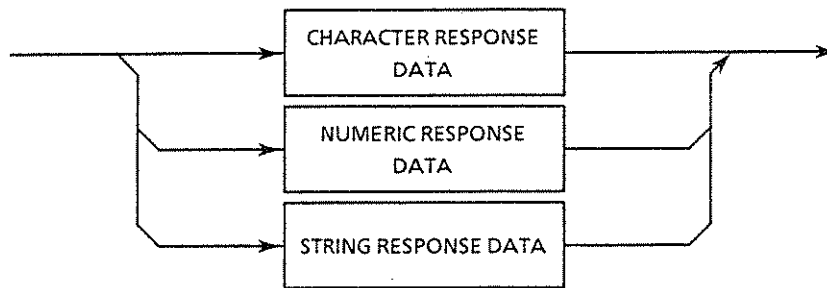


When a query is sent by the PRINT statement with one or more program queries, the response message also consists of one or more response message units.

#### (3) RESPONSE MESSAGE UNIT ( example )



**(4) RESPONSE DATA**

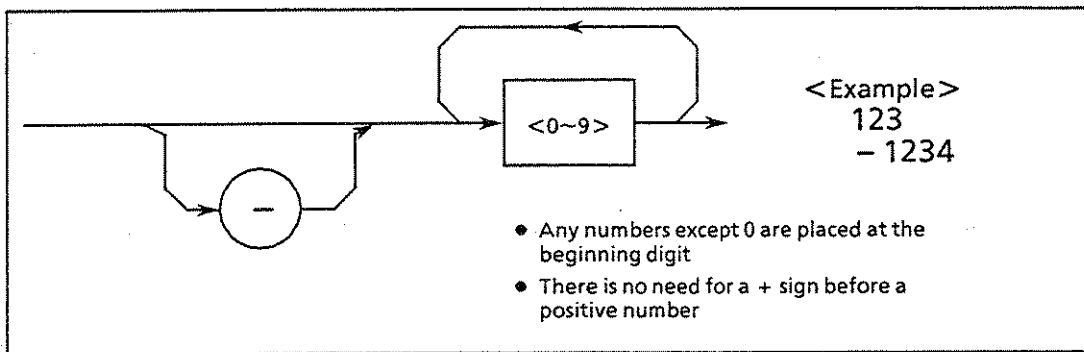


**(5) CHARACTER RESPONSE DATA**

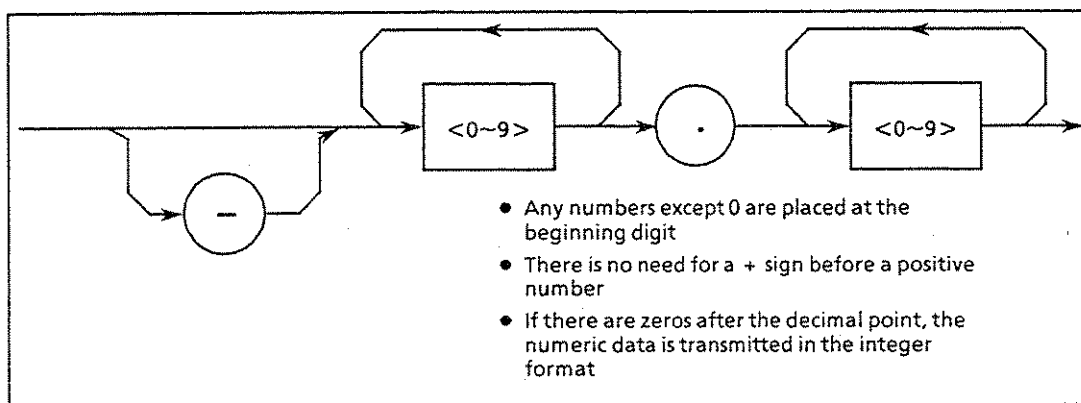
Character response data consists of the upper-case alphabetic characters from A to Z, lower-case alphabetic characters from a to z, the underline “\_”, and the numbers 0 to 9. They can be used in a specified combination.

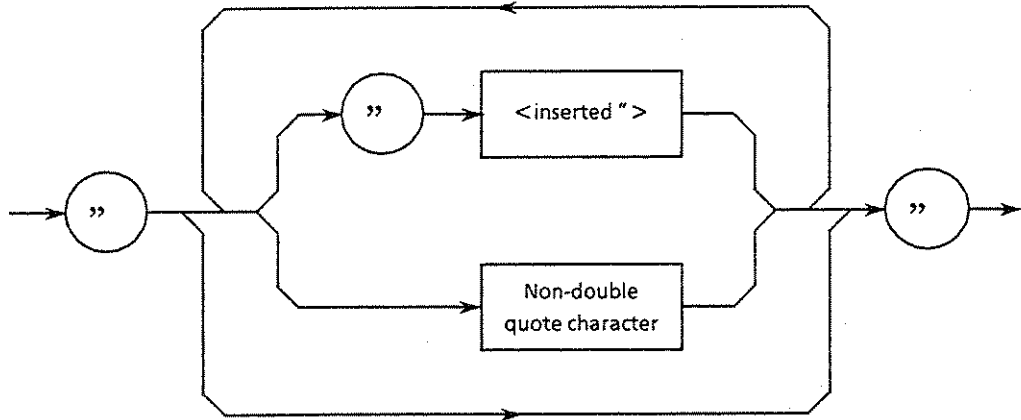
**(6) NUMERIC RESPONSE DATA**

< Integer format (NR1) >



< Fixed-point 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 transmitting binary format data and 2-bite binary data, see example 3-2 ( page 6-7 and page 6-9 ) in SECTION 6, "SAMPLE PROGRAM".

( Blank )



## SECTION 4

### STATUS STRUCTURE

This section describes the device-status reporting and its data structure defined by the IEEE488.2 standard. It also describes the synchronization techniques between a controller and devices.

#### TABLE OF CONTENTS

4.1	IEEE488.2 Standard Status Model .....	4-3
4.2	Status Byte (STB) Register .....	4-5
4.2.1	ESB and MAV summary messages .....	4-5
4.2.2	Device-dependent summary messages .....	4-6
4.2.3	Reading and clearing STB register .....	4-7
4.3	Service Request (SRQ) Enabling Operation .....	4-8
4.4	Standard Event Status Register .....	4-9
4.4.1	Bit definition of Standard Event Status Register .....	4-9
4.4.2	Reading, writing to, and clearing Standard Event Status Register .....	4-10
4.4.3	Reading, writing to, and clearing Standard Event Status Enable Register .....	4-10
4.5	Extended Event Status Register .....	4-11
4.5.1	Bit definition of Extended Event Status Register .....	4-12
4.5.2	Reading, writing to, and clearing Extended Event Status Register .....	4-13
4.5.3	Reading, writing to, and clearing Extended Event Status Enable Register .....	4-13
4.6	Techniques for Synchronizing MS2602A with a Controller .....	4-14
4.6.1	Wait for response after *OPC? query is sent .....	4-14
4.6.2	Wait for service request after *OPC is sent .....	4-15

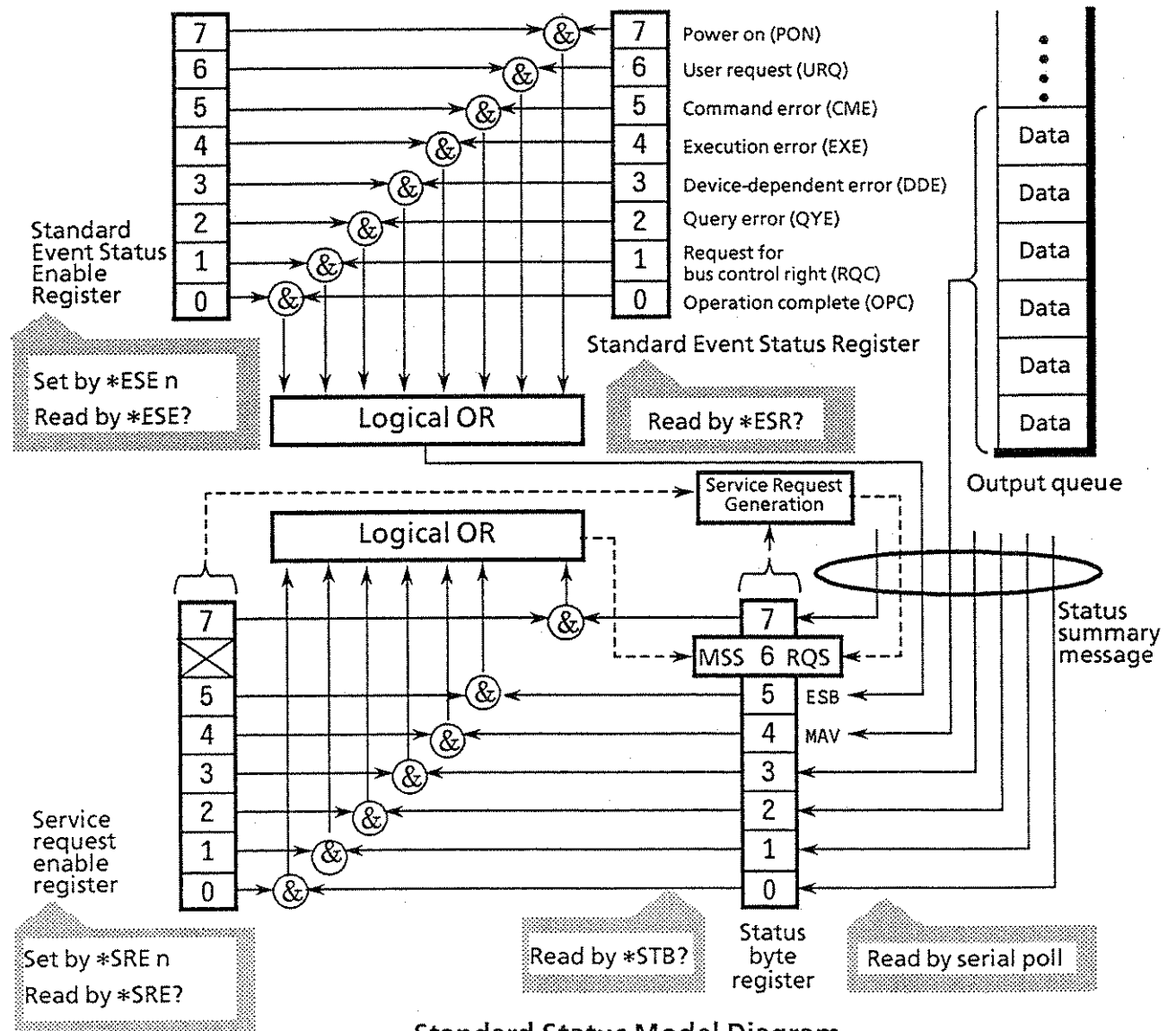
( Blank )

## SECTION 4 STATUS STRUCTURE

The status Byte (STB) sent by the controller is based on the IEEE488.1 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.

### 4.1 IEEE488.2 Standard Status Model

The diagram below shows the standard model for the status data structures stipulated in the IEEE488.2 standard.



Standard Status Model Diagram

In the status model, the IEEE488.1 status bytes are used as the lowest grade status. This status byte is composed of seven summary message bits from the higher grade status structure. In order to create these bits, the status data structure is composed of two types of register and queue models.

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 used for sequentially recording the waiting status values and data. The queue structure summary message becomes 1 if the queue is not empty and 0 if it is empty.

In IEEE488.2, 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
<p>The Standard Event Status Register has the structure of the previously described register model. In this register, the bits for 8 types of standard events encountered by a device are set as follows:</p> <ul style="list-style-type: none"> <li>① Power on</li> <li>② User request</li> <li>③ Command error</li> <li>④ Execution error</li> <li>⑤ Device-dependent error</li> <li>⑥ Query error</li> <li>⑦ Request for bus control right</li> <li>⑧ Operation complete</li> </ul> <p>The Logical OR output bit is represented by Status Byte Register bit 5 ( DIO6 ) as a summary message for the Event Status Bit ( ESB ).</p>	<p>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.</p>	<p>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.</p>

## 4.2 Status Byte (STB) Register

The STB register consists of the STB and RQS ( or MSS ) messages of the device.

### 4.2.1 ESB and MAV summary messages

This paragraph describes the ESB and MAV summary messages.

#### (1) ESB summary message

The ESB ( Event Summary Bit ) is a message defined by IEEE488.2, which uses bit 5 of the STB register. The ESB summary message bit becomes 1 when the setting permits events to occur if any one of the events recorded in the Standard Event Status Register becomes 1. Conversely, it becomes 0 if none of the recorded events occurs, even if events are set to occur.

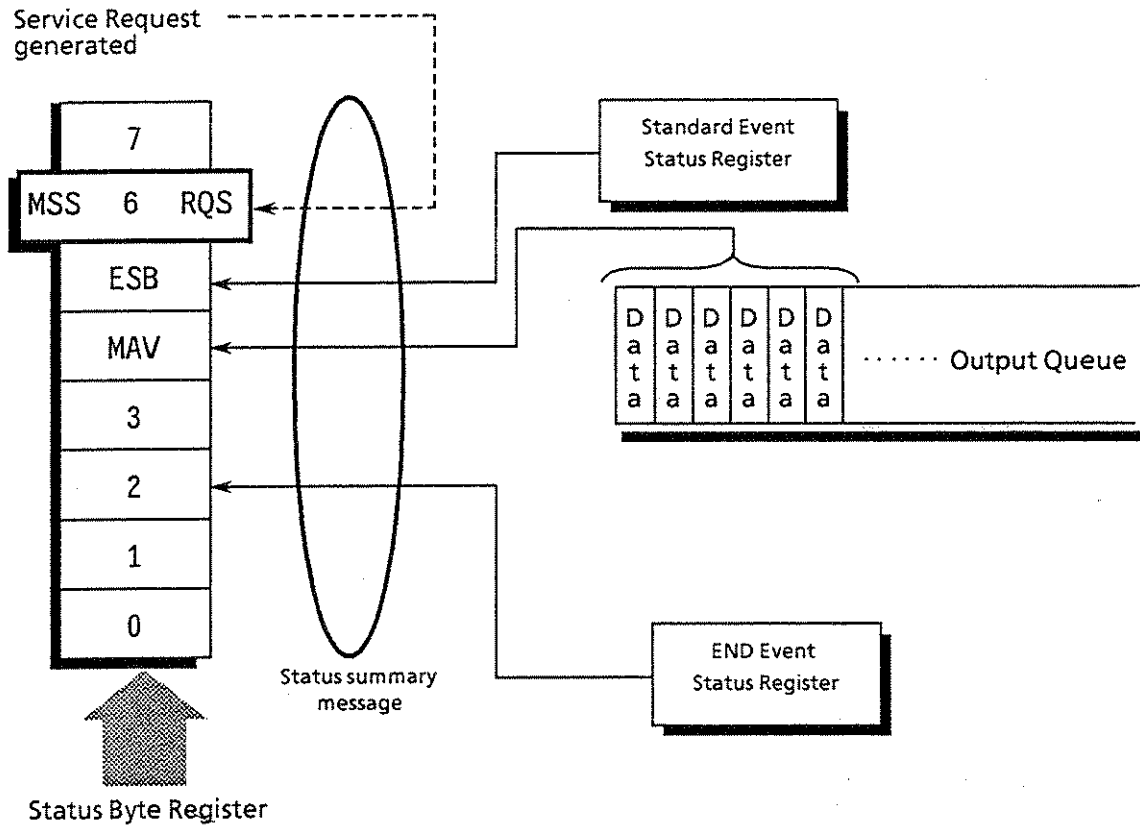
This bit becomes 0 when the ESR register is read out by the ESR? query or when cleared by the \*CLS command.

#### (2) MAV summary message

The MAV ( Message Available ) summary bit is a message defined by IEEE488.2, which uses bit 4 of the STB register. This bit status indicates whether the output queue is empty or not. The MAV summary message bit is set to 1 when a device is ready to receive a request for a response message from the controller, and to 0 when the output queue is empty. This message is used to synchronize the information exchange with the controller. For example, it is available when, after setting a query command to a device, the controller waits until MAV becomes 1. While the controller is waiting 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

As shown below, the MS2602A does not use bits 0, 1, 3, and 7, and uses bit 2 as the summary bit of the END Event Status Register.



### 4.2.3 Reading and clearing STB register

Serial polling or the \*STB common query allows the contents of the STB register to be read. The 488.1 STB message can be read by either method, but the value set to bit 6 is different for each method. The STB register contents can be cleared by the \*CLS command.

#### (1) Reading by serial polling

The IEEE488.1 serial polling allows the device to return a 7-bit status byte and an RQS message bit which conforms to IEEE488.1. The value of the status byte is not changed by serial polling. The device sets RQS message to 0 immediately after being polled.

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

The \*STB? common query requires the devices to send the contents of the STB register and the integer format response messages including the MSS ( Master Summary Status ) summary message. Thus, except bit 6 which represents the MSS summary message, the response to \*STB? is identical to that of serial polling.

#### (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 polling. It should not be taken as part of the status byte specified by IEEE488.1. MSS is configured by the overall logical OR in which the STB register and SRQ Enable ( SRE ) register are combined.

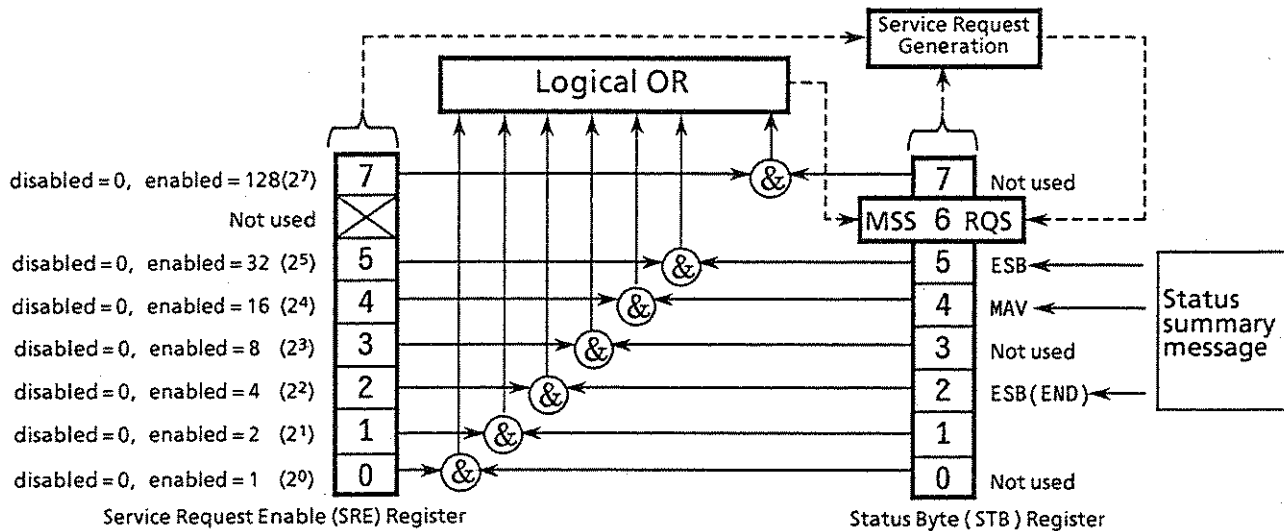
#### (4) Clearing the STB register by the \*CLS common command

The \*CLS common command clears all status data structure as well as the summary messages corresponding to them. The \*CLS command does not affect settings in the Enable Registers.

### 4.3 Service Request ( SRQ ) Enabling Operation

The bit status ( 0 or 1 ) of the Service Request Enable Register ( SRE ) determines which bit of the corresponding STB register may generate SRQ.

Bits in the Service Request Enable Register correspond to bits in the status byte register. If a bit in the Status Byte Register corresponding to an enabled bit in the Service Request Enable Register is set to 1, the device makes a service request to the controller with the RQS bit set to 1.



#### (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.

#### (2) Updating the SRE register

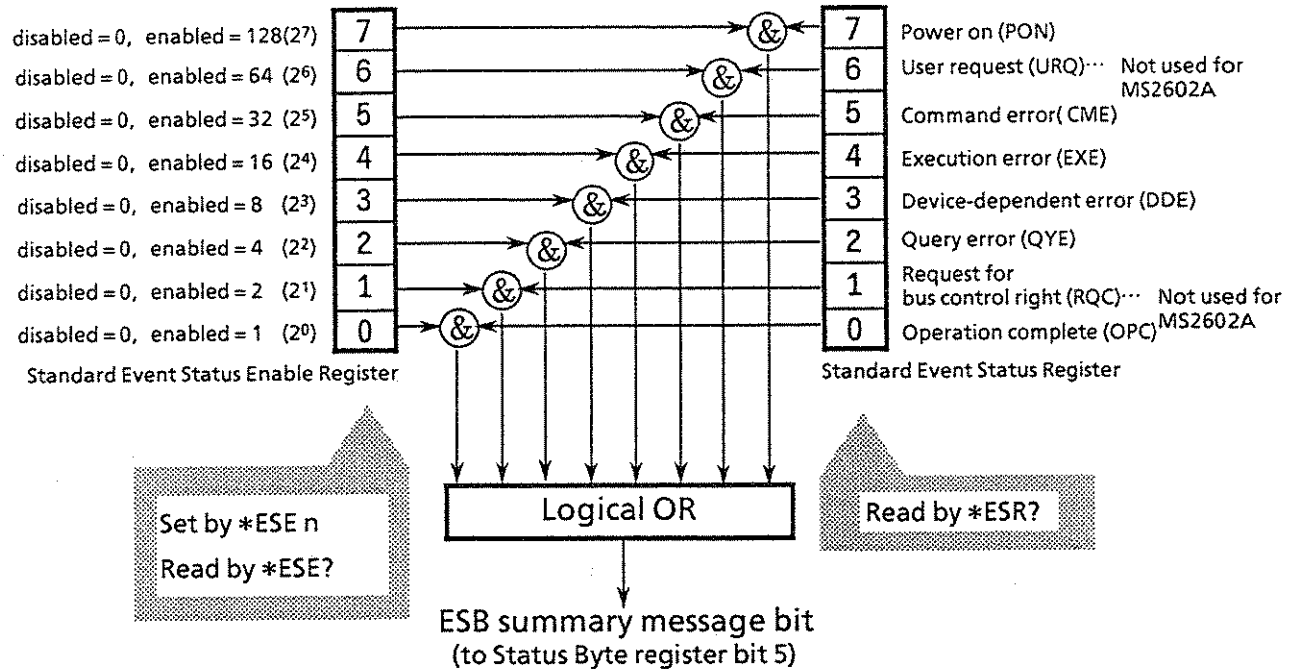
The SRE register is written to by using the \*SRE common command. An integer from 0 to 255 is assigned as a parameter to set the SRE register bit to 0/1. The value of bit 6 is ignored.



## 4.4 Standard Event Status Register

### 4.4.1 Bit definition of Standard Event Status Register

The diagram below shows the operation of the Standard Event Status Register.



The Standard Event Status Enable Register on the left is used to select which bits in the corresponding Event Register will cause a TRUE summary message when set.

Bit	Event name	Description
7	Power on (PON–Power on)	A transition from power-off to power-on occurred during the power-up procedure
6	(Not used)	
5	Command error (CME–Command Error)	An illegal program message or a misspelled command was received
4	Execution error (EXE–Execution Error)	A legal but unexecuted program was received
3	Device-dependent error (DDE–Device-dependent Error)	An error caused by other than CME, EXE, or QYE occurred (parameter etc.)
2	Query error (QYE–Query Error)	An attempt was made to read data in an empty Output Queue. Or, before data in the Output Queue was read, the data had already been lost.
1	(Not used)	
0	Operation complete (OPC–Operation Complete)	This bit becomes 1 when this instrument has processed the *OPC command

#### 4.4.2 Reading, writing to, and clearing Standard Event Status Register

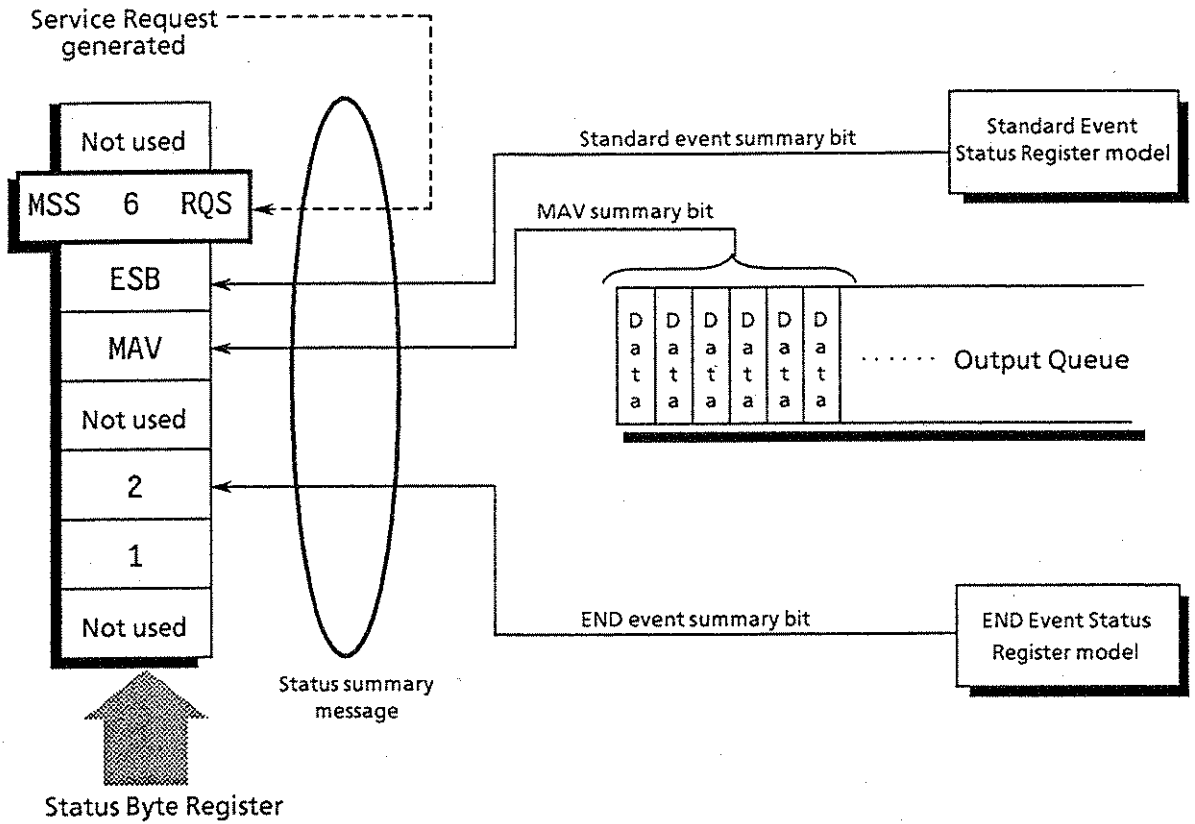
Reading	The *ESR? common query is used to read the ESR register, which is cleared after being read. The response message is integer-format data for which the sum of the binary-weighted event bit is converted to decimals.
Writing	With the exception of clearing, it is impossible to write to the register from outside
Clearing	The register is cleared in the following cases. ① When a *CLS command is received ② When the power is turned on, bit 7 is set to ON, and the other bits are cleared to 0 ③ An event is read for the *ESR? query command

#### 4.4.3 Reading, writing to, and clearing Standard Event Status Enable Register

Reading	The register is read by the *ESE? common query. The response message is an integer-format data for which the sum of the binary-weighted event bit is converted to decimals.
Writing	The register is written to by the *ESE common command
Clearing	The register is cleared in the following cases. ① When an *ESE command with a data value of 0 is received ② When the power is turned on The Standard Event Status Enable Register is not affected by the following. ① When the device clear function status of IEEE488.1 is changed ② When a *RST common command is received ③ When a *CLS common command is received

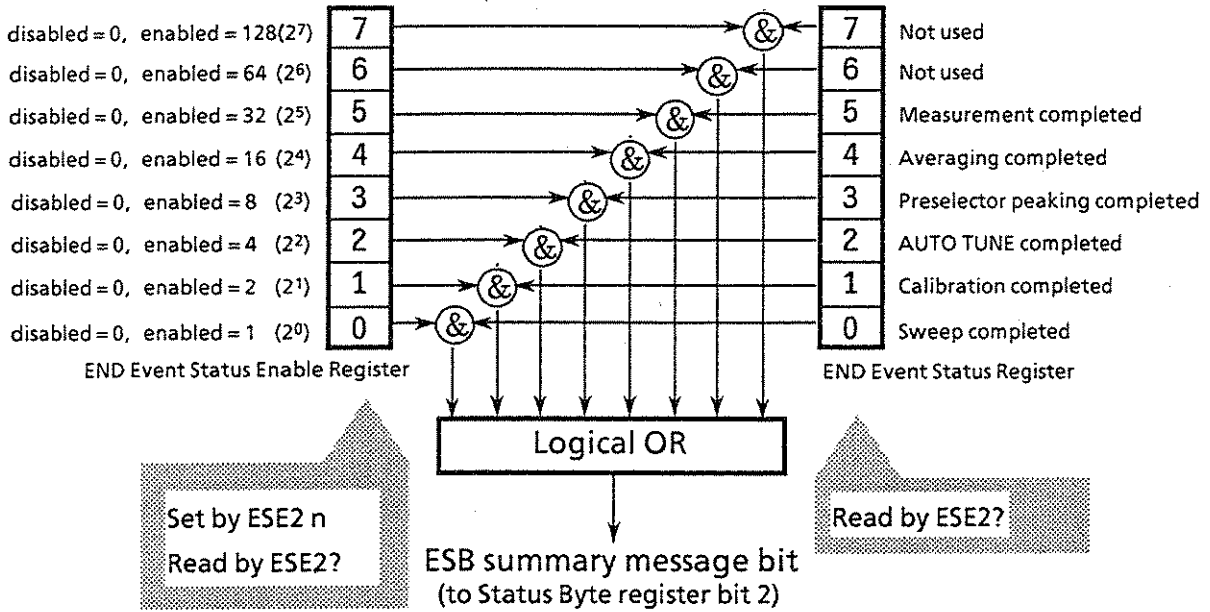
### 4.5 Extended Event Status Register

As shown below, bits 0, 1, 3, and 7 are unused and bit 2 is assigned to the END event summary bit as the status-summary bit supplied by the extended register model.



