

MG3670B

**Digital Modulation
Signal Generator**

**Operation Manual
(GPIB Remote Control)**

(Blank)

WARNING

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

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

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

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

■ SAFETY CONSIDERATIONS:

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

Labels used in this manual:

WARNING

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

CAUTION

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

Labels or symbols used on / in the product:



: This symbol indicates hazardous voltages. Be careful (not used in this instrument).



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



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

CERTIFICATION

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

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

WARRANTY

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

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

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

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

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

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

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

MEMORY BACK-UP BATTERY REPLACEMENT

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

STORAGE MEDIUM

This equipment stores data and programs using backed - up memories . Data and programs may be lost due to improper use or failure.

ANRITSU CANNOT COMPENSATE FOR ANY MEMORY LOSS.

Please pay careful attention to the following points.

(Backed-up memory)

- Isolate the memory from static electricity.

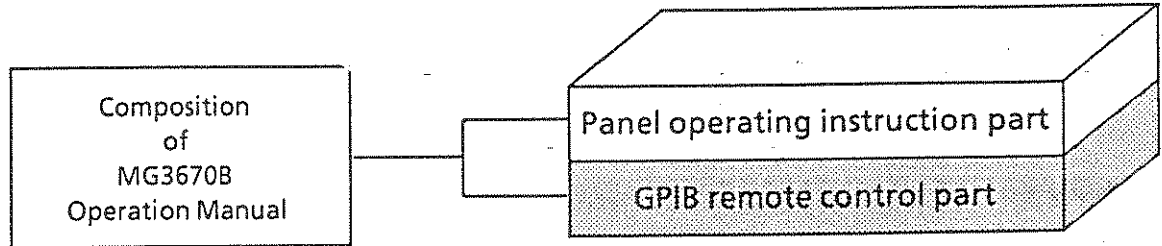
Note: The battery life is about 7 years. Early battery replacement is recommended.

(Blank)

ABOUT THIS MANUAL

(1) Composition of MG3670B Operation Manuals

The MG3670B Digital Modulation Signal Generator operation manual of the standard type is composed of the following two parts. Use them properly according to the usage purpose.



Panel operating instruction part:

Panel Operating Instructions: Provides information on the MG3670B outline, preparation before use, panel description, basic operation, performance tests, calibration, storage / transportation, and quick reference for screen transition and error messages.

GPIB remote control part:

Provides information on the MG3670B remote control which conforms with IEEE488.2 standards. To assist creating GPIB programs, this manual gives examples of N₈₈ BASIC language programs that run on the NEC PC9800 series of personal computers.

The operation manuals of the MG0301C ($\pi/4$ DQPSK Modulation Unit) and the MG0302A (GMSK Modulation Unit) are prepared and sold separately. The operations of the MG0303B (Burst Function Unit) are explained in each the MG0301C/MG0302A operation manual description related to the burst function.

(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

SECTION 1	GENERAL	1-1
	1.1 General	1-3
	1.2 Functions of GPIB	1-3
	1.3 Examples of System Configuration Using GPIB	1-4
	1.4 Specifications	1-5
SECTION 2	DEVICE MESSAGE LIST	2-1
	2.1 IEEE 488.2 Common Commands and MG3670B Supported	2-3
	2.2 MG3670B Device Message List	2-5
	2.2.1 How to read the MG3670B device message tables	2-5
	2.2.2 MG3670B device message list	2-6
	2.2.3 Device message list compatible with the MG3633A	2-13
SECTION 3	CONNECTING BUS AND SETTING ADDRESS	3-1
	3.1 Connecting Devices with GPIB Cables	3-3
	3.2 Checking and Setting GPIB address	3-4
SECTION 4	INITIAL SETTINGS	4-1
	4.1 Bus Initialization by the IFC Statement	4-4
	4.2 Initialization for Message Exchange by DCL and SDC Bus Commands	4-5
	4.3 Device Initialization by the *RST Command	4-6
	4.4 Device Initialization by the PRE Command	4-9
	4.5 Device Status at Power-on	4-9
SECTION 5	DEVICE MESSAGE FORMAT	5-1
	5.1 General Description	5-3
	5.2 Program Message Format	5-3
	5.3 Response Message Format	5-9
SECTION 6	COMMON COMMANDS	6-1
	6.1 Classification by Function of Common Commands Supported by the MG3670B	6-3
	6.2 Common Commands List	6-4

SECTION 7	STATUS STRUCTURE	7-1
7.1	IEEE488.2 Standard Status Model	7-3
7.2	Status Byte (STB) Register	7-5
7.2.1	ESB and MAV summary messages	7-5
7.2.2	Device-dependent summary messages	7-6
7.2.3	Reading and clearing STB register	7-7
7.3	Service Request (SRQ) Enabling Operation	7-8
7.4	Standard Event Status Register	7-9
7.4.1	Bit definition of Standard Event Status Register	7-9
7.4.2	Reading, writing to, and clearing Standard Event Status Register	7-10
7.4.3	Reading, writing to, and clearing Standard Event Status Enable Register	7-10
7.5	Extended Event Status Register	7-11
7.5.1	Bit definition of Extended Event Status Register	7-12
7.5.2	Bit definition of Extended END Event Status Register	7-13
7.5.3	Reading, writing to, and clearing Extended Event Status Register	7-14
7.5.4	Reading, writing to, and clearing Extended Event Status Enable Register	7-14
7.6	Techniques for Synchronizing MG3670B with a Controller	7-15
7.6.1	Wait for response after *OPC? query is sent	7-15
7.6.2	Wait for service request after *OPC is sent	7-16
SECTION 8	DETAILS OF MG3670B DEVICE MESSAGES	8-1
8.1	Device Messages Specific to the MG3670B	8-4
8.2	Control by MG3633A Device Messages	8-45
8.2.1	Frequency	8-45
8.2.2	Output level	8-56
8.2.3	Save / recall	8-75
8.2.4	Others	8-80
SECTION 9	SAMPLE PROGRAMS	9-1
9.1	Precautions on Creating the GPIB Program	9-3
9.2	Sample Programs	9-4
9.2.1	Initializing MG3670B	9-4
9.2.2	Setting frequency and output level	9-5
9.2.3	Setting frequency using increment step frequency	9-6
9.2.4	Calibrating by external I, Q signal input	9-7

APPENDIX A ASCII CODE TABLE A-1
APPENDIX B COMPARISON TABLE OF CONTROLLER'S GPIB INSTRUCTIONS B-1

SECTION 1 GENERAL

This section outlines the GPIB functions of the MG3670B Digital Modulation Signal Generator.

TABLE OF CONTENTS

1.1	General	1-3
1.2	Functions of GPIB	1-3
1.3	Examples of System Configuration Using GPIB	1-4
1.4	Specifications	1-5

(Blank)

SECTION 1 GENERAL

1.1 General

The MG3670B Digital Modulation Signal Generator, when combined with an external controller, can automate your measurement system. For this purpose the MG3670B is equipped with a GPIB interface bus (IEEE std 488.2 1987) as a standard feature.

1.2 Functions of GPIB

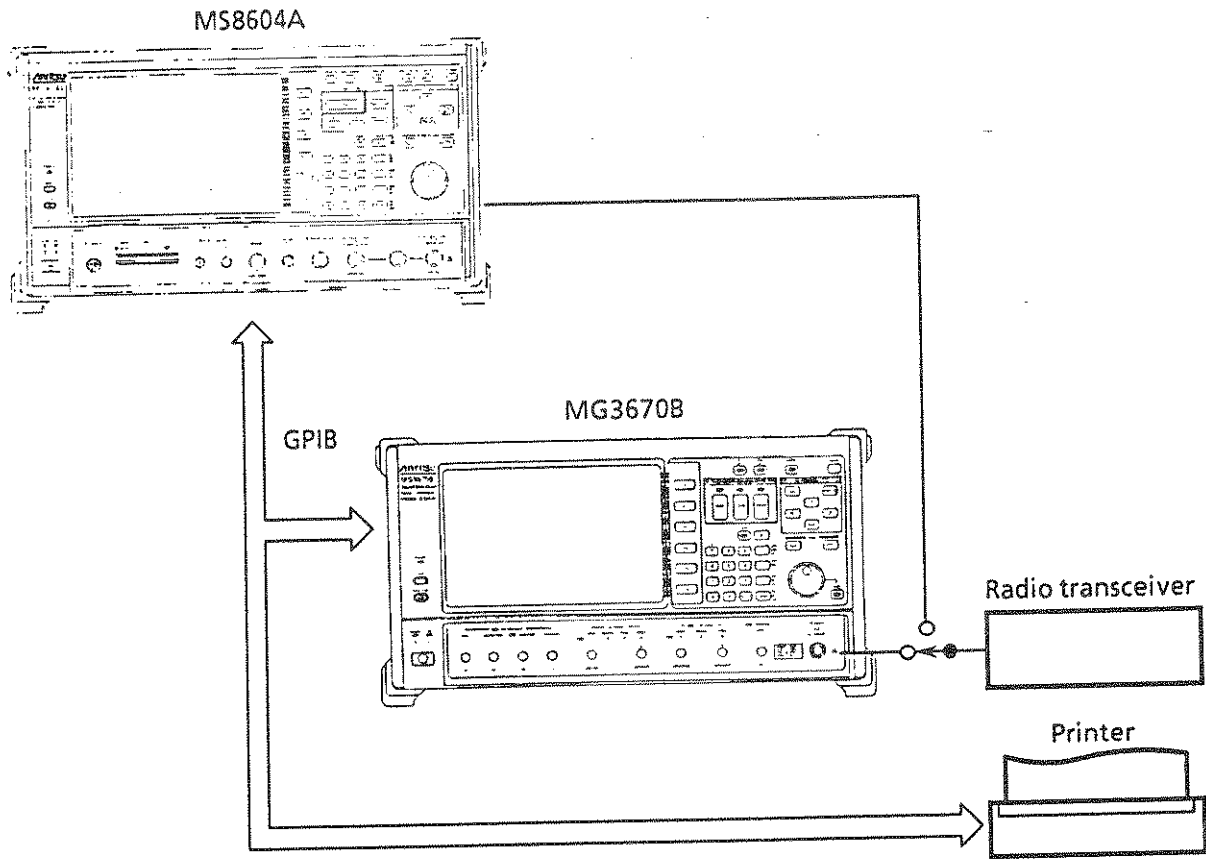
The functions of the MG3670B GPIB are as follows:

- (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 MG3670B is combined with a personal computer and other measuring instruments

1.3 Examples of System Configuration Using GPIB

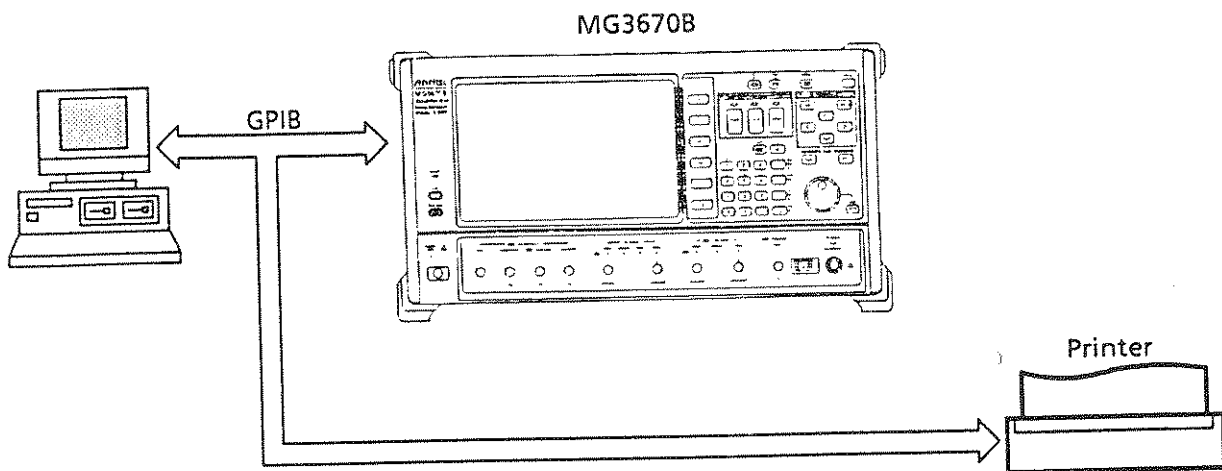
(1) PTA control by the MS8604A

The waveforms and parameters etc. measured by controlling the MG3670B with the MS8604A via GPIB can be output to the printer.



(2) Control by the host computer

The parameters etc. can be output to the printer by controlling the MG3670B with the host computer via GPIB.



1.4 Specifications

The MG3670B GPIB provides 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.
DT0	Device trigger functions are not provided.
C0	Controller functions are not provided.

(Blank)

SECTION 2 DEVICE MESSAGE LIST

This section lists the device messages of the MG3670B.

TABLE OF CONTENTS

2.1	IEEE 488.2 Common Commands and MG3670B Supported Commands	2-3
2.2	MG3670B Device Message List	2-5
2.2.1	How to read the MG3670B device message tables	2-5
2.2.2	MG3670B device message list	2-6
2.2.3	Device message list compatible with the MG3633A	2-13

TELEPHONE DIVISION

...

...

...

...

...

...

...

(Blank)

2.1 IEEE 488.2 Common Commands and MG3670B Supported Commands

The table below lists 39 types of common commands specified in the IEEE 488.2 standard. IEEE 488.2 common commands which are supported by the MG3670B are indicated with © symbol in the table.

Mnemonic	Command name	IEEE488.2 Standard	MG3670B supported commands
*AAD	Accept Address Command	Optional	
*CAL?	Calibration Query	Optional	
*CLS	Clear Status Command	Mandatory	©
*DDT	Define Device Trigger Command	Optional	
*DDT?	Define Device Trigger Query	Optional	
*DLF	Disable Listener Function Command	Optional	
*DMC	Define Macro Command	Optional	
*EMC	Enable Macro Command	Optional	
*EMC?	Enable Macro Query	Optional	
*ESE	Standard Event Status Enable Command	Mandatory	©
*ESE?	Standard Event Status Enable Query	Mandatory	©
*ESR?	Standard Event Status Register Query	Mandatory	©
*GMC?	Get Macro Contents Query	Optional	
*IDN?	Identification Query	Mandatory	©
*IST?	Individual Status Query	Optional	
*LMC?	Learn Macro Query	Optional	
*LRN?	Learn Device Setup Query	Optional	
*OPC	Operation Complete Command	Mandatory	©
*OPC?	Operation Complete Query	Mandatory	©
*OPT?	Option Identification Query	Optional	
*PCB	Pass Control Back Command	Mandatory if other than C0	
*PMC	Purge Macro Command	Optional	
*PRE	Parallel Poll Register Enable Command	Optional	
*PRE?	Parallel Poll Register Enable Query	Optional	
*PSC	Power On Status Clear Command	Optional	
*PSC?	Power On Status Clear Query	Optional	
*PUD	Protected User Data Command	Optional	
*PUD?	Protected User Data Query	Optional	
*RCL	Recall Command	Optional	
*RDT	Resource Description Transfer Command	Optional	
*RDT?	Resource Description Transfer Query	Optional	

SECTION 2 DEVICE MESSAGE LIST

Mnemonic	Command name	IEEE488.2 Standard	MG3670B supported commands
*RST	Reset Command	Mandatory	⊙
*SAV	Save Command	Optional	
*SRE	Service Request Enable Command	Mandatory	⊙
*SRE?	Service Request Enable Query	Mandatory	⊙
*STB?	Read Status Byte Query	Mandatory	⊙
*TRG	Trigger Command	Mandatory if DT1	
*TST?	Self Test Query	Mandatory	⊙
*WAI	Wait to Continue Command	Mandatory	⊙

Note: The first character of a common command or query starts with the asterisk symbol().*

2.2 MG3670B Device Message List

The tables on the following pages list the command messages, query messages, and response messages specific to the MG3670B.

2.2.1 How to read the MG3670B device message tables

(1) Command Messages (Command Msg)/Query Message (Query Msg)

(a) Upper-case:	Reserved word
(b) Number:	Reserved word (Number code)
(c) Lower-case argument	
f (Frequency):	Numeric data (NR1,NR2,NR3 format)
Suffix code:	GHZ, GZ, MHZ, MZ, KHZ, KZ, HZ, None = HZ
ℓ_1 (Level):	Numeric data (NR1,NR2,NR3 format)
Suffix code:	DBM, DM, DBU, DU, V, MV, UV, None = DBM
ℓ_2 (Level):	Numeric data (NR1,NR2,NR3 format)
Suffix code:	DB None = DB
ℓ_3 (Level):	Numeric data (NR1,NR2,NR3 format)
Suffix code:	V, MV, UV None = MV
n (Integral number):	Numeric data (NR1 format)
h (hexadecimal number):	Numeric data (hexadecimal data)
s:	Character string enclosed by double quotation marks (" ") or single quotation marks (' ')

(2) Response message(Response Msg)

(a) Upper-case:	Reserved word
(b) Number:	Reserved word (Number code)
(c) Lower-case argument	
f (Frequency):	Numeric data (NR1 format)
Suffix code:	HZ
ℓ_1 (Level):	Numeric data (NR2 format)
Suffix code:	DBM, DBU, V, MV, UV (Transfers the unit which is represented in the current output level units)
ℓ_2 (Level):	Numeric data (NR2 format)
Suffix code:	DB
ℓ_3 (Level):	Numeric data (NR2 format)
Suffix code:	MV
n (Integral number):	Numeric data (NR1 format)
h (hexadecimal number):	Numeric data (hexadecimal data)

Note: ● In the device message list compatible with the MG3633A, the netted messages, which overlap with the MG3670B, can be used for MG3670B.

- If the header is set to Off, then the response message header and the numeric data suffix code are not transferred. (Initial setting: Header Off)
- means a space of one character.
- \emptyset indicates a zero.

2.2.2 MG3670B device message list

(1) Frequency

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Sets frequency	Frequency	FREQ_ f	FREQ?	FREQ_ f
Sets frequency using incremental step	Frequency Incremental Step Value	FIS_ f	FIS?	FIS_ f
	Up	FRS_ UP	_____	_____
	Down	FRS_ DN	_____	_____
Sets frequency using rotary knob	Frequency Knob			
	Up Down	FRK_ UP FRK_ DN	_____ _____	_____ _____
Sets frequency resolution	Frequency Resolution			
	1 Hz	FRR_ 1HZ	FRR?	FRR_ 1HZ
	10 Hz	FRR_ 10HZ	FRR?	FRR_ 10HZ
	100 Hz	FRR_ 100HZ	FRR?	FRR_ 100HZ
	1 kHz	FRR_ 1KHZ	FRR?	FRR_ 1KHZ
	10 kHz	FRR_ 10KHZ	FRR?	FRR_ 10KHZ
	100 kHz	FRR_ 100KHZ	FRR?	FRR_ 100KHZ
	1 MHz	FRR_ 1MHZ	FRR?	FRR_ 1MHZ
	10 MHz	FRR_ 10MHZ	FRR?	FRR_ 10MHZ
	100 MHz	FRR_ 100MHZ	FRR?	FRR_ 100MHZ
	1 GHz	FRR_ 1GHZ	FRR?	FRR_ 1GHZ
Right	FRR_ R	FRR?	FRR_ 1HZ~1GHZ	
Left	FRR_ L	FRR?	FRR_ 1HZ~1GHZ	
Turns relative frequency mode on or off	Frequency Relative Reference Value	_____	FRLR?	FRLR_ f
	Value	_____	FRLV?	FRLV_ f
	On	FRL_ ON	FRL?	FRL_ ON
	Off	FRL_ OFF	FRL?	FRL_ OFF
Sets frequency offset	Frequency Offset			
	Value	FOS_ f	FOS?	FOS_ f
	On	FOF_ ON	FOF?	FOF_ ON
Off	FOF_ OFF	FOF?	FOF_ OFF	

(2) Output level

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Sets output level	Output Level	OLVL $_l_1$	OLVL?	OLVL $_l_1$
Sets output level using incremental Step	Output Level Incremental Step			
	Value	OIS $_l_2$	OIS?	OIS $_l_2$
	Up	OLS $_UP$	_____	_____
	Down	OLS $_DN$	_____	_____
Sets output level using rotary knob	Output Level Knob			
	Up	OLK $_UP$	_____	_____
	Down	OLK $_DN$	_____	_____
Sets output level resolution	Output Level Resolution			
	0.1 dB	OLR $_0.1DB$	OLR?	OLR $_0.1DB$
	1 dB	OLR $_1DB$	OLR?	OLR $_1DB$
	10 dB	OLR $_10DB$	OLR?	OLR $_10DB$
	Right	OLR $_R$	OLR?	OLR $_0.1DB\sim10DB$
	Left	OLR $_L$	OLR?	OLR $_0.1DB\sim10DB$
Turns output level Continuous Mode on or off	Output Level Continuous Mode			
	On	OCNT $_ON$	OCNT?	OCNT $_ON$
	Off	OCNT $_OFF$	OCNT?	OCNT $_OFF$
Turns relative output Level mode on or off	Output Level Relative Reference Value	_____	ORLR?	ORLR $_l_1$
	Value	_____	ORLV?	ORLV $_l_2$
	On	ORL $_ON$	ORL?	ORL $_ON$
	Off	ORL $_OFF$	ORL?	ORL $_OFF$
Sets output level offset	Output Level Offset			
	Value	OOS $_l_2$	OOS?	OOS $_l_2$
	On	OOF $_ON$	OOF?	OOF $_ON$
	Off	OOF $_OFF$	OOF?	OOF $_OFF$
Turns output level on or off	Output Level			
	On	LVL $_ON$	LVL?	LVL $_ON$
	Off	LVL $_OFF$	LVL?	LVL $_OFF$
Selects voltage display	Voltage Display			
	Electro Motive Force (EMF)	VDSPL $_EMF$	VDSPL?	VDSPL $_EMF$
	Terminated (TERM)	VDSPL $_TERM$	VDSPL?	VDSPL $_TERM$
Changes output level unit	Output Level Unit			
	Change to dBm	OLDBM	_____	_____
	Change to dB μ	OLDBU	_____	_____
	Change to V	OLV	_____	_____
Calibrates output level	Calibration	CAL	_____	_____

(3) Modulation

(Main instrument)

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Turns modulation on or off	Modulation			
	On	MOD _ ON	MOD?	MOD _ ON
	Off	MOD _ OFF	MOD?	MOD _ OFF

(4) IF / RF Setup

Measurement parameter		Command Msg	Query Msg	Response Msg	
Brief Function	Control Item				
Sets output level Upper limit	Output Level Limit				
	Value	OLM _ l ₁	OLM?	OLM _ l ₁	
	On	OLL _ ON	OLL?	OLL _ ON	
	Off	OLL _ OFF	OLL?	OLL _ OFF	
Sets reference frequency	Reference Frequency				
	10 MHz	REF _ 10MHZ	REF?	REF _ 10MHZ	
	13 MHz	REF _ 13MHZ	REF?	REF _ 13MHZ	
Sets I, Q levels	I, Q Level	50 Ω 500 mVrms	IQL _ 500MV	IQL?	IQL _ 500MV
		CMOS	IQL _ CMOS	IQL?	IQL _ CMOS
	CMOS Amplitude		CAPL _ l ₃	CAPL?	CAPL _ l ₃
			COS _ l ₃	COS?	COS _ l ₃
	CMOS DC Offset				
Turns I, Q input trim on or off	I, Q Input Trim	On	ITR _ ON	ITR?	ITR _ ON
		Off	ITR _ OFF	ITR?	ITR _ OFF
Turns I, Q output trim on or off	I, Q Output Trim	On	OTR _ ON	OTR?	OTR _ ON
		Off	OTR _ OFF	OTR?	OTR _ OFF
Sets pulse modulation signal input	Pulse Modulation	Int	PM _ INT	PM?	PM _ INT
		Ext	PM _ EXT	PM?	PM _ EXT
Sets pulse modulation signal polarity	Pulse Modulation Polarity	Positive	PMP _ POS	PMP?	PMP _ POS
		Negative	PMP _ NEG	PMP?	PMP _ NEG

(5) Base Band Setup

Measurement parameter		Command Msg	Query Msg	Response Msg	
Brief Function	Control Item				
Sets data and data clock source	Mod Input Data Int Ext	MID┐INT	MID?	MID┐INT	
		MID┐EXT	MID?	MID┐EXT	
	Mod Input Data Clock Int Ext	MIC┐INT	MIC?	MIC┐INT	
		MIC┐EXT	MIC?	MIC┐EXT	
Sets external modulation input data polarity	Ext Mod Input Data Positive Negative	EID┐POS	EID?	EID┐POS	
		EID┐NEG	EID?	EID┐NEG	
	Data Clock Rise Fall	EIC┐RISE	EIC?	EIC┐RISE	
		EIC┐FALL	EIC?	EIC┐FALL	
	Symbol Clock Rise Fall	EIS┐RISE	EIS?	EIS┐RISE	
		EIS┐FALL	EIS?	EIS┐FALL	
	Burst Gate Positive Negative	EIB┐POS	EIB?	EIB┐POS	
		EIB┐NEG	EIB?	EIB┐NEG	
	Sets external modulation output data polarity	Ext Mod Output Data Positive Negative	EOD┐POS	EOD?	EOD┐POS
			EOD┐NEG	EOD?	EOD┐NEG
Data Clock Rise Fall		EOC┐RISE	EOC?	EOC┐RISE	
		EOC┐FALL	EOC?	EOC┐FALL	
Symbol Clock Rise Fall		EOS┐RISE	EOS?	EOS┐RISE	
		EOS┐FALL	EOS?	EOS┐FALL	
Burst Gate Positive Negative		EOB┐POS	EOB?	EOB┐POS	
		EOB┐NEG	EOB?	EOB┐NEG	

(Continued)

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Sets burst trigger input/output edge	Burst Trigger			
	Input			
	Rise	BTI┐RISE	BTI?	BTI┐RISE
	Fall	BTI┐FALL	BTI?	BTI┐FALL
	Output			
	Rise	BTO┐RISE	BTO?	BTO┐RISE
	Fall	BTO┐FALL	BTO?	BTO┐FALL
Selects the pattern sync output signal	Pattern Sync Output			
	PN Clock	PSYNC┐PNCLK	PSYNC?	PSYNC┐PNCLK
	PN Gate	PSYNC┐PNGAT	PSYNC?	PSYNC┐PNGAT
	RF Gate	PSYNC┐RFGAT	PSYNC?	PSYNC┐RFGAT

(6) Save / Recall

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Saves or recalls frequency memory	Frequency Memory			
	Save	FSAV┐n	┐┐┐┐	┐┐┐┐┐┐
	Recall	FRCL┐n	┐┐┐┐	┐┐┐┐┐┐
Saves or recalls parameter memory	Parameter Memory			
	Save	PSAV┐n[,s]†	┐┐┐┐	┐┐┐┐┐┐
	Recall	PRCL┐n	┐┐┐┐	┐┐┐┐┐┐

† Entering a title (s) may be omitted.

(7) Control Command

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Turns buzzer switch on or off	Buzzer			
	On	BUZ┐ON	┐┐┐┐	┐┐┐┐┐┐
	Off	BUZ┐OFF	┐┐┐┐	┐┐┐┐┐┐
Turns display switch on or off	Display On / Off			
	On	DSPL┐ON	┐┐┐┐	┐┐┐┐┐┐
	Off	DSPL┐OFF	┐┐┐┐	┐┐┐┐┐┐
Resets RPP circuit	RPP Reset	RS	┐┐┐┐	┐┐┐┐┐┐
Initializes device in level 3	Preset	PRE	┐┐┐┐	┐┐┐┐┐┐

(8) Screen Transition

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Displays parameter setting Screen	Parameter	PRMTR	_____	_____
Displays burst internal modulation data	Burst Pattern	BURST	_____	_____
Displays IF/RF setup screen	IF / RF Setup	IFRF	_____	_____
Displays base band setup	Base Band Setup	BASE	_____	_____
Displays hardware check screen	Hardware Check	CHECK	_____	_____
Displays interface setup screen	Interface Setup	INTFC	_____	_____

(9) GPIB

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Turns GPIB response header on or off	Header			
	On Off	HEAD _ ON HEAD _ OFF	_____	_____
Sets GPIB terminator	Terminator			
	LF CR / LF	TRM _ 0 TRM _ 1	TRM? TRM?	TRM _ 0 TRM _ 1
Sets Extend Status Enable Register and Extend Status Register	Extend Status Enable Register (END)	ESE2 _ n	ESE2?	ESE2 _ n
	Extend Status Register (END)	_____	ESR2?	ESR2 _ n
	Extend Status Enable Register (ERR)	ESE3 _ n	ESE3?	ESE3 _ n
	Extend Status Register (ERR)	_____	ESR3?	ESR3 _ n

2.2.3 Device message list compatible with the MG3633A

(1) Frequency

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Sets frequency	Frequency	FR_f FC_f CF_f	FROA FCOA CFOA	FR_f FR_f FR_f
Sets frequency using incremental step	Frequency Incremental Step Value Up Down	FIS_f UFR DFR	FISOA _____ _____	FIS_f _____ _____
Sets frequency using rotary knob	Frequency Knob Up Down	TFR EFR	_____ _____	_____ _____
Sets frequency resolution	Frequency Resolution 1 Hz 10 Hz 100 Hz 1 kHz 10 kHz 100 kHz 1 MHz 10 MHz Right Left	R2 R3 R4 R5 R6 R7 R8 R9 FSR FSL	_____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____	_____ _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
Turns relative frequency mode on or off	Frequency Relative Reference Value Value On Off	_____ _____ FO FF	REFOA RLFOA _____ _____	REF_f RLF_f _____ _____
Sets frequency offset	Frequency Offset Value On Off	FOS_f SP12 SP11	FOSOA _____ _____	FOS_f _____ _____

(2) Output level

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Sets output level	Output Level	OL $\underline{\hspace{1cm}}$ l_1 AP $\underline{\hspace{1cm}}$ l_1	OLOA APOA	OL $\underline{\hspace{1cm}}$ l_1 OL $\underline{\hspace{1cm}}$ l_1
Sets output level using incremental Step	Output Level Incremental Step Value Up Down	OIS $\underline{\hspace{1cm}}$ l_2 UOL DOL	OISOA 	OIS $\underline{\hspace{1cm}}$ l_2
Sets output level using rotary knob	Output Level Knob Up Down	TOL EOL	 	
Sets output level resolution	Output Level Resolution 0.1 dB 1 dB 10 dB Right Left	LØ L1 L2 OSR OSL	 	
Turns output level Continuous Mode on or off	Output Level Continuous Mode On Off	LC LN	 	
Turns relative output Level mode on or off	Output Level Relative Reference Value Value On Off	 LO LF	RE00A RLO0A 	REO $\underline{\hspace{1cm}}$ l_1 RLO $\underline{\hspace{1cm}}$ l_2
Sets output level offset	Output Level Offset Value On Off	OOS $\underline{\hspace{1cm}}$ l_2 SPØ8 SPØ7	OOSO A 	OOS $\underline{\hspace{1cm}}$ l_2
Turns output level on or off	Output Level On Off	RO RF	 	
Selects voltage display	Voltage Display Electro-Motive Force (EMF) Terminated (TERM)	SPØ3 SPØ4	 	

(Continued)

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Changes output level unit	Output Level Unit	OLDBM	_____	_____
	Change to dBm	OLDM APDBM APDM	_____	_____
	Change to dB μ	OLDBU OLDU APDBU APDU	_____	_____
	Change to V	OLV APV	_____	_____
Sets output level Upper limit	Output Level Limit Value On Off	OLM $\dots l_1$ SP06 SP05	OLMOA _____ _____	OLM $\dots l_1$ _____ _____

(3) Save / Recall

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Saves or recalls frequency memory	Frequency Memory			
	Save Recall	FQnST FQnRC	_____ _____	_____ _____
Saves or recalls parameter memory	Parameter Memory			
	Save Recall	FNnST FNnRC	_____ _____	_____ _____

(4) Others

Measurement parameter		Command Msg	Query Msg	Response Msg
Brief Function	Control Item			
Resets RPP circuit	RPP Reset	RS	_____	_____
Initializes device in level 3	Preset	SP00	_____	_____
Turns buzzer switch on or off	Buzzer			
	On Off	SP02 SP01	_____ _____	_____ _____
Turns GPIB response header on or off	Header			
	On Off	SP60 SP61	_____ _____	_____ _____

SECTION 2 DEVICE MESSAGE LIST

(Blank)

SECTION 3

CONNECTING BUS AND SETTING 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

3.1	Connecting Devices with GPIB Cables	3-3
3.2	Checking and Setting GPIB address	3-4

(PAGE)

(Blank)

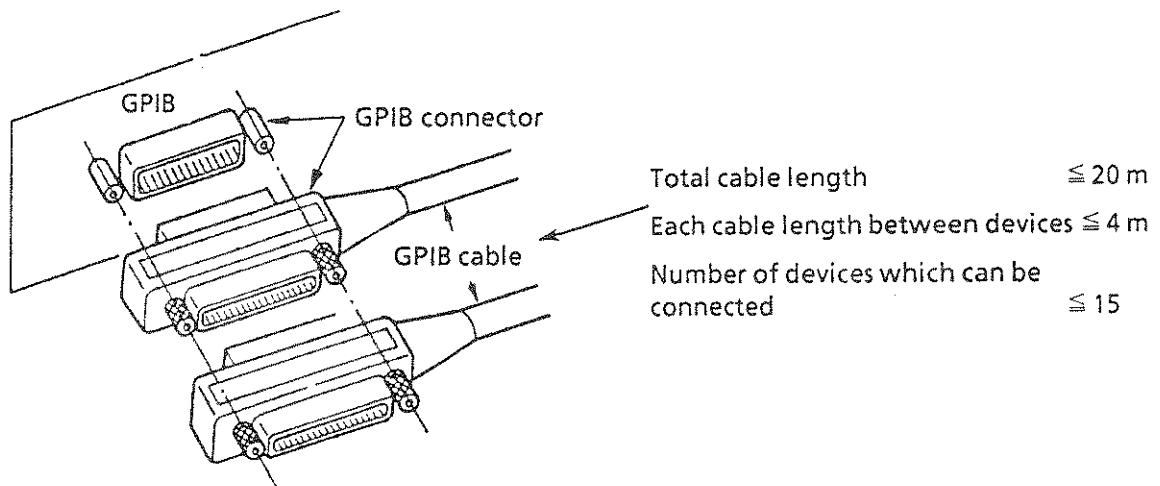
3.1 Connecting Devices with GPIB Cables

CAUTION

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


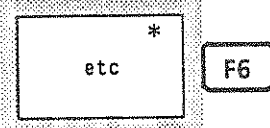
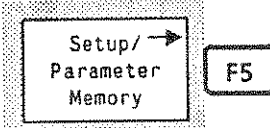
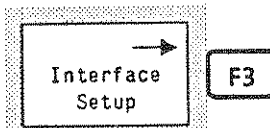

The rear panel has a connector for connecting GPIB cables.

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



3.2 Checking and Setting GPIB address

The following shows how to set GPIB address.

Step	Action	Verification
1		<p>Enters the frequency, output level, or modulation setting mode under the parameter setting screen and lights the Freq, Level, or Mod key's LED. (any one of the parameter setting screens including the 'Setup/Parameter Memory' F5 soft key can be used when the Freq, Level, or Mod panel key is pressed.)</p>
2		<p>Switches to screen transition menu 2.</p>
3		<p>Switches to screen transition menu 1 from any of the frequency setting mode/output level setting mode/modulation parameter setting mode in the parameter setting screen.</p>
4		<p>Switches to the Interface setup screen and displays a current GPIB address. Set the desired address. (For further details, see paragraph 4.6.3 in the Panel Operating Instruction Part of this manual)</p>
5		<p>Returns to the parameter setting screen (screen transition state).</p>

† If the Freq, Level, or Mod key's LED is already lit, there is no need to press its key.

Notes: GPIB address has been set to 00 at shipment from the factory.

SECTION 4 INITIAL SETTINGS

The MG3670B 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

4.1	Bus Initialization by the IFC Statement	4-4
4.2	Initialization for Message Exchange by DCL and SDC Bus Commands	4-5
4.3	Device Initialization by the *RST Command	4-6
4.4	Device Initialization by the *PRE Command	4-9
4.5	Device Status at Power-on	4-9

(Blank)

SECTION 4 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 (Device Clear) and SDC (Selected Device Clear) GPIB bus commands, which also nullify the function that reports to the controller that operation has completed	Can be combined with other levels, level 2 must be executed before level 3
3	Device initialization	The *RST or PRE 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.