### 4.5.1 Bit definition of Extended Event Status Register

The diagram below describes the operation, event-bit names, and their meanings of the END Event Status Register.



The END Event Status Enable Register on the left is used to select which bits in the corresponding Event Register will cause a TRUE summary message when set.

Bit	Event name	Description
7	(Not used)	(Not used)
6	(Not used)	(Not used)
5	Measurement completed	Calculation processing for measuring (frequency count, noise, etc.) has completed
4	Averaging completed	Sweeping the specified number of averaging rate has completed
3	Preselector peaking completed	Preselector peaking has completed
2	AUTO TUNE completed	AUTO TUNE has completed
1	Calibration completed	Any one of ALL CAL, LEVEL CAL, and FREQ CAL has completed
0	Sweep completed	Single sweep has completed or is in standby status

#### 4.5.2 Reading, writing to, and clearing Extended Event Status Register

Reading	The ESR2? common query is used to read the register, which is cleared after being read. The response message is an integer-format data for which the sum of the binary-weighted event bit is converted to decimals.
Writing	With the exception of clearing, it is impossible to write to the register from outside
Clearing	The register is cleared in the following cases. <ol style="list-style-type: none"> <li>① When a *CLS command is received</li> <li>② When the power is turned on</li> <li>③ An event is read for the ESR2? query command</li> </ol>

#### 4.5.3 Reading, writing to, and clearing Extended Event Status Enable Register

Reading	The register is read by the ESE2? common query. The response message is integer-format data for which the sum of the binary-weighted event bit is converted to decimals.
Writing	The register is written to by the ESE2 program command. Since bits 0 to 7 of the registers are weighted respectively to 1, 2, 4, 8, 16, 32, 64, and 128, the write data is transmitted as integer format data that is the sum of the desired-bit digits selected from the weighted values.
Clearing	The register is cleared in the following cases. <ol style="list-style-type: none"> <li>① When an ESE2 program command with a data value of 0 is received</li> <li>② When the power is turned on</li> </ol> <p>The Extended Event Status Enable Register is not affected by the following.</p> <ol style="list-style-type: none"> <li>① When the device clear function status of IEEE488.1 is changed</li> <li>② When a *RST common command is received</li> <li>③ When a *CLS common command is received</li> </ol>

## 4.6 Techniques for Synchronizing MS2602A with a Controller

The MS2602A 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 MS2602A and the controller.

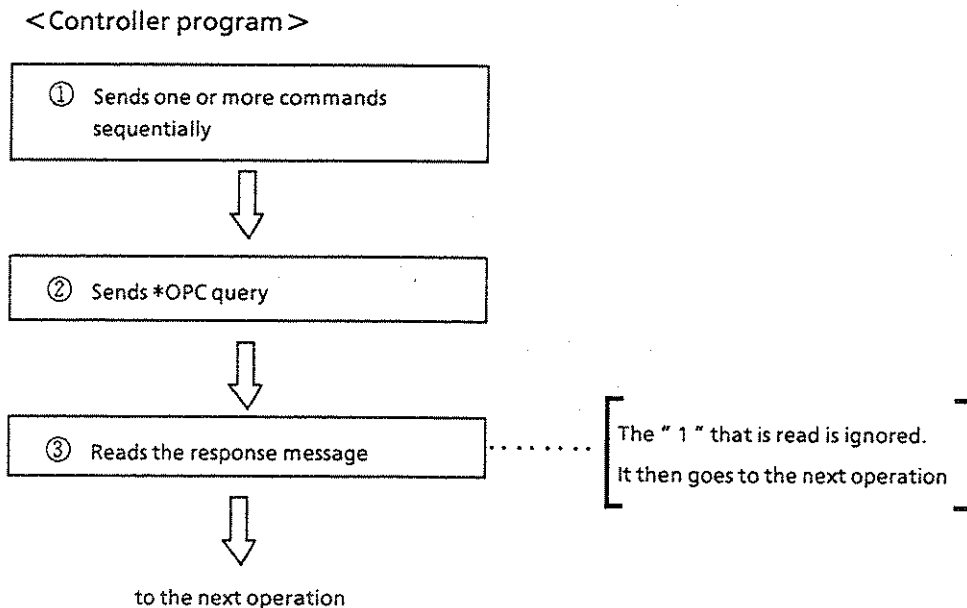
If the controller controls one or more devices and synchronizes with them, after all the commands specified for the MS2602A have been processed, the next commands must be sent to other devices.

There are two ways of synchronizing the MS2602A with the controller.

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

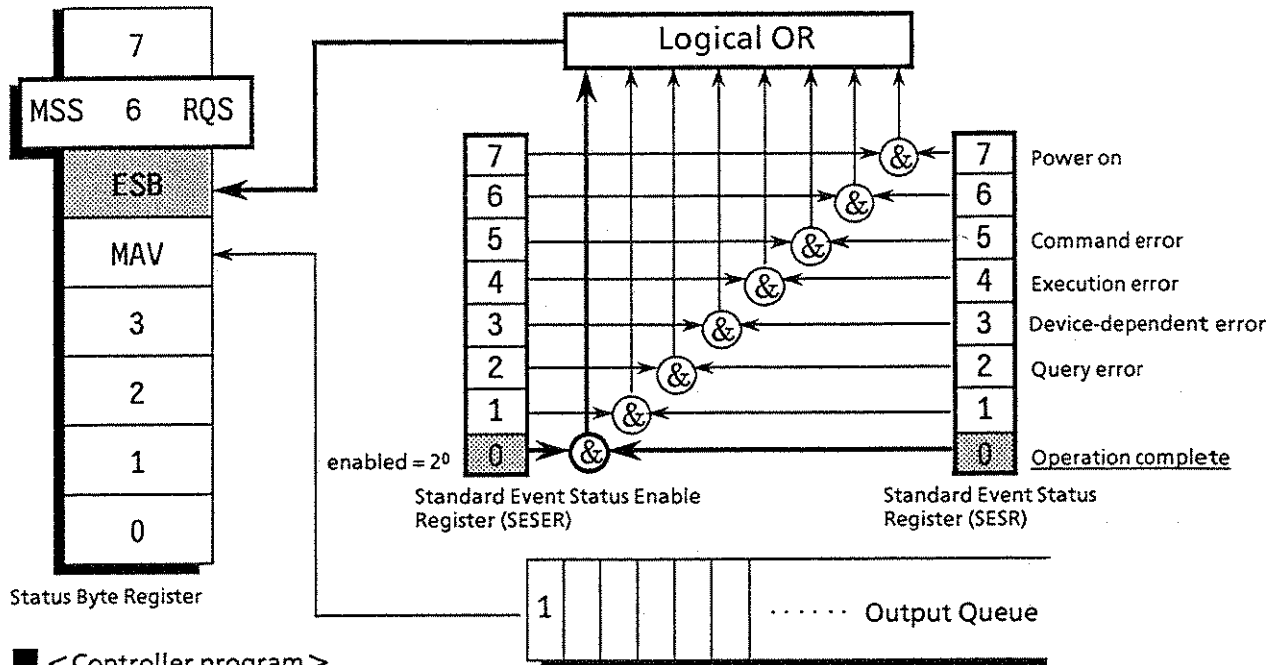
### 4.6.1 Wait for response after \*OPC? query is sent

The MS2602A outputs "1" as the response message when executing the \*OPC? query command. The controller is synchronized with the MS2602A by waiting for the response message to be entered.

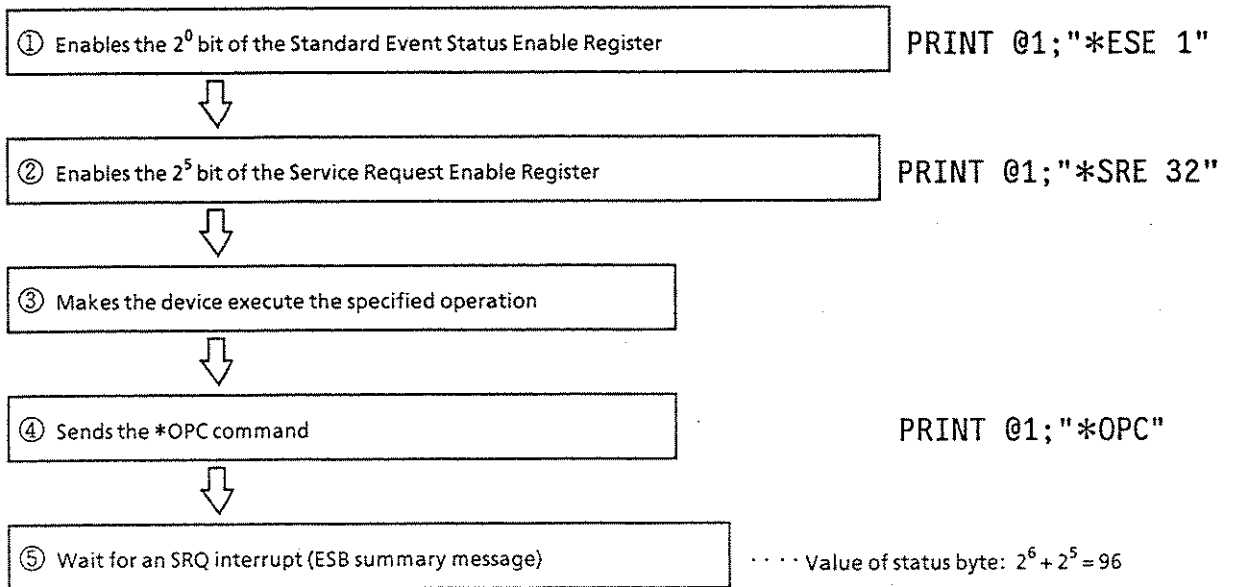


### 4.6.2 Wait for service request after \*OPC is sent

The MS2602A sets the operation-complete bit ( bit 0 ) to 1 when executing the \*OPC command. The controller is synchronized with the MS2602A by waiting for SRQ when the operation-complete bit is set for SRQ.



■ < Controller program >



( Blank )



## SECTION 5 INITIAL SETTINGS

The MS2602A initializes the GPIB interface system using three levels in accordance with the IEEE488.2 specifications. This section describes how these three level initializations are processed and how to instruct the initialization from the controller.

### 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 INI/IP Command .....	5-7
5.5	Device Status at Power-on .....	5-7

( Blank )

## SECTION 5 INITIAL SETTINGS

In the IEEE488.2 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 and SDC 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 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 MS2602A 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.                   |
| ⑥ Device functions .....                                     | Puts all functions associated with the message exchange in the idle state. The device continues to wait for a message from the controller.                    |

### CAUTION

*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 (MS2602A) whose address is 1 with level 3

#### ■ Explanation

The \*RST command is an IEEE488.2 common command which resets a device with level 3.

The \*RST (Reset) command is used to reset a device (MS2602A) to a specific initial state. For details of the items which are initialized and the initial settings, see Appendix A.

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

- ① *IEEE488.1 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*

☞ For details on the initial settings of the MS2602A, see Appendix A.

## 5.4 Device Initialization by the INI / IP Command

### ■ Syntax

---

INI

IP

---

### ■ Example ( program message )

PRINT @1;"INI" ..... Initializes the device ( MS2602A ) whose address is 1 with level 3

### ■ Explanation

The INI and IP commands are MS2602A device-dependent messages which initialize a device with level 3. For details of the items and the initial settings which are initialized by the INI / IP commands, see Appendix A.

## 5.5 Device Status at Power-on

When the power is switched on:

- ① The device status is set to the status when the power was last switched off.  
In the special case when the power is switched on immediately after delivery of the MS2602A, the MS2602A settings are those listed in the Initial Settings Table ( Appendix A ).
- 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.

( Blank )



## SECTION 6

### SAMPLE PROGRAMS

This section gives some examples of the N88-Basic program that controls the MS2602A from the NEC-PC9800 personal computer which is used as a controller.

#### TABLE OF CONTENTS

6.1	Precautions on Creating the GPIB Program .....	6-3
6.2	Sample Programs .....	6-4
6.2.1	Initializing MS2602A .....	6-4
6.2.2	Reading the frequency and level at marker point .....	6-6
6.2.3	Reading trace data .....	6-8
6.2.4	Transferring 2-byte binary data .....	6-11
6.2.5	Delta marker .....	6-13
6.2.6	Multimarker function .....	6-15
6.2.7	Gate functions .....	6-17
6.2.8	Saving and recalling data .....	6-20
6.2.9	Adjacent-channel leakage power measurement .....	6-22
6.2.10	Occupied frequency bandwidth measurement .....	6-23
6.2.11	Setting template data .....	6-24
6.2.12	Measuring template .....	6-26
6.2.13	Burst wave average power measurement .....	6-27

(Blank)

## SECTION 6 SAMPLE PROGRAMS

### 6.1 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 ( INI or *RST )</li> </ul>
2	Turn the device to the remote state of RWLS ( Remote With Lockout State ).	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 &H3F, &H11 to prevent it from returning to the local state. ( Turn every device to the locally controlled state with WBYTE &H3F, listener address, secondary address, &H01. )
3	Do not send any command ( related to the device ) other than the INPUT @ statement immediately after sending a query.	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.
4	Create a program that avoids an exception processing of the protocol	Avoid stoppage of execution ( caused by an error ) by means of providing a program with exception-processing section against exceptions that can be foreseen.
5	Confirm the interface functions of each device ( subset ).	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 IEEE488.2.

## 6.2 Sample Programs

## 6.2.1 Initializing MS2602A

&lt; Example 1 &gt; Initializes MS2602A

```

10 '-----
20 ' MS2602A 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 MS2602A address to variable SPA
90 PRINT @SPA;"INI" ..... Initializes MS2602A
100 END

```

The parameters initialized by the above program are shown in Appendix A. The table below lists part of Appendix A.

Table of MS2602A Device-Dependent Initial Settings ( part of Appendix A ) ( 1 / 2 )

Group	Brief function	Function item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
Frequency	Selects the mode for setting the frequency band	FREQUENCY MODE	START-STOP		
	Sets the start frequency	START FREQUENCY	0 Hz	-----	0 Hz
	Sets the center frequency	CENTER FREQUENCY	4.25 GHz		4.25 GHz
	Sets the stop frequency	STOP FREQUENCY	8.50 GHz	-----	8.50 GHz
	Sets the frequency span	FREQUENCY SPAN	8.50 GHz	*0 Hz	8.50 GHz
	Sets the center-frequency step size	CENTER FREQ STEP SIZE	1 GHz		
Level	Sets the reference level	REFERENCE LEVEL	- 10 dBm		
	Sets the reference-level step size	REF LEVEL STEP SIZE	AUTO / 1 div		
	Selects the mode for setting the Y axis	SCALE MODE	LOG	LOG	LOG
	Set the LOG scale	LOG SCALE	10 dB / div	10 dB / div	*10 dB / div
Coupled function	Selects the mode for setting the resolution bandwidth	RESOLUTION BANDWIDTH	AUTO	AUTO	*AUTO
	Selects the mode for setting the video bandwidth	VIDEO BANDWIDTH	AUTO	AUTO	*AUTO
	Selects the mode for setting the sweep time	SWEEP TIME	AUTO	AUTO	*AUTO
	Selects the mode for setting the RF attenuator	RF ATTENUATOR	AUTO		

Table of MS2602A Device-Dependent Initial Settings ( part of Appendix A ) ( 2 / 2 )

Group	Brief function	Function item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
Display mode	Selects the display format	DISPLAY MODE	TRACE-A		
Marker function	Selects the marker mode	MARKER MODE	NORMAL		
	Specifies the zone-marker center	ZONE MARKER CENTER	250 point	250 point	250 point
	Specifies the zone-marker width	ZONE MARKER WIDTH	501 point	*1 point	501 point
	Multimarker mode	MULTI MARKER MODE	OFF		

An initial value appended with "\*" represents a fixed value.

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' command is identical to the 'INI' command.

For general usage of INI and \*RST, first initialize the MS2602A device functions with the IP or INI command, then use the program commands to set only the functions to be changed. This prevents the MS2602A from being controlled while unnecessary functions are set.

## 6.2.2 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  ' MS2602A 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"
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

Sending data from talker → 000 501 251 000 → FREQ → 501 251 000

| ← ASCII codes including 12 characters → |  
 ||  
 30H 30H 30H 35H 30H 31H 32H 35H 31H 30H 30H 30H

### ■ LEVEL

Sending data from talker → -015.53 → LEVEL → -15.53

| ← ASCII codes including 7 characters → |  
 ||  
 2DH 30H 31H 35H 2EH 35H 33H

Program execution result of <Example 2>

MARKER    FREQ = 501.251MHz  
 LEVEL = -15.53dBm

## 6.2.3 Reading trace data

< Example 3-1 > Reads the trace level at all points when CF and SPAN are set to 500 MHz and 10 MHz respectively.

```

10  '-----
20  ' MS2602A GPIB SAMPLE PROGRAM
30  ' OUTPUT TRACE DATA (ASCII)
40  '-----
50  ISET IFC
60  ISET REN
70  CMD DELIM=0
80  LET SPA=1
90  PRINT @SPA;"INI"
100 DIM TRACE(500) ..... Declares array variable TRACE ( )
110 PRINT @SPA;"CNF 500MHZ"
120 PRINT @SPA;"SPF 10MHZ"
130 PRINT @SPA;"BIN 0" ..... Specifies ASCII as the response data format
140 PRINT @SPA;"SWP" ..... Executes a single sweep
150 FOR I=0 TO 500
160     PRINT @SPA;"XMA?" + STR$(I) + ",1" ..... Queries the trace data
170     INPUT @SPA;TRC$ ..... Reads the trace data
180     TRACE(I)=VAL(TRC$) ..... Converts ASCII into numeric data
190     PRINT USING "####.##dBm";TRACE(I)/100
200 NEXT I
210 END

```

The "BIN\_0" at line 130 is a command for specifying ASCII as the response data format. The ASCII or BINARY transfer format can be specified for the "XMA?", "XMB?", "XMG?", and "XMT?" queries for reading trace data. An example of BINARY format is shown in <Example 3-2>.



< Example 3-2 > Transfers data in BINARY format.

```

10  '-----
20  ' MS2602A GPIB SAMPLE PROGRAM
30  ' OUTPUT TRACE DATA (BINARY)
40  '-----
50  ISET IFC
60  ISET REN
70  CMD DELIM=0
80  PRINT @1;"INI"
90  DIM TRACE(500)
100 PRINT @1;"CNF 500MHZ"
110 PRINT @1;"SPF 10MHZ"
120 PRINT @1;"BIN 1" ..... Specifies BINARY as the response data format
130 PRINT @1;"TRM 1" ..... Sets the terminator to CR + LF
140 PRINT @1;"SWP" ..... Executes a single sweep
150 PRINT @1;"XMA? 0,501" ..... Queries the trace data
160 WBYTE &H3F,&H5F,&H20,&H41;
      ..... Specifies PC9801 as listener address 0, and MS2602A as a talker address 1
170 FOR I=0 TO 500
180 RBYTE ;UPRBYTE,LWRBYTE ..... Reads 2-byte binary data in sequence
190 TRACE(I)=UPRBYTE*256+LWRBYTE ..... Converts 2-byte data to decimals
200 IF UPRBYTE>=128 THEN TRACE(I)=TRACE(I)-65536
      ..... Converts negative data to decimals
210 PRINT USING "####.##dBm";TRACE(I)/100
220 NEXT I
230 RBYTE ;DMY1,DMY2 ..... Receives terminator data
240 WBYTE &H3F,&H5F; ..... Releases listener and talker
250 END

```

The "BIN\_1" at line 120 specifies BINARY as the response data format. The terminator indicating that the last data byte transmission is completed is set to CR + LF at line 130. After a single sweep is executed at line 140, the XMA? at line 150 queries the reading trace data. At line 160, the PC9801 ( address 0 ) is specified as a listener, and the MS2602A ( address 1 ) as a talker. At line 180, the 2-byte binary data is read by the PC9801 in sequence from the high-order byte to the low-order byte. At line 190, the 2-byte binary data is converted into decimals to assign them to variable TRACE ( 1 ). Line 200 allows data to be correctly read even if the data is negative. The FOR...NEXT statement at lines 170 to 220 converts negative data to decimals repeatedly 501 times.

At line 230, 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 240 is used to release the talker and listener.

Interface message at line 160

```

WBYTE  &H3F,  &H5F,  &H20,  &H41 ;
        [ ]    [ ]    [ ]    [ ]
        UNL   UNT   MLA   MTA

```

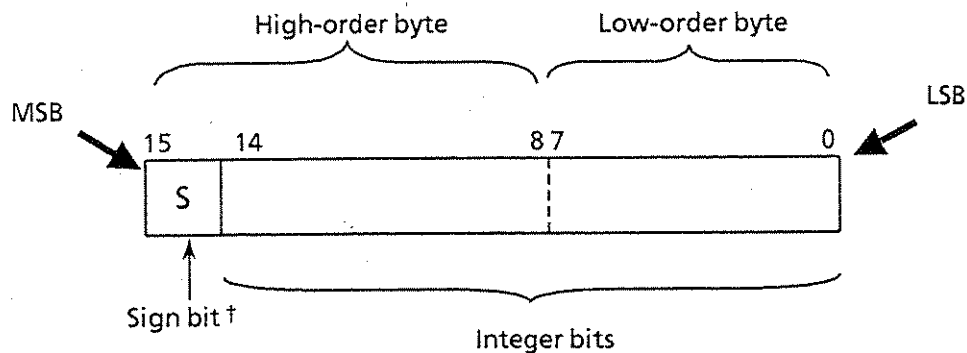
- UNL=Unlisten: Specifies all devices so that the listener status can be released.
- UNT=Untalk: Specifies all devices so that the talker status can be released.
- MLA=My Listen Address: If a device receives the listener address code ( listener address command – MLA ) corresponding to the address ( address 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.

### 6.2.4 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

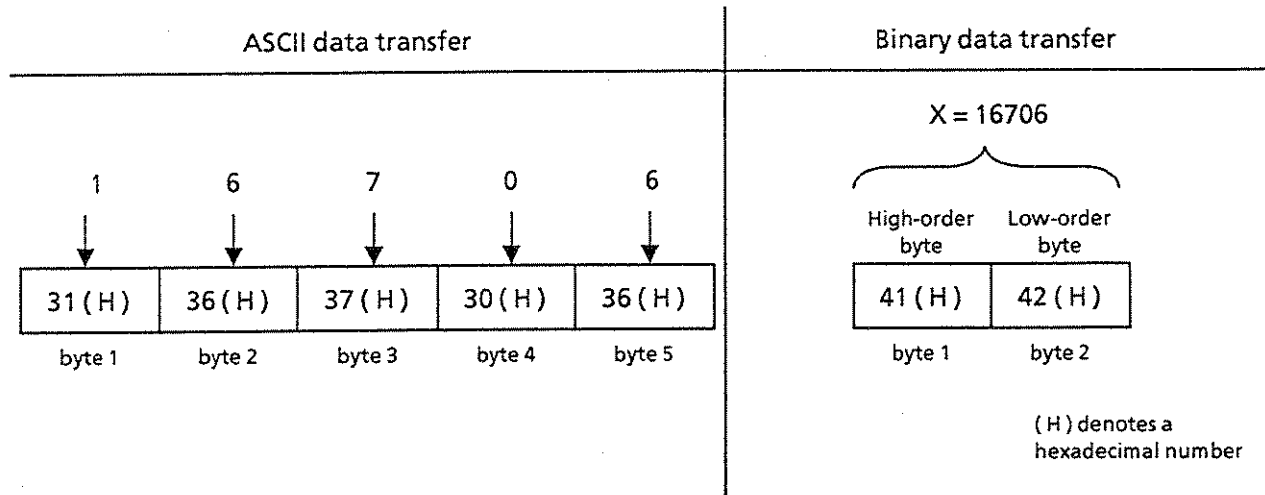
Sign bit: Bit 15 (MSB) is used  
 Integer bits: Bits 0 through 14 are used. Thus, 16 bits (2 bytes) are used.



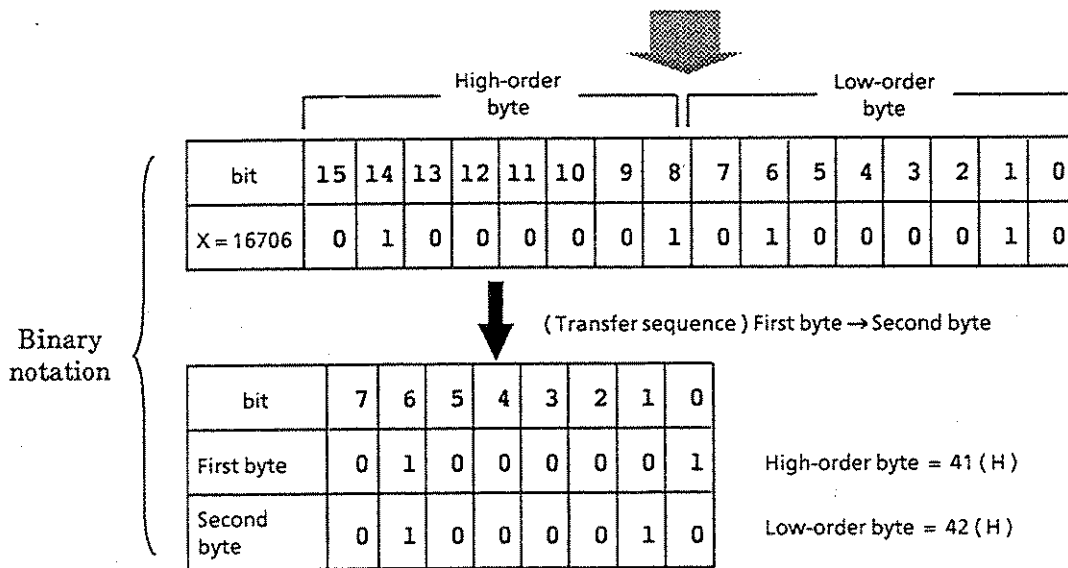
Internal Expression of 2-byte Binary Data

† 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.



$$16706 (D) = 4 \times 16^3 + 1 \times 16^2 + 4 \times 16^1 + 2 \times 16^0$$



### 6.2.5 Delta marker

< Example 4 > Using a delta marker, reads out the frequency and level differences between a peak point and the next peak point.

```

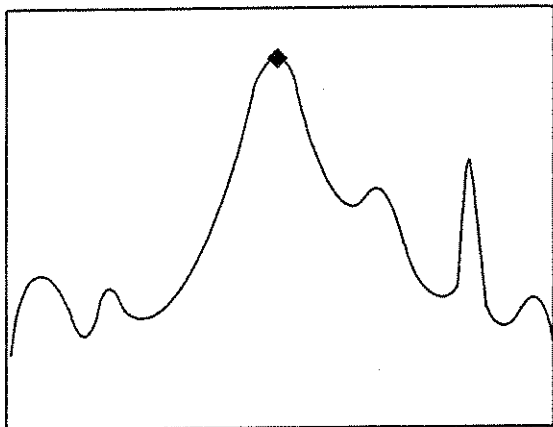
10  '-----
20  ' MS2602A GPIB SAMPLE PROGRAM
30  ' OUTPUT DELTA FREQ&LEVEL
40  '-----
50  ISET IFC
60  ISET REN
70  CMD DELIM=0
80  LET SPA=1
90  PRINT @SPA;"INI"
100 PRINT @SPA;"CNF 500MHZ"
110 PRINT @SPA;"SPF 100KHZ"
120 PRINT @SPA;"SWP" ..... Executes a single sweep
130 PRINT @SPA;"MKR 0" ..... Sets marker mode to NORMAL
140 PRINT @SPA;"MKS 0" ..... Executes PEAK SEARCH
150 PRINT @SPA;"MKR 1" ..... Sets marker mode to DELTA
160 PRINT @SPA;"MKS 1" ..... Executes NEXT PEAK SEARCH
170 PRINT @SPA;"MKF?" ..... Queries reading frequency difference
180 INPUT @SPA;DFREQ
190 PRINT @SPA;"MKL?" ..... Queries reading level difference
200 INPUT @SPA;DLEVEL
210 PRINT USING "DELTA FREQ=####.####kHz";DFREQ/1000
220 PRINT USING "      LEVEL=####.## dB";DLEVEL
230 END

```

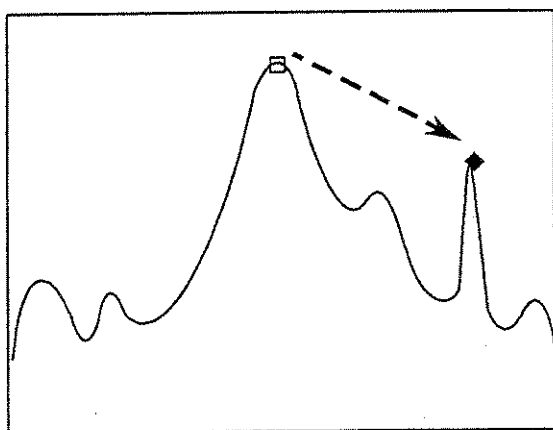
The "MKR\_1" at line 150 is used to set the marker mode to DELTA, so that the reference marker can also be set together to the current marker position.

The "MKS\_1" at line 160 sets the marker search to NEXT PEAK to move the current marker to NEXT PEAK point.

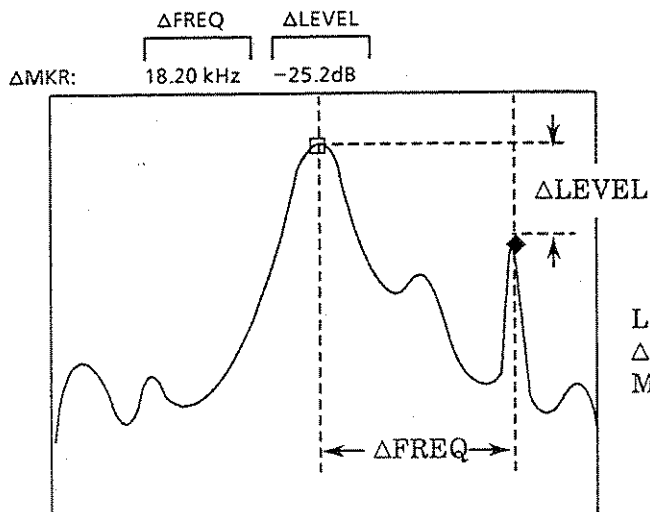
The "MKF?" and "MKL?" at lines 170 and 190 query reading the frequency and level at the current marker position while the marker mode is NORMAL. It is also used to query reading the frequency and level differences between the current marker and the reference marker while the marker mode is DELTA.



Executing PEAK SEARCH at line 140 allows the current marker to be set to the peak point



Line 150 allows the reference marker to be set together to the current marker position. Executing NEXT PEAK SEARCH at line 160 allows the current marker to move to the next peak point as shown to the left.



Lines 170 to 200 read out the  $\Delta$ FREQ and  $\Delta$ LEVEL displayed in the upper left of the MS2602A screen

### 6.2.6 Multimarker function

< Example 5-1 > Using the multimarker function, measures the frequency / level at 10 points in descending order.

```

10 '-----
20 ' MS2602A GPIB SAMPLE PROGRAM
30 ' MULTI MARKER HIGHEST10
40 '-----
50 ISET IFC
60 ISET REN
70 CMD DELIM=0
80 LET SPA=1
90 PRINT @SPA;"INI"
100 PRINT @SPA;"CNF 500MHZ"
110 PRINT @SPA;"SPF 100KHZ"
120 PRINT @SPA;"SWP"
130 PRINT @SPA;"MKMHI" ..... Executes multimarker mode HIGHEST 10
140 FOR I=1 TO 10
150     PRINT @SPA;"MKMP?" + STR$(I) ..... Queries reading frequency at the marker point
160     INPUT @SPA;FREQ
170     PRINT @SPA;"MKML?" + STR$(I) ..... Queries reading level at the marker point
180     INPUT @SPA;LEVEL
190     PRINT USING "MKR No. ## ####.####MHz ####.##dBm";
        I, FREQ/1E+06, LEVEL
200 NEXT I
210 END

```

The MS2602A multimarker function allows up to ten markers to be set at a time. The "MKMHI" at line 130 is used to set the multimarker to HIGHEST 10 mode which sets up to ten markers in descending order.

The frequency and level at each marker are read out by lines 140 to 200.

This program allows harmonics to be observed if the program is modified. < Example 5-2 > shows the program for observing the harmonics from a fundamental to the fifth order.

< Example 5-2 > Harmonic frequency measurement ( measures 500 MHz fundamental and up to its fifth order harmonics )

```

10  '-----
20  ' MS2602A GPIB SAMPLE PROGRAM
30  ' MULTI MARKER HARMONICS
40  '-----
50  ISET IFC
60  ISET REN
70  CMD DELIM=0
80  LET SPA=1
90  PRINT @SPA;"INI"
100 PRINT @SPA;"STF 0HZ" ..... Sets the start frequency
110 PRINT @SPA;"SOF 3000MHZ" ..... Sets the stop frequency
111 PRINT @SPA;"MKZF 500MHZ" ..... Sets the zone marker center frequency
120 PRINT @SPA;"SWP"
130 PRINT @SPA;"MKMHRM" ..... Sets multimarker mode to HARMONICS
140 FOR I=1 TO 5
150     PRINT @SPA;"MKMP?" + STR$(I)
160     INPUT @SPA;FREQ
170     PRINT @SPA;"MKML?" + STR$(I)
180     INPUT @SPA;LEVEL
190     PRINT USING "MKR No. ## ####.##MHz ####.##dBm";
        I, FREQ/1E+06, LEVEL
200 NEXT I
210 END

```

This program allows the frequency to be set using the START-STOP at line 100 and line 110. The "MKZF\_500MHZ" at line 111 moves the zone marker center to 500 MHz so that marker can capture a fundamental. ( In the initial state, the zone is positioned in the center of the screen. The "MKMHRM" at line 130 sets the multimarker to HARMONICS mode ( harmonic frequency measurement ). Respective frequencies and levels at five markers can be read out by setting the number of loops to 5 in the FOR...NEXT statement from line 140 to line 200. The other parts of this program are the same as < Example 5-1 > .



## 6.2.7 Gate functions

< Example 6 > Reads out spectrum data by observing the burst wave using the gate function.

```

10 '-----
20 ' MS2602A GPIB SAMPLE PROGRAM
30 ' GATE TRIGGER
40 '-----
50 ISET IFC
60 ISET REN
70 CMD DELIM=0
80 LET SPA=1
90 PRINT @SPA;"INI"
100 DIM TRACE(500)
110 PRINT @SPA;"CNF 500MHZ"
120 PRINT @SPA;"SPF 1MHZ"
130 PRINT @SPA;"RB 100KHZ" ..... Sets RBW
140 PRINT @SPA;"TRG 3" ..... Sets the trigger input to EXT
150 PRINT @SPA;"EXTTYPE 10V" ..... Sets EXT to INPUT 1
160 PRINT @SPA;"TRGLVL 0" ..... Sets the trigger level to 0
170 PRINT @SPA;"GD 50US" ..... GATE DEDAY = 50us
180 PRINT @SPA;"GL 400US" ..... GATE LENGTH = 400us
190 PRINT @SPA;"GE INT" ..... GATE END = INTERNAL
200 PRINT @SPA;"GATE ON" ..... Sets the GATE function to ON
210 FOR TIMER=0 TO 25000:NEXT TIMER ..... Sets the timer
220 FOR I=0 TO 500
230   PRINT @SPA;"XMA?" + STR$(I) + ",1"
240   INPUT @SPA;TRC$
250   TRACE(I)=VAL(TRC$)
260   PRINT USING "####.##dBm";TRACE(I)/100
270 NEXT I
280 END

```

When the burst waveform shown in Fig.6-1 is observed, the spectrum shown in Fig.6-2 ( a ) is output. This function can conveniently be used to observe the spectrum of the ON interval ( interval shown by A in Fig.6-1 ) in this waveform. The synchronous signal shown in Fig.6-1 is input to the " INPUT 1 " connector on the MS2602A rear panel. The spectrum is displayed using this synchronous signal and setting conditions.

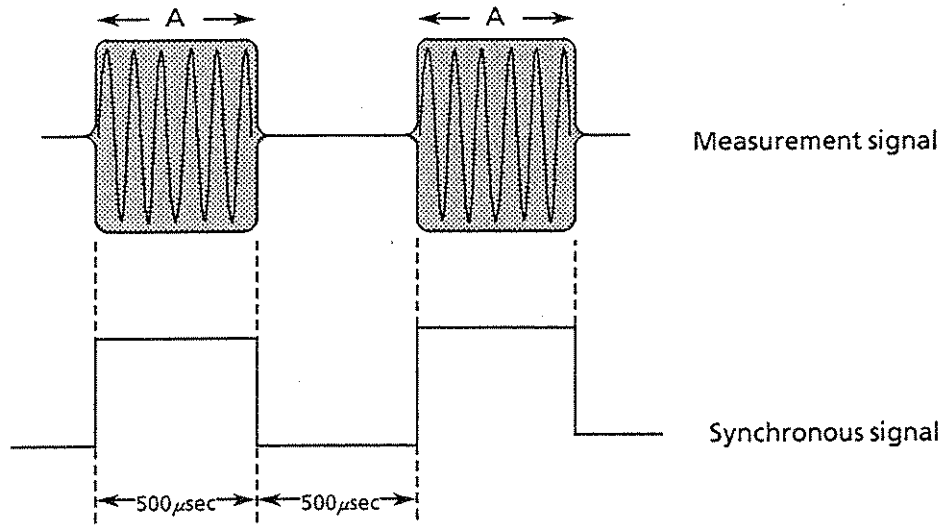


Fig.6-1 Burst Waveform

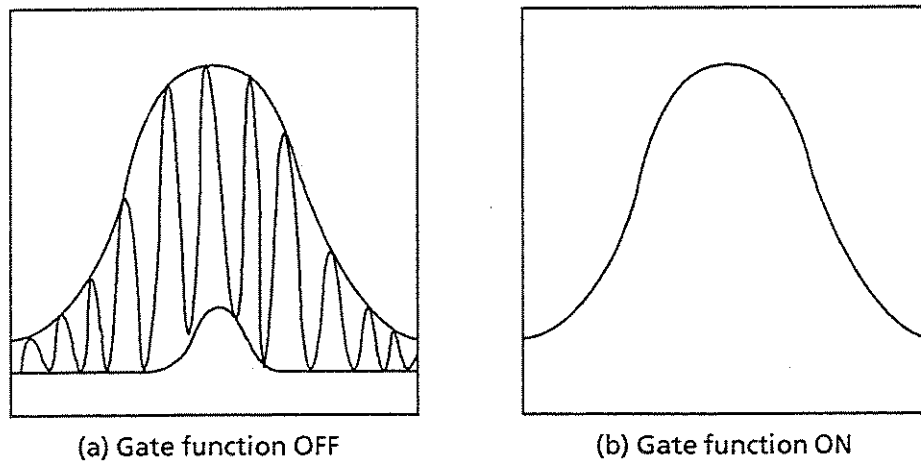


Fig.6-2 Burst Wave Spectrum

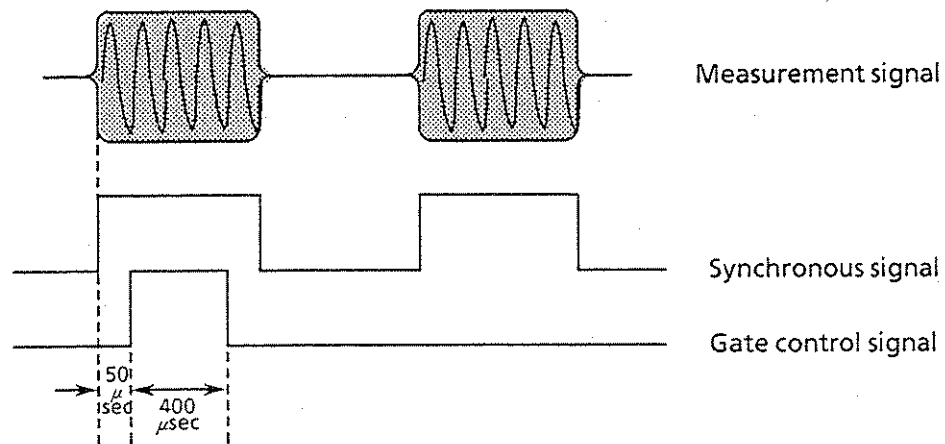


Fig.6-3 Sample Program for Gate-Control Signal Generation Timing

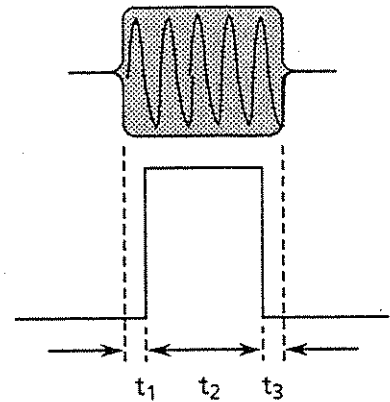
The RBW command at line 130 sets RBW to the optimum value depending on the GATE conditions ( GATE DELAY:  $t_1$ , GATE LENGTH:  $t_2$  ) as shown in Table 6-1 below.

The block from lines 140 to 160 sets the trigger signal, and the block from lines 170 to 190 sets the gate conditions. The gate function is set to ON at line 200 provided the timer is set because it takes time to form a perfect waveform which is fully connected.

The block from lines 220 to 270 allows trace data to be output by the " XMA? " query. The spectrum can be observed as shown in Fig. 6-2 ( b ) by executing this program.

Table 6-1 RBW Optimum Values

RBW	$t_1$	$t_2$	$t_3$
1 kHz	$\cong 2 \text{ msec}$	$\cong 20 \mu\text{sec}$	$\cong 1 \mu\text{sec}$
3 kHz	$\cong 600 \mu\text{sec}$		
10 kHz	$\cong 230 \mu\text{sec}$		
30 kHz	$\cong 200 \mu\text{sec}$		
100 kHz	$\cong 20 \mu\text{sec}$		
300 kHz	$\cong 15 \mu\text{sec}$		
1 MHz 3 MHz	$\cong 10 \mu\text{sec}$		



## 6.2.8 Saving and recalling data

< Example 7 > Saves and recalls data to and from PMC.

### ■ Saving data

```

1000 *SAVPMC
1010 '-----
1020 ' MS2602A GPIB SAMPLE PROGRAM
1030 ' SAVE TO PMC(FD) SUBROUTINE
1040 '-----
1050 INPUT "INPUT TITLE";TTL$ ..... Inputs title
1060 PRINT @SPA;"TITLE '"+TTL$+"'" ..... Sets title
1070 MEDIA$="IPMC" ..... MEDIA = INTPMC
1080 PRINT @SPA;"PMCS "+MEDIA$ ..... Sets the MEDIA
1090 INPUT "FILE No.";FILE ..... Inputs FILE No.
1100 PRINT @SPA;"SVM"+STR$(FILE) ..... Saves data
1100 RETURN

```

### ■ Recalling data

```

1200 *RCLPMC
1210 '-----
1220 ' MS2602A GPIB SAMPLE PROGRAM
1230 ' RECALL FROM PMC SUBROUTINE
1240 '-----
1250 MEDIA$="IPMC"
1260 PRINT @SPA;"PMCS "+MEDIA$
1270 INPUT "SELECT RECALL DATA 1=TRACE&PARAM 2=PARAM";RCD
..... Selects data to be recalled
1280 IF RCD=2 THEN RCDATA$="P" ELSE RCDATA$="TP"
1290 PRINT @SPA;"RDATA "+RCDATA$ ..... Sets data to be recalled
1300 INPUT "FILE No.";FILE ..... Inputs file No.
1310 PRINT @SPA;"RCM"+STR$(FILE) ..... Recalls data
1320 RETURN

```

These two programs are used as subroutines called from other programs. Each subroutine can be called by placing GOSUB \*SAVPMC or GOSUB \*RCLPMC at the line number where the program data is to be saved or restored.

< Example >

```

      :
200 PRINT @SPA;"SWP"
210 GOSUB *SAVPMC
      :
```

The block from lines 1050 to 1060 sets the title. When the saved data is displayed if the title has been set, this title is also displayed. This can conveniently be used to find data.

The block from lines 1070 to 1080 sets the media to be used for saving to the internal PMC. By changing the data at line 1070 or by specifying a media with the INPUT statement, it is possible to save data to an external PMC or floppy disk ( MC8104A is required ). FILE No. is input at line 1090 and data is saved to the FILE No. at line 1100.

Line 1270 selects the data to be recalled for trace data including parameters or parameters only. Line 1290 declares the item to be recalled to MS2602A, and the specified file is recalled at lines 1300 to 1310.

### 6.2.9 Adjacent-channel leakage power measurement

< Example 8 > Subroutine for adjacent-channel leakage power measurement

```

1000 *ADJ
1010 '-----
1020 ' MS2602A GPIB SAMPLE PROGRAM
1030 ' ADJ CH MEASURE SUBROUTINE
1040 '-----
1050 CH$="BOTH"
1060 CHBW$="8.5KHZ"
1070 CHSP$="12.5KHZ"
1075 CHSPF$="25.0KHZ"
1080 PRINT @SPA;"ADJCH "+CH$ ..... Sets the adjacent channel
1090 PRINT @SPA;"ADJCHBW "+CHBW$ ..... Sets the adjacent-channel bandwidth
1100 PRINT @SPA;"ADJCHSP "+CHSP$
1105 PRINT @SPA;"ADJCHSPF "+CHSPF$
..... Sets the adjacent-channel separation

1108 PRINT @SPA;"MADJMOD MOD" ..... Selects R: Total Power
1110 PRINT @SPA;"SWP"
1120 PRINT @SPA;"MEAS ADJ, EXE" ..... Measures the adjacent-channel leakage power
1130 PRINT @SPA;"RES?" ..... Queries the measurement data
1140 INPUT @SPA;LWLEV1, UPLEV1, LWLEV2, UPLEV2
1150 PRINT USING "UPPER SIDE CH1 LEVEL=####.##dBm"; UPLEV1
1160 PRINT USING "LOWER SIDE CH1 LEVEL=####.##dBm"; LWLEV1
1170 PRINT USING "UPPER SIDE CH2 LEVEL=####.##dBm"; UPLEV2
1180 PRINT USING "LOWER SIDE CH2 LEVEL=####.##dBm"; LWLEV2
1190 RETURN

```

This program is a subroutine, which requires the center frequency and frequency span to be set to appropriate values in the main program.

The block from lines 1050 to 1075 assigns adjacent-channel setting values to variables. Both the upper and lower channels to be measured here lie in the 8.5 kHz channel 1 width with 12.5 kHz channel separation channel 2 width with 25.0 kHz channel separation. The block from lines 1080 to 1105 sends the program commands for the adjacent channel to set the MS2602A. After the sweep is executed by the "SWP" command at line 1110, the adjacent-channel leakage power is measured at line 1120. Line 1130 queries reading the measured value at line 1140.

The program in <Example 8> for measuring a modulated wave relative to the total power can be changed to a program for measurement relative to the reference level by rewriting line 1108 as shown below:

```
1108 PRINT @SPA;"MADJMOD UNMD"
```

In this case, perform the following operations before activating this subroutine.

Put the input signal in the unmodulated state and execute PEAK → CF, PEAK → REF. Then return to the modulated state.

### 6.2.10 Occupied frequency bandwidth measurement

< Example 9 > Subroutine for occupied frequency bandwidth measurement using N% of POWER method

```

1000 *OBW
1010 '-----
1020 ' MS2602A GPIB SAMPLE PROGRAM
1030 ' OBW MEASURE SUBROUTINE
1040 '-----
1050 NPC=99 ..... Assigns 99 to variable NPC
1060 PRINT @SPA;"OBWN"+STR$(NPC) ..... Sets N%
1065 PRINT @SPA;"MOBW N" ..... Selects N% method
1070 PRINT @SPA;"DET SMP" ..... Sets the detection mode to SAMPLE
1080 PRINT @SPA;"VAVG 16" ..... Sets the averaging rate to 16
1090 PRINT @SPA;"VAVG ON" ..... Sets averaging to ON
1100 FOR I=1 TO 16
1110 PRINT @SPA;"SWP" ..... Repeats the sweep 16 times
1120 NEXT I
1130 PRINT @SPA;"MEAS OBW, EXE" ..... Measures the occupied frequency bandwidth
1140 PRINT @SPA;"RES?"
1150 INPUT @SPA;OBWFREQ,CNTRFRQ
1160 PRINT USING "CENTER FREQ=####.###MHz";CNTRFRQ/1E+06
1170 PRINT USING "##%BW FREQ=####.###kHz";NPC,OBWFREQ/1000!
1180 RETURN

```

Line 1050 assigns the N% value to a simple numeric variable NPC to set N = 99% in < Example 9 > by sending the OBWN command for setting the occupied frequency bandwidth to MS2602A at line 1060 to 1065. Line 1070 sets the detection mode to SAMPLE. Lines 1080 and 1090 set the averaging rate to 16 and averaging to ON respectively.

The FOR...NEXT statement at lines 1100 to 1120 sends the " SWP " command 16 times to repeat the sweep by the required number of times for averaging processing. Line 1130 measures the occupied frequency bandwidth of the averaging-processed waveform. Line 1140 queries reading the occupied frequency bandwidth and the center frequency of the frequency bandwidth at line 1150.

To make a measurement using X dB DOWN, rewrite four lines as shown below:

```

...
1050 XDB=25
1060 PRINT @SPA;"OBWXDB"+STR$(XDB)
1065 PRINT @SPA;"MOBW XDB"
...
1170 PRINT USING "##dB DOWN FREQ=####.###kHz";XDB,
      OBWFREQ/1000
...

```

## 6.2.11 Setting template data

&lt; Example 10 &gt; Subroutine for template data

```

1000 *SETTMP
1010 '-----
1020 ' MS2602A GPIB SAMPLE PROGRAM
1030 ' SET TEMPLATE DATA SUBROUTINE
1040 '-----
1050 TMPNO=1 ..... Assigns template No.1 to variable TMPNO
1060 PRINT @SPA;"MTEMP "+STR$(TMPNO)..... Sets the template No.
1070 PRINT @SPA;"MTEMPINI UP1"
1080 PRINT @SPA;"MTEMPINI LW1"
1090 PRINT @SPA;"MTEMPL UP1" ..... Selects LIMIT LINE 1 UPPER
1100 RESTORE *LMTUP1 ..... Specifies the data line for reading
1110 READ N ..... Reads the number of template data points
1120 FOR I=1 TO N
1130     READ TMS,LEV$ ..... Reads the template data (time, level)
1140     PRINT @SPA;"MTEMPIN"+STR$(I)+", "+TMS+", "+LEV$
..... Sets the template data

1150 NEXT I
1160 PRINT @SPA;"MTEMPL LW1" ..... Selects LIMIT LINE 1 LOWER
1170 RESTORE *LMTLW1 ..... Specifies the data line to be read
1180 READ N ..... Reads the number of template data points
1190 FOR J=1 TO N
1200     READ TMS,LEV$ ..... Reads the template data (time, level)
1210     PRINT @SPA;"MTEMPIN"+STR$(J)+", "+TMS+", "+LEV$
..... Sets the template data

1220 NEXT J
1230 *LMTUP1
1240 '== LIMIT LINE1 UPPER DATA ==
1250 DATA 8
1260 DATA "-50US", "-65.2DBM"
1270 DATA "-24US", "-65.2DBM"
1280 DATA "-24US", "18.8DBM"
1290 DATA "6.643MS", "18.8DBM"
1300 DATA "6.643MS", "-45.2DBM"
1310 DATA "6.714MS", "-45.2DBM"
1320 DATA "6.714MS", "-65.2DBM"
1330 DATA "6.950MS", "-65.2DBM"
1340 *LMTLW1
1350 '== LIMIT LINE1 LOWER DATA ==

```

..... Template data



```

1360 DATA 4
1370 DATA "95US", "-200DBM"
1380 DATA "95US", "0.8DBM"
1390 DATA "6.524MS", "0.8DBM"
1400 DATA "6.524MS", "-200DBM"
1410 RETURN

```

..... Template data

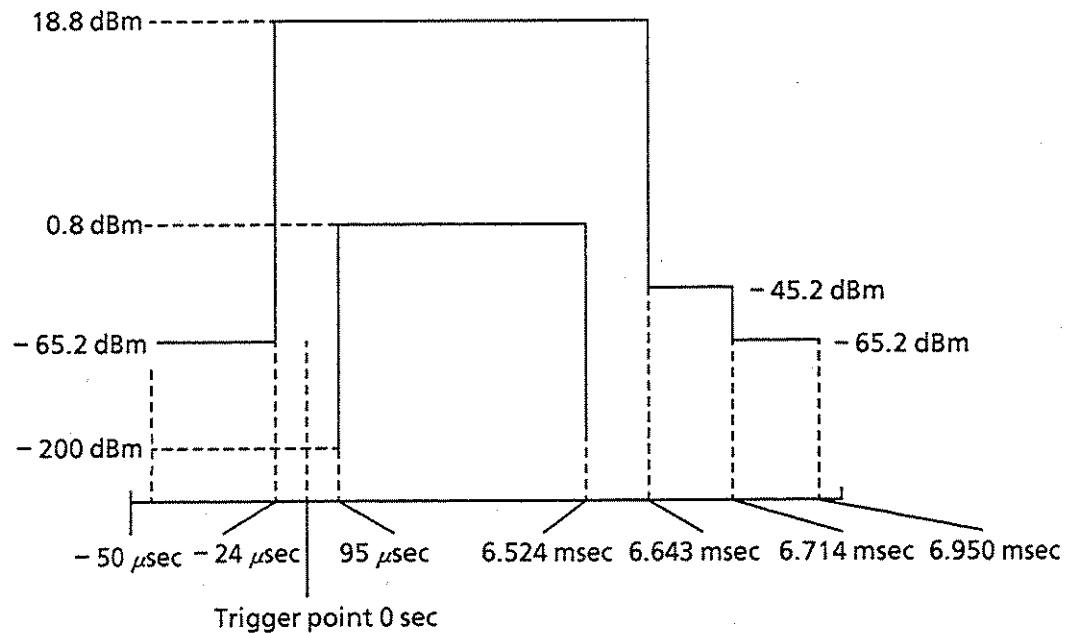


Fig. 6-4 Setting Data

The block from lines 1050 to 1060 selects the template No. to be set. The block from lines 1070 to 1080 initializes the current data settings. The block from lines 1090 to 1150 sets LIMIT LINE 1 UPPER. Line 1100 specifies the line where setting data is written.

Line 1110 reads the number of data points to set the number of loops to N in the FOR...NEXT statement at lines 1120 to 1150. Various data settings are read in the FOR...NEXT block.

The block from lines 1160 to 1220 sets LIMIT LINE 1 LOWER like the block from lines 1090 to 1150.

The block from lines 1230 to 1400 contains the DATA statements for setting the data included in these lines as template data. Line 1230 and line 1340 are label lines for the RESTORE statement. Each data item in line 1250 and line 1360 is numeric, and shows the number of data points. In the DATA statements following the DATA statement with this numeric data, the string expressions are listed as string data with units in order of time and level.

## 6.2.12 Measuring template

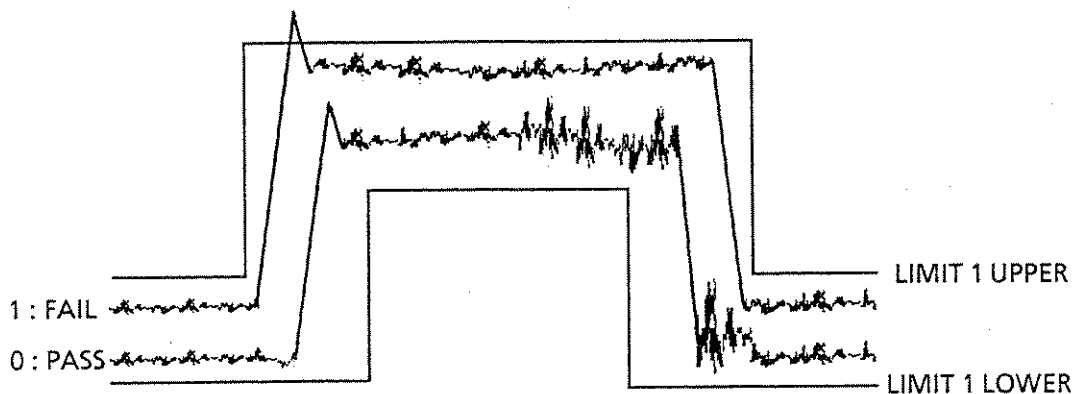
< Example 11 > Subroutine for template measurement

```

1000 *MEASTMP
1010 '-----
1020 ' MS2602A GPIB SAMPLE PROGRAM
1030 ' TEMPLATE MEASURE SUBROUTINE
1040 '-----
1050 PRINT @SPA;"MTEMP 1" ..... Sets the template No.
1060 PRINT @SPA;"TEMPSLCT UP1,ON" ..... Sets the template to ON
1070 PRINT @SPA;"TEMPSLCT LW1,ON"
1080 PRINT @SPA;"SWP"
1090 PRINT @SPA;"MEAS TEMP,CHECK" ..... Measures the template
1100 PRINT @SPA;"RES?"
1110 INPUT @SPA;CHK1$,CHK2$
1120 PRINT "LIMIT LINE 1"
1130 IF CHK1$="0" THEN PRINT "CHECK PASS!" ELSE PRINT "CHECK FAIL!"
1140 RETURN

```

This subroutine checks whether or not a burst signal waveform exits within a go / no-go line using the set template data. Line 1050 specifies the template No. used for a go / no-go decision. Line 1060 and line 1070 specify LIMIT 1 UPPER and LIMIT 1 LOWER as limit lines respectively. Line 1090 executes template measurement, and line 1100 queries reading data at 1110.