4.1 Bus Initialization by the IFC Statement

■ Example

```
ISSET 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)

4.2 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 MG3670B 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*
-

4.3 Device Initialization by the *RST Command

■ Syntax

*RST

■ Example

PRINT @1;"*RST" Initializes the device (MG3670B) 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 (MG3670B) to a specific initial state.

*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*

The following tables list the initial settings of MG3670B.

Group	Preset item	Initial setting data
Frequency	Frequency	10 MHz
	Incremental step frequency	1 MHz
	Frequency resolution	1 Hz
	Relative frequency display mode	Off
	Frequency offset mode	Off
	Frequency offset value	0 Hz
Output	Output level	-30 dBm
	Incremental step output level	1 dB
	Output level resolution	0.1 dB
	Continuous mode	Off
	Relative level display mode	Off
	Output level (On / Off)	On
	Output level offset mode	Off
	Output level offset value	0 dB
	Open-circuit / terminated voltage display	Open-circuit voltage display
Modulation	Modulation	Off

SECTION 4 INITIAL SETTINGS

Group	Preset item	Initial setting data
IF/RF settings	Reference frequency	10 MHz
	Output level upper limit	-10 dBm
	Output level upper limit (On/Off)	Off
	I, Q LEVEL	50 Ω 500 mVrms CMOS 50 mVrms CMOS DC Offset 2500 mV
	I, Q trim setting	Input: Off Output: Off
	Pulse modulation	Pulse Modulation: Int Pulse Modulation Polarity: Positive
Baseband settings	Data	Int
	Data Clock	Int
	EXT Mod Input	Data: Positive Data Clock: Rise Symbol Clock: Rise Burst Gate: Positive
	EXT Mod Output	Data: Positive Data Clock: Rise Symbol Clock: Rise Burst Gate: Positive Burst Trigger Input: Rise Burst Trigger Output: Rise Pattern Sync Output: PN Clock
Others		Panel lock: Released Buzzer On/Off: On Display On/Off: On Rotary knob hold On/Off: Off

4.4 Device Initialization by the PRE Command

■ Syntax

```
PRE
```

■ Example

```
PRINT @1;"PRE" ..... Initializes the device ( MG3670B ) whose address is 1 with level 3
```

■ Explanation

The PRE command is MG3670B device-dependent messages which initialize a device with level 3.

The items and the initial settings which are initialized by the PRE commands are the same as those of the *RST command listed in paragraph 4.3.

4.5 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.
- 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 5 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

5.1	General Description	5-3
5.2	Program Message Format	5-3
5.3	Response Message Format	5-9

TRANSDUCER INFORMATION

TRANSDUCER INFORMATION

(Blank)

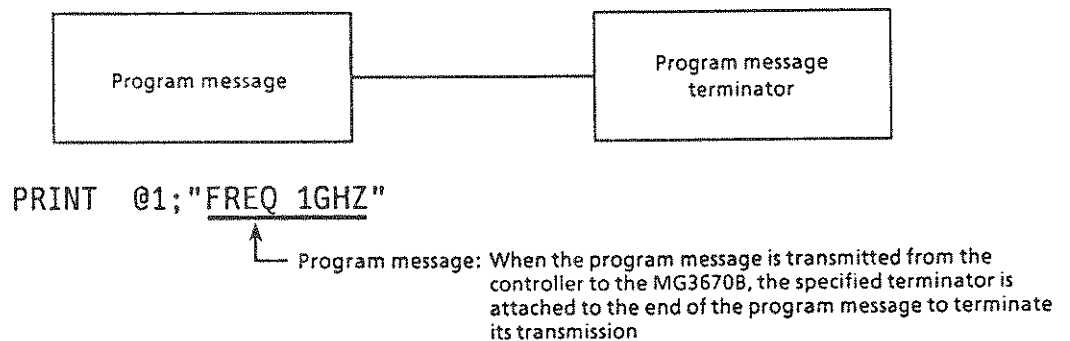
SECTION 5 DEVICE MESSAGE FORMAT

5.1 General Description

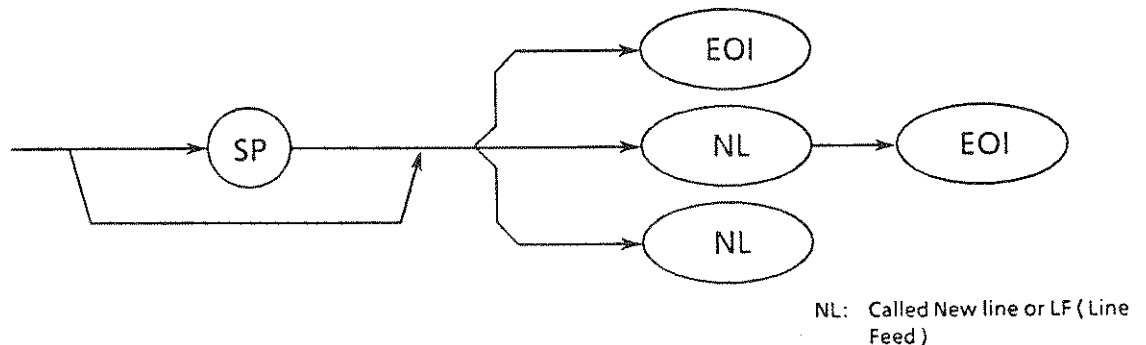
The device messages are data messages that are transmitted between the controller and devices. There are two types of data messages: program messages that are transferred from the controller to the MG3670B (device), and response messages that are sent from the MG3670B (device) to the controller. There are also two types of program commands and program queries in the program message. The program command is used to set the MG3670B's parameters and to instruct it to execute processing. The program query is used to query the values of parameters and measured results.

5.2 Program Message Format

To transfer the program messages from the controller program to the MG3670B using the PRINT statement, the program message formats are defined as follows.

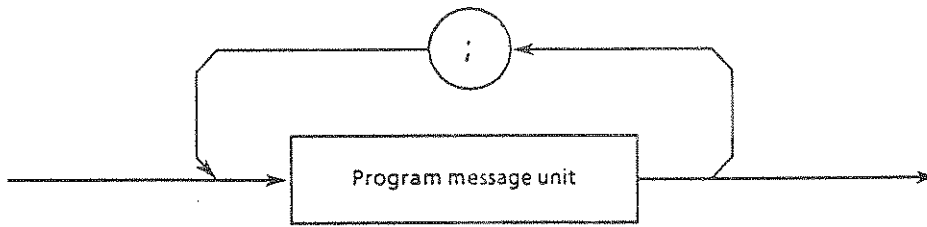


(1) Program message terminator



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

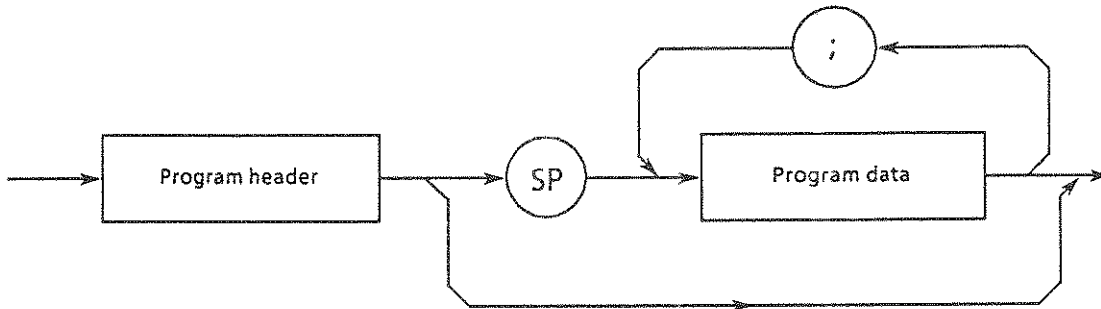
(2) Program message



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

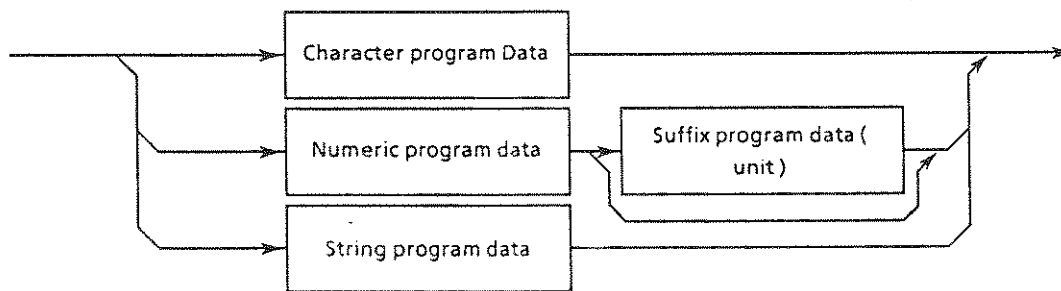
<Example> PRINT @1;"FREQ 1GHZ;OLVL 0DBM"

(3) Program message unit



- The IEEE488.2 common command has a leading asterisk “*” that is always placed before the program header.
- The program query has a trailing question mark “?” that is always added at the end of the program header.

(4) Program data



(5) Character program data

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

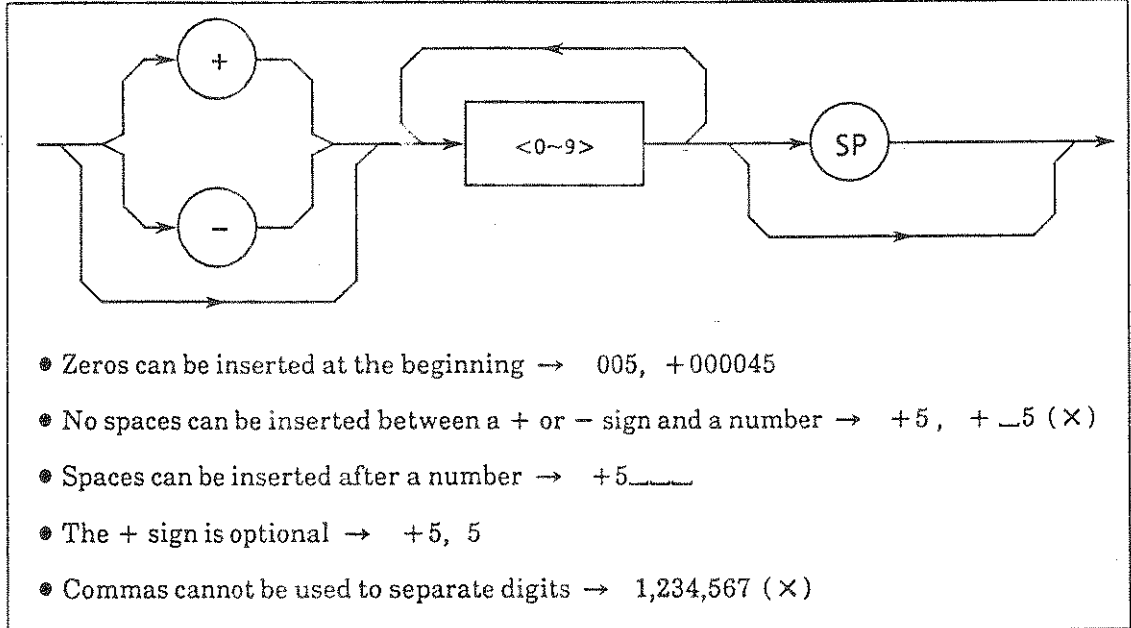
<Example>

PRINT @1 ; "VDSPL TERM" Displays an output level voltage as the terminated voltage

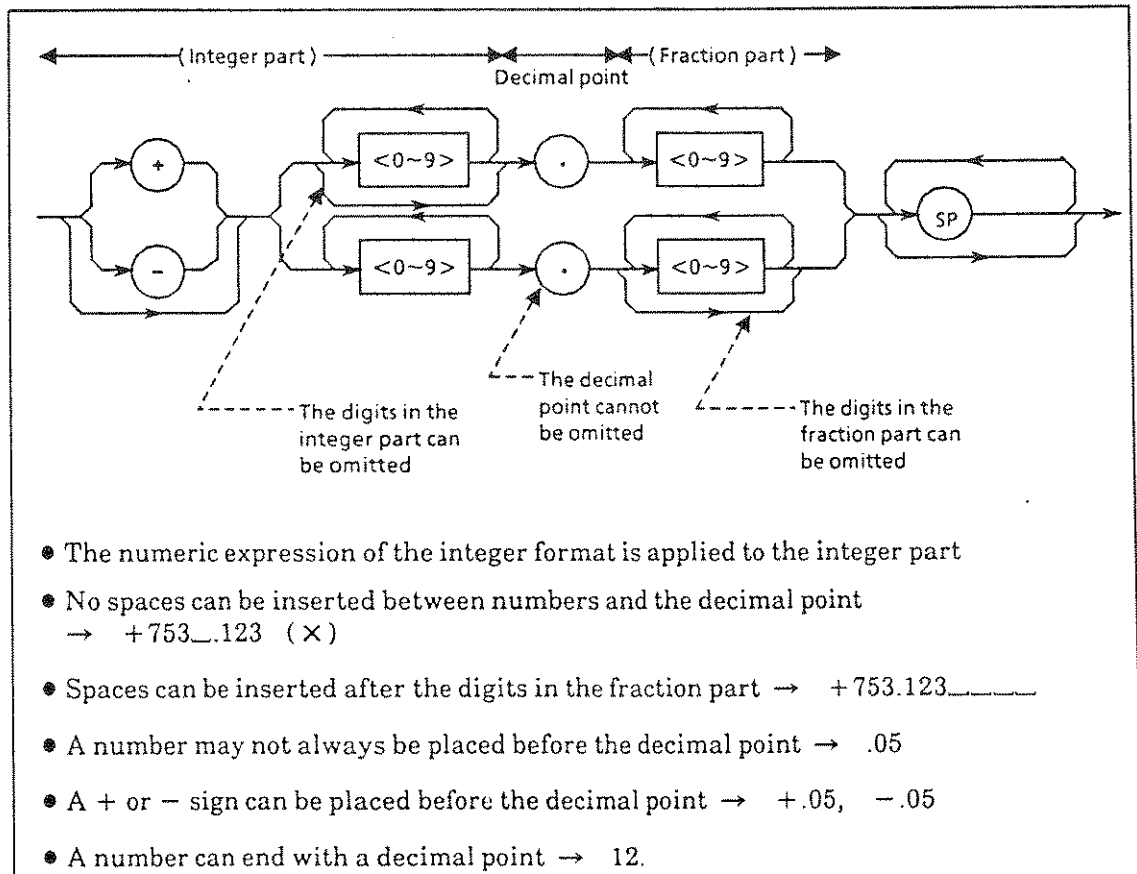
(6) Numeric program data

Numeric program data has four types of formats: integer format (NR1), fixed-point format (NR2), floating point format (NR3), and hexadecimal number.

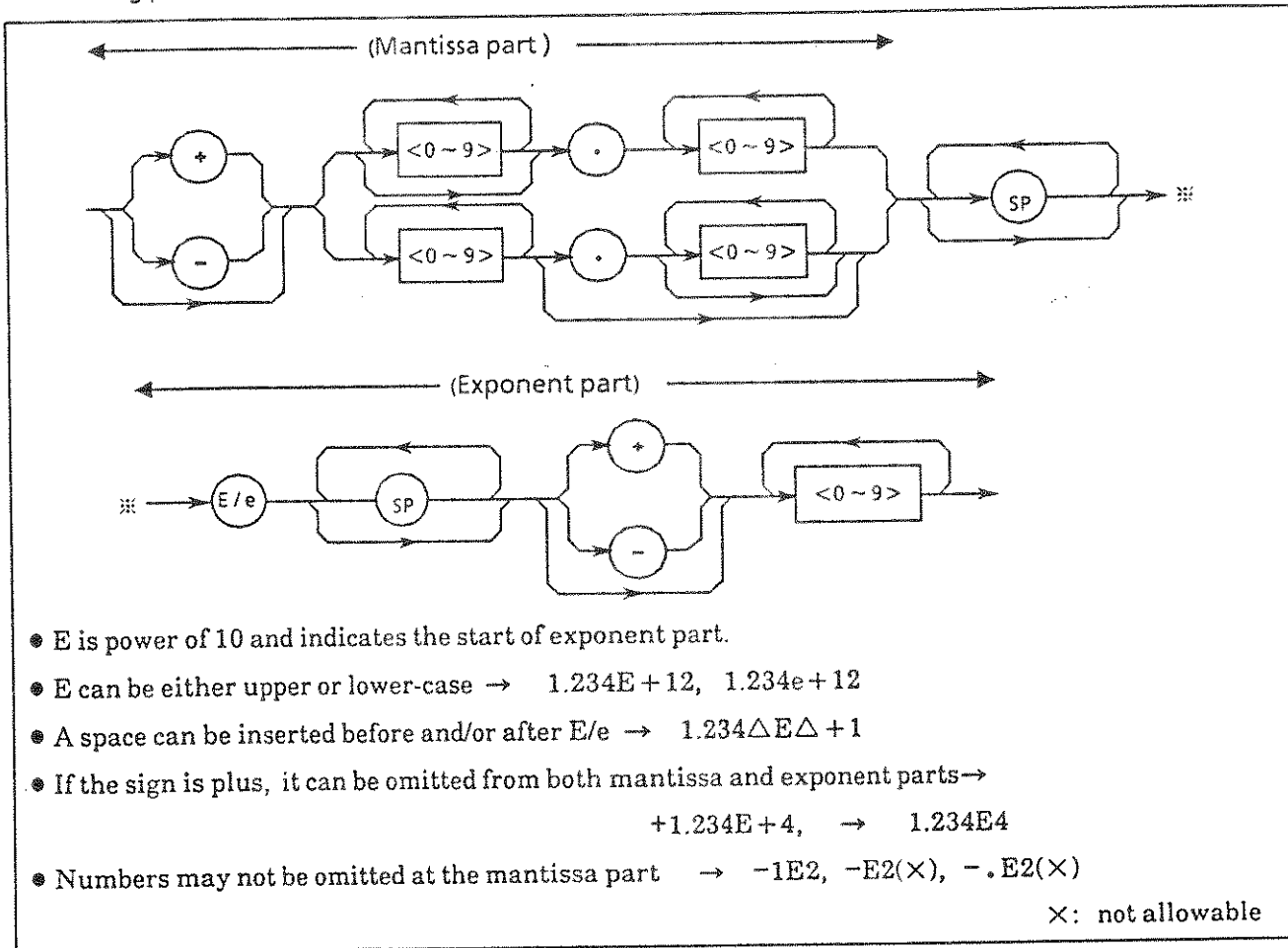
< Integer Format (NR1) >



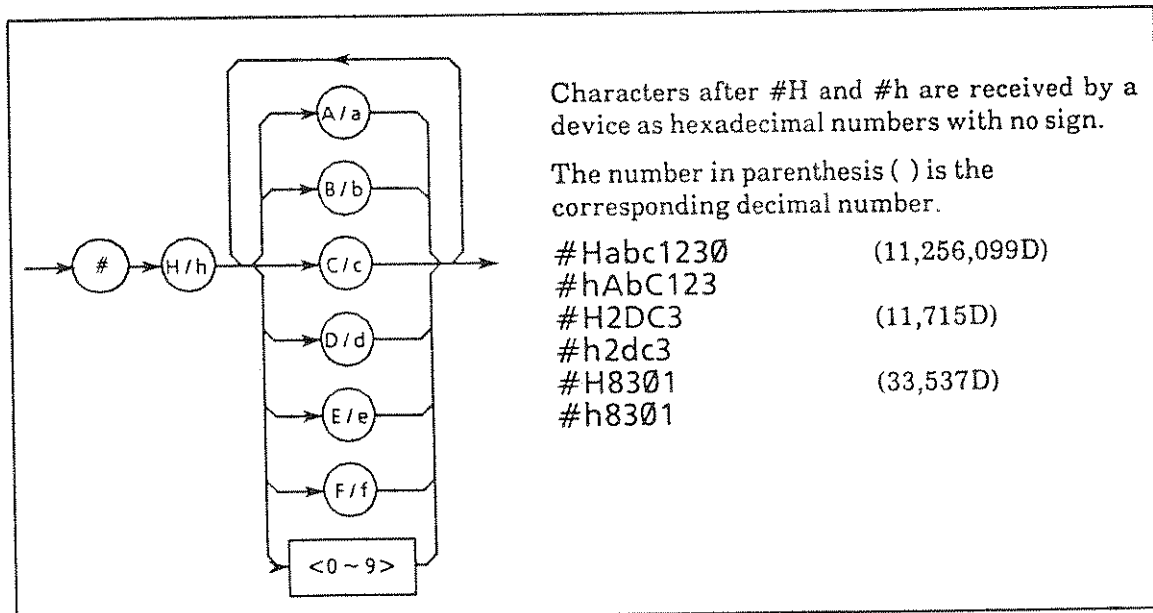
< Fixed-Point Format (NR2) >



< Floating point format (NR3) >



< Hexadecimal Number >



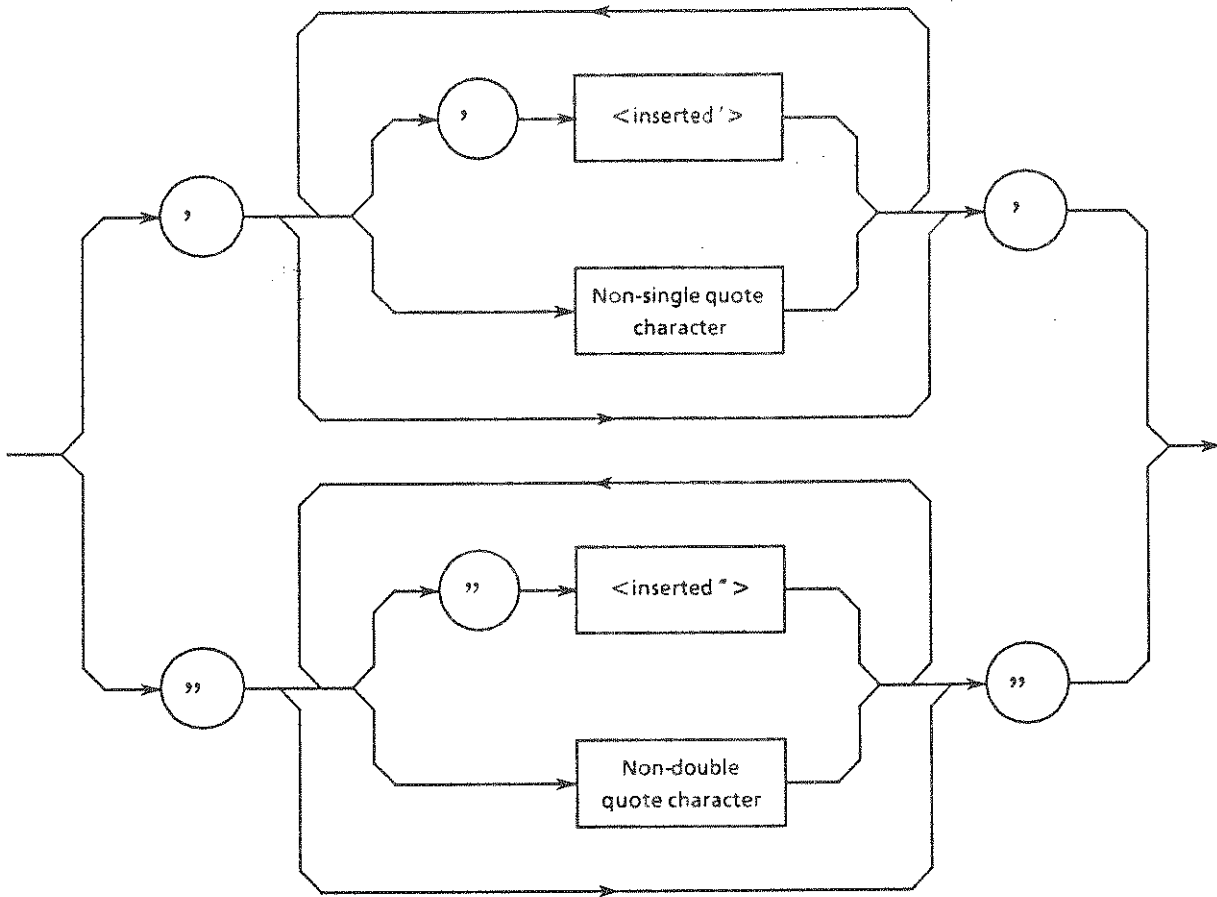
(7) Suffix program data (unit)

The table below shows the suffixes used for the MG3670B.

Table of MG3670B Suffix Codes

Classification	Unit	Suffix code
Frequency	GHz	GHZ , GZ
	MHz	MHZ , MZ
	kHz	KHZ , KZ
	Hz	HZ
Output level	dB	DB
	dBm	DBM , DM
	dB μ	DBU , DU
	V	V
	mV	MV
	μ	UV

(8) STRING PROGRAM DATA



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

```
PRINT @ 1; "PSAV 1,'MG367ØB'"
```

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

```
PRINT @ 1; "PSAV 1,'DSG''1'''"
```

Executing PSAVE results in DSG ' 1 '.

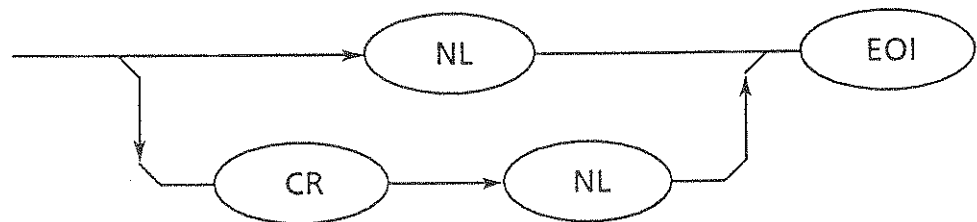
Note: To use the double quotation mark " in the PRINT statement, specify CHR\$(&H22).

5.3 Response Message Format

To transfer the response messages from the MG3670B to the controller using the INPUT statement, the response message formats are defined as follows.

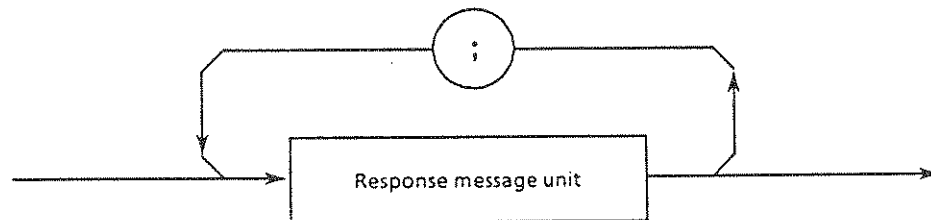


(1) Response message terminator



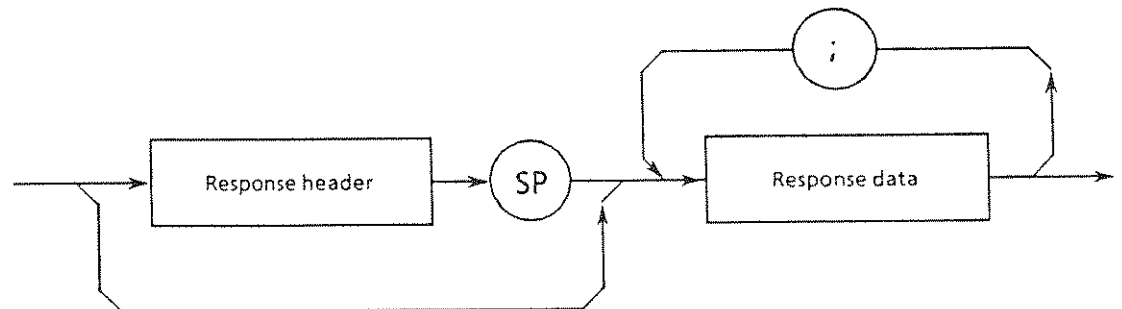
The response message terminator to be used depends on the TRM command.

(2) Response message

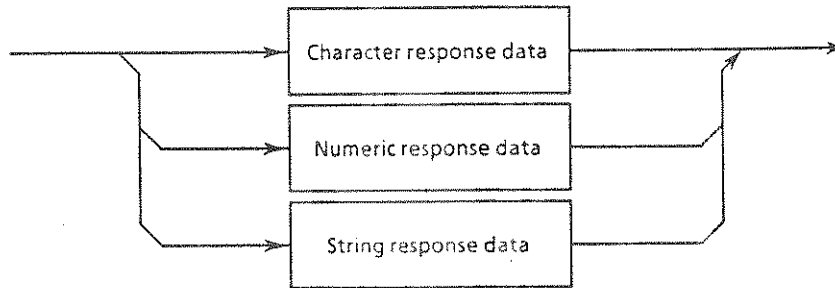


When a query is sent by the PRINT statement with one or more program queries, the response message also consists of one or more response message units.

(3) Response message unit (example)



(4) Response data

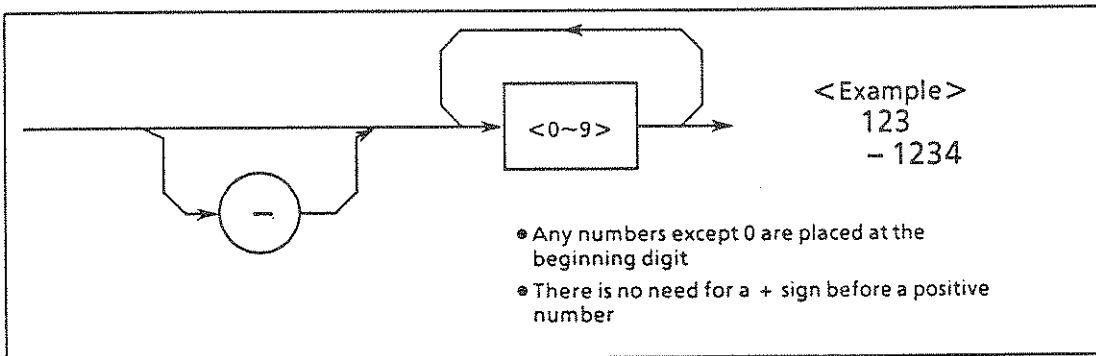


(5) character response data

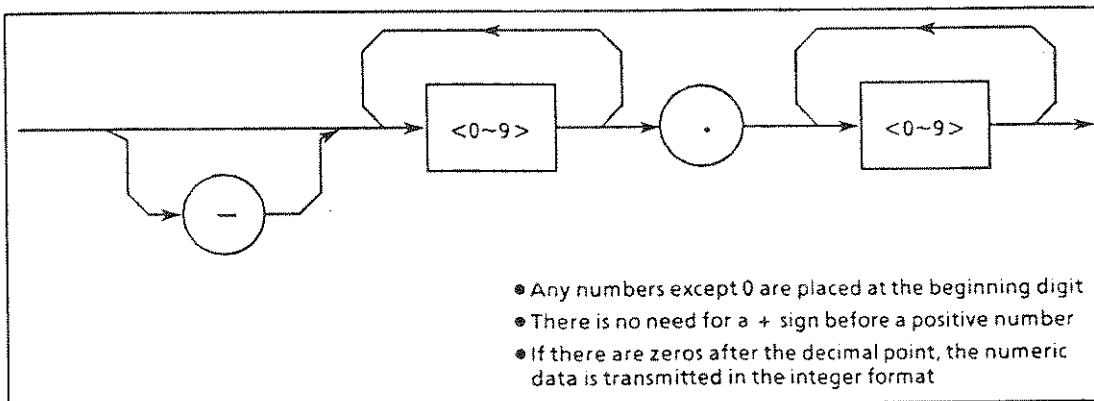
Character response data consists of the upper-case alphabetic characters from A to Z, lower-case alphabetic characters from a to z, the underline “_”, and the numbers 0 to 9. They can be used in a specified combination.

(6) Numeric response data

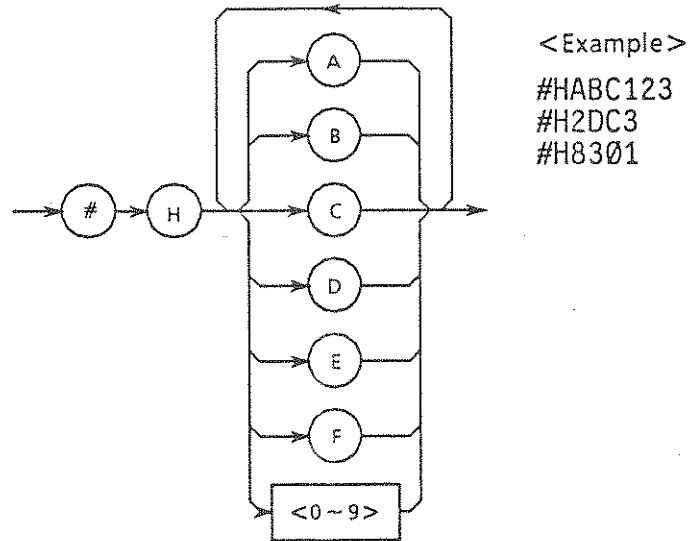
< Integer format (NR1) >



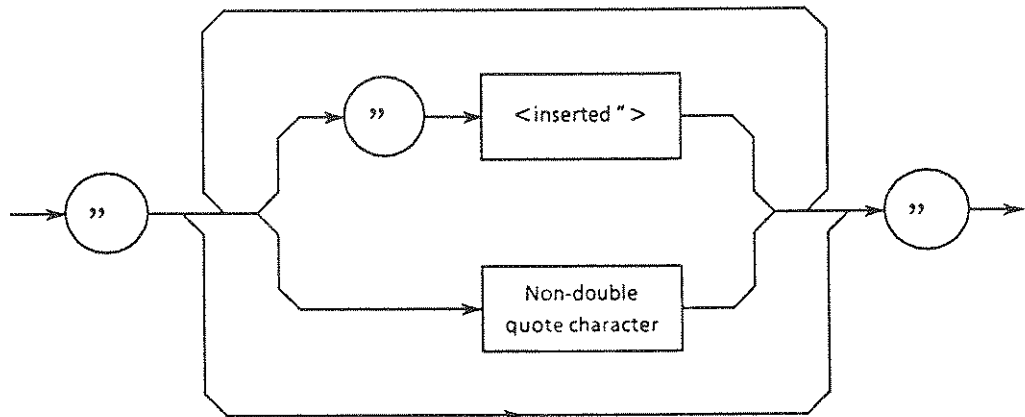
< Fixed-point format (NR2) >



< Hexadecimal Number >



(7) String response data



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

(Blank)

SECTION 6 COMMON COMMANDS

This section describes the common commands and common query commands specified in the IEEE 488.2 standard. Like device messages, common commands are a type of data message used in the bus data mode. They can be used for all measuring instruments, including those made by other companies, as long as they conform to the IEEE 488.2 standard. IEEE 488.2 common commands must start with an *.

TABLE OF CONTENTS

6.1	Classification by Function of Common Commands Supported by the MG3670B	6-3
6.2	Common Commands List	6-4

(Blank)

6.1 Classification by Function of Common Commands Supported by the MG3670B

The table below shows the classification by function of the IEEE 488.2 common commands supported by the MG3670B. Supported commands are listed on the following pages in alphabetical order.

Group	Function	Mnemonic
System data	Data specific to each device connected to the GPIB system, e.g. manufacturer, model, serial number, etc.	*IDN?
Internal operation	Device internal control: ① Resetting device in level 3 ② Device self testing and error detection	*RST *TST?
Synchronization	Synchronization of device to controller by: ① Waiting for a service request ② Waiting for a response from the device output queue ③ Forced sequential execution	*OPC *OPC? *WAI
Status and event	A status byte consists of seven single-bit summary messages. The summary bits of the message are supplied by the standard event register, the output queue, and the extended event register or extended queue. Three commands and four queries are available to set or clear the data in the registers and queues, to enable or disable them and to obtain the settings status of the registers.	*CLS *ESE *ESE? *ESR? *SRE *SRE? *STB?

6.2 Common Commands List

Command	Function
*CLS	Clears the Status Byte Register
*ESE NR1	Sets or clears the Standard Event Status Enable Register
*ESE?	Returns the current value of the Standard Event Status Enable Register
*ESR?	Returns the current value in the Standard Event Status Register
*IDN?	<p>Returns manufacturer's name, model name, serial numbers, firmware level</p> <p>Returns manufacturer's name · model name · serial numbers · firmware level</p>
*OPC	Sets the status of bit 0 of the Standard Event Status Register when device operation is completed
*OPC?	Sets 1 in the output queue to generate a MAV summary message when device operation has been completed
*RST	Resets device in third level
*SRE NR1	Sets or clears the status of bits in the Service Request Enable Register
*SRE?	Returns the current value of the Service Request Enable Register
*STB?	Returns the current values of status bytes including MSS bits
*TST?	Returns self-test error results
*WAI	Keeps the next command on stand-by if the device is currently executing a command

***CLS Clear Status Command**

(Clears the Status Byte Register)

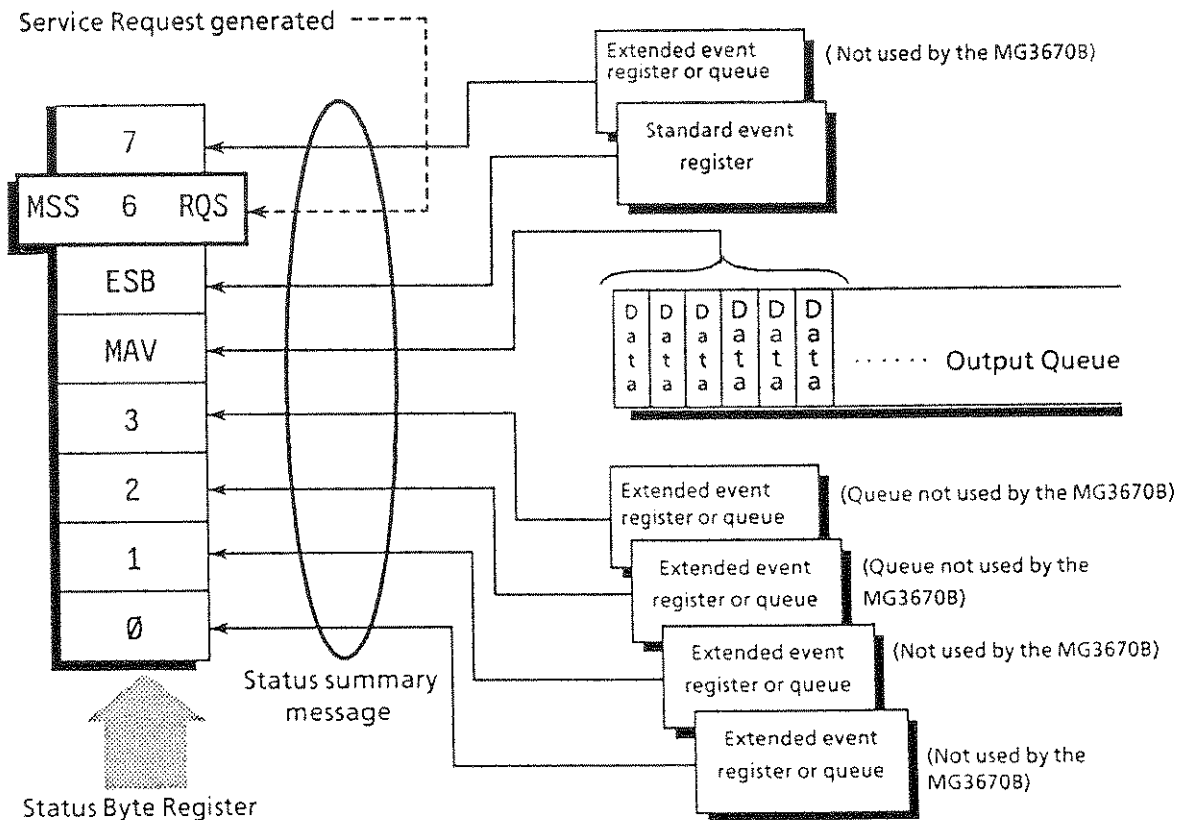
Format***CLS****Example**`30 PRINT @ADR;"*CLS"`**Explanation**

The *CLS common command clears all status data (i.e their event registers and queues) except for the output queue and its MAV summary messages. It also clears the summary messages corresponding to these structures.

In the example below, the output queue and its MAV summary messages are also cleared.

That is to say, if a *CLS command is sent after a <Program message terminator> or before <Query message unit> elements, all status bytes are cleared. This command also clears all unread messages in the output queue.

*CLS has no effect on settings in enable registers.



***ESE Standard Event Status Enable Command**

(Sets or clears the Standard Event Status Enable Register)

Format

*ESE <Header separator> <Decimal numeric program data>

In this format:

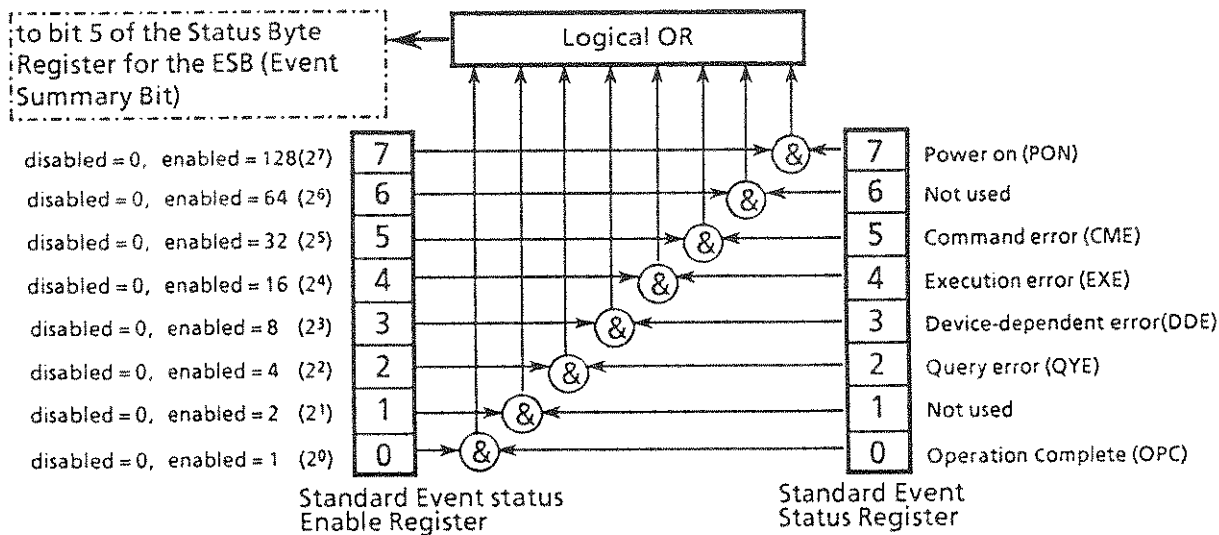
<Decimal numeric program data> = Value rounded to an integer from 0 to 255 (Binary weighted with a base value of 2)

Example

PRINT @ADR;"*ESE 20"! Sets bits 2 and 4 of Enable Register

Explanation

The program data is the sum of weighted bit-digit values when the weighted value for bits to be enabled are selected from among the values $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$ or $2^7=128$; corresponding to the Standard Event Enable Register bits 0, 1, 2, 3, 4, 5, 6 or 7. The value of bits to be disabled is 0.



***ESE? Standard Event Status Enable Query**

(Returns current value of Standard Event Status Enable Register)

Format

*ESE?

Example

20 is the response if *ESE? is sent after executing *ESE 20

Explanation

Returns NR1, the value of the Standard Event Status Enable Register.