When part of a waveform is beyond LIMIT LINE, a response of " 1 " is generated to indicate FAIL. When the waveform is not beyond LIMIT LINE, a response of " 0 " is generated to indicate PASS.

### 6.2.13 Burst wave average power measurement

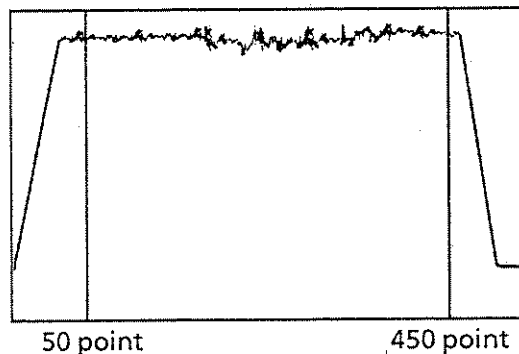
< Example 12 > Subroutine for burst wave average power measurement

```

1000 *MEASPWR
1010 '-----
1020 ' MS2602A GPIB SAMPLE PROGRAM
1030 ' BURST POWER MEAS SUBROUTINE
1040 '-----
1050 PWSTART=50 ..... Measurement-starting point = 50 point
1060 PWSTOP=450 ..... Measurement-stopping point = 450 point
1070 PRINT @SPA;"PWRSTART"+STR$(PWSTART) ..... Sets the starting point
1080 PRINT @SPA;"PWRSTOP"+STR$(PWSTOP) ..... Sets the stopping point
1090 PRINT @SPA;"SWP"
1100 PRINT @SPA;"MEAS POWER, EXE" ..... Measures the power
1110 PRINT @SPA;"RES?"
1120 INPUT @SPA;PWRDB PWRW
1130 PRINT USING "####.##dBm ####.##mW";PWRDB,PWRW/1E+09
1140 RETURN

```

This program is a subroutine that measures the burst wave average power. Line 1050 and line 1060 assign 50 and 450 to numeric variables PWSTART and PWSTOP respectively to set the measurement start and stop points on the screen display at lines 1070 and 1080. After the sweep is executed at line 1090, the average power is measured at line 1100. Data can be obtained as a value with dBm units or pW units.



When a waveform is displayed on the screen as shown in the left diagram (TIME domain), the average power between 50 point and 450 point is measured

Before calling the subroutine, set the center frequency and time delay.

< Example >

```

100 PRINT @SPA;"CNF 500MHZ" ..... Sets the center frequency
110 PRINT @SPA;"SPF 0HZ" ..... Sets the frequency span to 0 Hz for time domain
120 PRINT @SPA;"TSP 550US" ..... Sets the time span
130 PRINT @SPA;"TDLY 25US" ..... Sets the time delay
140 GOSUB *MEASPWR

```

( Blank )

## SECTION 7

### TABLE OF DEVICE MESSAGES

This section gives the tables of device messages, which can be used for the MS2602A, classified by the function type according to the contents. For detailed descriptions of commands, see SECTION 8, "DETAILED DESCRIPTION OF COMMANDS."

#### TABLE OF CONTENTS

Frequency / Amplitude .....	7-3
Display function .....	7-6
Signal search .....	7-11
Marker function .....	7-11
Coupled function .....	7-15
Sweep function .....	7-18
Save / Recall .....	7-20
Hard copy .....	7-21
Measure function .....	7-22
Sound .....	7-27
Parameter setting .....	7-27
System setting .....	7-28
Calibration .....	7-28
GPIB .....	7-28
RS-232C .....	7-29
Title .....	7-29
CAL / UNCAL .....	7-29
Spectrum data .....	7-30
PMC .....	7-30
PTA control .....	7-30
Others .....	7-31
Common command and event status .....	7-32

**( Blank )**

Table of MS2602A Device Message (1/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Frequency/ Amplitude</b>	<b><u>FREQUENCY / AMPLITUDE</u></b>			
<b>● Frequency</b>	<b><u>FREQUENCY</u></b>			
Selects the mode for setting frequency band	FREQ MODE CENTER-SPAN START-SPAN START-STOP	FRQ_0 FRQ_1 FRQ_2	FRQ? FRQ? FRQ?	FRQ_0 FRQ_1 FRQ_2
Sets center frequency	CENTER FREQ	CNF_f CF_f	CNF? CF?	CNF_f f
Steps up center frequency	FREQ STEP UP	FUP CF_UP	_____ _____	_____ _____
Steps down center frequency	FREQ STEP DOWN	FDN CF_DN	_____ _____	_____ _____
Sets start frequency	START FREQ	STF_f FA_f	STF? FA?	STF_f f
Sets stop frequency	STOP FREQ	SOF_f FB_f	SOF? FB?	SOF_f f
Sets frequency step size	FREQ STEP SIZE	FSS_f SS_f	FSS? SS?	FSS_f f
Sets scroll step size	SCROLL STEP SIZE 1 div 2 div 5 div 10 div	SSS_1 SSS_2 SSS_5 SSS_10	SSS? SSS? SSS? SSS?	SSS_1 SSS_2 SSS_5 SSS_10
Setting the Fine Adj	FINE ADJ ON OFF	FADJ_ON FADJ_OFF	_____ FADJ?	_____ OFF
Setting the Fine Adj frequency	ADJ	FADJ_f	FADJ?	f
<b>● Span</b>	<b><u>SPAN</u></b>			
Sets frequency span	FREQ SPAN	SPF_f SP_f	SPF? SP?	SPF_f f
Steps up frequency span	FREQ SPAN STEP UP	SPU SP_UP	_____ _____	_____ _____
Steps down frequency span	FREQ SPAN STEP DOWN	SPD SP_DN	_____ _____	_____ _____
Sets to full span	FULL SPAN	FS	_____	_____

Table of MS2602A Device Message (2/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Frequency/ Amplitude</b>				
<b>● Span</b>				
Sets to zero span	ZERO SPAN	SPF_0	SPF?	SPF_0
Selects one of the bands	BAND SELECT AUTO : 0 Hz to 8.5 GHz	BNDC_AUTO HNLOCK_OFF HNUNLK	BNDC? HNLOCK?	AUTO HNLOCK_OFF
	0 : 0 Hz to 2.0 GHz	BNDC_0 HNLOCK_0 HN_0	BNDC? HNLOCK? HN?	0 HNLOCK_ON 0
	1- : 1.7 to 7.5 GHz	BNDC_1- HNLOCK_1 HN_1	BNDC? HNLOCK? HN?	1- HNLOCK_ON 1
	1+ : 6.5 to 8.5 GHz	BNDC_1+ HNLOCK_2 HN_2	BNDC? HNLOCK? HN?	1+ HNLOCK_ON 2
<b>● Level</b>				
Sets reference level	REFERENCE LEVEL	RLV_l RL_l	RLV? RL?	RLV_l l
Steps up reference level	REF LEVEL STEP UP	RL_UP	—	—
Steps down reference level	REF LEVEL STEP DOWN	RL_DN	—	—
Sets LOG scale step size	LOG SCALE STEP SIZE MANUAL	LSS_l	LSS?	LSS_l
	AUTO	LSSA_1 LSSA_2 LSSA_5 LSSA_10	LSSA? LSSA? LSSA? LSSA?	LSSA_1 LSSA_2 LSSA_5 LSSA_10
	1 div			
	2 div			
	5 div			
Sets LOG scale	LOG SCALE RANGE	SCL_0 LG_1DB	SCL? LG?	SCL_0 1
	1 dB / div	SCL_1 LG_2DB	SCL? LG?	SCL_1 2
	2 dB / div	SCL_2 LG_5DB	SCL? LG?	SCL_2 5
	5 dB / div	SCL_3 LG_10DB	SCL? LG?	SCL_3 10
	10 dB / div	SCALE UP LG_UP	—	—
	SCALE DOWN	LG_DN	—	—



Table of MS2602A Device Message (3/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<p>■ <u>Frequency/Amplitude</u></p> <p>● <u>Level</u></p> <p>Sets LIN scale</p>	<p><u>FREQUENCY / AMPLITUDE</u></p> <p><u>AMPLITUDE</u></p> <p>SCALE LIN RANGE Switching LIN scale 1% / div 2% / div 5% / div 10% / div</p>	<p>LN LG_0 SCL_4 SCL_5 SCL_6 SCL_7</p>	<p>— — SCL? SCL? SCL? SCL?</p>	<p>— — SCL_4 SCL_5 SCL_6 SCL_7</p>
<p>Sets display unit system</p>	<p><u>DISPLAY UNIT</u></p> <p>dBm</p> <p>dB<math>\mu</math>V</p> <p>dBmV</p> <p>V</p> <p>dB<math>\mu</math>V(emf)</p> <p>W</p>	<p>UNT_0 AUNITS_DBM KSA UNT_1 AUNITS_DBUV KSC UNT_2 AUNITS_DBMV KSB UNT_3 AUNITS_V KSD UNT_4 AUNITS_DBUVE UNT_5 AUNITS_W</p>	<p>UNT? AUNITS? — UNT? AUNITS? — UNT? AUNITS? — UNT? AUNITS? — UNT? AUNITS? — UNT? AUNITS?</p>	<p>UNT_0 DBM — UNT_1 DBUV — UNT_2 DBMV — UNT_3 V — UNT_4 DBUVE — UNT_5 W</p>
<p>● <u>Display line</u></p> <p>Sets Display line to ON/OFF</p> <p>Sets Display line level</p> <p>Selects displaying absolute value/value relative to the display line (marker level, waveform data)</p>	<p><u>DISPLAY LINE</u></p> <p>OFF ON</p> <p><u>DISPLAY LINE LEVEL</u></p> <p>ABS / REL ABS REL TRACE-A ABS REL TRACE-B ABS REL TRACE-TIME ABS REL TRACE-BG ABS REL</p>	<p>DL_OFF DL_ON</p> <p>DL_l</p> <p>DSPLV_ABS DSPLV_REL DSPLVM_TRA, ABS DSPLVM_TRA, REL DSPLVM_TRB, ABS DSPLVM_TRB, REL DSPLVM_TRTIME, ABS DSPLVM_TRTIME, REL DSPLVM_TRBG, ABS DSPLVM_TRBG, REL</p>	<p>DL? — DL?</p> <p>DSPLV? DSPLV? DSPLVM?_TRA DSPLVM?_TRA DSPLVM?_TRB DSPLVM?_TRB DSPLVM?_TRTIME DSPLVM?_TRTIME DSPLVM?_TRBG DSPLVM?_TRBG</p>	<p>OFF — l</p> <p>ABS REL ABS REL ABS REL ABS REL ABS REL</p>

Table of MS2602A Device Message (4/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■</b> <u>Frequency/Amplitude</u>	<b><u>FREQUENCY/AMPLITUDE</u></b>			
● <u>Reference level offset</u>	<b><u>REFERENCE LEVEL OFFSET</u></b>			
Offset	OFFSET OFF ON	ROFFSET_OFF ROFFSET_ON	ROFFSET? _____	OFF _____
Offset value	OFFSET VALUE	ROFFSET_ℓ	ROFFSET?	ℓ
● <u>Correction factor relevance</u>	<b><u>CORRECTION</u></b>			
Selects the type of correction factor	CORRECTION FACTOR SELECT OFF  ON CORR1 CORR2 CORR3 CORR4 CORR5	CORR_OFF CORR_0 CORR_ON CORR_1 CORR_2 CORR_3 CORR_4 CORR_5	_____ CORR? _____ CORR? CORR? CORR? CORR? CORR?	_____ CORR_0 _____ CORR_1 CORR_2 CORR_3 CORR_4 CORR_5
Registers correction factor	CORRECTION FACTOR <sup>†</sup> ENTRY	CORD_n, f, ℓ	CORD?_n	CORD?_n, f, ℓ
Registers correction factor label	CORRECTION FACTOR <sup>†</sup> LABEL ENTRY	CORRLABEL_n, "text"	CORRLABEL ?_n	"text"
Initializes correction factor	CORRECTION FACTOR <sup>†</sup> INITIALIZATION	CORC	_____	_____
<b>■</b> <u>Display function</u>	<b><u>DISPLAY</u></b>			
● <u>Display mode</u>	<b><u>DISPLAY FUNCTION</u></b>			
Selects display format	DISPLAY FORMAT TRACE-A TRACE-B TRACE-TIME TRACE-A/B (A & B) TRACE-A/B (A/B) TRACE-A/BG (BG > A) TRACE-A/BG (BG < A) TRACE-A/TIME (TIME > A) TRACE-A/TIME (TIME < A)	DFMT_A DFMT_B DFMT_TIME DFMT_AB1 DFMT_AB2 DFMT_ABG1  DFMT_ABG2  DFMT_ETIME1 DFMT_ETIME2	DFMT? DFMT? DFMT? DFMT? DFMT? DFMT?  DFMT?  DFMT? DFMT?	A B TIME AB1 AB2 ABG1  ABG2  ETIME1 ETIME2

<sup>†</sup> It is impossible to set manually because of the commands used only for GPIB.

Table of MS2602A Device Message (5/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<p>■ <u>Display function</u></p> <p>● <u>Waveform writing</u></p> <p>Controls writing waveform to trace A</p> <p>Controls writing waveform to trace B</p> <p>Controls writing waveform to trace BG</p> <p>Controls writing waveform to trace TIME</p>	<u>DISPLAY</u>			
	<u>WRITE SWITCH</u>			
	TRACE-A WRITE SWITCH			
	VIEW	AWR_Ø AWR_OFF VIEW_TRA	— AWR? —	— AWR_OFF —
	WRITE	AWR_1 AWR_ON CLRW_TRA A1	— AWR? — —	— AWR_ON — —
	TRACE-B WRITE SWITCH			
	VIEW	BWR_Ø BWR_OFF VIEW_TRB	— BWR? —	— BWR_OFF —
	WRITE	BWR_1 BWR_ON CLRW_TRB B1	— BWR? — —	— BWR_ON — —
	TRACE-BG WRITE SWITCH			
	VIEW	BGWR_Ø BGWR_OFF VIEW_TRBG	— BGWR? —	— BGWR_OFF —
	WRITE	BGWR_1 BGWR_ON CLRW_TRBG	— BGWR? —	— BGWR_ON —
	TRACE-TIME WRITE SWITCH			
VIEW	TMWR_Ø TMWR_OFF VIEW_TRTIME	— TMWR? —	— TMWR_OFF —	
WRITE	TMWR_1 TMWR_ON CLRW_TRTIME	— TMWR? —	— TMWR_ON —	

Table of MS2602A Device Message (6/30)

Parameter		Program command	Query	Response
Brief function	Control item			
■ <u>Display function</u>	<u>DISPLAY</u>			
● <u>Storage mode</u>	<u>STORAGE MODE</u>			
Selects the mode for processing trace A waveform	TRACE MODE (A) NORMAL MAX HOLD	AMD_0 AMD_1 MXMH_TRA A2	AMD? AMD? _____ _____	AMD_0 AMD_1 _____ _____
	AVERAGE MIN HOLD CUMULATIVE OVER WRITE	AMD_2 AMD_3 AMD_4 AMD_5	AMD? AMD? AMD? AMD?	AMD_2 AMD_3 AMD_4 AMD_5
Selects the mode for processing trace B waveform	TRACE MODE (B) NORMAL MAX HOLD	BMD_0 BMD_1 MXMH_TRB B2	BMD? BMD? _____ _____	BMD_0 BMD_1 _____ _____
	AVERAGE MIN HOLD CUMULATIVE OVER WRITE	BMD_2 BMD_3 BMD_4 BMD_5	BMD? BMD? BMD? BMD?	BMD_2 BMD_3 BMD_4 BMD_5
Selects the mode for processing trace TIME waveform	TRACE MODE (TIME) NORMAL MAX HOLD AVERAGE MIN HOLD CUMULATIVE OVER WRITE	TMMD_0 TMMD_1 TMMD_2 TMMD_3 TMMD_4 TMMD_5	TMMD? TMMD? TMMD? TMMD? TMMD? TMMD?	TMMD_0 TMMD_1 TMMD_2 TMMD_3 TMMD_4 TMMD_5
Average processing	AVERAGE OFF	VAVG_0 VAVG_OFF KSH	_____ _____ _____	_____ _____ _____
	ON	VAVG_1 VAVG_ON KSG	_____ _____ _____	_____ _____ _____

Table of MS2602A Device Message (7/30)

Parameter		Program command	Query	Response
Brief function	Control item			
■ <u>Display function</u>	<u>DISPLAY</u>			
● <u>Storage mode</u>	<u>STORAGE MODE</u>			
Number of trace averaged	NUMBER of TRACE AVERAGE 4 8 16 32 128 n	AVR_0 AVR_1 AVR_2 AVR_3 AVR_4 VAVG_n	AVR? AVR? AVR? AVR? AVR? VAVG?	AVR_0 AVR_1 AVR_2 AVR_3 AVR_4 n
	DETECTION MODE POS PEAK	DET_0 DET_POS	— DET?	— POS
	SAMPLE	DET_1 DET_SMP	— DET?	— SMP
	NEG PEAK	DET_2 DET_NEG	— DET?	— NEG
Selects detection mode	TRACE-A DETECTION MODE POS PEAK SAMPLE NEG PEAK	DETM_TRA, POS DETM_TRA, SMP DETM_TRA, NEG	DETM?_TRA DETM?_TRA DETM?_TRA	POS SMP NEG
Selects detection mode	TRACE-B DETECTION MODE POS PEAK SAMPLE NEG PEAK	DETM_TRB, POS DETM_TRB, SMP DETM_TRB, NEG	DETM?_TRB DETM?_TRB DETM?_TRB	POS SMP NEG
	TRACE-TIME DETECTION MODE POS PEAK SAMPLE NEG PEAK	DETM_TRTIME, POS DETM_TRTIME, SMP DETM_TRTIME, NEG	DETM?_TRTIME DETM?_TRTIME DETM?_TRTIME	POS SMP NEG

Table of MS2602A Device Message (8/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<p>■ <u>Display function</u></p> <p>● <u>Time</u></p> <p>Sets time delay in the time axis sweep mode</p> <p>Sets time span in the time axis sweep mode</p> <p>Sets time expand mode to ON or OFF</p> <p>Sets time expand mode to ON or OFF</p> <p>Sets the start time of time expand</p> <p>Sets the magnified range of time expand</p> <p>Sets the function for monitoring the trace TIME waveform</p> <p>Sets the bandwidth for demodulating FM</p> <p>Sets the coupling to AC or DC to monitor FM waveform</p> <p>● <u>A/B</u></p> <p>Active marker Trace</p>	<p><u>DISPLAY</u></p> <p><u>TIME</u></p> <p>DELAY TIME</p> <p>TIME SPAN</p> <p>EXPAND ZONE OFF ON</p> <p>EXPAND OFF ON</p> <p>ZONE START</p> <p>ZONE SPAN</p> <p>FM/TRIGGER MONITOR OFF FM MONITOR EXT TRIGGER MONITOR</p> <p>FM RANGE 2 kHz / div 20 kHz / div 200 kHz / div</p> <p>COUPLING AC COUPLING DC COUPLING</p> <p>ACTIVE MARKER TRACE TRACE A TRACE B</p>	<p>TDLY_t</p> <p>TSP_t</p> <p>TZONE_0 TZONE_OFF TZONE_1 TZONE_ON</p> <p>TEXPAND_0 TEXPAND_OFF TEXPAND_1 TEXPAND_ON</p> <p>TZSTART_t</p> <p>TZSP_t</p> <p>SPFUNC_OFF SPFUNC_FM SPFUNC_EXT</p> <p>FMRNG_2KHZ FMRNG_20KHZ FMRNG_200KHZ</p> <p>COUPLE_AC COUPLE_DC</p> <p>MKTRACE_TRA MKTRACE_TRB</p>	<p>TDLY?</p> <p>TSP?</p> <p>TZONE? TZONE? TZONE?</p> <p>TEXPAND? TEXPAND? TEXPAND?</p> <p>TZSTART?</p> <p>TZSP?</p> <p>SPFUNC? SPFUNC? SPFUNC?</p> <p>FMRNG? FMRNG? FMRNG?</p> <p>COUPLE? COUPLE?</p>	<p>t</p> <p>t</p> <p>OFF ON</p> <p>OFF ON</p> <p>t</p> <p>t</p> <p>OFF FM EXT</p> <p>2000 20000 200000</p> <p>AC DC</p>

Table of MS2602A Device Message (9/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Signal search</b>	<b><u>SIGNAL SEARCH</u></b>			
Sets the maximum peak point within BG to center frequency	AUTO TUNE	ATUN	—	—
Sets the maximum peak point to center frequency	PEAK to CF	PCF	—	—
Sets the maximum peak point to REF level	PEAK to REF	PRL	—	—
Shifts the spectrum in the right or left direction	SCROLL			
	LEFT	SCR_0 SCR_LEFT	— —	— —
	RIGHT	SCR_1 SCR_RIGHT	— —	— —
<b>■ Marker function</b>	<b><u>MARKER</u></b>			
Selects marker mode	MARKER MODE			
	NORMAL	MKR_0 M2	MKR? —	MKR_0 —
	DELTA	MKR_1 MKD	MKR? —	MKR_1 —
Specifies zone marker center position as a point	OFF	M3 MKR_2 MKOFF MKOFF_ALL M1	MKR? — — — —	MKR_2 — — —
	ZONE POSITION (point)	MKZ_p MKP_p	MKZ? MKP?	MKZ_p p
	ZONE POSITION (freq or time)			
Specifies zone marker center position as a frequency or time	FREQ SET	MKZF_f MKN_f	MKZF? MKN?	f f
	UP	MKN_UP	—	—
	DOWN	MKN_DN	—	—
TIME SET	UP	MKZF_t MKN_t	MKZF? MKN?	t t
	DOWN	MKN_UP	—	—
	DOWN	MKN_DN	—	—



Table of MS2602A Device Message (10/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Marker function</b>	<b><u>MARKER</u></b>			
Specifies zone marker width as a point	ZONE WIDTH (point)	MZW_p	MZW?	MZW_p
Specifies zone marker width as a frequency	ZONE WIDTH (freq)	MZWF_f	MZWF?	f
Specifies zone marker width as a div	ZONE WIDTH (div) SPOT 0.5 div 1 div 2 div 5 div 10 div	MKW_1 MKW_0 MKW_5 MKW_6 MKW_7 MKW_2	MKW? MKW? MKW? MKW? MKW? MKW?	MKW_1 MKW_0 MKW_5 MKW_6 MKW_7 MKW_2
<b>● Marker function</b>	<b><u>MARKER FUNCTION</u></b>			
Moves marker frequency to center frequency	MKR to CF	MKR_3 MKCF E2	_____ _____ _____	_____ _____ _____
Sets level at marker to REF level	MKR to REF	MKR_4 MKRL E4	_____ _____ _____	_____ _____ _____
Sets marker frequency to CF step	MKR to CFstep	MKR_5 MKSS E3	_____ _____ _____	_____ _____ _____
Sets delta marker frequency to span	MKR to SPAN	MKR_6 MKSP KSO	_____ _____ _____	_____ _____ _____
Sets zone frequency to span	ZONE to SPAN	MKR_7	_____	_____
<b>● Multimarker</b>	<b><u>MULTI MARKER</u></b>			
Multimarker	MULTI MARKER OFF	MKMULTI_0 MKMULTI_OFF	_____ MKMULTI?	_____ OFF
	ON	MKMULTI_1 MKMULTI_ON	_____ MKMULTI?	_____ ON



Table of MS2602A Device Message (11/30)

Parameter		Program command	Query	Response
Brief function	Control item			
■ <u>Marker function</u>	<u>MARKER</u>			
● <u>Multimarker</u>	<u>MULTI MARKER</u>			
Multimarker mode	MULTI MARKER MODE Registers multimarkers on the peak point in descending level order from the maximum level down to the tenth. Registers multimarkers on the harmonic frequency ranging from the reference multimarker frequency up to the tenth.	MKMHI  MKMHRM	_____  _____	_____  _____
Selects multimarker	SELECT MULTI MARKER The n th marker: Sets to OFF  Sets to ON	MKSLCT_n, 0 MKSLCT_n, OFF MKSLCT_n, 1 MKSLCT_n, ON	_____ MKSLCT?_n _____ MKSLCT?_n	_____ OFF _____ ON
Selects the active marker of the multimarkers	ACTIVE MARKER	MKACT_n	MKACT?	n
Specifies the frequency of the designated multimarker number	MARKER POSITION	MKMP_n, f	MKMP?_n	f
Clears all the registered multimarkers	CLEAR MULTI MARKER	MKMCL	_____	_____
Multimarker list	MULTI MARKER LIST OFF  ON	MKLIST_0 MKLIST_OFF MKLIST_1 MKLIST_ON	_____ MKLIST? _____ MKLIST?	_____ OFF _____ ON
Reads out the multimarker level	MULTI MARKER LEVEL QUERY	_____	MKML?n	ℓ

Table of MS2602A Device Message (12/30)

Parameter		Program command	Query	Response
Brief function	Control item			
■ <u>Marker function</u>	<u>MARKER</u>			
● <u>Peak search</u>	<u>PEAK SEARCH</u>			
Peak search mode	PEAK SERCH MODE PEAK	MKS_0 MKPK MKPK_HI E1	_____ _____ _____ _____	_____ _____ _____ _____
	NEXT PEAK	MKS_1 MKPK_NH	_____ _____	_____ _____
	DIP	MKS_2 MKMIN	_____ _____	_____ _____
	NEXT RIGHT PEAK	MKS_9 MKPK_NR	_____ _____	_____ _____
	NEXT LEFT PEAK	MKS_10 MKPK_NL	_____ _____	_____ _____
	NEXT DIP	MKS_11	_____	_____
● <u>Input position</u>	<u>INPUT POSITION</u>			
Reads out reference marker position	REFERENCE MARKER POSITION	_____	RMK?	RMK_p
Reads out current marker position	CURRENT MARKER POSITION	_____	CMK?	CMK_p
Reads out frequency at marker point	MARKER FREQ QUERY FREQ TIME	_____ _____	MKF? MKF?	f t
Reads out level at marker point	MARKER LEVEL	_____ _____	MKL? MKA?	ℓ ℓ

Table of MS2602A Device Message (13/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Coupled function</b>	<b><u>COUPLED FUNCTION</u></b>			
Sets resolution bandwidth	RESOLUTION BANDWIDTH			
	MANUAL	ARB_0	ARB?	ARB_0
	AUTO	ARB_1	ARB?	ARB_1
		RB_AUTO	—	—
		CR	—	—
	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
	RBW UP	RB_UP	—	—
	RBW DOWN	RB_DN	—	—

Table of MS2602A Device Message (14/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Coupled function</b>	<b><u>COUPLED FUNCTION</u></b>			
Sets video bandwidth	VIDEO BANDWIDTH			
	MANUAL	AVB_0	AVB?	AVB_0
	AUTO	AVB_1	AVB?	AVB_1
		VB_AUTO	—	—
		CV	—	—
	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
	VBW UP	VB_UP	—	—
	VBW DOWN	VB_DN	—	—
Sets the ratio of VBW and RBW (where VBW = AUTO)	VBW / RBW RATIO			
	RATIO=r	VBR_r	VBR?	r

Table of MS2602A Device Message (15/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Coupled function</b>	<b>COUPLED FUNCTION</b>			
Sets sweep time	SWEEP TIME MANUAL AUTO	AST_0 AST_1 ST_AUTO CT	AST? AST? ____ ____	AST_0 AST_1 ____ ____
	SWEEP_TIME SET TIME=t	SWT_t ST_t	SWT? ST?	SWT_t t
	UP DOWN	ST_UP ST_DN	____ ____	____ ____
Sets RF attenuator to MANUAL or AUTO mode	RF ATTENUATOR MANUAL AUTO	AAT_0 AAT_1 AT_AUTO CA	AAT? AAT? ____ ____	AAT_0 AAT_1 ____ ____
Sets RF attenuator	0 dB	ATT_0 AT_0	ATT? AT?	ATT_0 0
	5 dB	ATT_6 AT_5	ATT? AT?	ATT_6 5
	10 dB	ATT_1 AT_10	ATT? AT?	ATT_1 10
	15 dB	ATT_7 AT_15	ATT? AT?	ATT_7 15
	20 dB	ATT_2 AT_20	ATT? AT?	ATT_2 20
	25 dB	ATT_8 AT_25	ATT? AT?	ATT_8 25
	30 dB	ATT_3 AT_30	ATT? AT?	ATT_3 30
	35 dB	ATT_9 AT_35	ATT? AT?	ATT_9 35
	40 dB	ATT_4 AT_40	ATT? AT?	ATT_4 40
	45 dB	ATT_10 AT_45	ATT? AT?	ATT_10 45
	50 dB	ATT_5 AT_50	ATT? AT?	ATT_5 50
	55 dB	ATT_11 AT_55	ATT? AT?	ATT_11 55
	UP DOWN	AT_UP AT_DN	____ ____	____ ____
Sets RBW, VBW, and sweep time to AUTO mode	RBW, VBW / SWEEP TIME AUTO	BSAUTO	____	____
Sets coupled function to AUTO mode	COUPLED FUNCTION AUTO	AUTO	____	____

Table of MS2602A Device Message (16/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Sweep function</b>	<b><u>SWEEP CONTROL</u></b>			
Sets zone sweep to ON or OFF	ZONE SWEEP OFF	PSW_0 PSW_OFF	PSW?	PSW_OFF
	ON	PSW_1 PSW_ON	PSW?	PSW_ON
Sets tracking function	TRACKING OFF	MKTRACK_0 MKTRACK_OFF	MKTRACK?	OFF
	ON	MT0 MKTRACK_1 MKTRACK_ON MT1	MKTRACK?	ON
Gate function	GATE MODE OFF	GATE_0 GATE_OFF	GATE?	OFF
	ON	GATE_1 GATE_ON	GATE?	ON
Sets gate delay time	GATE DELAY TIME	GD_t	GD?	t
Sets gate length	GATE LENGTH	GL_t	GL?	t
Sets ending gate interval internally or externally	GATE END INTERNAL	GE_INT	GE?	INT
	EXTERNAL	GE_EXT	GE?	EXT
Sets trigger mode (Sets trigger source / trigger switch)	TRIGGER MODE FREERUN	TRG_0 TM_FREE	TRG? TM?	TRG_0 FREE
	VIDEO	TRG_1 TM_VID	TRG? TM?	TRG_1 VID
	LINE	TRG_2 TM_LINE	TRG? TM?	TRG_2 LINE
	EXT	TRG_3 TM_EXT	TRG? TM?	TRG_3 EXT
	TV	TRG_6 TM_TV	TRG? TM?	TRG_6 TV
	Sets trigger switch	TRIGGER SWITCH FREERUN TRIGGERD	TRGS_FREE TRGS_TRGD	TRGS? TRGS?