```
30 PRINT @ADR;"*ESE?"
40 INPUT @ADR;NR1
50 PRINT NR1
```

Response message

NR1 = 0 to 255

***ESR?: Standard Event Status Register Query**

(Returns the current value in the Standard Event Status Register)

Format***ESR?****Example**

```

30 PRINT @ADR;"*ESR?"
40 INPUT @ADR;NR1!
50 PRINT NR1

```

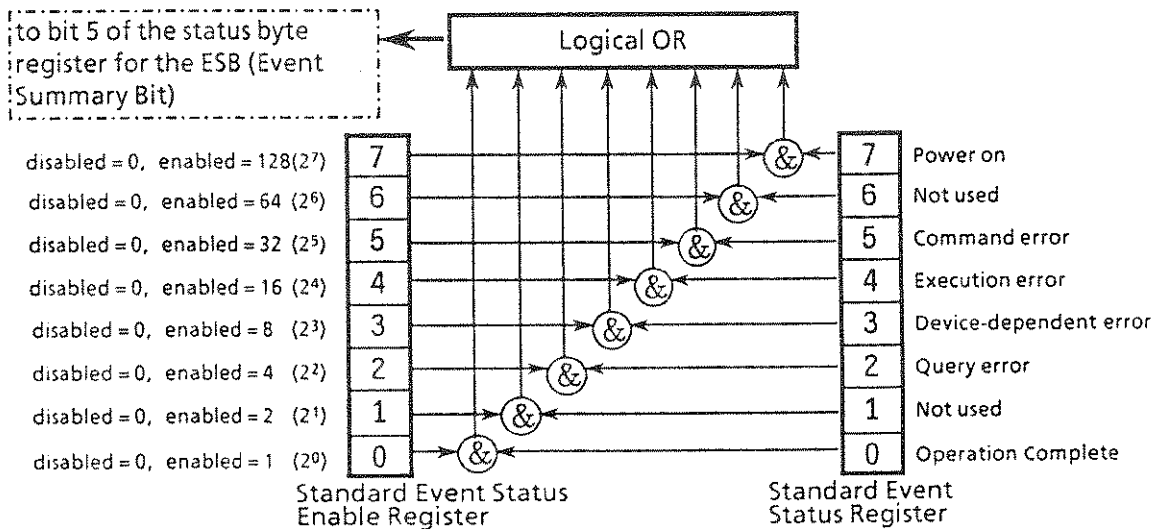
A command error is issued if the value of the variable is 32

Response Message

NR1 = 0 to 255

Explanation

The current value of the Standard Event Status Register is returned by NR1. NR1 is the total of weighted bit-digit values of bits (enabled by the standard event status enable register) which are selected from among the values $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$ or $2^7=128$: corresponding to the Standard Event Status Register bits 0, 1, 2, 3, 4, 5, 6 or 7. This register is cleared when the response is read (e.g. line 40).



***IDN? Identification Query**

(Returns manufacturer name, model name etc. of product)

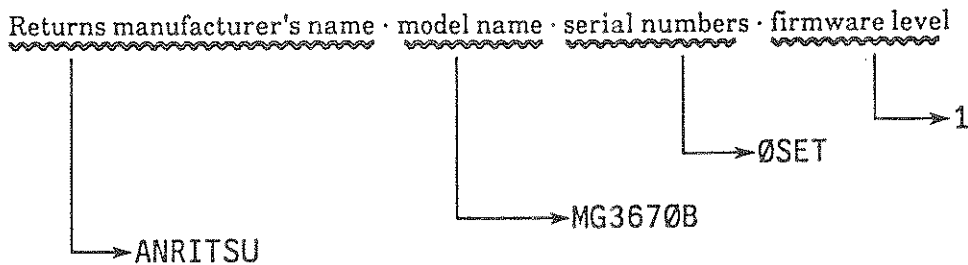
■ **Format**

*IDN?

■ **Example**

```
30 PRINT @ADR;"*IDN?"
40 INPUT @ADR;IDEN$! ..... Stores names of manufacturer, model, etc.
```

■ **Explanation**



A response message comprising the four fields shown below is returned.

- ① Field 1 Manufacturer's name (Anritsu)
- ② Field 2 Model name
- ③ Field 3 (usually 0)
- ④ Field 4 Firmware version (In the case of Anritsu, versions are from 1 to 99. The version is 1 when no changes have been made to hardware and software.)

■ **Response message**

A response message comprising of the above four fields separated by commas is sent by <Arbitrary ASCII response data>.

<field 1>, <field 2>, <field 3>, <field 4>

For the above example:

ANRITSU, MG3670B, 0, 1

The total length of a response message is ≤ 72 characters

***OPC Operation Complete Command**

(Sets the status of bit 0 of the Standard Event Status Register when device operation is completed)

Format

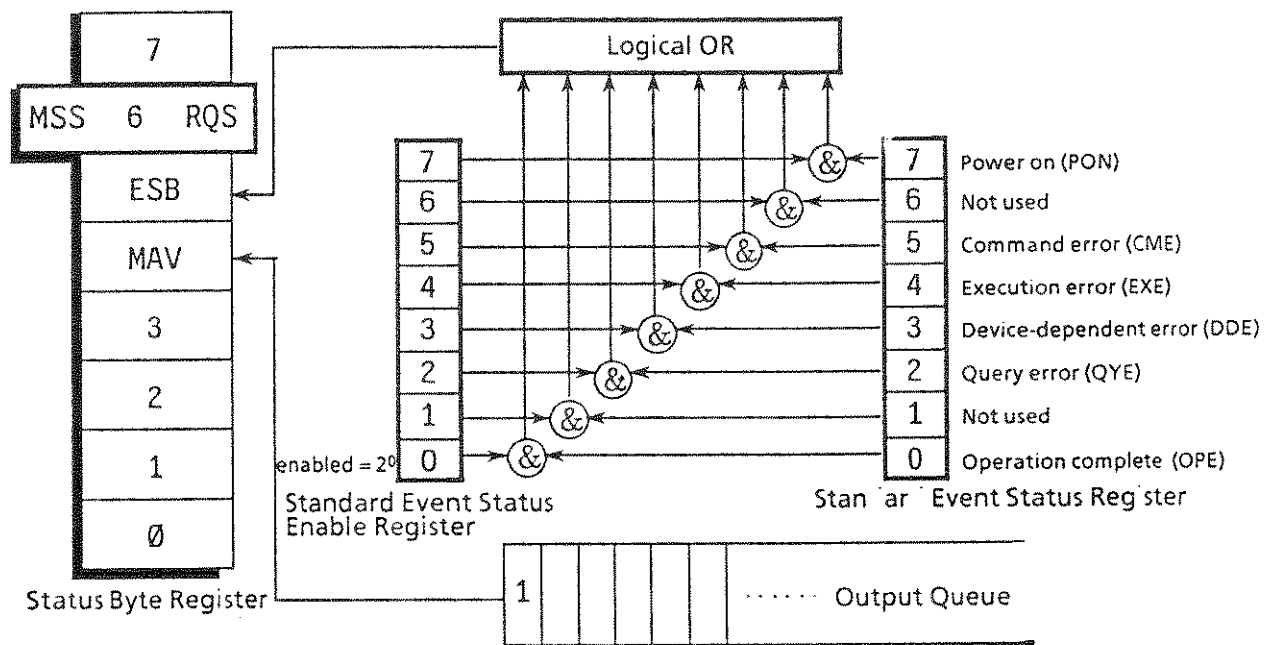
*OPC

Example

```
PRINT @ADR;"*OPC"
```

Explanation.

Sets the status of bit 0, i.e. the operation complete bit, in the Standard Event Status Register when selected device operations have been completed. This is an overlap command.

***OPC? Operation Complete Query**

(Sets 1 in the output queue to generate a MAV summary message when device operation has been completed)

Format

*OPC?

Example

```
30 PRINT @ADR;"*OPC?"
40 INPUT @ADR;N
```

Explanation

When selected device operations have been completed, sets 1 in the output queue and waits for the MAV summary message to be generated.

Response message

A single ASCII-encoded byte for '1' (31hex) is returned by <NR1 numeric response data>.

*RST Reset Command

(Resets device in third level)

■ Format

*RST

■ Example

PRINT @ADR;"*RST" Initializes only 3-addressed device in the third level

■ Explanation

The *RST command resets a device in the third level. (See Section 4.) The third-level initialization items are as follows:

- ① The device-dependent functions and condition are set to a known state that is independent of the past-use history of the device. The MG3670B is set as shown in paragraph 4.3.
- ② The device is forced to OCIS (Operation Complete Command Idle State). As a result, the operation-complete bit cannot be set in the Standard Event Status Register.
- ③ The device is forced to OQIS (Operation Complete Query Idle State). As a result, the operation-complete bit 1 cannot be set in the output queue. The MAV bit in the Status Byte Register is cleared.

The *RST command does not affect the items listed below

- ① IEEE 488.1 interface state
- ② Device address
- ③ Output queue
- ④ Service Request Enable Register
- ⑤ Standard Event Status Enable Register
- ⑥ Power-on-status-clear flag setting
- ⑦ Calibration data affecting device specification

***SRE Service Request Enable Command**

(Sets status of bits in the Service Request Enable Register)

Format

*SRE <Header separator> <Decimal numeric program data>

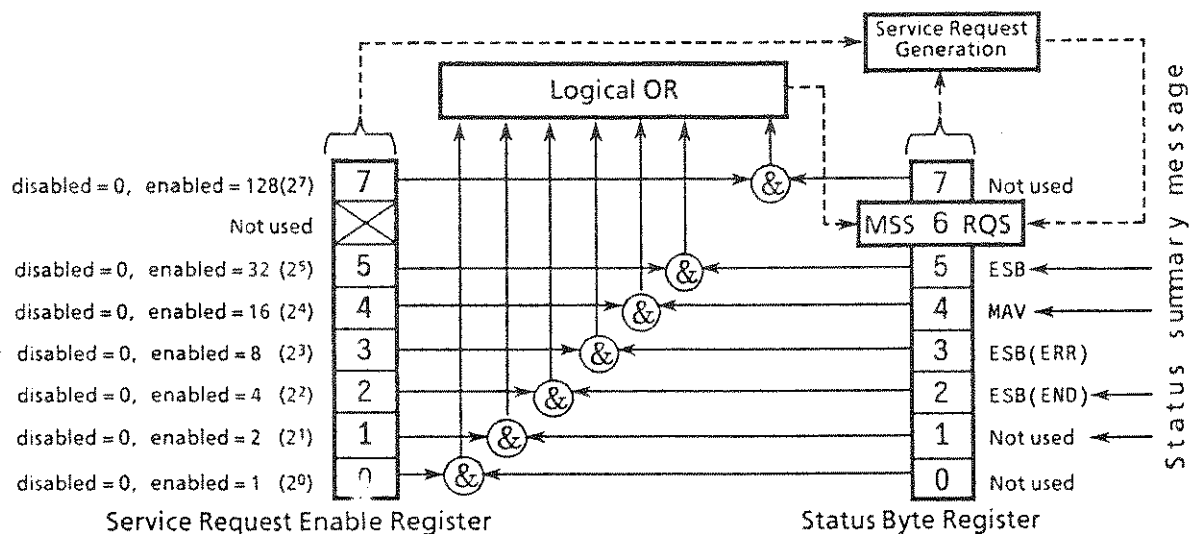
<Decimal numeric program data> = Values rounded to an integer from 0 to 255 (binary weighted with a base value of 2)

Example

PRINT @ADR;"*SRE 16"! Sets bit 4 (16=0001 0000) of the enable register

Explanation

The program data is the sum of weighted bit-digit values when the weighted value for bits to be enabled are selected from among the values $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$ or $2^7=128$: corresponding to the Service Request Enable Register bits 0, 1, 2, 3, 4, 5, 6 or 7. The value of bits to be disabled is 0.

***SRE? Service Request Enable Query**

(Returns the current value of the Service Request Enable Register)

Format

*SRE?

Example

A 16 is sent in response if *SRE? is sent after executing *SRE 16.

Explanation

Returns NR1, the value of the Service Request Enable Register.

```
30 PRINT @ADR;"*SRE?"
40 INPUT @ADR;NR1
50 PRINT NR1
```

Response message

As NR1 (bit 6 : RQS bit) cannot be set, NR1 = 0 to 63 or 128 to 191)

*STB? Read Status Byte Command

(Returns the current values of status bytes including MSS bits)

■ Format

*STB?

■ Example

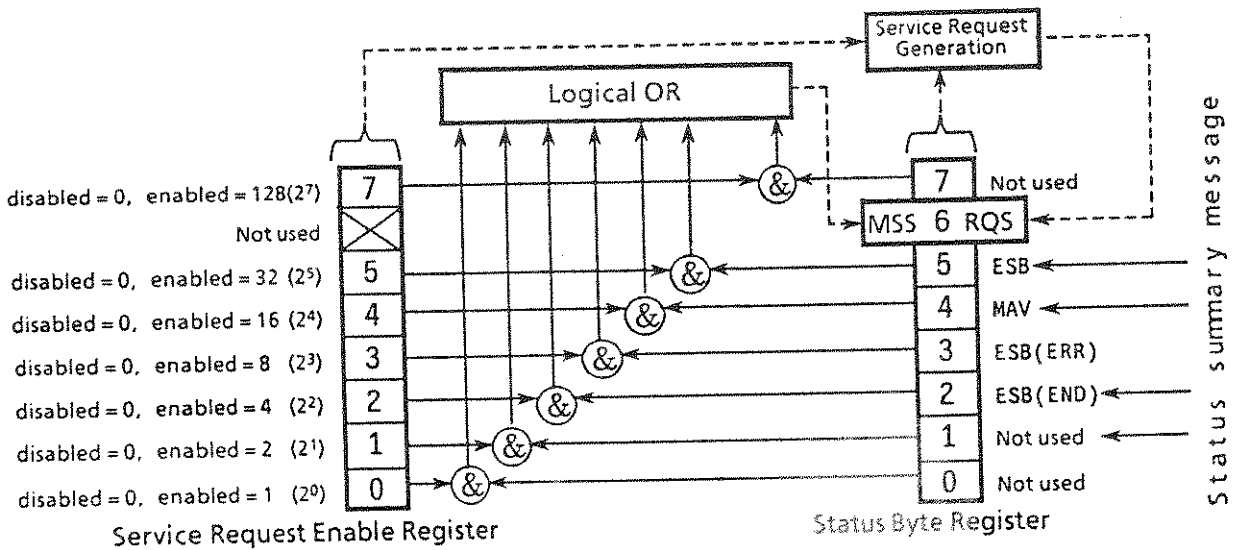
```
30 PRINT @ADR;"*STB?"
40 INPUT @ADR;NR1
50 PRINT NR1
```

■ Explanation

The *STB? query returns the total of the binary weighted values of the Status Byte Register and of the MSS summary message with <NR1 numeric response data >.

■ Response message

The response message is a <NR1 numeric response data > integer in the range 0 to 255 representing the total of the binary weighted values of the bits in the Status Byte Register. Status Byte Register bits 0 to 5 and 7 are weighted to 1, 2, 4, 8, 16, 32 and 128, respectively, and the MSS (Master Summary Status) bit to 64. MSS message indicates that a request has at least one cause. The table below shows the conditions for the MG3670B's status byte register.



Bit	Bit weight	Bit name	Status-byte-register conditions
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	ESB(ERR)	0 = Event status not generated 1 = Event status generated
2	4	ESB(END)	0 = Event status not generated 1 = Event status generated
1	2	—	0 = Not used
0	1	—	0 = Not used

***TST? Self-test Query**

(Returns self-test error results.)

■ Format***TST?****■ Example**

```

30 PRINT @ADR;"*TST?"
40 INPUT @ADR;NR1
50 PRINT NR1

```

■ Explanation

The *TST? query generates an internal self-test and places the test results into the output queue to indicate whether or not the device completed the self-test without detected errors. Operator intervention is not required to execute the self-test. The MG3670B returns the results of the self-test executed at power-on. A bit with detected errors is set to 1.

The MG3670B test range is as follows:

```

bit 0: CPU test
bit 1: ROM test
bit 2: Non-backup RAM test
bit 3: Backup RAM test
bit 4: ACRTC test
bit 5: GPIB test
bit 6: Programmable timer test
bit 7: Burst-unit ROM test
bit 8: Burst-unit RAM test
bit 9 to 15:(Not used)

```

■ Response message

The response message is sent by <NR1 numeric response data>. The data range is 0 to 4095.

NR1 = 0 Indicates that self-test completed without errors detected

NR1 ≠ 0 Indicates that self-test not completed or completed with errors detected

*WAI Wait-to-Continue Command

(Keeps the next command on stand-by if the device is currently executing a command)

■ Format

*WAI

■ Example