Table of MS2602A Device Message (17/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Sweep function</b>	<b><u>SWEEP CONTROL</u></b>			
Sets trigger source	TRIGGER SOURCE VIDEO LINE EXT TV	TRGSOURCE_VID TRGSOURCE_LINE TRGSOURCE_EXT TRGSOURCE_TV	TRGSOURCE? TRGSOURCE? TRGSOURCE? TRGSOURCE?	VID LINE EXT TV
Sets the external trigger level type (when trigger source = EXT)	EXT TRIGGER TYPE INPUT1 (±10 V) INPUT2 (TTL)	EXTTYPE_10V EXTTYPE_TTL	EXTTYPE? EXTTYPE?	10V TTL
Selects TV system (when trigger source = TV)	TV TYPE PAL NTSC	TVSTND_PAL TVSTND_NTSC	TVSTND? TVSTND?	PAL NTSC
Selects TV horizontal synchronous signal (when trigger source = TV)	TV SYNCHRONIZING SIGNAL VERTICAL HORIZONTAL-EVEN HORIZONTAL-ODD	TVSFRM_VERTICAL TVSFRM_EVEN  TVSFRM_ODD	TVSFRM? TVSFRM?  TVSFRM?	VERTICAL EVEN  ODD
Selects the number of TV horizontal synchronous signal line (when trigger source = TV)	TV H-SYNC LINE	TVLINE_a	TVLINE?	a
Sets the trigger threshold level for starting sweep	TRIGGER LEVEL	TRGLVL_ℓ	TRGLVL?	ℓ
Selects the slope for detecting the sweep-starting trigger (when trigger source = TV, EXT)	TRIGGER SLOPE RISE FALL	TRGSLP_RISE TRGSLP_FALL	TRGSLP? TRGSLP?	RISE FALL
Sets sweep mode to single	SINGLE SWEEP MODE	SNGLS S2	— —	— —

Table of MS2602A Device Message (18/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Sweep function</b>		<b><u>SWEEP CONTROL</u></b>		
Executes / checks single sweep	SINGLE SWEEP / SWEEP STATUS			
	Executing single sweep	SWP TS	— —	— —
	Checking sweep status	—	SWP? SWP?	SWP_0 SWP_1
	Sweep completed	—		
	Sweep in progress			
Continuous sweep mode	CONTINUOUS SWEEP MODE	CONTS S1	— —	— —
Stops sweep	SWEEP STOP	SWSTOP	—	—
Restarts sweep	SWEEP RESTART	SWSTART	—	—
<b>■ Save / Recall</b>		<b><u>SAVE / RECALL</u></b>		
Recalls data from internal memory	RECALL DATA FROM INTERNAL MEMORY	RGRC_r RC_r	— —	— —
Recalls data from PMC (floppy disk)	RECALL DATA FROM PMC (FD)	RCM_r	—	—
Saves data in internal memory	SAVE DATA INTO INTERNAL MEMORY	RGSV_s SV_s	— —	— —
Saves data on PMC (floppy disk)	SAVE DATA INTO PMC (FD)	SVM_s	—	—
Displays the directory of the recall memory	MEMORY DIRECTORY	RGDIR	—	—
Sets the recall data	RECALLED DATA TRACE&PARAM PARAM ONLY	RDATA_TP RDATA_P	RDATA? RDATA?	TP P



Table of MS2602A Device Message (19/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Hard copy</b>	<b><u>HARD COPY</u></b>			
Starts direct plot	DIRECT PLOT START	PLS_0 PLOT PRINT	— — —	— — —
<b>● Controls hard copy</b>	<b><u>COPY CONTROL</u></b>			
Direct plotting device selection Selects the plotter Selects the printer	DIRECT PLOT DEVICE PLOTTER HP-GL GP-GL PRINTER VP-800 HP-2225 MC8104A UA-455A	PMOD_0 PMOD_1  PMOD_2 PMOD_3 PMOD_4 PMOD_5	PMOD? PMOD?  PMOD? PMOD? PMOD? PMOD?	PMOD_0 PMOD_1  PMOD_2 PMOD_3 PMOD_4 PMOD_5
Sets the printer GPIB address	PRINTER ADDRESS SET	PRIA_a	PRIA?	a
Sets the plotter GPIB address	PLOTTER ADDRESS SET	PLTA_a	PLTA?	a
Sets the size of the paper output from the plotter	DIRECT PLOT SIZE A4 A3	PLF_0 PLF_1	PLF? PLF?	PLF_0 PLF_1
Sets the size of the plot	PLOT AREA FULL SIZE QUATER SIZE	PLTARA_FULL PLTARA_QTR	PLTARA? PLTARA?	FULL QTR
Sets the location of the plot on the paper	PLOT LOCATION renewed automatically Fixed in upper left-corner Fixed in upper right-corner Fixed in lower left-corner Fixed in lower right-corner	PLTLC_AUTO PLTLC_UPLEFT PLTLC_UPRIGHT PLTLC_LOWLEFT PLTLC_LOWRIGHT	PLTLC? PLTLC? PLTLC? PLTLC? PLTLC?	AUTO UPLEFT UPRIGHT LOWLEFT LOWRIGHT
Selects one of the plot items to a plotter	DIRECT PLOT OUTPUT ITEM ALL TRACE ONLY SCALE ONLY	PLI_0 PLI_1 PLI_2	PLI? PLI? PLI?	PLI_0 PLI_1 PLI_2
Selects "UPPER LEFT" for the plot location on the paper (Only at AUTO ADVANCE)	PLOTTER LOCATION PRESET	PLTHOME	—	—

Table of MS2602A Device Message (20/30)

Parameter		Program command	Query	Response
Brief function	Control item			
■ <u>Measure function</u>	<u>MEASURE</u>			
	Sets measure function to OFF	MEASURE FUNCTION ALL OFF	MEAS_OFF —	—
● <u>Frequency measurement</u>	<u>FREQ MEASURE</u>			
	Measures frequency	FREQ MEASURE OFF	MKC_0 MC_OFF MKFC_0 MKFC_OFF MEAS_FREQ, OFF	MKC_0 — 0 — —
		ON	MKC_1 MC_ON MKFC_0 MKFC_ON MEAS_FREQ, ON	MKC_1 — 1 — —
	Transferring measured results	—	RES?	f
Sets counter to the specified resolution	COUNT RESOLUTION			
	1 Hz	CRS_0 MKFCR_1HZ	CRS? MKFCR?	CRS_0 1
	10 Hz	CRS_1 MKFCR_10HZ	CRS? MKFCR?	CRS_1 10
	100 Hz	CRS_2 MKFCR_100HZ	CRS? MKFCR?	CRS_2 100
	1 kHz	CRS_3 MKFCR_1KHZ	CRS? MKFCR?	CRS_3 1000
	FREQ UP	MKFCR_UP	—	—
	FREQ DOWN	MKFCR_DN	—	—
● <u>Noise measurement</u>	<u>NOISE MEASURE</u>			
	Measures noise	NOISE MEASURE OFF ON ABSOLUTE executed C/N RATIO executed	MEAS_NOISE, OFF MEAS_NOISE, ON MEAS_NOISE, ABS MEAS_NOISE, CN	— — — —
	Transferring measured results	—	RES?	e
Calculation method	ABSOLUTE	MNOISE_ABS	MNOISE?	ABS
	C/N RATIO	MNOISE_CN	MNOISE?	CN

Table of MS2602A Device Message (21/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Measure function</b>	<b><u>MEASURE</u></b>			
<b>● Occupied frequency bandwidth measurement</b>	<b><u>OBW MEASURE</u></b>			
Measures occupied frequency bandwidth	OBW MEASURE Executes calculation Executes (X dB DOWN) Executes (N%) Transferring measured results (f <sub>1</sub> : Occupied bandwidth f <sub>2</sub> : Center frequency)	MEAS_OBW, EXE MEAS_OBW, XDB MEAS_OBW, N _____	_____ _____ _____ RES?	_____ _____ _____ f <sub>1</sub> , f <sub>2</sub>
Calculation method	X dB DOWN method N% method	MOBW_XDB MOBW_N	MOBW? MOBW?	XDB N
Sets the conditions of occupied frequency bandwidth	OBW VALUE x dB n%	OBWXDB_xDB OBWN_n	OBWXDB? OBWN?	x n
<b>● Adjacent channel measurement</b>	<b><u>ADJACENT CH MEASURE</u></b>			
Measures adjacent channel	ADJACENT CH MEASURE Executes calculation Executes (UNMODULATED CARRIER) Executes (MODULATED CARRIER) Transferring measured results (ℓ <sub>L1</sub> : CH1 lower sideband ℓ <sub>U1</sub> : CH1 upper sideband ℓ <sub>L2</sub> : CH2 lower sideband ℓ <sub>U2</sub> : CH2 upper sideband)	MEAS_ADJ, EXE MEAS_ADJ, UNMD  MEAS_ADJ, MOD _____	_____ _____ _____ RES?	_____ _____ _____ ℓ <sub>L1</sub> , ℓ <sub>U1</sub> , ℓ <sub>L2</sub> , ℓ <sub>U2</sub>
Selects adjacent channel	ADJACENT CH SELECT BOTH SIDES UPPER SIDE LOWER SIDE OFF	ADJCH_BOTH ADJCH_UP ADJCH_LOW ADJCH_OFF	ADJCH? ADJCH? ADJCH? ADJCH?	BOTH UP LOW OFF
Sets adjacent channel bandwidth	ADJACENT CH BANDWIDTH	ADJCHBW_f	ADJCHBW?	f
Sets adjacent channel 1 separation	ADJACENT CH1 SEPALATION	ADJCHSP_f	ADJCHSP?	f
Sets adjacent channel 2 separation	ADJACENT CH2 SEPALATION	ADJCHSPF_f	ADJCHSPF?	f

Table of MS2602A Device Message (22/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Measure function</b>	<b><u>MEASURE</u></b>			
<b>● Adjacent channel measurement</b>	<b><u>ADJACENT CH MEASURE</u></b>			
Selects calculation method	R: TOTAL POWER (MOD) R: REF LEVEL (UNMOD)	MADJMOD_MOD MADJMOD_UNMD	MADJMOD? MADJMOD?	MOD UNMD
Sets graph display to ON/OFF	GRAPH OFF ON	MADJGRAPH_OFF MADJGRAPH_ON	MADJGRAPH? MADJGRAPH?	OFF ON
Sets channel center line display to ON/OFF	CHANNEL CENTER LINE OFF ON	MADJCTRLN_OFF MADJCTRLN_ON	MADJCTRLN? MADJCTRLN?	OFF ON
Sets channel range line to ON/OFF	CHANNEL BAND LINE OFF ON	MADJBWLN_OFF MADJBWLN_ON	MADJBWLN? MADJBWLN?	OFF ON
<b>● Template measurement</b>	<b><u>TEMPLATE</u></b>			
Measures template	TEMPLATE MEASURE OFF ON CHECK TEMP Transferring measured results (c <sub>1</sub> : LIMIT1 check result c <sub>2</sub> : LIMIT2 check result)	MEAS_TEMP,OFF MEAS_TEMP,ON MEAS_TEMP,CHECK _____	_____ _____ _____ RES?	_____ _____ _____ C1, C2 (PASS = 0, FAIL = 1)
Moves template	TEMPLATE MOVE MOVE X MOVE Y SAVE CANCEL	TEMPMVX_t TEMPMVY_l TEMPMSV TEMPMCL	TEMPMVX? TEMPMVY? _____ _____	t l _____ _____

Table of MS2602A Device Message (23/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<ul style="list-style-type: none"> <li>■ <u>Measure function</u></li> <li>● <u>Template measurement</u></li> </ul>	<p><u>MEASURE</u></p> <p><u>TEMPLATE</u></p>			
Selects template	<p>SELECT TEMPLATE</p> <p>No.</p> <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p>	<p>TEMP_1</p> <p>TEMP_2</p> <p>TEMP_3</p> <p>TEMP_4</p> <p>TEMP_5</p>	<p>TEMP?</p> <p>TEMP?</p> <p>TEMP?</p> <p>TEMP?</p> <p>TEMP?</p>	<p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p>
Selects LIMIT line	<p>SELECT LIMIT LINE</p> <p>LIMIT1 UPPER</p> <p>OFF</p> <p>ON</p> <p>LIMIT2 UPPER</p> <p>OFF</p> <p>ON</p> <p>LIMIT1 LOWER</p> <p>OFF</p> <p>ON</p> <p>LIMIT2 LOWER</p> <p>OFF</p> <p>ON</p>	<p>TEMPSLCT_UP1, 0</p> <p>TEMPSLCT_UP1, OFF</p> <p>TEMPSLCT_UP1, 1</p> <p>TEMPSLCT_UP1, ON</p> <p>TEMPSLCT_UP2, 0</p> <p>TEMPSLCT_UP2, OFF</p> <p>TEMPSLCT_UP2, 1</p> <p>TEMPSLCT_UP2, ON</p> <p>TEMPSLCT_LW1, 0</p> <p>TEMPSLCT_LW1, OFF</p> <p>TEMPSLCT_LW1, 1</p> <p>TEMPSLCT_LW1, ON</p> <p>TEMPSLCT_LW2, 0</p> <p>TEMPSLCT_LW2, OFF</p> <p>TEMPSLCT_LW2, 1</p> <p>TEMPSLCT_LW2, ON</p>	<p>_____</p> <p>TEMPSLCT?UP1</p> <p>_____</p> <p>TEMPSLCT?UP1</p> <p>_____</p> <p>TEMPSLCT?UP2</p> <p>_____</p> <p>TEMPSLCT?UP2</p> <p>_____</p> <p>TEMPSLCT?LW1</p> <p>_____</p> <p>TEMPSLCT?LW1</p> <p>_____</p> <p>TEMPSLCT?LW2</p> <p>_____</p> <p>TEMPSLCT?LW2</p>	<p>_____</p> <p>OFF</p> <p>_____</p> <p>ON</p> <p>_____</p> <p>OFF</p> <p>_____</p> <p>ON</p> <p>_____</p> <p>OFF</p> <p>_____</p> <p>ON</p> <p>_____</p> <p>OFF</p> <p>_____</p> <p>ON</p> <p>_____</p> <p>OFF</p> <p>_____</p> <p>ON</p>
<ul style="list-style-type: none"> <li>● <u>Power measurement</u></li> </ul>	<p><u>POWER MEASURE</u></p>			
Measures power	<p>POWER MEASURE MEASURE</p> <p>Transferring measured results</p> <p>(ℓ: dBm value</p> <p>w: pW value)</p>	<p>MEAS_POWER, EXE</p> <p>_____</p>	<p>_____</p> <p>RES?</p>	<p>_____</p> <p>ℓ, w</p>
Sets the point at which power measurement starts	<p>POWER MEASURE START</p>	<p>PWRSTART_p</p>	<p>PWRSTART?</p>	<p>p</p>
Sets the point at which power measurement ends	<p>POWER MEASURE STOP</p>	<p>PWRSTOP_p</p>	<p>PWRSTOP?</p>	<p>p</p>

Table of MS2602A Device Message (24/30)

Parameter		Program command	Query	Response
Brief function	Control item			
■ <u>Measure function</u>	<u>MEASURE</u>			
● <u>Template management function</u>	<u>MANAGE TEMPLATE</u>			
Measures template number	SELECT TEMPLATE No. 1 2 3 4 5	MTEMP_1 MTEMP_2 MTEMP_3 MTEMP_4 MTEMP_5	MTEMP? MTEMP? MTEMP? MTEMP? MTEMP?	1 2 3 4 5
Selects LIMIT line	SELECT LIMIT LINE LIMIT1 UPPER LIMIT2 UPPER LIMIT1 LOWER LIMIT2 LOWER	MTEMPL_UP1 MTEMPL_UP2 MTEMPL_LW1 MTEMPL_LW2	MTEMPL? MTEMPL? MTEMPL? MTEMPL?	UP1 UP2 LW1 LW2
Sets the level data by distinguishing relative value from absolute value	TEMPLATE LEVEL MODE ABSOLUTE RELATIVE	MTEMPREL_OFF MTEMPREL_ON	MTEMPREL? MTEMPREL?	OFF ON
Adds the template data by 1 point	INSERT TEMPLATE POINT DATA	MTEMPIN_p, t, ℓ	—	—
Replaces the template data by 1 point	REPLACE TEMPLATE POINT DATA	MTEMPRP_p, t, ℓ	—	—
Reads out the template data by 1 point	READ TEMPLATE POINT DATA	—	MTEMPPD?_p	t, ℓ
Deletes the template data by 1 point	DELETE TEMPLATE POINT DATA	MTEMPDEL_p	—	—
Initializes the template data	INITIATE LINE/TEMPLATE LIMIT1 UPPER LIMIT2 UPPER LIMIT1 LOWER LIMIT2 LOWER	MTEMPINI_UP1 MTEMPINI_UP2 MTEMPINI_LW1 MTEMPINI_LW2	— — — —	— — — —

Table of MS2602A Device Message (25/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<p>■ <u>Measure function</u></p> <p>● <u>Template management function</u></p> <p>Specifies how the template management screen is displayed</p> <p>Sets template label</p> <p>● <u>Measure window display position</u></p> <p>Specifies display position</p> <p>■ <u>Sound</u></p> <p>Selects the function for monitoring the sound from the detector output</p> <p>Adjusts the volume of sound monitor</p> <p>■ <u>Parameter setting</u></p>	<p><u>MEASURE</u></p> <p><u>MANAGE TEMPLATE</u></p> <p>DISPLAY TEMPLATE MODE GRAPH LIST</p> <p>TEMP LABEL</p> <p>WINDOW POSITION UPPER RIGHT LOWER LEFT LOWER MIDDLE LOWER RIGHT</p> <p><u>SOUND</u></p> <p>AM / FM SOUND MONITOR OFF AM FM</p> <p>AM / FM SOUND MONITOR VOLUME</p> <p><u>PARAMETER</u></p> <p>PARAMETER OFF PARAMETER LIST1 PARAMETER LIST2</p>	<p>MTEMPDSP_GRAPH MTEMPDSP_LIST</p> <p>MTEMPLABEL_n, 'text'</p> <p>WINDPOS_UPRIGHT WINDPOS_LOWLEFT WINDPOS_LOWMID WINDPOS_LOWRIGHT</p> <p>MON_OFF MON_AM MON_FM</p> <p>MONVOL_v</p> <p>PARAM_OFF PARAM_1 PARAM_2</p>	<p>MTEMPDSP? MTEMPDSP?</p> <p>MTEMPLABEL?n</p> <p>WINDPOS? WINDPOS? WINDPOS? WINDPOS?</p> <p>MON? MON? MON?</p> <p>MONVOL?</p> <p>PARAM? PARAM?</p> <p>PARAM?</p>	<p>GRAPH LIST</p> <p>text</p> <p>UPRIGHT LOWLEFT LOWMID LOWRIGHT</p> <p>OFF AM FM</p> <p>v</p> <p>OFF 1 2</p>



Table of MS2602A Device Message (26/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ System setting</b>		<b><u>SYSTEM</u></b>		
Sets the auto sweep time	AUTO SWEEP TIME FAST NORMAL	ASWT_FAST ASWT_SLOW	ASWT? ASWT?	FAST SLOW
Sets the data point	DATA POINT 501 1002	DPOINT_NRM DPOINT_DOUBLE	DPOINT? DPOINT?	NRM DOUBLE
Sets the coupled functions commonly or independently between frequency domain or time domain	COUPLE MODE COMMON INDEPENDENT	VBCOUPLE_COM VBCOUPLE_IND	VBCOUPLE? VBCOUPLE?	COM IND
Setting the display format of the date	DATE DISP YY/MM/DD MMM-DD-YY DD-MMM-YY	DDTYP_0 DDTYP_1 DDTYP_2	DDTYP? DDTYP? DDTYP?	DDTYP_0 DDTYP_1 DDTYP_2
<b>■ Calibration</b>		<b><u>CALIBRATION</u></b>		
Executes calibration with the internal CAL signal	CALIBRATION ALL FREQ LEVEL	CAL_0 CAL_1 CAL_2	— — —	— — —
Adjusts bias of preselector	PRESELECTOR TUNE MANUAL AUTO  PRESET	PRESEL_a PRESEL_AUTO PP PRESEL_PRESET	PRESEL? — — —	a — — —
<b>■ GPIB</b>		<b><u>GPIB</u></b>		
Sets GPIB2 self address	GPIB2 SELF ADDRESS	GPIA_a	GPIA?	a
Sets the GPIB address of MC8104A	MC8104A ADDRESS	DSUA_a	DSUA?	a
Sets the time out period of the talker function	GPIB TIME OUT	GTOUT_t	GTOUT?	t



Table of MS2602A Device Message (27/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ RS-232C<sup>†</sup></b>				
Sets baud rate	<b>RS-232C</b>			
	BAUD RATE			
	300	BAUD_300	BAUD?	300
	600	BAUD_600	BAUD?	600
	1200	BAUD_1200	BAUD?	1200
	2400	BAUD_2400	BAUD?	2400
Sets parity	4800	BAUD_4800	BAUD?	4800
	9600	BAUD_9600	BAUD?	9600
	PARITY			
	EVEN	PRTY_EVEN	PRTY?	EVEN
Sets data bit	ODD	PRTY_ODD	PRTY?	ODD
	OFF	PRTY_OFF	PRTY?	OFF
Sets stop bit	DATA BIT			
	7bit	DATB_7	DATB?	7
Sets the period of reception time out	8bit	DATB_8	DATB?	8
	STOP BIT			
	1bit	STPB_1	STPB?	1
Sets the period of reception time out	1.5bit	STPB_1.5	STPB?	1.5
	2bit	STPB_2	STPB?	2
Sets the period of reception time out	TIME OUT	TOUT_t	TOUT?	t
<b>■ Title</b>				
Title entry	<b>TITLE</b>			
	TITLE ENTRY	TITLE_'text' KSE_'text'	TITLE?	text
Title display	TITLE DISPLAY			
	OFF	TTL_0 TTL_OFF	TTL?	TTL_OFF
	ON	TTL_1 TTL_ON	TTL?	TTL_ON
	DATE	TTL_2 TTL_DATE	TTL?	TTL_DATE
<b>■ CAL/UNCAL</b>				
Couple failure	<b>CAL/UNCAL</b>			
	UNCAL			
	UNCAL DISPLAY			
	OFF	UNC_0 UNC_OFF	UNC?	UNC_OFF
Couple failure	ON	UNC_1 UNC_ON	UNC?	UNC_ON
	UNCAL STATUS			
Couple failure	NORMAL	---	UCL?	UCL_0
	UNCAL	---	UCL?	UCL_1

<sup>†</sup> All the commands controlling RS-232C are available only when the option 02 is installed.

Table of MS2602A Device Message (28/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Spectrum data</b>	<b><u>SPECTRUM DATA</u></b>			
Trace A memory	TRACE-A MEMORY	XMA_p, b	XMA?_p, d	b
Trace B memory	TRACE-B MEMORY	XMB_p, b	XMB?_p, d	b
Trace BG memory	TRACE-BG MEMORY	XMG_p, b	XMG?_p, d	b
Trace TIME memory	TRACE-TIME MEMORY	XMT_p, b	XMT?_p, d	b
Selects ASCII / Binary	ASCII DATA	BIN_0	—	—
		BIN_OFF	—	—
	BINARY DATA	BIN_1	—	—
		BIN_ON	—	—
<b>■ PMC</b>	<b><u>PMC</u></b>			
Selects internal or external PMC (floppy disk)	PMC SELECT Internal PMC External PMC1 External PMC2 External FD	PMCS_IPMC PMCS_EPMC1 PMCS_EPMC2 PMCS_EFD	PMCS? PMCS? PMCS? PMCS?	IPMC EPMC1 EPMC2 EFD
<b>■ PTA control†</b>	<b><u>PTA CONTROL</u></b>			
Switches PTA function to ON/OFF	PTA SWITCH OFF	PTA_OFF PTA_0	— PTA?	— PTA_0
	ON	PTA_ON PTA_1	— PTA?	— PTA_1
Selects the mode for controlling PTA via GPIB	PTL I/O MODE† OFF INPUT (COMMAND PROGRAM) OUTPUT (PROGRAM)	PTL_0 PTL_1 —	— — PTL?	— — text
Sets GPIB 2 self address	GPIB (2) SELF ADDRESS	GPIA_a	GPIA?	GPIA_a
Writes/reads to and from the dual port memory	DUAL-PORT MEMORY† READ / WRITE	PMY_a, "b"	PMY_a, c	"b"
Selects the control port for GPIB	CONTROL PORT SELECT† GPIB (1) GPIB (2)	PORT_1 PORT_2	PORT? PORT?	PORT_1 PORT_2

† All the commands controlling PTA are available only when the option 04 or 05 is installed.

Table of MS2602A Device Message (29/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>■ Others</b>	<b><u>ETC.</u></b>			
Terminator	TERMINATOR LF CR/LF	TRM_0 TRM_1	____ ____	____ ____
Initializes measurement control parameters with level 3	INITIALIZE	INI IP	____ ____	____ ____
Sounds the buzzer	BUZZER ON	BZR	____	____
Buzzer switch	BUZZER SWITCH OFF  ON	BEP_0 BEP_OFF BEP_1 BEP_ON	____ ____ ____ ____	____ ____ ____ ____
Sets the clock in the measuring instrument to the specified time	TIMER SET DATE TIME	DATE_ yy, mm, dd TIME_ hh, mm, ss	DATE? TIME?	yy, mm, dd hh, mm, ss
Reads the live time counted with an integrating meter	TIME COUNT READ	____	TMCNT?	t(hr)

Table of MS2602A Device Message (30/30)

Parameter		Program command	Query	Response
Brief function	Control item			
<b>Common command and event status</b>	<b><u>GPIB COMMON COMMAND</u></b> <b><u>EVENT STATUS</u></b>			
Clears Status Byte Register	CLEAR STATUS COMMAND	*CLS	—	—
Sets bit in Service Request Register	SERVICE REQUEST ENABLE	*SRE_n	*SRE?	n
Reads out the current value of Status Byte	READ STATUS BYTE	—	*STB?	n
Executes single sweep	TRIGGER COMMAND	*TRG	—	—
Executes self test	SELF TEST	—	*TST	n
Keeps the next command on stand-by during a device command execution	WAIT TO CONTINUE	*WAI	—	—
Returns the manufacturer name, model name etc. of the product	IDENTIFICATION QUERY	—	*IDN?	ANRITSU...
Resets the device with level 3	RESET COMMAND	*RST	—	—
Synchronization mode between device and controller	OPERATION COMPLETE WAITING FOR SERVICE REQUEST WAITING FOR OUTPUT QUEUE IN DEVICE	*OPC	—	—
Sets or clears Standard Event Status Enable Register	STANDARD EVENT STATUS ENABLE	*ESE_n	*ESE?	n
Reads out Standard Event Status Enable Register	STANDARD EVENT STATUS REGISTER	—	*ESR?	n
Controls masking Extended Event Status	EVENT STATUS ENABLE	ESE2_n	ESE2?	n
Reads out Extended Event Status	EVENT STATUS REGISTER	—	ESR2?	n

## SECTION 8

### DETAILED DESCRIPTION OF COMMANDS

This section describes in detail the device messages and response messages for the MS2602A. These messages are listed in alphabetical order as shown below.