```
PRINT @ADR;"*WAI"
```

■ Explanation

The *WAI common command executes overlap commands as sequential commands.

An command or query (sent from the controller to a device) is called an overlap command if the next command can start execution while it is executing some function in the device.

Executing the *WAI command (after an overlap command) set the next command on hold and permits it to execute its function once the first command has finished. This is the same as sequential commands.

However, since overlap commands are not available with the MG3670B, so this command is not necessary.

SECTION 7 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

7.1	IEEE488.2 Standard Status Model	7-3
7.2	Status Byte (STB) Register	7-5
7.2.1	ESB and MAV summary messages	7-5
7.2.2	Device-dependent summary messages	7-6
7.2.3	Reading and clearing STB register	7-7
7.3	Service Request (SRQ) Enabling Operation	7-8
7.4	Standard Event Status Register	7-9
7.4.1	Bit definition of Standard Event Status Register	7-9
7.4.2	Reading, writing to, and clearing Standard Event Status Register	7-10
7.4.3	Reading, writing to, and clearing Standard Event Status Enable Register	7-10
7.5	Extended Event Status Register	7-11
7.5.1	Bit definition of Extended Event Status Register	7-12
7.5.2	Bit definition of Extended END Event Status Register	7-13
7.5.3	Reading, writing to, and clearing Extended Event Status Register	7-14
7.5.4	Reading, writing to, and clearing Extended Event Status Enable Register	7-14
7.6	Techniques for Synchronizing MG3670B with a Controller	7-15
7.6.1	Wait for response after *OPC? query is sent	7-15
7.6.2	Wait for service request after *OPC is sent	7-16

TABLE OF CONTENTS

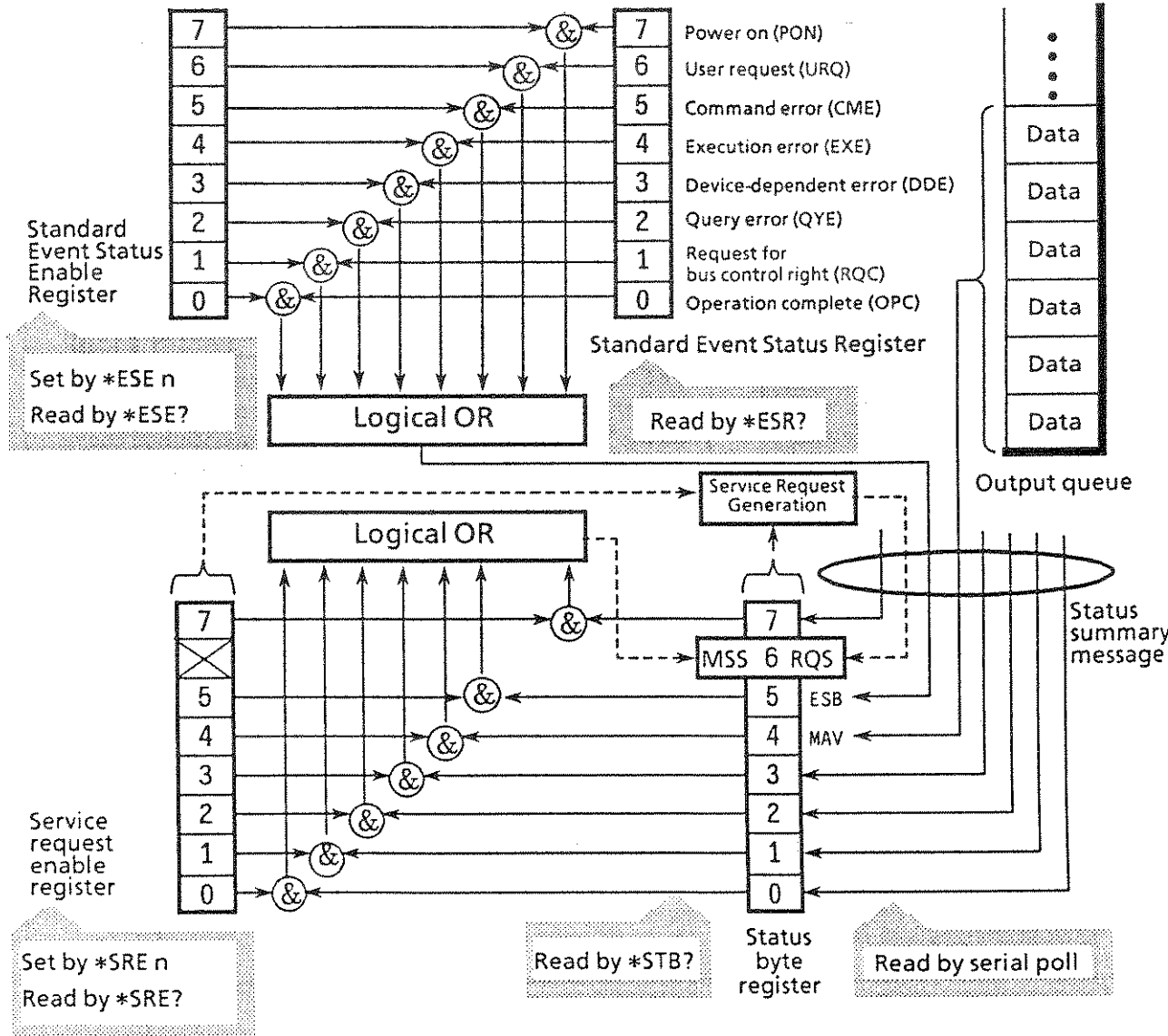
(Blank)

SECTION 7 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.

7.1 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
<p>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.</p>	<p>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.</p>

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"> ① Power on ② User request ③ Command error ④ Execution error ⑤ Device-dependent error ⑥ Query error ⑦ Request for bus control right ⑧ Operation complete <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>

7.2 Status Byte (STB) Register

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

7.2.1 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.

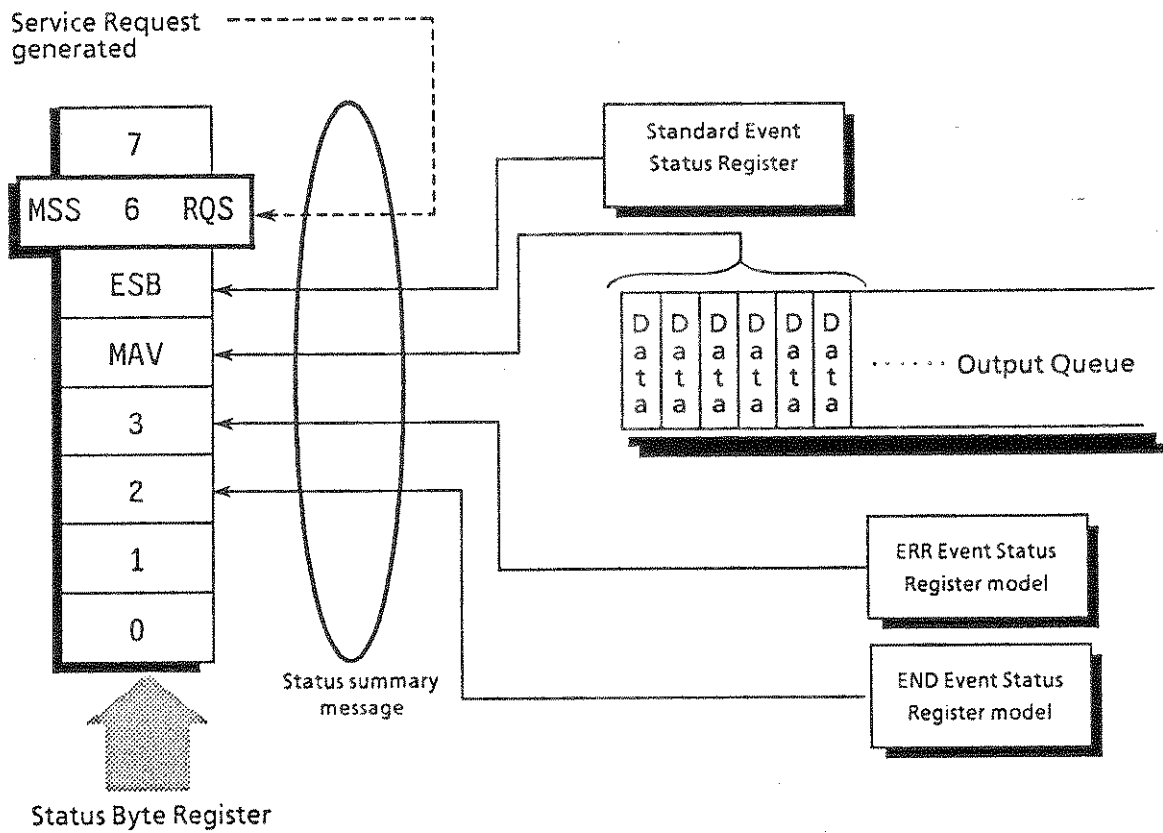
This bit becomes 0 when the ESR register is read out by the ESR? query or when cleared by the *CLS command.

(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.

7.2.2 Device-dependent summary messages

As shown below, the MG3670B does not use bits 0, 1, and 7, and uses bits 2 and 3 as the summary bit of the Event Status Registers.



7.2.3 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.

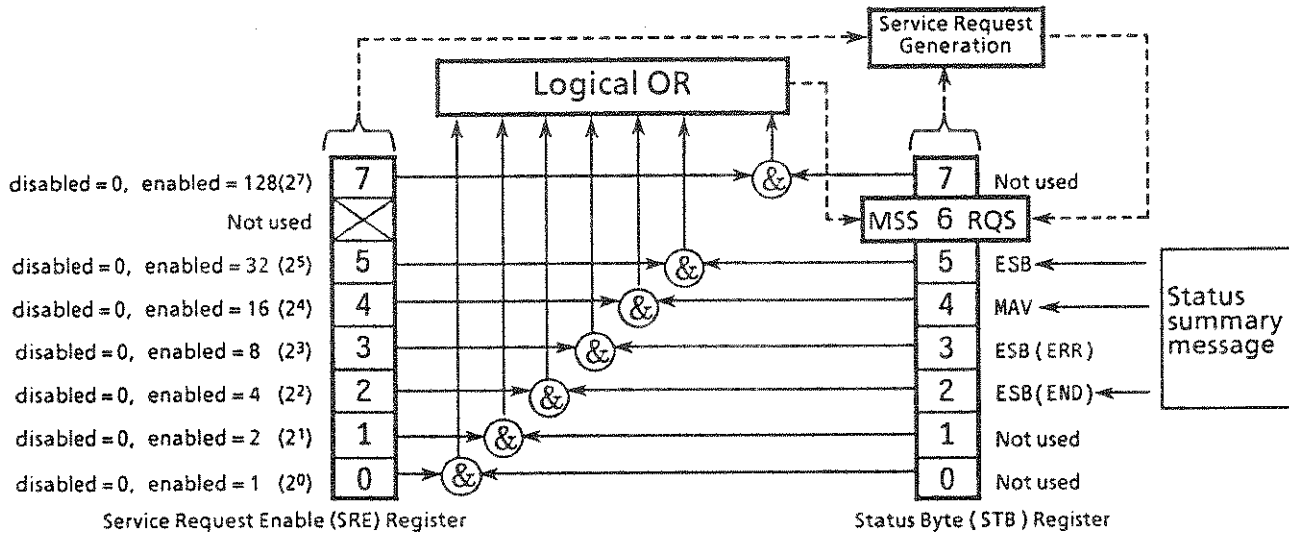
(4) Clearing the STB register by the *CLS common command

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.

7.3 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.

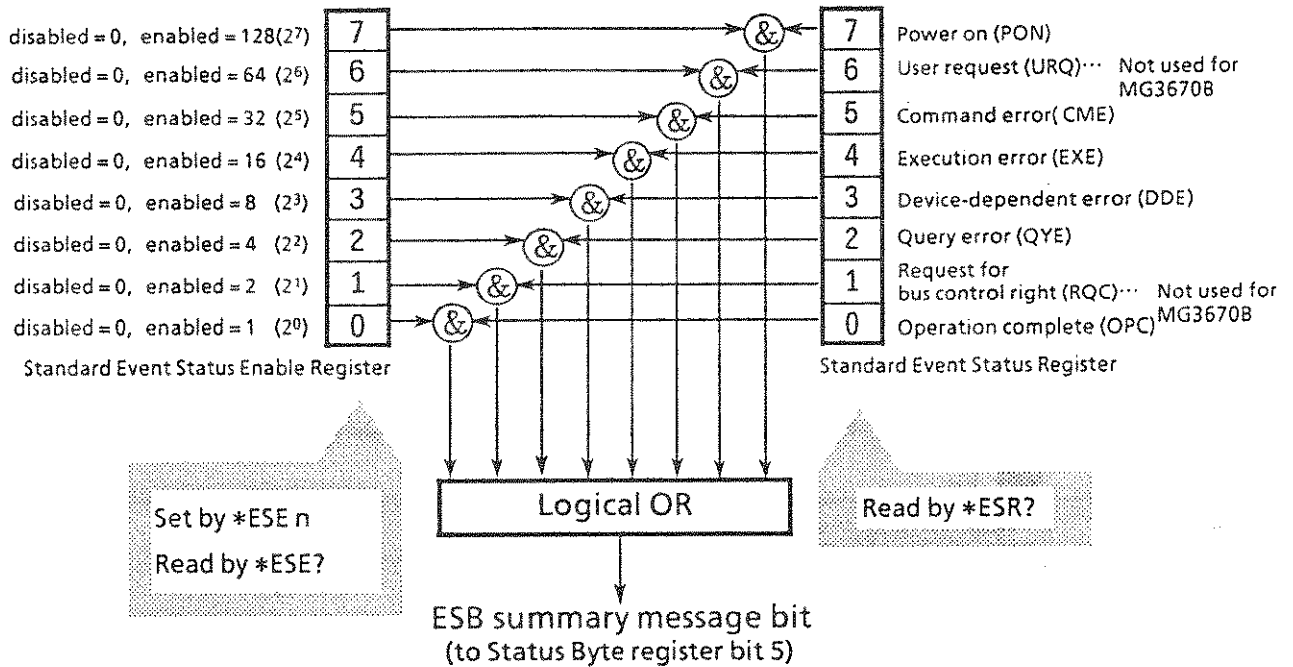
(2) Updating the SRE register

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.

7.4 Standard Event Status Register

7.4.1 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

7.4.2 Reading, writing to, and clearing Standard Event Status Register

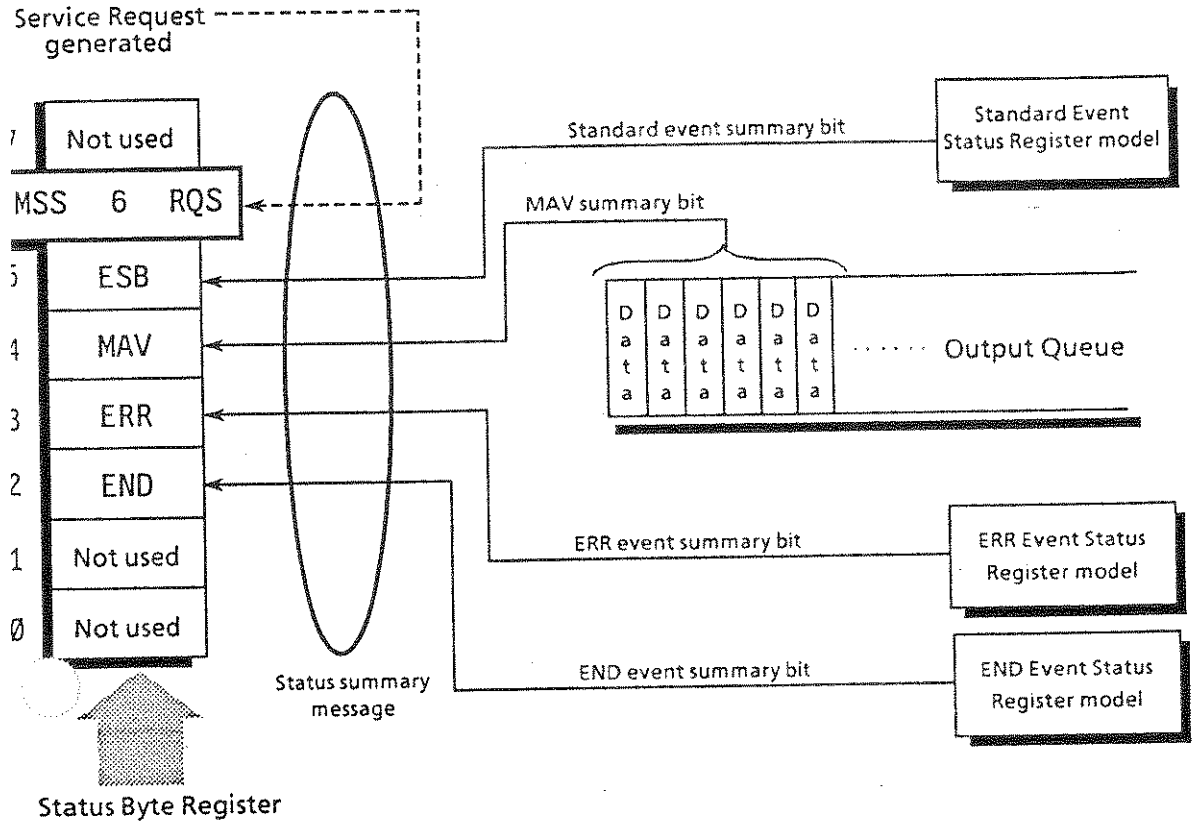
Reading	The *ESR? common query is used to read the ESR register, which is cleared after being read. The response message is integer-format data for which the sum of the binary-weighted event bit is converted to decimals.
Writing	With the exception of clearing, it is impossible to write to the register from outside
Clearing	The register is cleared in the following cases. <ul style="list-style-type: none"> 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

7.4.3 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. <ul style="list-style-type: none"> ① When an *ESE command with a data value of 0 is received ② When the power is turned on <p>The Standard Event Status Enable Register is not affected by the following.</p> <ul style="list-style-type: none"> 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

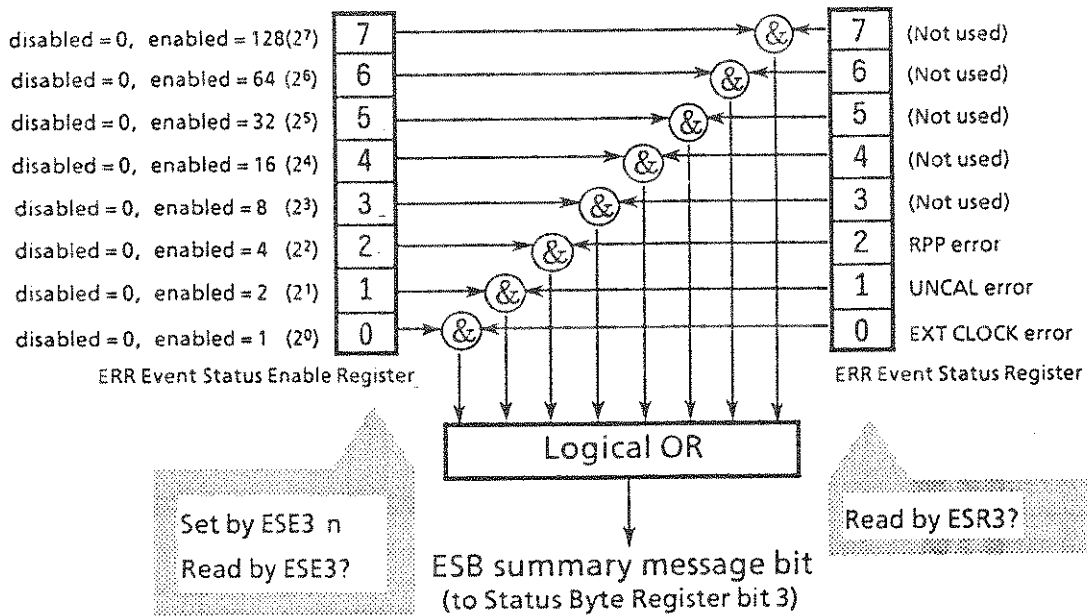
Extended Event Status Register

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



7.5.1 Bit definition of Extended ERR Event Status Register

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

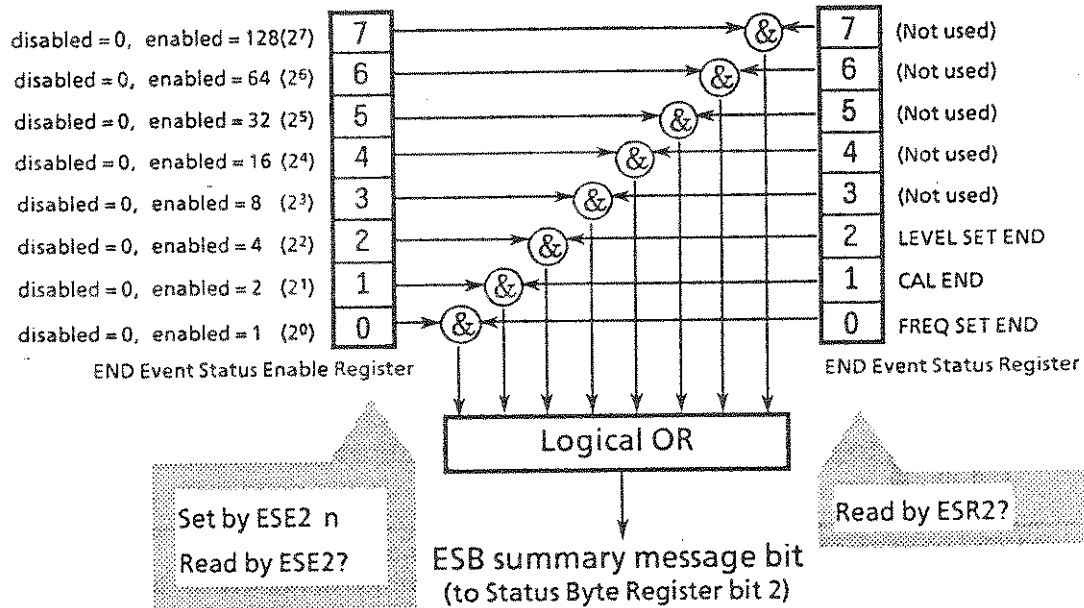


The ERR Event Status Enable Register on the left is used to select which bits in the corresponding Event Register will cause a TRUE summary message when set.

Bit	Event name	Description
7	(Not used)	(Not used)
6	(Not used)	(Not used)
5	(Not used)	(Not used)
4	(Not used)	(Not used)
3	(Not used)	(Not used)
2	RPP error	Sets 1 when the RPP (Reverse Power Protection) circuit has operated
1	UNCAL error	Sets 1 when output level has reached to an UNCAL level
0	EXT CLOCK error	Sets 1 when external baseband clock input signal improper during external data clock selection

7.5.2 Bit definition of Extended END 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	(Not used)	(Not used)
4	(Not used)	(Not used)
3	(Not used)	(Not used)
2	LEVEL SET END	Sets 1 when output level setting completed
1	CAL END	Sets 1 when output level calibration completed
0	FREQ SET END	Sets 1 when frequency setting completed

7.5.3 Reading, writing to, and clearing Extended Event Status Register

Reading	The ESR2? or ESR3? 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. ① When a *CLS command is received ② When the power is turned on ③ An event is read for the ESR2? or ESR3? query command

7.5.4 Reading, writing to, and clearing Extended Event Status Enable Register

Reading	The register is read by the ESE2? or ESE3? 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? or ESE3? 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. ① When an ESE2 or ESE3 program command with a data value of 0 is received ② When the power is turned on The Extended 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

7.6 Techniques for Synchronizing MG3670B with a Controller

The MG3670B 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 MG3670B and the controller.

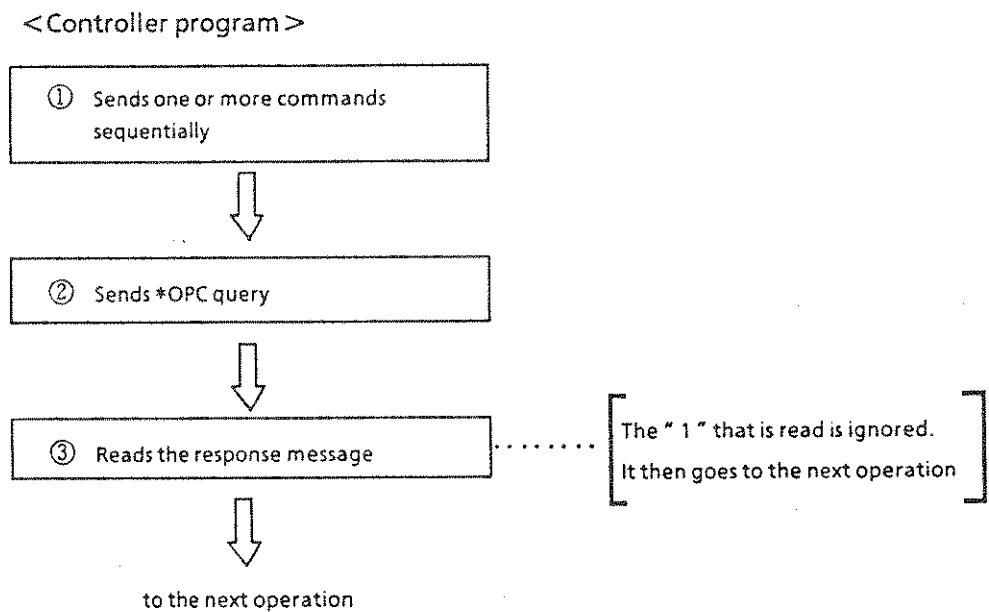
If the controller controls one or more devices and synchronizes with them, after all the commands specified for the MG3670B have been processed, the next commands must be sent to other devices.

There are two ways of synchronizing the MG3670B with the controller.

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

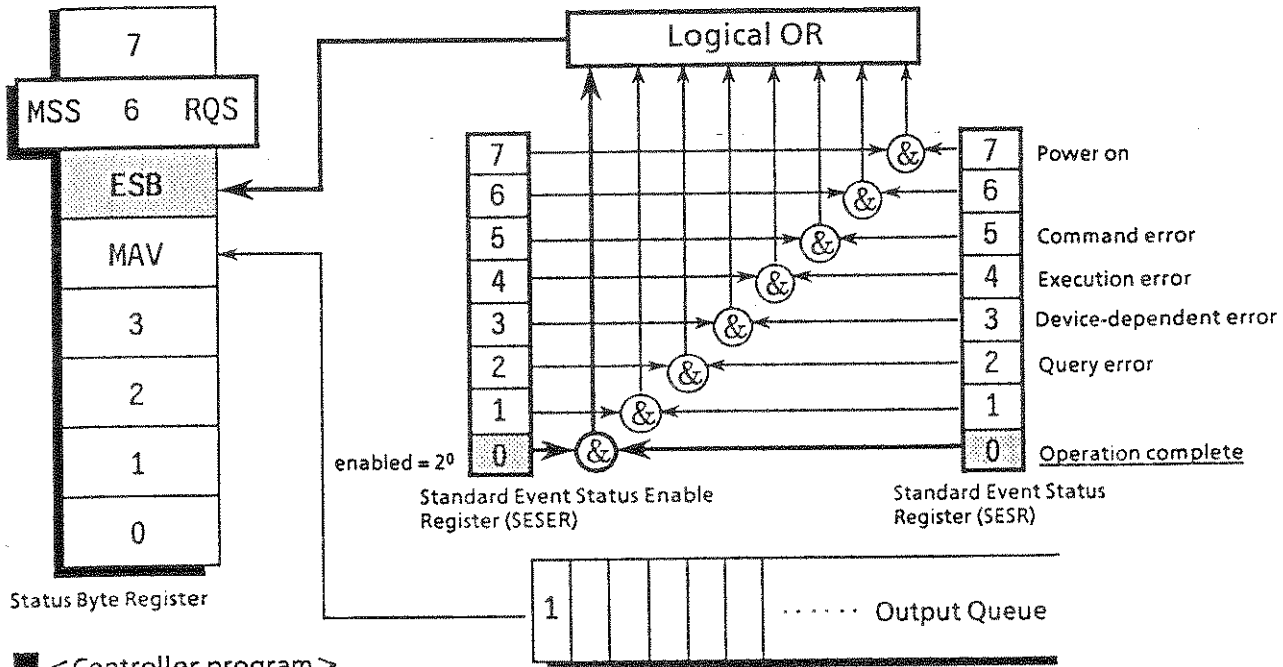
7.6.1 Wait for response after *OPC? query is sent

The MG3670B outputs "1" as the response message when executing the *OPC? query command. The controller is synchronized with the MG3670B by waiting for the response message to be entered.



7.6.2 Wait for service request after *OPC is sent

The MG3670B sets the operation-complete bit (bit 0) to 1 when executing the *OPC command. The controller is synchronized with the MG3670B by waiting for SRQ when the operation-complete bit is set for SRQ.



■ < Controller program >

- ① Enables the 2⁰ bit of the Standard Event Status Enable Register PRINT @1; "*ESE 1"
- ↓
- ② Enables the 2⁵ bit of the Service Request Enable Register PRINT @1; "*SRE 32"
- ↓
- ③ Makes the device execute the specified operation
- ↓
- ④ Sends the *OPC command PRINT @1; "*OPC"
- ↓
- ⑤ Wait for an SRQ interrupt (ESB summary message) ... Value of status byte: 2⁶ + 2⁵ = 96

SECTION 8

DETAILS OF MG3670B DEVICE MESSAGES

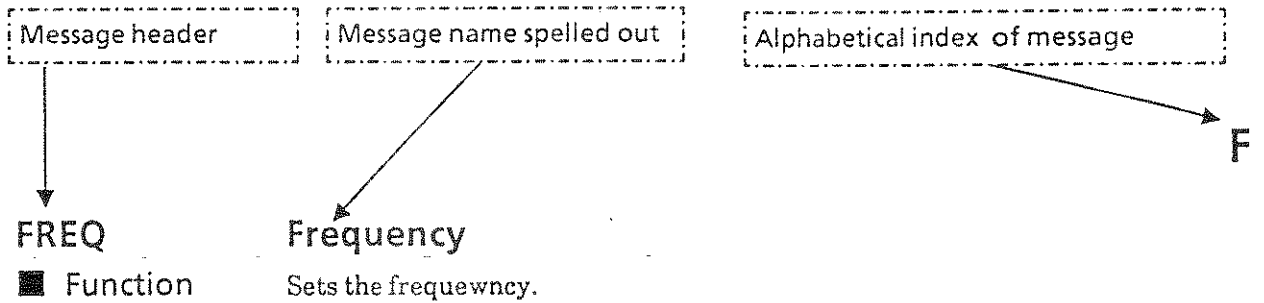
This section lists the device messages specific to the MG3670B, which are given in Section 2, in alphabetical order and also lists the messages compatible with the MG3633A in functional types to explain the detailed functions of each device message.

TABLE OF CONTENTS

8.1	Device Messages Specific to the MG3670B	8-4	FRL	8-20
	BASE	8-5	FRLR?	8-20
	BTI	8-5	FRLV	8-21
	BTO	8-6	FRR	8-21
	BURST	8-6	FRS	8-22
	BUZ	8-7	FSAV	8-22
	CAL	8-8	HEAD	8-23
	CAPL	8-8	IFRF	8-24
	CHECK	8-9	INTFC	8-24
	COS	8-9	IQL	8-25
	DSPL	8-10	ITR	8-25
	EIB	8-11	LVL	8-26
	EIC	8-11	MIC	8-27
	EID	8-12	MID	8-27
	EIS	8-12	MOD	8-28
	EOB	8-13	OCNT	8-29
	EOC	8-13	OIS	8-29
	EOD	8-14	OLDBM	8-30
	EOS	8-14	OLDBU	8-30
	ESE2	8-15	OLK	8-31
	ESE3	8-15	OLL	8-31
	ESR2?	8-16	OLM	8-32
	ESR3?	8-16	OLR	8-32
	FIS	8-17	OLS	8-33
	FOF	8-17	OLV	8-33
	FOS	8-18	OLVL	8-34
	FRCL	8-18	OOF	8-34
	FREQ	8-19	OOS	8-35
	FRK	8-19	ORL	8-35

ORLR?	8-36	OLDBU, OLDU, APDBU, APDU	8-71
ORLV?	8-36	OLV, APV	8-72
OTR	8-37	OLM	8-73
PM	8-38	SP06, SP05	8-74
PMP	8-38		
PRCL	8-39	8.2.3 Save/recall	8-75
PRE	8-39	FQaST	8-76
PRMTR	8-40	FQaRC	8-77
PSAV	8-40	FNaST	8-78
PSYNC	8-41	FNaRC	8-79
REF	8-42		
RS	8-42	8.2.4 Others	8-80
TRM	8-43	RS	8-81
VDSPL	8-44	SP00	8-82
		SP02, SP01	8-83
		SP60, SP61	8-84
8.2 Control by MG3633A			
Device Messages	8-45		
8.2.1 Frequency	8-45		
FR, FC, CF	8-46		
FIS	8-47		
UFR, DFR	8-48		
TFR, EFR	8-49		
R2, R3, R4, R5, R6, R7, R8, R9,			
FSR, FSL	8-50		
REFOA	8-51		
RLFOA	8-52		
FO, FF	8-53		
FOS	8-54		
SP12, SP11	8-55		
8.2.2 Output level	8-56		
OL, AP	8-57		
OIS	8-58		
UOL, DOL	8-59		
TOL, EOL	8-60		
L0, L1, L2, OSR, OSL	8-61		
LC, LN	8-62		
REOOA	8-63		
RLOOA	8-64		
LO, LF	8-65		
OOS	8-66		
SP08, SP07	8-67		
RO, RF	8-68		
SP03, SP04	8-69		
OLDBM, OLDM, APDBM, APDM	8-70		

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



Header	Program command	Query	Response
FREQ	FREQ_a	FREQ?	FREQ_a (a=0 to 2250 000 000 Hz)

- Value of a 0 Hz to 2.25 GHz
 - Suffix code HZ: Hz
 - KHZ, KZ: kHz
 - MHZ, MZ: MHz
 - GHZ, GZ: GHz
 - None: Hz
- The left side of the colon is part of the program or response data
 ● The right side of the colon describes the data
- Initial setting 10MHZ
 - Example FREQ_123MHZ

8.1 Device Messages Specific to the MG3670B

BASE Base Band Setup

- **Function** Switches to the base band setup screen.
(When the modulation unit is supported, this function is available.)

Header	Program command	Query	Response
BASE	BASE	—	—

- **Suffix code** None
- **Example** BASE

BTI Burst-Trigger Input Edge

- **Function** Sets the burst trigger input edge.
(When the modulation unit and the burst function unit are supported, this function is available)

Header	Program command	Query	Response
BTI	BTI_a	BTI?	BTI_a

- **Value of a** RISE: Rising edge
FALL: Falling edge
- **Suffix code** None
- **Initial setting** RISE
- **Example** BTI_FALL

B

BTO Burst-Trigger Output Edge

- **Function** Sets the burst trigger output edge.
(When the modulation unit and the burst function unit are supported, this function is available)

Header	Program command	Query	Response
BTO	BTO_a	BTO?	BTO_a

- **Value of a** RISE: Rising edge
FALL: Falling edge
- **Suffix code** None
- **Initial setting** RISE
- **Example** BTO_FALL

BURST Burst Pattern

- **Function** Switches to the burst internal modulation data setting screen.
(When the modulation unit and the burst function unit are supported, this function is available)

Header	Program command	Query	Response
BURST	BURST	—	—

- **Suffix code** None
- **Example** BURST

BUZ Buzzer On / Off

■ Function Turns the buzzer switch On or Off.

Header	Program command	Query	Response
BUZ	BUZ_a	—	—

■ Value of a ON: On
OFF: Off

■ Suffix code None

■ Initial setting ON

■ Example BUZ_aOFF

C.

CAL Calibration

■ Function Calibrates output level.

Header	Program command	Query	Response
CAL	CAL	—	—

■ Suffix code None

■ Example CAL

CAPL CMOS Amplitude

■ Function Sets the CMOS level when I,Q signal level has been selected for CMOS.

Header	Program command	Query	Response
CAPL	CAPL_a	CAPL?	CAPL_a (a = 50 to 500 mVrms)

■ Value of a 50 to 500mVrms (50mVrms step)

■ Suffix code V: Vrms
MV: mVrms
UV: μ Vrms
None: mVrms

■ Initial setting 500MV

■ Example CAPL_50MV

CHECK Hardware Check

■ **Function** Switches to the hardware check screen.

Header	Program command	Query	Response
CHECK	CHECK	—	—

■ **Suffix code** None

■ **Example** CHECK

COS CMOS DC Offset

■ **Function** Sets the CMOS level DC offset value when I,Q signal level has been selected for CMOS.

Header	Program command	Query	Response
COS	COS_a	COS?	COS_a (a=0 to 4000 mV)

■ **Value of a** 0 to 4000mV (1mV step)

■ **Suffix code**
 V: V
 MV: mV