#### TABLE OF CONTENTS

AAT .....	8-6	CLRW .....	8-22
ADJCH .....	8-6	CMK ? .....	8-22
ADJCHBW .....	8-7	CNF .....	8-23
ADJCHSP .....	8-7	CONTS .....	8-23
ADJCHSPF .....	8-8	CORC .....	8-24
AMD .....	8-8	CORD .....	8-24
ARB .....	8-9	CORR .....	8-25
AST .....	8-9	CORRLABEL .....	8-25
ASWT .....	8-10	COUPLE .....	8-26
AT .....	8-10	CR .....	8-26
ATT .....	8-11	CRS .....	8-27
ATUN .....	8-11	CT .....	8-27
AUNITS .....	8-12	CV .....	8-28
AUTO .....	8-12	DATE .....	8-28
AVB .....	8-13	DDTYP .....	8-29
AVR .....	8-13	DET .....	8-29
AWR .....	8-14	DETM .....	8-30
A1 .....	8-14	DFMT .....	8-31
A2 .....	8-15	DL .....	8-32
BEP .....	8-15	DPOINT .....	8-33
BGWR .....	8-16	DSPLV .....	8-33
BIN .....	8-16	DSPLVM .....	8-34
BMD .....	8-17	DSUA .....	8-34
BNDC .....	8-17	ESE2 .....	8-35
BSAUTO .....	8-18	ESR2 ? .....	8-35
BWR .....	8-18	EXTTYPE .....	8-36
BZR .....	8-19	E1 .....	8-36
B1 .....	8-19	E2 .....	8-37
B2 .....	8-20	E3 .....	8-37
CA .....	8-20	E4 .....	8-38
CAL .....	8-21	FA .....	8-38
CF .....	8-21	FADJ .....	8-39

## TABLE OF CONTENTS ( Continued )

FB	8-39	MKFC	8-61
FDN	8-40	MKFCR	8-61
FMRNG	8-40	MKL ?	8-62
FRQ	8-41	MKLIST	8-62
FS	8-41	MKMCL	8-63
FSS	8-42	MKMHI	8-63
FUP	8-42	MKMHRM	8-64
GATE	8-43	MKMIN	8-64
GD	8-43	MKML ?	8-65
GE	8-44	MKMP	8-65
GL	8-44	MKMULTI	8-66
GPIA	8-45	MKN	8-66
GTOUT	8-45	MKOFF	8-67
HN	8-46	MKP	8-67
HNLOCK	8-46	MKPK	8-68
HNUNLK	8-47	MKR	8-68
INI	8-47	MKRL	8-69
IP	8-48	MKS	8-69
KSA	8-48	MKSLCT	8-70
KSB	8-49	MKSP	8-70
KSC	8-49	MKSS	8-71
KSD	8-50	MKTRACE	8-71
KSE	8-50	MKTRACK	8-72
KSG	8-51	MKW	8-72
KSH	8-51	MKZ	8-73
KSO	8-52	MKZF	8-73
LG	8-52	MNOISE	8-74
LN	8-53	MOBW	8-74
LSS	8-53	MON	8-75
LSSA	8-54	MONVOL	8-75
MADJBWLN	8-54	MTEMP	8-76
MADJCTRLN	8-55	MTEMPDEL	8-76
MADJGRAPH	8-55	MTEMPDSP	8-77
MADJMOD	8-56	MTEMPIN	8-77
MC	8-56	MTEMPINI	8-78
MEAS	8-57	MTEMPPL	8-78
MKA ?	8-58	MTEMPLABEL	8-79
MKACT	8-58	MTEMPPD ?	8-79
MKC	8-59	MTEMPREL	8-80
MKCF	8-59	MTEMPRP	8-80
MKD	8-60	MT0	8-81
MKF ?	8-60	MT1	8-81

## TABLE OF CONTENTS ( Continued )

MXMH .....	8-82	RLV .....	8-106
MZW .....	8-82	RMK ? .....	8-107
MZWF .....	8-83	ROFFSET .....	8-107
M1 .....	8-83	SCL .....	8-108
M2 .....	8-84	SCR .....	8-108
M3 .....	8-84	SNGLS .....	8-109
OBWN .....	8-85	SOF .....	8-109
OBWXDB .....	8-85	SP .....	8-110
PARAM .....	8-86	SPD .....	8-110
PCF .....	8-86	SPF .....	8-111
PLF .....	8-87	SPFUNC .....	8-111
PLI .....	8-87	SPU .....	8-112
PLOT .....	8-88	SS .....	8-112
PLS .....	8-88	SSS .....	8-113
PLTA .....	8-89	ST .....	8-113
PLTARA .....	8-89	STF .....	8-114
PLTHOME .....	8-90	SV .....	8-114
PLTLC .....	8-90	SVM .....	8-115
PMCS .....	8-91	SWP .....	8-115
PMOD .....	8-91	SWSTART .....	8-116
PMY .....	8-92	SWSTOP .....	8-116
PORT .....	8-92	SWT .....	8-117
PP .....	8-93	S1 .....	8-117
PRESEL .....	8-93	S2 .....	8-118
PRIA .....	8-94	TDLY .....	8-118
PRINT .....	8-94	TEMP .....	8-119
PRL .....	8-95	TEMPMCL .....	8-119
PSW .....	8-95	TEMPMSV .....	8-120
PTA .....	8-96	TEMPMVX .....	8-120
PTL .....	8-96	TEMPMVY .....	8-121
PWRSTART .....	8-97	TEMPSLCT .....	8-121
PWRSTOP .....	8-97	TEXPAND .....	8-122
RB .....	8-98	TIME .....	8-122
RBW .....	8-98	TITLE .....	8-123
RC .....	8-99	TM .....	8-123
RCM .....	8-99	TMCNT ? .....	8-124
RDATA .....	8-100	TMMD .....	8-124
RES ? .....	8-101	TMWR .....	8-125
RGDIR .....	8-103	TRG .....	8-125
RGRC .....	8-103	TRGLVL .....	8-126
RGSV .....	8-104	TRGS .....	8-126
RL .....	8-105	TRGSLP .....	8-127

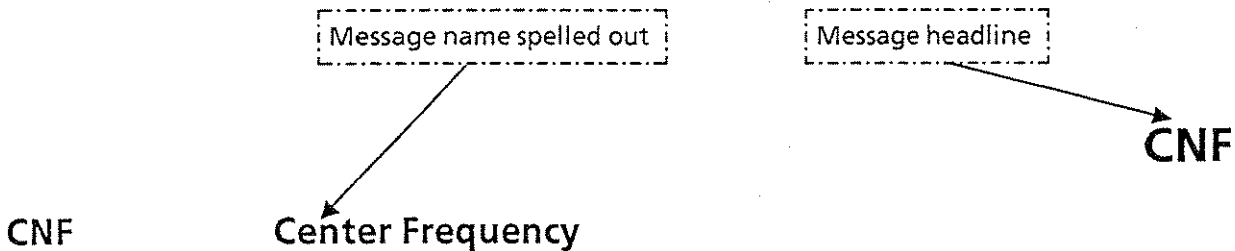
## TABLE OF CONTENTS ( Continued )

TRGSOURCE .....	8-127
TRM .....	8-128
TS .....	8-128
TSP .....	8-129
TTL .....	8-129
TVLINE .....	8-130
TVSFRM .....	8-130
TVSTND .....	8-131
TZONE .....	8-131
TZSP .....	8-132
TZSTART .....	8-132
UCL ? .....	8-133
UNC .....	8-133
UNT .....	8-134
VAVG .....	8-134
VB .....	8-135
VBCOUPLE .....	8-135
VBR .....	8-136
VBW .....	8-136
VIEW .....	8-137
WINDPOS .....	8-137
XMA .....	8-138
XMB .....	8-138
XMG .....	8-139
XMT .....	8-139
*CLS .....	8-140
*ESE .....	8-140
*ESR ? .....	8-141
*IDN ? .....	8-141
*OPC .....	8-142
*OPC ? .....	8-142
*RST .....	8-143
*SRE .....	8-143
*STB ? .....	8-144
*TRG .....	8-144
*TST ? .....	8-145
*WAI .....	8-145



## SECTION 8 DETAILED DESCRIPTION OF COMMANDS

The pages that follow explain the device messages in alphabetical order in the format shown below.



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

Header	Program command message	Program query message	Response message
CNF	CNF_f	CNF?	CNF_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 <sup>0</sup> )	} • The left side of the colon is part of the program or response data • The right side of the colon describes the data
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	MS2602A device-dependent initial setting value
CNF_50MHZ	
CNF?	

## AAT

### AAT RF Attenuator

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

Header	Program command	Query	Response
AAT	AAT_a a=0, 1	AAT?	AAT_a a=0, 1

- **Value of a** 0: MANUAL  
1: AUTO
- **Suffix code** None
- **Initial setting** 1: AUTO
- **Example** AAT\_1

## ADJCH

### ADJCH Adjacent CH Select

- **Function** Selects the subject channel to be calculated for an adjacent channel.

Header	Program command	Query	Response
ADJCH	ADJCH_sw sw=BOTH, UP, LOW, OFF	ADJCH?	SW sw=BOTH, UP, LOW, OFF

- **Value of sw** BOTH: BOTH SIDES  
UP: UPPER SIDE  
LOW: LOWER SIDE  
OFF: OFF
- **Suffix code** None
- **Initial setting** BOTH: BOTH SIDES
- **Example** ADJCH\_BOTH  
ADJCH\_LOW

**ADJCHBW****ADJCHBW**      **Adjacent CH Bandwidth**

- **Function**      Sets the bandwidth of adjacent channel.

Header	Program command	Query	Response
ADJCHBW	ADJCHBW_f	ADJCHBW?	f f= 10 to 9999990 Transfers the data without suffix code as if represented in units of Hz

- **Value of f**      10 Hz to 9.99999 MHz  
( 10 Hz resolution , a number of less than 10 Hz digits is truncated. )
- **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**      8.5KHZ:      8.5 kHz
- **Example**      ADJCHBW\_8.5KHZ

**ADJCHSP****ADJCHSP**      **Adjacent CH1 Sepalation**

- **Function**      Sets the separation of adjacent channel 1.

Header	Program command	Query	Response
ADJCHSP	ADJCHSP_f	ADJCHSP?	f f= 0 to 9999990 Transfers the data without suffix code as if represented in units of Hz

- **Value of f**      10 Hz to 9.99999 MHz  
( 10 Hz resolution , provided a number of less than 10 Hz digits is truncated. )
- **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**      12.5KHZ:      12.5 kHz
- **Example**      ADJCHSP\_12.5kHz

## ADJCHSPF

### ADJCHSPF Adjacent CH2 Separation

- **Function** Sets the separation of adjacent channel 2.

Header	Program command	Query	Response
ADJCHSPF	ADJCHSPF_f	ADJCHSPF?	f f=0 to 9999990 Transfers the data without suffix code as if represented in units of Hz

- **Value of f** 10 Hz to 9999990 Hz  
(10 Hz resolution, a number of less than 10 Hz digits is truncated.)
- **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** 25.0KHZ
- **Example** ADJCHSPF\_25.0kHz

## AMD

### AMD Trace A Storage Mode

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

Header	Program command	Query	Response
AMD	AMD_a a=0 to 5	AMD?	AMD_a a=0 to 5

- **Value of a**
  - 0: NORMAL
  - 1: MAX HOLD
  - 2: AVERAGE
  - 3: MIN HOLD
  - 4: CUMULATIVE
  - 5: OVER WRITE
- **Suffix code** None
- **Initial setting** 0: NORMAL
- **Example** AMD\_0

**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** 0: MANUAL  
1: AUTO
- **Suffix code** None
- **Initial setting** 1: AUTO
- **Example** ARB\_0  
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** 0: MANUAL  
1: AUTO
- **Suffix code** None
- **Initial setting** 1: AUTO
- **Example** AST\_0  
AST\_1

## ASWT

### ASWT Auto Sweep Time

- **Function** Sets the auto sweep time.

Header	Program command	Query	Response
ASWT	ASWT_sw sw = FAST, SLOW	ASWT?	SW sw = FAST, SLOW

- **Value of sw** FAST: FAST  
SLOW: NORMAL
- **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_a a= AUTO, UP, DN, 0 to 55 5 step	AT?	a a=0 to 55

- **Value of a** AUTO: AUTO  
UP: UP  
DN: DOWN  
Ø to 55(5step): 0 to 55dB (5dB step)
- **Suffix code** None: dB  
DB: dB
- **Initial setting** Calculated value when AUTO is selected for ATT
- **Example** AT\_1Ø  
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**

0:	0 dB	6:	5 dB
1:	10 dB	7:	15 dB
2:	20 dB	8:	25 dB
3:	30 dB	9:	35 dB
4:	40 dB	10:	45 dB
5:	50 dB	11:	55 dB

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

**ATUN****ATUN Auto Tune**

- **Function** Detects the maximum peak point within the frequency band specified in the BG (BackGround) band and displays its spectrum in the center of the screen in CENTER-SPAN mode.

Header	Program command	Query	Response
ATUN	ATUN	—	—

- **Example** ATUN

## 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, W	AUNITS?	a a = DBM, DBUV, DBMV, DBUVE, V, W

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

## AUTO

### AUTO Coupled Function All Auto

- **Function** Executes all coupled functions (RBW, VBW, SWT, ATT) 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**    0: MANUAL  
                    1: AUTO  
                    2: OFF

- **Suffix code**    None

- **Initial setting** 1: AUTO

- **Example**        AVB\_0  
                    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**    0: 4 times  
                    1: 8 times  
                    2: 16 times  
                    3: 32 times  
                    4: 128 times

- **Suffix code**    None

- **Initial setting** 1: 8 times

- **Example**        AVR\_0  
                    AVR\_3

## AWR

### AWR Trace A Write Switch

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

Header	Program command	Query	Response
AWR	AWR_a a=ON, 1, OFF, 0	AWR?	AWR_a 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 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 waveform to be processed in MAX HOLD mode (same function as AMD\_1).

Header	Program command	Query	Response
A2	A2	—	—

- **Example** A2

**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

## BGWR

### BGWR Trace BG Write Switch

- **Function** Controls writing the waveform data to trace BG.

Header	Program command	Query	Response
BGWR	BGWR_sw sw = ON, 1, OFF, 0	BGWR?	BGWR_sw sw = ON, OFF

- **Value of sw**
  - ON: TRACE BG WRITE ON ( Same function as CLRW\_TRBG )
  - 1: TRACE BG WRITE ON ( Same function as CLRW\_TRBG )
  - OFF: TRACE BG WRITE OFF ( Same function as VIEW\_TRBG )
  - Ø: TRACE BG WRITE OFF ( Same function as VIEW\_TRBG )
- **Suffix code** None
- **Initial setting** ON: TRACE BG WRITE ON
- **Example** BGWR\_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

**BMD****BMD Trace B Storage Mode**

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

Header	Program command	Query	Response
BMD	BMD_a a=0 to 5	BMD?	BMD_a a=0 to 5

- **Value of a**
  - Ø: NORMAL
  - 1: MAX HOLD
  - 2: AVERAGE
  - 3: MIN HOLD
  - 4: CUMULATIVE
  - 5: OVER WRITE

- **Suffix code** None

- **Initial setting** Ø: NORMAL

- **Example** BMD\_Ø

**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

## BWR

### BWR      Trace B Write Switch

- **Function**      Controls writing the waveform data to trace B.

Header	Program command	Query	Response
BWR	BWR_a      a=ON, 1, OFF, 0	BWR?	BWR_a      a=ON, OFF

- **Value of a**      ON:      TRACE B WRITE ON ( Same function as CLRW\_TRB )  
                          1:      TRACE B WRITE ON ( Same function as CLRW\_TRB )  
                          OFF:    TRACE B WRITE OFF ( Same function as VIEW\_TRB )  
                          Ø:      TRACE B WRITE OFF ( Same function as VIEW\_TRB )
- **Suffix code**      None
- **Initial setting**    1:    TRACE B WRITE ON
- **Example**      BWR\_Ø

**BZR****BZR**                      **Sounds Buzzer**

■ **Function**                      Sounds the buzzer.

Header	Program command	Query	Response
BZR	BZR	—	—

■ **Example**                      BZR

**B1****B1**                              **Trace B Write ON**

■ **Function**                      Clears trace B waveform data to set the write mode to ON (same function as BWR\_1, CLRW\_TRB).

Header	Program command	Query	Response
B1	B1	—	—

■ **Example**                      B1

**B2****B2 Trace B Max Hold**

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

Header	Program command	Query	Response
B2	B2	—	—

- **Example** B2

**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**     $\emptyset$ : All  
                       1: Frequency  
                       2: Level
- **Suffix code**    None
- **Example**        CAL\_ $\emptyset$

## 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 850000000 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 (  $10^0$  )        a: None  
                           HZ:        Hz (  $10^0$  )  
                           KHZ, KZ:    kHz (  $10^3$  )  
                           MHZ, MZ:    MHz (  $10^6$  )  
                           GHZ, GZ:    GHz (  $10^9$  )
- **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	—	—

- **Value of tr**
  - TRA: Trace A (same function as AWR\_1)
  - TRB: Trace B (same function as BWR\_1)
  - TRBG: Trace BG (same function as BGWR\_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_f	CNF?	CNF_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 ( $10^0$ )
  - HZ: Hz ( $10^0$ )
  - KHZ, KZ: kHz ( $10^3$ )
  - MHZ, MZ: MHz ( $10^6$ )
  - GHZ, GZ: GHz ( $10^9$ )
- **Initial setting** Value of f = 4.25 GHz
- **Example**
  - CNF\_123456
  - CNF\_50MHZ
  - CNF?

**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

## CORC

### CORC Correction Factor Initialization

- **Function** Initializes the correction factor currently selected by the CORR command.

Header	Program command	Query	Response
CORC	CORC	—	—

- **Example** CORC

All frequency data and level data are initialized. The initialized data is used as the correction value of 0 dB in each frequency range.

## CORD

### CORD Correction Factor Entry

- **Function** Registers the correction factor currently selected by the CORR command. If the correction factor is selected for OFF, it is not valid.

Header	Program command	Query	Response
CORD	CORD_n, f, ℓ n=0 to 149 f= 0 to 400 GHz ℓ= -100.00 to +100.00 dB (incremented by 0.01 dB steps)	CORD?_n	CORD_n, f, ℓ n=0 to 149 f= 0 to 400 GHz (no units) ℓ= -100.00 to +100.00 dB (incremented by 0.01 dB steps)

- **Value of n** 0 to 149
- **Value of f** 0 to 400 GHz
- **Value of ℓ** -100.00 to +100.00 dB (incremented by 0.01 dB steps)
- **Suffix code**
  - f: None: Hz (10<sup>0</sup>)
  - HZ: Hz (10<sup>0</sup>)
  - KHZ: kHz (10<sup>3</sup>)
  - MHZ: MHz (10<sup>6</sup>)
  - GHZ: GHz (10<sup>9</sup>)
  - ℓ: None: dB
  - DB: dB

- **Example** CORD\_0, 1MHZ, 10  
CORD\_1, 2000000, 10  
If (fn - 1 < fn < fn + 1) is not satisfied when (n - 1 < n < n + 1), an error occurs.

**CORR****CORR Correction Factor Select**

- **Function** Selects the type of correction factor.

Header	Program command	Query	Response
CORR	CORR_a a=0 to 5, OFF	CORR?	CORR_a a=0 to 5

- **Value of a**
  - OFF: OFF
  - Ø : OFF
  - 1 : CORR1
  - 2 : CORR2
  - 3 : CORR3
  - 4 : CORR4
  - 5 : CORR5
- **Suffix code** None
- **Initial setting** Ø: OFF (provided the correction factor already registered is not initialized)
- **Example**
  - CORR\_Ø
  - CORR\_2
  - CORR\_4

**CORRLABEL****CORRLABEL Correction Factor Label**

- **Function** Registers the name of the correction factor currently selected by the CORR command.

Header	Program command	Query	Response
CORRLABEL	CORRLABEL_n, text	CORRLABEL?_n	"text"

- **Value of text** A string of up to 30 characters enclosed by double or single quotation marks.
- **Value of n** 1 to 5
- **Suffix code** None
- **Example**
  - CORRLABEL\_1, "CORRECTION FACTOR"
  - CORRLABEL\_2, 'MS26Ø2A'

## COUPLE

### COUPLE Coupling Mode

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

Header	Program command	Query	Response
COUPLE	COUPLE_sw	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

### 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

## CRS

**CRS**      **Count Resolution**

- **Function**      Selects the frequency resolution of frequency counter.

Header	Program command	Query	Response
CRS	CRS_a                      a=0 to 3	CRS?	CRS_a                      a=0 to 3

- **Value of a**      0: 1 Hz  
                          1: 10 Hz  
                          2: 100 Hz  
                          3: 1 kHz
- **Suffix code**    None
- **Initial setting** 3: 1 kHz
- **Example**        CRS\_0  
                          CRS\_3

## CT

**CT**      **Sweep Time Auto**

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

Header	Program command	Query	Response
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

---

## DATE

### DATE Date

- **Function** Sets the clock built in MS2602A 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



**DDTYP****DDTYP      DATE DISP**

- **Function**      Set the display format of the date.

Header	Program command	Query	Response
DDTYP	DDTYP_a                      a=0,1,2	DDTYP_a	DDTYP_a                      a=0,1,2

- **Value of a**      Ø:    YY/MM/DD  
                       1:    MMM-DD-YY  
                       2:    DD-MMM-YY
- **Initial setting** YY/MM/DD
- **Example**        DDTYP\_1

**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

## 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	DETM?_tr	SW sw=POS,SMP,NEG

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

**DFMT****DFMT Display Format**

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

Header	Program command	Query	Response
DFMT	DFMT_sw	DFMT?	SW sw=A, B, TIME, AB1, AB2, ABG1, ABG2, ATIME1, ATIME2

- **Value of sw**
  - A: Trace A
  - B: Trace B
  - TIME: Trace TIME
  - AB1: Trace A / Trace B ( A & B )
  - AB2: Trace A / Trace B ( A / B )
  - ABG1: Trace A / Trace BG ( BG > A )
  - ABG2: Trace A / Trace BG ( BG < A )
  - ATIME1: Trace A / Trace TIME ( TIME > A )
  - ATIME2: Trace A / Trace TIME ( TIME < A )

- **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_sw sw = NRM, DOUBLE	DPOINT?	SW sw = NRM, DOUBLE

- **Value of sw** NRM: 501 points  
DOUBLE: 1002 points
- **Suffix code** None
- **Initial setting** NRM: 501 points (provided the address already allocated is not initialized)
- **Example** DPOINT\_NRM

**DSPLV****DSPLV Marker Level Absolute / Relative**

- **Function** Specifies the marker level in the absolute value display or in the relative value display when seen from the display line.

Header	Program command	Query	Response
DSPLV	DSPLV_sw	DSPLV?	SW

- **Value of sw** ABS: Absolute value  
REL: Relative value
- **Suffix code** None
- **Initial setting** ABS: Absolute value
- **Example** DSPLV\_REL

## DSPLVM

### DSPLVM Marker Level Absolute / Relative

- **Function** With the trace mode specified, also specifies the marker level in the absolute value display or in the relative value display when seen from the display line.

Header	Program command	Query	Response
DSPLVM	DSPLVM_tr,sw	DSPLVM?_tr	sw

- **Value of tr** TRA :Trace A  
TRB :Trace B  
TRTIME:Trace Time  
TRBG :Trace BG
- **Value of sw** ABS: Absolute value  
REL: Relative value
- **Suffix code** None
- **Initial setting** ABS: Absolute value
- **Example** DSPLVM\_TRA, REL

## DSUA

### DSUA DSU (Data Storage Unit) Address

- **Function** Sets the GPIB address of the MC8104A Data Storage Unit.

Header	Program command	Query	Response
DSUA	DSUA_a a = 0 to 30	DSUA?	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** DSUA\_20

**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

**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?

## EXTTYPE

### EXTTYPE Ext Trigger Input Type

- **Function** Chooses the level input from the external trigger when EXT is selected for trigger source.

Header	Program command	Query	Response
EXTTYPE	EXTTYPE_sw	EXTTYPE?	SW sw = 10V, TTL

- **Value of sw** 10V: INPUT 1 ( $\pm 10V$ )  
TTL: INPUT 2 (TTL)
- **Suffix code** None
- **Initial setting** 10V: INPUT 1 ( $\pm 10V$ )
- **Example** EXTTYPE\_10V  
EXTTYPE\_TTL

## E1

### 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****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****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 <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 = 0 Hz
- **Example** FA\_1GZ

**FADJ****FADJ****FINE ADJ**

- **Function** Set the ON/OFF of FINE ADJ, and frequency.

Header	Program command	Query	Response
FADJ	FADJ_ON FADJ_OFF FADJ_f	_____ FADJ? FADJ?	_____ FADJ_OFF f (Output the value without units or in Hz) (Return the frequency value when it is ON, and return "FADJ_OFF" when it is OFF.)

- **Value of f** -50 to 50 kHz
- **Suffix code** None: Hz  
KHZ, KZ: kHz
- **Initial setting** -1.5 kHz
- **Example** FADJ\_1KHZ

**FB****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<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** 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

**FRQ**      **Frequency Mode**

- **Function**      Selects the mode for setting the FG frequency band.

Header	Program command	Query	Response
FRQ	FRQ_a                      a=0 to 2	FRQ?	FRQ_a                      a=0 to 2

- **Value of a**      0: CENTER-SPAN  
1: START-SPAN  
2: START-STOP
- **Suffix code**    None
- **Initial setting** 2: START-STOP
- **Example**        FRQ\_0  
                      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 Center Frequency Step Up

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

Header	Program command	Query	Response
FUP	FUP	—	—

- **Example** FUP

**GATE****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 sw=ON, 1, OFF, 0	GATE?	SW sw=ON, OFF

- **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 t=0 to 65500 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  
EXT: EXTERNAL
- **Suffix code** None
- **Initial setting** INT: INTERNAL
- **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



**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 Talker time out**

- **Function** Sets the time out of the GPIB talker function (plotter/printer output, data output from PTA, etc)

Header	Program command	Query	Response
GTOUT	GTOUT_t	GTOUT?	t

- **Value of t** 1to255: 1 sec to 255 sec  
0: No time out (infinite wait state)
- **Suffix code** None
- **Initial setting** 30: 30 sec
- **Example** GTOUT\_60

# HN

## 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_a 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_a a=0 to 2, OFF	HNLOCK?	HNLOCK_b 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

**INI****INI Initialize**

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

Header	Program command	Query	Response
INI	INI	—	—

- **Example** INI

**IP****IP Initialize**

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

Header	Program command	Query	Response
IP	IP	—	—

- **Example** IP

**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****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****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

## KSE

### 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\_ "MS2602A"  
KSE\_ 'SPECTRUM ANALYZER'

**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****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

### 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 Y axis scale to linear mode.

Header	Program command	Query	Response
LN	LN	—	—

- **Example**                      LN

LSS

**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

### 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

## MADJBWLN

### MADJBWLN ADJ-CH Band Line

- **Function** Turns the display of the adjacent channel range line ON/OFF.

Header	Program command	Query	Response
MADJBWLN	MADJBWLN_sw	MADJBWLN?	sw

- **Value of sw**
  - OFF : OFF
  - ON : ON
- **Suffix code** None
- **Initial setting** OFF : OFF
- **Example** MADJBWLN\_OFF

## MADJCTRLN

### MADJCTRLN ADJ-CH Center Line

- **Function** Turns the display of the adjacent channel center line ON/OFF.

Header	Program command	Query	Response
MADJCTRLN	MADJCTRLN_sw	MADJCTRLN?	sw

- **Value of sw** OFF:OFF  
ON :ON
- **Suffix code** None
- **Initial setting** ON:ON
- **Example** MADJCTRLN\_OFF

## MADJGRAPH

### MADJGRAPH Adjacent CH Graph

- **Function** Specifies the graph display function of Adj-CH measure with the ON/OFF parameter.

Header	Program command	Query	Response
MADJGRAPH	MADJGRAPH_a a=ON, OFF	MADJGRAPH?	a

- **Value of a** OFF:GRAPH OFF  
ON :GRAPH ON
- **Suffix code** None
- **Initial setting** ON:Graph ON
- **Example** MADJGRAPH\_ON

## MADJMOD

### MADJMOD ADJ-CH Measure Method

- **Function** Selects the calculation method of Adj-CH measure.

Header	Program command	Query	Response
MADJMOD	MADJMOD_sw sw = MOD, UNMD	MADJMOD?	SW

- **Value of sw** MOD : Reference = Total Power (Mod method)  
UNMD : Reference = REF LEVEL (un-mod method)
- **Suffix code** None
- **Initial setting** MOD : R : Total Power
- **Example** MADJMOD\_MOD

## MC

### MC Frequency Counter

- **Function** Turns ON or OFF the function for measuring the marker frequency during display with the counter (same function as MEAS\_FREQ).

Header	Program command	Query	Response
MC	MC_a	—	—

- **Value of a** ON: ON  
OFF: OFF
- **Suffix code** None
- **Initial setting** OFF: OFF
- **Example** MC\_ON  
MC\_OFF

**MEAS****MEAS Measure Function**

- **Function** Executes each item of the Measure functions when specified.

Header	Program command	Query	Response
MEAS	MEAS_data1,data2	MEAS?	data1 data1 = OFF, FREQ, NOISE, OBW, ADJ, TEMP, POWER

■ **Value of data1, data2**

Format1: Specifies the measure item and whether to switch it ON/OFF or execute it

- OFF : Measure OFF
- FREQ, ON : Frequency count ON
- FREQ, OFF : Frequency count OFF
- NOISE, ON : Noise calculation ON
- NOISE, OFF : Noise calculation OFF
- OBW, EXE : Executes the OBW calculation
- ADJ, EXE : Executes the ADJ-CH calculation
- TEMP, ON : Sets template line display to ON
- TEMP, OFF : Sets template line display to OFF
- TEMP, CHECK : Executes template check (and sets template line display to ON)
- POWER, EXE : Executes the burst power calculation

Format2: Specifies the measure item and calculation system, then whether to switch it ON/OFF or execute it

- NOISE, ABS : Sets noise calculation (Absolute method) to ON
- NOISE, CN : Sets noise calculation (C/N ratio method) to ON
- OBW, XDB : Executes OBW calculation (X dB down method)
- OBW, N : Executes OBW calculation (N% method)
- ADJ, UNMD : Executes ADJ-CH calculation (R:Ref Level method)
- ADJ, MOD : Executes ADJ-CH calculation (R:Total Power method)

## 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 W 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 W** No unit, Level data with 1 pw unit (when unit system for marker level is W), Resolution 1 aW
- **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

**MKC****MKC**      **Frequency Counter**

- **Function**      Turns ON or OFF the function for measuring the marker frequency during display with the counter (same function as MEAS\_FREQ).

Header	Program command	Query	Response
MKC	MKC_a                      a=0, 1	MKC?	MKC_a                      a=0, 1

- **Value of a**      Ø: OFF  
                          1: ON
- **Suffix code**    None
- **Initial setting** Ø: OFF
- **Example**        MKC\_Ø  
                          MKC\_1

**MKCF****MKCF**      **Marker to CF**

- **Function**      Sets the marker frequency to the center frequency (same function as MKR\_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?



## MKFC

**MKFC**      **Frequency Counter**

- **Function**      Turns ON or OFF the function for measuring the marker frequency during display with the counter (same function as MEAS\_FREQ).

Header	Program command	Query	Response
MKFC	MKFC_a      a=ON, 1, OFF, 0	MKFC?	a      a=0, 1

- **Value of a**      ON:      ON  
                          1:      ON  
                          OFF:    OFF  
                          Ø:      OFF
- **Suffix code**    None
- **Initial setting** Ø: OFF
- **Example**      MKFC\_Ø  
                          MKFC\_ON

## MKFCR

**MKFCR**      **Count Resolution**

- **Function**      Selects the frequency resolution of frequency counter.

Header	Program command	Query	Response
MKFCR	MKFCR_f MKFCR_a	MKFCR?	f f=1, 10, 100, 1000 Transfers the data without suffix code as if represented in units of Hz

- **Value of f**      1 Hz  
                          10 Hz  
                          100 Hz  
                          1 kHz
- **Value of a**      UP:    UP  
                          DN:    DOWN
- **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** 1 kHz
- **Example**      MKFCR\_1HZ  
                          MKFCR\_UP

## 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?	ℓ v W 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 W** No unit, Level data with 1 pW unit (when display unit system for marker level is W), Resolution 1 aw
- **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 Multi Marker**

- **Function** 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 Multi Marker

- **Function** 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

### 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?****MKML? Multi Marker List Query ( Level )**

- **Function** Reads out the level data at multimarkers.

Header	Program command	Query	Response
MKML	_____	MKML?_a	ℓ v w 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 w** No unit, Level data with 1 pW unit (when display unit system for marker level is W), Resolution 1 aw
- **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 OFF: OFF  
1: ON Ø: OFF
- Suffix code None
- Initial setting OFF: OFF
- Example MKMULTI\_ON

## MKN

### 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 ( $10^0$ ) 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 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

### 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**
  - 0: 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** 0: NORMAL

- **Example** MKR\_0



**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**
- Ø:    SEARCH PEAK ( MAX )
  - 1:    SEARCH NEXT PEAK
  - 2:    SEARCH DIP ( MIN )
  - 9:    SEARCH NEXT RIGHT PEAK
  - 1Ø:  SEARCH NEXT LEFT PEAK
  - 11:  SEARCH NEXT DIP

- **Suffix code**    None

- **Example**      MKS\_Ø  
                      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  
Ø: OFF
- **Suffix code** None
- **Initial setting** OFF: OFF
- **Example** MKSLCT\_3, ON

## MKSP

### 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****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

**MKTRACE****MKTRACE**      **Active Marker Trace**

- **Function**      Specifies the trace for displaying the marker when the display format is trace A/B.

Header	Program command	Query	Response
MKTRACE	MKTRACE_tr	MKTRACE?	tr

- **Value of tr**      TRA: Trace A  
                          TRB: Trace B
- **Suffix code**      None
- **Initial setting**    TRA: Trace A
- **Example**          MKTRACE\_TRB

## MKTRACK

### MKTRACK Tracking ON / OFF

- **Function** Sets the signal tracking function to ON or OFF.

Header	Program command	Query	Response
MKTRACK	MKTRACK_SW sw=ON, 1, OFF, 0	MKTRACK?	SW sw=ON, OFF

- **Value of sw**
  - ON: ON
  - 1: ON
  - OFF: OFF
  - Ø: OFF
- **Suffix code** None
- **Initial setting** OFF: OFF
- **Example** MKTRACK\_ON

## MKW

### MKW Zone Marker Width

- **Function** Specifies the zone marker width in the div unit.

Header	Program command	Query	Response
MKW	MKW_a a=0 to 2, 5 to 7	MKW?	MKW_a a=0 to 2, 5 to 7

- **Value of a**
  - Ø: 0.5 div
  - 1: Spot
  - 2: 10 div
  - 5: 1 div
  - 6: 2 div
  - 7: 5 div
- **Suffix code** None
- **Initial setting** 5: 1 div
- **Example**
  - MKW\_1
  - MKW\_5

**MKZ****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=0 to 500	MKZ?	MKZ_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****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_f MKZF_t	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 ( $10^0$ )  
HZ: Hz ( $10^0$ )  
KHZ, KZ: kHz ( $10^3$ )  
MHZ, MZ: MHz ( $10^6$ )  
GHZ, GZ: GHz ( $10^9$ )  
t: None: msec  
US:  $\mu$ sec  
MS: msec  
S: sec
- **Example** MKZF\_100MHZ  
MKZF\_1200000000

## MNOISE

### MNOISE Noise Measure Method

- **Function** Selects the calculation method of noise measure.

Header	Program command	Query	Response
MNOISE	MNOISE_sw sw = ABS, CN	MNOISE?	SW

- **Value of sw** ABS : Absolute method  
CN : C/N Ratio method
- **Suffix code** None
- **Initial setting** ABS : Absolute
- **Example** MNOISE\_ABS

## MOBW

### MOBW OBW Measure Method

- **Function** Selects the calculation method of OBW.

Header	Program command	Query	Response
MOBW	MOBW_sw sw = XDB, N	MOBW?	SW

- **Value of sw** XDB : X dB down method  
N : N% method
- **Suffix code** None
- **Initial setting** N : N% method
- **Example** MOBW\_N

**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 (1step)
- **Suffix code**      None
- **Initial setting**    Value of v = 10
- **Example**          MONVOL\_10

## MTEMP

### MTEMP Select Template

- **Function** Selects one of the 1 to 5 templates used for template management functions.

Header	Program command	Query	Response
MTEMP	MTEMP_n n=1 to 5	MTEMP?	n n=1 to 5

- **Value of n** 1 to 5 ( template No. )
- **Suffix code** None
- **Initial setting** 1
- **Example** MTEMP\_1

---

## MTEMPDEL

### MTEMPDEL Delete Template

- **Function** Deletes the template data by 1 point.

Header	Program command	Query	Response
MTEMPDEL	MTEMPDEL_p	—	—

- **Value of p** 1 to 32 ( Point No. )
- **Suffix code** None
- **Initial setting** (None)
- **Example** MTEMPDEL\_10



**MTEMPDSP****MTEMPDSP**      **Template Display Mode**

- **Function**                      Specifies how the template management screen is displayed.

Header	Program command	Query	Response
MTEMPDSP	MTEMPDSP_sw	MTEMPDSP?	SW                      sw = GRAPH, LIST

- **Value of sw**      GRAPH: GRAPH  
LIST: LIST
- **Suffix code**      None
- **Initial setting**      LIST
- **Example**              MTEMPDSP\_GRAPH

**MTEMPIN****MTEMPIN**      **Insert Point**

- **Function**                      Adds the template data by 1 point.

Header	Program command	Query	Response
MTEMPIN	MTEMPIN_p, t, ℓ	—	—

- **Value of p**              1 to 32 ( Point No. )
- **Value of t**              - 1000 sec to 1000 sec
- **Value of ℓ**              - 200.00 dBm to 200.00 dBm ( ABSOLUTE )  
- 200.00 dB to 200.00 dB ( RELATIVE )
- **Suffix code**              p: None  
t: None: msec                      ℓ: None: dB or dBm  
US: μsec                              DB, DBM, DM: dB or dBm  
MS: msec  
S: sec
- **Initial setting**      (None)
- **Example**                      MTEMPIN\_3, 10MS, -20.5DBM

## MTEMPINI

### MTEMPINI Initiate Line / Template

- **Function** Initializes the template limit line data.

Header	Program command	Query	Response
MTEMPINI	MTEMPINI_a	—	—

- **Value of a** UP1: LIMIT 1 UPPER  
UP2: LIMIT 2 UPPER  
LW1: LIMIT 1 LOWER  
LW2: LIMIT 2 LOWER
- **Suffix code** None
- **Example** MTEMPINI\_UP1

## MTEMPL

### MTEMPL Select Line

- **Function** Selects the type of limit lines used for template management functions.

Header	Program command	Query	Response
MTEMPL	MTEMPL_a	MTEMPL?	a

- **Value of a** UP1: LIMIT 1 UPPER  
UP2: LIMIT 2 UPPER  
LW1: LIMIT 1 LOWER  
LW2: LIMIT 2 LOWER
- **Suffix code** None

**MTEMPLABEL****MTEMPLABEL** Template Label

- **Function** Specifies the template label (name)

Header	Program command	Query	Response
MTEMPLABEL	MTEMPLABEL_n, text	MTEMPLABEL?n	text

- **Value of n** 1 to 5 (Template No.)
- **Value of text** character string within 24 words enclosed by double quotation marks or single quotation marks.
- **Suffix code** None
- **Initial setting** (None)
- **Example** MTEMPLABEL\_1, "RCR-28"  
MTEMPLABEL\_2, 'CHECKØ1'

**MTEMPPD?****MTEMPPD?** Read Limit Line Point Data

- **Function** Reads out the template data by 1 point.

Header	Program command	Query	Response
MTEMPPD?	_____	MTEMPPD?_p p=1 to 32	t ℓ t = -1000000000 to 1000000000 Transfers the data without suffix code as if represented in units of μsec ℓ = -200.00 to 200.00 Transfers the data without suffix code as if represented in units of dB

- **Value of p** 1 to 32 (Point No.)
- **Suffix code** None
- **Initial setting** (None)
- **Example** MTEMPPD?\_ℓ

## MTEMPREL

### MTEMPREL Template Level Mode

- **Function** Allows the template level data to be set either in relative values or in absolute values.

Header	Program command	Query	Response
MTEMPREL	MTEMPREL_sw sw=ON,OFF	MTEMPREL?	sw sw=ON,OFF

- **Value of sw** ON: RELATIVE  
OFF: ABSOLUTE
- **Suffix code** None
- **Initial setting** OFF: ABSOLUTE
- **Example** MTEMPREL\_ON

## MTEMPRP

### MTEMPRP Replace Point

- **Function** Replaces the template data by 1 point.

Header	Program command	Query	Response
MTEMPRP	MTEMPRP_p, t, ℓ	—	—

- **Value of p** 1 to 32 ( Point No. )
- **Value of t** - 1000 sec to 1000 sec
- **Value of ℓ** - 200.00 dBm to 200.00 dBm ( ABSOLUTE )  
- 200.00 dB to 200.00 dB ( RELATIVE )
- **Suffix code** p: None ℓ: None: dB or dBm  
t: None: msec DB, DBM, DM: dB or dBm  
US: μsec  
MS: msec  
S: sec
- **Initial setting** ( None )
- **Example** MTEMPRP\_3, 10MS, -20.5DBM

**MT0****MT0 Tracking OFF**

■ **Function** Sets the signal tracking function to OFF.

Header	Program command	Query	Response
MT0	MT0	—	—

■ **Example** MT0

**MT1****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	—	—

- **Value of tr** TRA: Trace A  
TRA: Trace B
- **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****MZWF Zone Marker Width**

- **Function** Specifies the zone marker width on the X axis in the frequency or time units.

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



**OBWN****OBWN      OBW N% Value**

- **Function**      Sets the conditions of the occupied frequency bandwidth in % unit.

Header	Program command	Query	Response
OBWN	OBWN_r	OBWN?	r r = 1 to 99 Transfers the data without suffix code as if represented in % unit

- **Value of r**      1 to 99 (1 step): 1 to 99% (1% step)
- **Suffix code**      None
- **Initial setting**      99%
- **Example**      OBWN\_80

**OBWXDB****OBWXDB      OBW XdB Value**

- **Function**      Sets the conditions of the occupied frequency bandwidth in dB unit.

Header	Program command	Query	Response
OBWXDB	OBWXDB_ℓ	OBWXDB?	ℓ ℓ = 1 to 100 Transfers the data without suffix code as if represented in units of dB

- **Value of ℓ**      1 to 100 (1 step): 1 to 100 dB (1 dB step)
- **Suffix code**      None: dB  
DB: dB
- **Initial setting**      25 dB
- **Example**      OBWXDB\_6DB

## 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** Ø, 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

**PLF****PLF Plotting Paper Form**

- **Function** Specifies the paper size for the plotter

Header	Program command	Query	Response
PLF	PLF_sw	PLF?	PLF_sw

- **Value of sw** Ø:A4  
1:A3
- **Suffix code** None
- **Initial setting** Ø:A4
- **Example** PLF\_1

**PLI****PLI Direct Plot Output Item For Plotter**

- **Function** Selects one of the items (waveform only, scale only, etc) to be plotted directly.

Header	Program command	Query	Response
PLI	PLI_a a=0 to 2	PLI?	PLI_a a=0 to 2

- **Value of a** Ø: ALL  
1: TRACE ONLY  
2: SCALE ONLY
- **Suffix code** None
- **Initial setting** Ø: ALL (Provided the address already allocated is not initialized)
- **Example** PLI\_Ø

## PLOT

### PLOT Direct Plot Start

■ Function Starts direct plotting.

Header	Program command	Query	Response
PLOT	PLOT	—	—

■ Example PLOT

## PLS

### PLS Direct Plot Start

■ Function Starts direct plotting.

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

■ Value of a  $\emptyset$ : PLOT START

■ Suffix code None

■ Example PLS\_ $\emptyset$

**PLTA****PLTA Direct Plot Plotter Address**

- **Function** Sets the GPIB address of the plotter for direct plotting.

Header	Program command	Query	Response
PLTA	PLTA_a a=0 to 30	PLTA?	PLTA_a a=0 to 30

- **Value of a** 0 to 30
- **Suffix code** None
- **Initial setting** Value of a = 18 (provided the GPIB address already allocated is not initialized)
- **Example** PLTA\_0

**PLTARA****PLTARA Plotting Size**

- **Function** Specifies the size of the plotting area.

Header	Program command	Query	Response
PLTARA	PLTARA_sw	PLTARA?	sw

- **Value of sw** FULL : total  
QTR : 1/4 size
- **Suffix code** None
- **Initial setting** FULL : total
- **Example** PLTARA\_QTR

## PLTHOME

### PLTHOME Set Home Position

- **Function** Initializes the printing position to the upper left-corner when the LOCATION selected is AUTO.

Header	Program command	Query	Response
PLTHOME	PLTHOME	—	—

---

## PLTLC

### PLTLC Plot Location

- **Function** Specifies the print location when the X-Y plotter is set to plot on a quarter of the paper area.

Header	Program command	Query	Response
PLTLC	PLTLC_a	PLTLC?	a

- **Value of a**
  - AUTO: Allows the MS2602A to determine the print position automatically and print out to it.
  - UPLEFT: Upper left
  - UPRIGHT: Upper right
  - LOWLEFT: Lower left
  - LOWRIGHT: Lower right

## PMCS

## PMCS

## PMC Select

## ■ Function

Selects one from the PMC built in the MS2602A, 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  
 (provided the address already allocated is not initialized)

## ■ Example

PMCS\_IPMC  
 PMCS\_EFD

## PMOD

## PMOD

## Printer Type

## ■ Function

Selects the type of printer for direct plotting.  
 32 bytes × 32 memories

Header	Program command	Query	Response
PMOD	PMOD_a a=0 to 5	PMOD?	PMOD_a a=0 to 5

## ■ Value of a

0: Printer ... HP-GL  
 1: Printer ... GP-GL  
 2: Printer ... VP-800 (EPSON system)  
 3: Printer ... HP2225 (Hewlett Packard)  
 4: Printer ... MC8104A (Anritsu)  
 5: Printer ... UA455A

## ■ Suffix code

None

## ■ Initial setting

3: Printer ... HP2225

## ■ Example

PMOD\_2  
 PMOD\_4

## PMY

### PMY Dual-Port Memory

- **Function** Writes to the dual port memory or reads from the memory.

Header	Program command	Query	Response
PMY	PMY_a, b a=0 to 31 b=data	PMY?_a, c a=0 to 31 c=1 to 32	b b=data

- **Value of a** Dual port number : 0 to 31
- **Value of b** Data enclosed in double or single quotation marks
- **Value of c** The number of data items read from the dual port memory : 1 to 32
- **Example** PMY\_0, "50"  
PMY?\_0, 1

## PORT

### PORT Control Port Select

- **Function** Specifies the GPIB port for controlling the printer, plotter, MC8104A, etc.

Header	Program command	Query	Response
PORT	PORT_a a=1,2	PORT?	PORT_a a=1,2

- **Value of a** 1: GPIB1  
2: GPIB2
- **Suffix code** None
- **Initial setting** 1: GPIB1 (provided the port already allocated is not initialized)
- **Example** PORT\_1



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

## PRIA

### 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

**PRL****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****PSW Zone Sweep**

- **Function** Sets the zone sweep to ON or OFF.

Header	Program command	Query	Response
PSW	PSW_sw sw=ON, 1, OFF, 0	PSW?	PSW_sw sw=ON, OFF

- **Value of sw** ON: ON  
1: ON  
OFF: OFF  
Ø: OFF
- **Suffix code** None
- **Initial setting** OFF: OFF
- **Example** PSW\_ON

## PTA

### PTA PTA Switch / PTA Status

- **Function** Sets the PTA to ON/OFF.  
Reads whether PTA is BUSY or READY.  
(PTA OFF resets the PTA program.)

Header	Program command	Query	Response
PTA	PTA_a	PTA?	PTA_a a=0,1

- **Value of a**
  - Ø :OFF
  - 1 :ON
  - OFF:OFF
  - ON :ON
- **Suffix code** None
- **Initial setting** Ø: OFF (provided the PTA OFF is not affected by the INI command)
- **Examples**
  - PTA\_Ø
  - PTA? → PTA\_Ø...When PTA is Ready, PTA 0 is transferred.
  - PTA? → PTA\_1...When PTA is Busy, PTA 1 is transferred.

## PTL

### PTL PTL I / O Mode

- **Function** Selects the mode for controlling PTA via GPIB.

Header	Program command	Query	Response
PTL	PTL_a a=0,1	PTL?	text

- **Suffix code** None
- **Initial setting** OFF (provided the mode already allocated is not initialized)
- **Examples**
  - PTL\_Ø : OFF
  - PTL\_1 : Input (A mode to transfer a command or statement to PTA)
  - PTL? : Output (A mode to transfer a statement from PTA to an external device)

## PWRSTART

### PWRSTART Power Measure Start Point

- **Function** Specifies the point at which measurement is started to measure burst power.

Header	Program command	Query	Response
PWRSTART	PWRSTART_p	PWRSTART?	p p = 0 to 500

- **Value of p** 0 to 500
- **Suffix code** None
- **Initial setting** 0 point
- **Example** PWRSTART\_100

## PWRSTOP

### PWRSTOP Power Measure Stop Point

- **Function** Specifies the point at which measurement is ended to measure burst power.

Header	Program command	Query	Response
PWRSTOP	PWRSTOP_p	PWRSTOP?	p p = 0 to 500

- **Value of p** 0 to 500
- **Suffix code** None
- **Initial setting** 0 point
- **Example** PWRSTOP\_400

**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

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 16	—	—

- **Value of a** 1 to 16 ( memory No. )
- **Suffix code** None
- **Example** RC\_1

RCM

**RCM Recall Data from PMC**

- **Function** Recalls measurement conditions ( parameter ) and measured results ( trace ) from PMC or floppy disk.

Header	Program command	Query	Response
RCM	RCM_ a a = 1 to 99	—	—

- **Value of a** 1 to 99 ( File No. )
- **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  
                        P: Parameter Only
- **Suffix code**     None
- **Initial setting** TP: Trace & Parameter  
                        (provided the address already allocated is not initialized)
- **Example**        RDATA\_TP



**RES?****RES?****Measure Result**■ **Function**

Reads out the results which are measured and calculated by the measure functions.

Header	Program command	Query	Response
RES	—	RES?	data 1 data 1, data 2 data 1, data 2, data 3, data 4

■ **Values of data 1, data 2, data 3 and data 4**

Measure Control item (corresponding command)	Corresponding Response	Value of data1	Value of data2	Value of data3	Value of data4
When the measure item or sub item is OFF	OFF	Not transferred	Not transferred	—	—
FREQ COUNT (MEAS__FREQ,ON)	f	As value of f, transfers the data without suffix code as if represented in units of 1 Hz. Resolution: 1 Hz	—	—	—
NOISE MEASURE (MEAS__NOISE,ABS) (MEAS__NOISE,C/N)	ℓ	As value of ℓ, transfers the data without suffix code as if represented in units of 1 dB (dBm/ch, dBm/Hz, dBc/ch, dBc/Hz). Resolution: 0.01 dB	—	—	—
OBW MEASURE (MEAS__OBW,XDB) (MEAS__OBW,N)	f <sub>1</sub> , f <sub>2</sub>	As value (occupied bandwidth) of f <sub>1</sub> , transfers the data without suffix code as if represented in units of 1 Hz. Resolution: 1 Hz	As value (center frequency) of f <sub>2</sub> , transfers the data without suffix code as if represented in units of 1 Hz. Resolution: 1 Hz	—	—

SECTION 8 DETAILED DESCRIPTION OF COMMANDS

Measure Control item (corresponding command)	Corresponding Response	Value of data1	Value of data2	Value of data3	Value of data4
ADJ CH MEASURE (MEAS_ADJ, UNMD) (MEAS_ADJ, MOD)	$\ell_{L1}, \ell_{U1}, \ell_{L2}, \ell_{U2}$	As value ( lower channel of CH SEPA 1 ) of $\ell_{L1}$ , transfers the data without suffix code as if represented in units of 1 dB. Resolution: 0.01 dB	As value ( upper channel of CH SEPA 2) of $\ell_{U1}$ , transfers the data without suffix code as if represented in units of 1 dB. Resolution: 0.01 dB	As value ( lower channel of CH SEPA 2 ) of $\ell_{L2}$ , transfers the data without suffix code as if represented in units of 1 dB. Resolution: 0.01 dB	As value ( upper channel of CH SEPA 2 ) of $\ell_{U2}$ , transfers the data without suffix code as if represented in units of 1 dB. Resolution: 0.01 dB
When the measure item or sub item is OFF	OFF	Not transferred	Not transferred	—	—
TEMPLATE (MEAS_TEMP, CHECK)	$C_1, C_2$	As value ( Limit 1 check result ) of $C_1$ , 0: PASS, 1: FAIL	As value ( Limit 2 check result ) of $C_2$ , 0: PASS, 1: FAIL	—	—
BURST POWER MEASURE (MEAS_POWER, EXE)	$\ell, w$	As value ( dB values ) of $\ell$ , transfers the data without suffix code as if represented in units of 1 dBm. Resolution: 0.01 dBm	As value ( pW values ) of $w$ , transfers the data without suffix code as if represented in units of 1 pW. Resolution: 1 pW	—	—

When the MEASURE function has caused a calculation error or execution error, the issued value is indicated as "\*\*\*\*".

■ Example RES?

**RGDIR****RGDIR**      **Memory Directory**

- **Function**      Displays the directory of recall memory.

Header	Program command	Query	Response
RGDIR	RGDIR	—	—

- **Example**      RGDIR

**RGRC****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 16	—	—

- **Value of a**      1 to 16 (Memory No.)
- **Suffix code**    None
- **Example**        RGRC\_1

## RGSV

### RGSV Save Data into Internal Memory

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

Header	Program command	Query	Response
RGSV	RGSV_a a = 1 to 16	—	—

- **Value of a** 1 to 16 ( Memory No. )
- **Suffix code** None
- **Example** RGSV\_1

RL

**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 $\mu\text{V}$ unit is always selected in LIN mode.

- **Value of ℓ** Value equivalent to  $-100 \text{ dBm}$  to  $+30 \text{ dBm}$  (  $0.1 \text{ dB}$  step )
- **Value of a** UP: LEVEL STEP UP  
DN: LEVEL STEP DOWN
- **Suffix code** None: Available for the current scale unit, provided  $\mu\text{V}$  unit is always selected in LIN mode  
  
DB, DBM, DM: dBm  
DBMV: dBmV  
DBUV: dB $\mu\text{V}$   
DBUVE: dB $\mu\text{V}$  ( emf )  
V: V  
MV: mV  
UV:  $\mu\text{V}$   
W: W  
MW: mW  
UW:  $\mu\text{W}$   
NW: nW  
PW: pW  
FW: fW
- **Initial setting** Value of ℓ =  $-10 \text{ dBm}$
- **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_ $\ell$	RLV?	RLV_ $\ell$ $\ell$ : Transfers the data without suffix code as if represented in the current scale units ( always $\mu$ V unit in LIN mode ).

- **Value of  $\ell$**  Value equivalent to  $-100$  to  $+30$  dBm  
( 0.1 dB step )
- **Suffix code** None: Available for the current scale unit, provided  $\mu$ V unit is always selected in LIN mode
  - DB , DBM , DM: dBm
  - DBMV: dBmV
  - DBUV: dB $\mu$ V
  - DBUVE: dB $\mu$ V ( emf )
  - V: V
  - MV: mV
  - UV:  $\mu$ V
  - W: W
  - MW: mW
  - UW:  $\mu$ W
  - NW: nW
  - PW: pW
  - FW: fW
- **Initial setting** Value of  $\ell = -10$  dBm
- **Example**

```
RLV_-100DBM
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?

**ROFFSET****ROFFSET Ref. Level Offset**

- **Function** Turns the reference level offset ON/OFF and sets the offset value.

Header	Program command	Query	Response
ROFFSET	ROFFSET_sw ROFFSET_ℓ	ROFFSET?	OFF ℓ

- **Value of sw** ON :ON  
OFF :OFF
- **Value of ℓ** -100.0 dB to +100.0 dB (0.1 dB step)
- **Suffix code** None :dB  
DB, DBM, DM: dB
- **Initial setting** ∅: 0 dB
- **Example** ROFFSET\_OFF  
ROFFSET\_2∅DB

## SCL

### 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**
- |                              |                            |
|------------------------------|----------------------------|
| Ø: 1 dB / div ( LOG SCALE )  | 4: 1% / div ( LIN SCALE )  |
| 1: 2 dB / div ( LOG SCALE )  | 5: 2% / div ( LIN SCALE )  |
| 2: 5 dB / div ( LOG SCALE )  | 6: 5% / div ( LIN SCALE )  |
| 3: 10 dB / div ( LOG SCALE ) | 7: 10% / div ( LIN SCALE ) |

- **Suffix code** None

- **Initial setting** 3: 10 dB / div ( LOG SCALE )

- **Example** SCL\_Ø  
SCL\_5

## SCR

### SCR Scroll

- **Function** Scrolls the displayed spectrum to the right or left in the specified scroll step size.

Header	Program command	Query	Response
SCR	SCR_a	—	—

- **Value of a**
- |                     |
|---------------------|
| Ø: SCROLL LEFT      |
| LEFT: SCROLL LEFT   |
| 1: SCROLL RIGHT     |
| RIGHT: SCROLL RIGHT |

- **Suffix code** None

- **Example** SCR\_Ø  
SCR\_RIGHT



**SNGLS****SNGLS**      **Single Sweep Mode**

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

Header	Program command	Query	Response
SNGLS	SNGLS	—	—

- **Example**      SNGLS

**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<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** 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

**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.

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 ( the 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

## SSS

**SSS**                      **Scroll Step Size**

- **Function**                      Sets the scroll step size.

Header	Program command	Query	Response
SSS	SSS_a	SSS?	SSS_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**                2:    2 div
- **Example**                        SSS\_1

## ST

**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

### 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 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 = 0 Hz
- **Example**
  - STF\_123MHZ
  - STF\_45.6KHZ

## SV

### 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 16	—	—

- **Value of a** 1 to 16 (Memory No.)
- **Suffix code** None
- **Example** SV\_1

**SVM****SVM Save Data into PMC**

- **Function** Saves the measurement conditions ( parameter ) and measured results ( trace ) to PMC or floppy disk.

Header	Program command	Query	Response
SVM	SVM_s s=1 to 99	—	—

- **Value of s** 1 to 99 ( File No. )
- **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 MS2602A 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** Ø: Sweep completed  
1: Sweep in progress
- **Example** SWP  
SWP?

## SWSTART

### SWSTART Restart Sweep

■ **Function** Restarts the sweep.

Header	Program command	Query	Response
SWSTART	SWSTART	—	—

■ **Example** SWSTART

## SWSTOP

### SWSTOP Stop Sweep

■ **Function** Stops the sweep.

Header	Program command	Query	Response
SWSTOP	SWSTOP	—	—

■ **Example** SWSTOP



## 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

## 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

### S2 Sweep Mode ( Single )

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

Header	Program command	Query	Response
S2	S2	—	—

- **Example** S2

## 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** 0: 0 sec
- **Example** TDLY\_20MS

**TEMP****TEMP      Select Template**

- **Function**      Selects one of the function templates.

Header	Program command	Query	Response
TEMP	TEMP_n n=1 to 5	TEMP?	n n=1 to 5

- **Value of n**      1 to 5 (Template No.)
- **Suffix code**    None
- **Initial setting** 1
- **Example**        TEMP\_1

**TEMPMCL****TEMPMCL      Cancel Moving Value**

- **Function**      Returns a template movement to 0.

Header	Program command	Query	Response
TEMPMCL	TEMPMCL	—	—

- **Example**        TEMPMCL

## TEMPMSV

### TEMPMSV Save Moved Template Data

- **Function** Stores the moved template data in the original template area.

Header	Program command	Query	Response
TEMPMSV	TEMPMSV	—	—

- **Example** TEMPMSV

## TEMPMVX

### TEMPMVX Template Move X

- **Function** Moves the template line along the X axis.

Header	Program command	Query	Response
TEMPMVX	TEMPMVX_t t = -1000 sec to 1000 sec	TEMPMVX?	t

- **Value of t** -1000 sec to 1000 sec
- **Suffix code**
  - None : msec
  - US :  $\mu$ sec
  - MS : msec
  - S : sec
- **Initial setting** 0:0 sec
- **Example** TEMPMVX\_10MS

## TEMPMVY

### TEMPMVY Template Move Y

- **Function** Moves the template line along the Y axis.

Header	Program command	Query	Response
TEMPMVY	TEMPMVY_ $\ell$	TEMPMVY?	$\ell$

- **Value of  $\ell$**  –200.00 dB to 200.00 dB
- **Suffix code** None : dB  
DB, DBM, DM : dB
- **Initial setting**  $\emptyset$  : 0 dB
- **Example** TEMPMVY\_ $\ell$ -2.5dB

## TEMPSLCT

### TEMPSLCT Template Limit Line Select

- **Function** Selects the Limit Line used for evaluating the measured results by the template functions.

Header	Program command	Query	Response
TEMPSLCT	TEMPSLCT_ $a$ , sw	TEMPSLCT?_ $a$	sw sw = ON, OFF

- **Value of  $a$**  UP1: LIMIT1 UPPER      LW1: LIMIT1 LOWER  
UP2: LIMIT2 UPPER      LW2: LIMIT2 LOWER
- **Value of sw** ON: ON  
1: ON  
OFF: OFF  
 $\emptyset$ : OFF
- **Suffix code** None
- **Initial setting** OFF
- **Example** TEMPSLCT\_UP1, ON

## TEXPAND

### TEXPAND Time Expand

- **Function** Turns ON or OFF the expand functions of trace time.

Header	Program command	Query	Response
TEXPAND	TEXPAND_SW 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

## TIME

### TIME Time

- **Function** Sets the clock built in MS2602A 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_text	TITLE?	text

- **text** Character string within 32 characters enclosed by double quotation marks or single quotation marks
- **Example** TITLE\_ "MS2602A"  
TITLE\_ 'SPECTRUM ANALYZER'

**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_a	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

## TMCNT?

### TMCNT? Time Count Read

- **Function** Reads in the values counted by the integrating meter which integrates the time over which electricity has been turned on.

Header	Program command	Query	Response
TMCNT?	_____	TMCNT?	t t = Transfers the data without suffix code as if represented in units of hr

- **Example** TMCNT?

## 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**
  - Ø: NORMAL
  - 1: MAX HOLD
  - 2: AVERAGE
  - 3: MIN HOLD
  - 4: CUMULATIVE
  - 5: OVER WRITE

- **Suffix code** None

- **Initial setting** Ø: NORMAL

- **Example** TMMD\_Ø



## 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  
Ø: OFF
- **Suffix code** None
- **Initial setting** ON: ON
- **Example** TMWR\_ON

## TRG

**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** Ø: FREERUN  
1: VIDEO  
2: LINE  
3: EXT  
6: TV
- **Suffix code** None
- **Initial setting** Ø: FREERUN
- **Example** TRG\_Ø

## TRGLVL

### TRGLVL Trigger Level

- **Function** Sets the threshold level of the trigger for starting the sweep when trigger source = VIDEO, EXT ( Input1 ) are selected

Header	Program command	Query	Response
TRGLVL	TRGLVL_ $\ell$	TRGLVL?	$\ell$

- **Value of  $\ell$**  -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  $\ell = 0.0$

- **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_sw      sw = FREE, TRGD	TRGS?	sw      sw = FREE, TRGD

- **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_sw	TRGSLP?	SW sw = RISE, FALL

- **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_sw	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

## TRM

### 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

## 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 100000000 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

**TTL****TTL**                      **Title Display Switch**

- **Function**                      Select the title display from among ON / OFF / DATE.

Header	Program command	Query	Response
TTL	TTL_SW    sw=ON, 1, OFF, 0, DATE, 2	TTL?	TTL_SW    sw=ON, OFF, DATE

- **Value of sw**                      ON: ON  
1: ON  
OFF: OFF  
Ø: OFF  
DATE: DATE  
2: DATE
- **Suffix code**                      None
- **Initial setting**                      OFF: OFF (provided the unit already registered is not initialized)
- **Example**                              TTL\_ON

## 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_sw sw = ON, 1, OFF, 0	TZONE?	SW sw = ON, OFF

- **Value of sw** ON: ON  
1: ON  
OFF: OFF  
Ø: 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_t	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            Ø: 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_sw                      sw=ON, 1, OFF, 0	UNC?	UNC_sw                      sw=ON, OFF

- Value of sw            ON: ON  
                          1: ON  
                          OFF: OFF  
                          Ø: OFF
- Suffix code            None
- Initial setting        ON: ON
- Example                UNC\_ON

# 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_a                      a=0 to 5	UNT?	UNT_a                      a=0 to 5

- **Value of a**    0: dBm  
                  1: dB $\mu$ V  
                  2: dBmV  
                  3: V  
                  4: dB $\mu$ V (emf)  
                  5: W
- **Suffix code**    None
- **Initial setting** 0: dBm
- **Example**        UNT\_0

# VAVG

## 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 1024

- **Value of sw**    ON:    ON  
                  1:    ON  
                  OFF: OFF  
                  0:    OFF
- **Value of a**    2 to 1024: Averaging rate to process
- **Suffix code**    None
- **Initial setting** 8:    8 times
- **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  
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 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 A, B, BG VBW / RBW RATIO = 1  
Trace TIME VBW / RBW RATIO = 1
- **Example** VBR\_1

## VBW

### 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****View**

- **Function** Stops writing the waveform data.

Header	Program command	Query	Response
VIEW	VIEW_tr	—	—

- **Value of tr**
  - TRA: Trace A
  - TRB: Trace B
  - TRBG: Trace BG
  - TRTIME: Trace TIME
- **Suffix code** None
- **Example** VIEW\_TRB

**WINDPOS****WINDPOS****Measure Window Position**

- **Function** Specifies the display position of the window for displaying the measure results.

Header	Program command	Query	Response
WINDPOS	WINDPOS_a	WINDPOS?	a

- **Value of a**
  - UPRIGHT : Upper right
  - LOWLEFT : Lower left
  - LOWMID : Lower center
  - LOWRIGHT : Lower right
- **Suffix code** None
- **Initial setting** UPRIGHT : Upper right
- **Example** WINDPOS\_LOWRIGHT

## XMA

### XMA Trace A Spectrum Data

■ **Function** Writes / reads the spectrum data to / from trace A ( main trace ) memory.

Header	Program command	Query	Response
XMA	XMA_p, b	XMA?_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. )

■ **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** 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 Trace B Spectrum Data

■ **Function** Writes / reads the spectrum data to / from trace B ( main trace ) memory.

Header	Program command	Query	Response
XMB	XMB_p, b	XMB?_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. )

■ **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** 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.

**XMG****XMG Trace BG Spectrum Data**

- **Function** Writes / reads the spectrum data to / from trace BG memory

Header	Program command	Query	Response
XMG	XMG_p, b	XMG?_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. )  
 ■ **Value of b** LOG scale used: Integers having 0.01 dBm unit (independent of display unit system)

$$\text{LIN scale used: } b = \frac{\text{Voltage value ( V )}}{\text{Reference level ( V )}} \times 10000$$

- **Value of d** 1 to 1002 ( number of points )  
 ■ **Example** XMG\_1, -2000  
 XMG?\_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 Spectrum Data**

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

Header	Program command	Query	Response
XMT	XMT_p, b	XMT?_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. )  
 ■ **Value of b** LOG scale used: Integers having 0.01 dBm unit ( independent of display unit system)

$$\text{LIN scale used: } b = \frac{\text{Voltage value ( V )}}{\text{Reference level ( V )}} \times 10000$$

- **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.

## \*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,MS2602A,0000,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 a MAV summary message when all pending selected device operations have been completed.

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

■ **Example**                \*OPC?

**\* RST****\* RST          Reset Command**

- **Function**          Resets the device in the third level.

Header	Program command	Query	Response
*RST	*RST	_____	_____

- **Example**          \*RST

**\* SRE****\* 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?****\*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	————	0= Not used
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 MS2602A executes a single sweep ( the 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**  $\emptyset$ : Indicates that the self-test completed without errors.  
 -32767 ~ -1,  
 1 ~ 32767: Indicates that the self-test did not complete, or completed with errors.

- **Example** \*TST?

**\*WAI****\*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

( Blank )

# APPENDIXES

## TABLE OF CONTENTS

APPENDIX A	TABLE OF MS2602A DEVICE-DEPENDENT INITIAL SETTINGS .....	A-1
APPENDIX B	ASCII*CODE TABLE .....	B-1
APPENDIX C	COMPARISON TABLE OF CONTROLLER'S GPIB INSTRUCTIONS .....	C-1

( Blank )



## A TABLE OF MS2602A DEVICE-DEPENDENT INITIAL SETTINGS

### Table of MS2602A Device-Dependent Initial Settings ( 1 / 6 )

Group	Brief function	Control item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
Frequency	Selects the mode for setting a frequency band	FREQUENCY MODE	START-STOP		
	Sets the start frequency	START FREQUENCY	0Hz	-----	0 Hz
	Sets the center frequency	CENTER FREQUENCY	4.25 GHz		4.25 GHz
	Sets the stop frequency	STOP FREQUENCY	8.50GHz	-----	8.50 GHz
	Sets the frequency span	FREQUENCY SPAN	8.50GHz	* 0 Hz	8.50 GHz
	Sets the center-frequency step size	CENTER FREQ STEP SIZE	1 GHz		
	Sets the scroll step size	SCROLL STEP SIZE	2 div		
	Selects the band	BAND SELECT	AUTO ( 0 to 8.5 GHz )		
Level	Sets the reference-level	REFERENCE LEVEL	-10 dBm		
	Sets the reference-level step size	REF LEVEL STEP SIZE	AUTO : 1div		
	Sets the scale mode	SCALE MODE	LOG	LOG	* LOG
	Sets the LOG scale	LOG SCALE	10 dB / div	10 dB / div	* 10 dB / div
	Sets the LIN scale	LIN SCALE	10% / div	10% / div	-----
	Sets the LOG unit system	LOG SCALE UNIT	Not initialized * RST: dBm		
	Sets the reference level offset	REF LEVEL OFFSET	OFF		
	Sets the reference level offset value	OFFSET VALUE	0 dB		
	Sets the display line	DISPLAY LINE	OFF		
	Sets the display line level	DISPLAY LINE LEVEL	-50 dBm		
	Selects the marker level ABS / REL	MARKER LEVEL ABS / REL	A : ABS B : ABS	ABS	ABS
	Sets the correction factor	CORRECTION	Not initialized * RST: OFF		
Sets the correction factor number	CORRECTION FACTOR No.	* RST: 1			

Table of MS2602A Device-Dependent Initial Settings ( 2 / 6 )

Group	Brief function	Control item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
Display mode	Selects display mode	DISPLAY MODE	TRACE-A		
	Selects the display format for TRACE-A / B	DISPLAY FORMAT (TRACE-A / B)	A & B	-----	-----
	Selects the display format for TRACE-A / BG	DISPLAY FORMAT (TRACE-A / BG)	SUB TRACE	-----	MAIN TRACE
	Selects the display format for TRACE-A / TIME	DISPLAY FORMAT (TRACE-A / TIME)	SUB TRACE	MAIN TRACE	-----
	Selects the mode for processing a waveform	TRACE STORAGE MODE	NORMAL	NORMAL	*NORMAL
	Number of traces averaged	AVERAGE No.	8 times		
	Selects the detection mode	DETECTION MODE	PEAK	SAMPLE	* PEAK
	Sets the delay time	DELAY TIME	-----	0 sec	-----
	Sets the time span	TIME SPAN	-----	# 200 msec	-----
	Sets the time expand zone to ON / OFF	EXPAND ZONE ON / OFF	-----	OFF	-----
	Sets expand mode to ON / OFF	EXPAND ON / OFF	-----	OFF	-----
	Sets the waveform monitor to ON / OFF	FM / TRIG MONITOR	-----	OFF	-----
	Sets the bandwidth for demodulating FM	FM RANGE	-----	200 kHz /div	-----
	Switches the coupling to AC/DC to monitor FM waveforms	FM COUPLING	-----	AC COUPLING	-----
	Sets the active marker when display mode is trace A / B	TRACE-A / B ACTIVE MKR	TRACE-A	-----	-----
	Selects the marker mode	MARKER MODE	NORMAL		
	Specifies the zone-marker center	ZONE MARKER CENTER	250 point	250 point	250 point
	Specifies the zone-marker width	ZONE MARKER WIDTH	51 point	* 1 point	501 point
	Sets the multimarker mode to ON / OFF	MULTI MARKER MODE	OFF		
	Sets the multimarker list to ON / OFF	MULTI MARKER LIST	OFF		
Sets the ' n ' th multimarker to ON / OFF	MULTI MARKER ON / OFF	Not initialized * RST: No.1 = ON, NO.2 to 10 = OFF			
Selects the active marker from the multimarkers	ACTIVE MARKER No.	Not initialized * RST: No.1			

Table of MS2602A Device-Dependent Initial Settings ( 3 / 6 )

Group	Brief function	Control item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
Sweep function	Sets the sweep mode	SWEEP MODE	CONTINUOUS		
	Sets the zone sweep to ON /OFF	ZONE SWEEP	OFF	-----	
	Sets the tracking function to ON / OFF	TRACKING SWEEP	OFF	-----	
	Sets the gate sweep function to ON / OFF	GATE SWEEP	OFF		-----
	Sets the gate delay time	GATE DELAY	0 sec		-----
	Sets the gate length	GATE LENGTH	1 msec		-----
	Sets the ending gate interval, either internally or externally	GATE END	INTERNAL		-----
	Sets the trigger switch mode	TRIGGER SWITCH	FREE RUN	FREE RUN	* FREE RUN
	Sets the trigger source	TRIGGER SOURCE	VIDEO		-----
	Sets the external trigger level type	TRIGGER SOURCE (EXT) type	INPUT1		-----
	Selects the TV system	TRIGGER SOURCE (TV)	NTSC		-----
	Selects TV horizontal synchronous signal	TRIG SOURCE(TV SYNC)	V-SYNC		-----
	Selects the number of TV horizontal synchronous signal lines	TV H-SYNC LINE No.	ODD 10		-----
	Selects the trigger slope	TRIGGER SLOPE	RISE		-----
Sets the trigger level	TRIGGER LEVEL	25%		-----	
Writing / reading waveform	Sets the trace write switch to ON / OFF	TRACE WRITE SWITCH	ON	ON	ON
	Sets the trace read switch to ON / OFF	TRACE READ SWITCH	ON	ON	ON
Coupled function	Selects the mode for setting the resolution bandwidth	RESOLUTION BANDWIDTH	AUTO	AUTO	* AUTO
	Selects the mode for setting the video bandwidth	VIDEO BANDWIDTH	AUTO	AUTO	* AUTO
	Selects the mode for setting the sweep time	SWEEP TIME	AUTO	AUTO	* AUTO
	Selects the mode for setting the RF attenuator	RF ATTENUATOR	AUTO		
	VBW/RBW ratio at VBW = AUTO	VBW/RBW RATIO	1	1	1

Table of MS2602A Device-Dependent Initial Settings ( 4 / 6 )

Group	Brief function	Control item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
SAVE / RECALL	Select data to be recalled	RECALLED DATA	Not initialized. When shipped from the factory: TRACE & PARAMETER		
Hard copy/plot	Select the printer device mode	PRINTER MODE	Not initialized. When shipped from the factory: HP2225		
	Sets the printer GPIB address	PRINTER GPIB ADDRESS	Not initialized. When shipped from the factory: 17		
	Selects the paper size for plotter	PLOTTER PAPER SIZE	Not initialized. When shipped from the factory: A4		
	Selects the output size from plotter	PLOTTER SIZE	Not initialized. When shipped from the factory: FULL		
	Selects the plot item	PLOT ITEM	Not initialized. When shipped from the factory: ALL		
Sound monitor	Selects the mode for monitoring the sound	AM / FM MONITOR	OFF		
	Adjusts the volume of the sound monitor	MONITOR VOLUME	10		
Measure function	Selects the item to be measured	MEASURE ITEM	OFF		
	Sets the counter to the specified resolution	COUNT RESOLUTION	1 kHz		
	Sets the occupied frequency bandwidth to N%	OBW N% VALUE	Not initialized *RST: 99%		
	Sets the occupied frequency to X dB	OBW XdB VALUE	Not initialized *RST: 25 dB		
	Selects the adjacent channel	ADJACENT CH SELECT	Not initialized *RST: BOTH SIDES		
	Sets the adjacent separation 1	ADJACENT CH SEPARATION 1	Not initialized *RST: 12.5 kHz		
	Sets the adjacent separation 2	ADJACENT CH SEPARATION 2	Not initialized *RST: 25.0 kHz		
	Sets the adjacent channel bandwidth	ADJACENT CH BANDWIDTH	Not initialized *RST: 8.5 kHz		
	Selects the template	SELECT TEMPLATE	Not initialized *RST: No.1		
	Selects the template level	TEMPLATE LEVEL	Not initialized *RST: ABSOLUTE		
	Sets the template management function	MANAGE TEMPLATE	Not initialized. Initialized to OFF at power-on.		
	Selects the noise measurement method	NOISE MEASURE METHOD	Not initialized. *RST: ABS		

Table of MS2602A Device-Dependent Initial Settings ( 5 / 6 )

Group	Brief function	Control item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
Measure function	Selects the occupied frequency bandwidth measurement method	OBW MEASURE METHOD	Not initialized. *RST: N %		
	Selects the adjacent channel leakage power measurement method	ADJ-CH MEASURE METHOD	*RST: R: TOTAL POWER		
	Sets the adjacent channel leakage power graph display	ADJ-CH GRAPH	*RST: ON		
	Sets the adjacent channel center line display	ADJ-CH CENTER LINE	*RST: ON		
	Sets the adjacent channel band line display	ADJ-CH BAND LINE	*RST: OFF		
	Sets the window display position for measured results	MEASURE WINDOW POSITION	UPRIGHT		
	Sets the point at which burst power measurement starts	BURST POWER MEASURE START POINT	0 point		
	Sets the point at which burst power measurement ends	BURST POWER MEASURE STOP POINT	0 point		
System setting	Sets the number of data points	DATA POINT NUMBER	Not initialized. When shipped from the factory: 501 point		
	Sets the coupled functions to COMMON or INDEPENDENT between frequency or time domain	COUPLE MODE (COMMON / INDEPENDENT)	Not initialized. When shipped from the factory: COMMON		
Calibration	Sets the preselector peaking bias	PRESELECTOR BIAS	Not initialized. When shipped from the factory: 0		
GPIB	Sets the GPIB 2 self address	GPIB2 SELF ADDRESS	Not initialized. When shipped from the factory: 16		
	Sets the DSU (MC8104A) address	DATA STORAGE UNIT ADDRESS	Not initialized. When shipped from the factory: 19		
Title	Sets the title output to ON / OFF	TITLE ON / OFF	Not initialized. When shipped from the factory: ON		
	Selects the title data	TITLE DATA	Not initialized. When shipped from the factory: ALL SPACE		
CAL / UNCAL	Displays couple failure	UNCAL DISPLAY	Not initialized. Initialized to ON at power-on.		

Table of MS2602A Device-Dependent Initial Settings ( 6 / 6 )

Group	Brief function	Control item	Initial setting data		
			TRACE-A,B	TRACE-TIME	TRACE-BG
Spectrum data/PMC / ETC	Selects the response data for ASCII / BINARY	RESPONSE DATA	Not initialized. When shipped from the factory: ASCII		
	Selects the media for PMC / floppy disk	MEDIA SELECT	Not initialized. When shipped from the factory: INT PMC		
	Selects the terminator for LF / CR + LF	TERMINATOR	Not initialized. When shipped from the factory: LF		
	Sets the buzzer switch to ON / OFF	BUZZER SWITCH	Not initialized. Initialized to ON at power-on.		

**Notes:**

- In place of the parameters not initialized by the INI command (GPIB) or Preset key, the initial settings (Indicated by \*RST) initialized by the \*RST command (GPIB) are listed in the above table. In place of the parameters not initialized by the \*RST command, the values when shipped from the factory are listed.

- An initial value marked with ' \* ' represents a fixed value.
- An initial value marked with ' # ' represents the value at COUPLE MODE = COMMON.



## B ASCII\*CODE TABLE

B7 B6 B5 BITS B4 B3 B2 B1	0 0		0 1		1 0		1 1		1 0		1 1					
	CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE			
0 0 0 0	0	NUL	20	DLE	40	SP	60	0	100	@	120	P	140	,	160	p
0 0 0 1	1	SOH	21	DC1	41	!	61	1	101	A	121	Q	141	a	161	q
0 0 1 0	2	STX	22	DC2	42	"	62	2	102	B	122	R	142	b	162	r
0 0 1 1	3	ETX	23	DC3	43	#	63	3	103	C	123	S	143	c	163	s
0 1 0 0	4	EOT	24	DC4	44	\$	64	4	104	D	124	T	144	d	164	t
0 1 0 1	5	ENO	25	NAK	45	%	65	5	105	E	125	U	145	e	165	u
0 1 1 0	6	ACK	26	SYN	46	&	66	6	106	F	126	V	146	f	166	v
0 1 1 1	7	BEL	27	ETB	47	'	67	7	107	G	127	W	147	g	167	w
1 0 0 0	8	BS	28	CAN	48	(	68	8	108	H	128	X	148	h	168	x
1 0 0 1	9	HT	29	EM	49	)	69	9	109	I	129	Y	149	i	169	y
1 0 1 0	10	LF	30	SUB	50	*	70	:	110	J	130	Z	150	j	170	z
1 0 1 1	11	VT	31	ESC	51	÷	71	;	111	K	131	[	151	k	171	{
1 1 0 0	12	FF	32	FS	52	,	72	<	112	L	132	\	152	l	172	
1 1 0 1	13	CR	33	GS	53	-	73	=	113	M	133	]	153	m	173	}
1 1 1 0	14	SO	34	RS	54	.	74	>	114	N	134	^	154	n	174	~
1 1 1 1	15	SI	35	US	55	/	75	?	115	O	135	_	155	o	175	rubout (DEL)
		Address command		Universal command		Listen address				Talk address					Secondary address or command	

**KEY**

octal	25	PPU
		ASCII character
hex	15	21

GPIB code  
decimal

\* USA Standard Code for Information Interchange





**Notes:**

- ① MSG = INTERFACE MESSAGE (Sent by ATN of True: Low level.)
- ② b<sub>1</sub> = DI01 ..... b<sub>7</sub> = DI07 (b<sub>1</sub> through b<sub>7</sub> correspond to DI01 to DI07 sequence.)

**Table of Address Assignments**

Address character		Address switch setting						Primary address	Factory address set device
Talk	Listen	5	4	3	2	1			
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>	10 Decimal		
1 0	0 1	↓	↓	↓	↓	↓			
@	SP	0	0	0	0	0	0	0	
A	!	0	0	0	0	1	1	1	
B	"	0	0	0	1	0	0	2	
C	#	0	0	0	1	1	1	3	
D	\$	0	0	1	0	0	0	4	
E	%	0	0	1	0	1	1	5	
F	&	0	0	1	1	0	0	6	
G	'	0	0	1	1	1	1	7	
H	(	0	1	0	0	0	0	8	
I	)	0	1	0	0	1	1	9	
J	*	0	1	0	1	0	0	10	
K	+	0	1	1	0	1	1	11	
L	,	0	1	1	1	0	0	12	
M	-	0	1	1	1	1	1	13	
N	.	0	1	1	1	0	0	14	
O	/	0	1	1	1	1	1	15	
P	0	1	0	0	0	0	0	16	
Q	1	1	0	0	0	1	1	17	
R	2	1	0	0	1	0	0	18	
S	3	1	0	0	1	1	1	19	
T	4	1	0	1	0	0	0	20	
U	5	1	0	1	0	1	1	21	
V	6	1	0	1	1	0	0	22	
W	7	1	0	1	1	1	1	23	
X	8	1	1	0	0	0	0	24	
Y	9	1	1	0	0	1	1	25	
Z	:	1	1	0	1	0	0	26	
[	;	1	1	0	1	1	1	27	
\	<	1	1	1	0	0	0	28	
]	=	1	1	1	1	0	1	29	
^	>	1	1	1	1	1	0	30	
?	—	1	1	1	1	1	1	31	

**Table of Interface Message Group**

D	D	D	D	D	D	D	D	D	D	Interface message group (G)
0	0	0	0	0	0	0	0	0	0	Addressed command G
0	0	1	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>1</sub>	b <sub>1</sub>	Universal command G
0	1	0	1	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>1</sub>	Listen address G
0	1	1	1	1	1	1	1	1	1	Unlisten (UNL)
1	0	0	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>1</sub>	b <sub>1</sub>	Talker address G
1	0	1	1	1	1	1	1	1	1	Untalk (UNF)
1	1	1	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>1</sub>	b <sub>1</sub>	Secondary command G

- GTL Go to Local
- SDC Select Device Clear
- PPC Parallel Poll Configure
- GET Group Execute Trigger
- TCT Take Control
- LL0 Local Lockout
- (ACG) Addressed Command Group
- (UCG) Universal Command Group
- (LAG) Listen Address Group
- (TAG) Talk Address Group
- (PCG) Primary Command Group
- (SGG) Secondary Command Group
- DCL Device Clear
- PPU Parallel Poll Unconfigure
- SPE Serial Poll Enable
- SPD Serial Poll Disable
- UNL Unlisten
- UNT Untalk

(Blank)

## C COMPARISON TABLE OF CONTROLLERS' GPIB INSTRUCTIONS

Function	Controller			
	PACKET V	PC9800	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	PC9800	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	RCT @ device number	WBYTE,talker address,&H09;	CALL IBRSV ( )	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( )	