 UV: μ V
 None: mV

■ **Initial setting** 2500MV

■ **Example** COS_1000MV

D

DSPL Display On / Off

■ Function Turns the Display switch On or Off.

Header	Program command	Query	Response
DSPL	DSPL_a	_____	_____

■ Value of a ON: On
OFF: Off

■ Suffix code None

■ Initial setting ON

■ Example DSPL_OFF

EIB External-Modulation-Input Burst-Gate Polarity

- **Function** Sets the burst gate signal polarity of external modulation input data.
(When the modulation unit and the burst function unit are supported, this function is available)

Header	Program command	Query	Response
EIB	EIB_a	EIB?	EIB_a

- **Value of a** POS: Positive
NEG: Negative
- **Suffix code** None
- **Initial setting** POS
- **Example** EIB_POS

EIC External-Modulation-Input Data Clock Edge

- **Function** Sets the clock signal edge of external modulation input data.
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
EIC	EIC_a	EIC?	EIC_a

- **Value of a** RISE: Rising edge
FALL: Falling edge
- **Suffix code** None
- **Initial setting** RISE
- **Example** EIC_FALL

E

EID External-Modulation-Input Data Polarity

- **Function** Sets the data signal polarity of external modulation input data.
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
EID	EID_a	EID?	EID_a

- **Value of a** POS: Positive
NEG: Negative
- **Suffix code** None
- **Initial setting** POS
- **Example** EID_POS

EIS External-Modulation-Input Symbol Clock Edge

- **Function** Sets the symbol clock signal edge of external modulation input data.
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
EIS	EIS_a	EIS?	EIS_a

- **Value of a** RISE: Rising edge
FALL: Falling edge
- **Suffix code** None
- **Initial setting** RISE
- **Example** EIS_FALL

EOB External-Modulation-Output Burst-Gate Polarity

- **Function** Sets the burst gate signal polarity of external modulation output data.
(When the modulation unit and the burst function unit are supported, this function is available)

Header	Program command	Query	Response
EOB	EOB_a	EOB?	EOB_a

- **Value of a** POS: Positive
NEG: Negative
- **Suffix code** None
- **Initial setting** POS
- **Example** EOB_POS

EOC External-Modulation-Output Data Clock Edge

- **Function** Sets the data clock signal edge of external modulation output data.
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
EOC	EOC_a	EOC?	EOC_a

- **Value of a** RISE: Rising edge
FALL: Falling edge
- **Suffix code** None
- **Initial setting** RISE
- **Example** EOC_FALL

E

EOD External-Modulation-Output Data Polarity

- **Function** Sets the data signal polarity of external modulation output data.
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
EOD	EOD_a	EOD?	EOD_a

- **Value of a** POS: Positive
NEG: Negative
- **Suffix code** None
- **Initial setting** POS
- **Example** EOD_POS

EOS External-Modulation-Output Symbol Clock Edge

- **Function** Sets the symbol clock signal edge of external modulation output data.
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
EOS	EOS_a	EOS?	EOS_a

- **Value of a** RISE: Rising edge
FALL: Falling edge
- **Suffix code** None
- **Initial setting** RISE
- **Example** EOS_FALL

ESE2 Event Status Enable Register (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	ESE2?	ESE2_a (a=0 to 255)

- **Value of a** 0 to 255:
Represents the sum of weighted bit-digit values when the weighted value for bits to be enabled are selected from among the values $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$ or $2^7=128$; corresponding to the END Event Status Enable Register bits 0, 1, 2, 3, 4, 5, 6 or 7.
- **Suffix code** None
- **Example** ESE2_3

ESE3 Event Status Enable Register (ERR)

- **Function** Allows the ERR Event Status Enable Register to select which bit in the corresponding Event Register causes a TRUE ESB summary message bit 3 when set.

Header	Program command	Query	Response
ESE3	ESE3_a	ESE3?	ESE3_a (a=0 to 255)

- **Value of a** 0 to 255:
Represents the sum of weighted bit-digit values when the weighted value for bits to be enabled are selected from among the values $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$ or $2^7=128$; corresponding to the ERR Event Status Enable Register bits 0, 1, 2, 3, 4, 5, 6 or 7.
- **Suffix code** None
- **Example** ESE3_5

E

ESR2? Event Status Register (END)

- **Function** Allows the sum of the binary-weighted event bit value 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?	ESR2_ a (a = 0 to 255)

- **Value of a** 0 to 255
- **Suffix code** None
- **Example** ESR2?

ESR3? Event Status Register (ERR)

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

Header	Program command	Query	Response
ESR3?	_____	ESR3?	ESR3_ a (a = 0 to 255)

- **Value of a** 0 to 255
- **Suffix code** None
- **Example** ESR3?

FIS Frequency-Incremental-Step Value

■ **Function** Sets the frequency incremental step value

Header	Program command	Query	Response
FIS	FIS_a	FIS?	FIS_a (a = 1 to 2250 000 000 Hz)

■ **Value of a** 1Hz to 2.25GHz

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

■ **Initial setting** 1MHZ

■ **Example** FIS_12.5KHZ

FOF Frequency-Offset On / Off

■ **Function** Turns the frequency offset mode On or Off

Header	Program command	Query	Response
FOF	FOF_a	FOF?	FOF_a

■ **Value of a** ON: On
 OFF: Off

■ **Suffix code** None

■ **Initial setting** OFF

■ **Example** FOF_ON

F

FOS Frequency-Offset Value

■ Function Sets the frequency Offset value

Header	Program command	Query	Response
FOS	FOS_a	FOS?	FOS_a (a = -2250 000 000 to 2250 000 000 Hz)

■ Value of a -2.25 to 2.25GHz

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

■ Initial setting 0HZ

■ Example FOS_25KHZ

FRCL Frequency-Memory Recall

■ Function Recalls frequency from frequency memory.

Header	Program command	Query	Response
FRCL	FRCL_a	---	---

■ Value of a 0 to 999

■ Suffix code None

■ Example FRCL_15

FREQ **Frequency**

- **Function** Sets the frequency.

Header	Program command	Query	Response
FREQ	FREQ_a	FREQ?	FREQ_a (a = 0 to 2250 000 000 Hz)

- **Value of a** 0 Hz to 2.25 GHz
- **Suffix code** HZ: Hz
 KHZ, KZ: kHz
 MHZ, MZ: MHz
 GHZ, GZ: GHz
 None: Hz
- **Initial setting** 10MHz
- **Example** FREQ_123MHz

FRK **Frequency Rotary-Knob Up / Down**

- **Function** Increases (or decreases) the frequency by frequency Resolution.

Header	Program command	Query	Response
FRK	FRK_a	---	---

- **Value of a** UP: Up
 DN: Down
- **Suffix code** None
- **Example** FRK_UP

F**FRL****Frequency-Relative On / Off**

- **Function** Turns the relative frequency mode On or Off.

Header	Program command	Query	Response
FRL	FRL_a	FRL?	FRL_a

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

FRLR?**Frequency-Relative Reference Value**

- **Function** Reads out the reference frequency of relative frequency mode.

Header	Program command	Query	Response
FRLR?	_____	FRLR?	FRLR_a (a = 0 to 2250 000 000 Hz)

- **Value of a** 0 Hz to 2.25 GHz
- **Suffix code** HZ: Hz
- **Example** FRLR?

FRLV Frequency-Relative Value

■ Function Reads out the relative frequency.

Header	Program command	Query	Response
FRLV	_____	FRLV?	FRLV_a (a = -2250 000 000 to 2250 000 000 Hz)

■ Value of a -2.25 to 2.25 GHz

■ Suffix code HZ: Hz

■ Example FRLV?

FRR Frequency Resolution

■ Function Sets the frequency resolution.

Header	Program command	Query	Response
FRR	FRR_a	FRR?	FRR_a

■ Value of a

1HZ:	1 Hz
10HZ:	10 Hz
100HZ:	100 Hz
1KHZ:	1 kHz
10KHZ:	10 kHz
100KHZ:	100 kHz
1MHZ:	1 MHz
10MHZ:	10 MHz
100MHZ:	100 MHz
1GHZ:	1 GHz
R:	Shifts the frequency resolution to right digit
L:	Shifts the frequency resolution to left digit

■ Suffix code None

■ Initial setting 1HZ

■ Example FRR_1KHZ

F

FRS Frequency-Incremental-Step Up / Down

■ **Function** Increases (or decreases) the frequency by frequency incremental step value.

Header	Program command	Query	Response
FRS	FRS_a	—	—

■ **Value of a** UP: Up
DN: Down

■ **Suffix code** None

■ **Example** FRS_UP

FSAV Frequency-Memory Save

■ **Function** Saves frequency to frequency memory.

Header	Program command	Query	Response
FSAV	FSAV_a	—	—

■ **Value of a** 0 to 999

■ **Suffix code** None

■ **Example** FSAV_15

HEAD Header On / Off

■ **Function** Selects whether or not to add header to response message.

Header	Program command	Query	Response
HEAD	HEAD_a	—	—

■ **Value of a** ON: On
 OFF: Off

■ **Suffix code** None

■ **Initial setting** OFF

■ **Example** HEAD_ON

IFRF

IF / RF Setup

- **Function** Switches to IF/RF setup screen.

Header	Program command	Query	Response
IFRF	IFRF	—	—

- **Suffix code** None

- **Example** IFRF

INTFC

Interface Setup

- **Function** Switches to interface setup screen.

Header	Program command	Query	Response
INTFC	INTFC	—	—

- **Suffix code** None

- **Example** INTFC

IQL I, Q Level

- **Function** Sets the I,Q signal levels.

Header	Program command	Query	Response
IQL	IQL_a	IQL?	IQL_a

- **Value of a** 500MV: 50 Ω 500 mVrms
CMOS: CMOS level

- **Suffix code** None
- **Initial setting** 500MV
- **Example** IQL_CMOS

ITR I, Q Input Trim On / Off

- **Function** Turns on or off the offset and balance adjustment function of I,Q input signals.

Header	Program command	Query	Response
ITR	ITR_a	ITR?	ITR_a

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

L .

LVL Output-Level On / Off

■ **Function** Turns the output level on or off.

Header	Program command	Query	Response
LVL	LVL_a	LVL?	LVL_a

■ **Value of a** ON: On
OFF: Off

■ **Suffix code** None

■ **Initial setting** ON

■ **Example** LVL_OFF

MIC Modulation Input Data Clock

- **Function** Sets the modulation data clock signal to either internal (Int) or external (Ext).
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
MIC	MIC_a	MIC?	MIC_a

- **Value of a** INT: Internal data clock signal
EXT: External data clock signal
- **Suffix code** None
- **Initial setting** INT
- **Example** MIC_EXT

MID Modulation Input Data

- **Function** Sets the modulation data signal to either internal (Int) or external (Ext).
(When the modulation unit is supported, this function is available)

Header	Program command	Query	Response
MID	MID_a	MID?	MID_a

- **Value of a** INT: Internal data signal
EXT: External data signal
- **Suffix code** None
- **Initial setting** INT
- **Example** MID_EXT

M.

MOD Modulation On / Off

■ Function Turns the modulation function on or off.

Header	Program command	Query	Response
MOD	MOD_a	MOD?	MOD_a

■ Value of a ON: On
OFF: Off

■ Suffix code None

■ Initial setting OFF

■ Example MOD_ON

OCNT Output-Level Continuous-Mode On / Off

■ Function Turns the output level continuous mode on or off.

Header	Program command	Query	Response
OCNT	OCNT_a	OCNT?	OCNT_a

■ Value of a ON: On
OFF: Off

■ Suffix code None

■ Initial setting OFF

■ Example OCNT_ON

OIS Output-Level-Incremental-Step Value

■ Function Sets the output level incremental step value.

Header	Program command	Query	Response
OIS	OIS_a	OIS?	OIS_a (a=0.1 to 156 dB)

■ Value of a 0.1 to 156 dB (0.1dB step)

■ Suffix code DB: dB

■ Initial setting 1DB

■ Example OIS_60DB

O

OLDBM Output-Level Unit to dBm

■ Function Changes output level units to dBm.

Header	Program command	Query	Response
OLDBM	OLDBM	—	—

■ Suffix code None

■ Example OLDBM

OLDBU Output-Level Unit to dB μ

■ Function Changes output level units to dB μ .

Header	Program command	Query	Response
OLDBU	OLDBU	—	—

■ Suffix code None

■ Example OLDBU

OLK Output-Level Rotary-Knob Up / Down

■ Function Increase(or decrease) the output level by output level Resolution.

Header	Program command	Query	Response
OLK	OLK_a	_____	_____

- Value of a UP: Up
DN: Down
- Suffix code None
- Example OLK_UP

OLL Output-Level-Limit On / Off

■ Function Turns the output level limit mode on or off.

Header	Program command	Query	Response
OLL	OLL_a	OLL?	OLL_a

- Value of a ON: On
OFF: Off
- Suffix code None
- Initial setting OFF
- Example OLL_ON

O.

OLM Output-Level-Limit Value

■ Function Sets the output level limit value.

Header	Program command	Query	Response
OLM	OLM_a	OLM?	OLM_a (a: Transfer the unit which is represented in the current output level units.)

■ Value of a

When unit is DBM,DM: -143 to 13 dBm

When unit is DBU,DU: -30 to 126 dB μ ---open voltage
-36 to 120 dB μ ---terminated voltage

When unit is V,MV,UV: 0.032 μ V to 2 V ---open voltage
0.016 μ V to 1 V ---terminated voltage

■ Suffix code

DBM, DM: dBm
DBU, DU: dB μ
V: V
MV: mV
UV: μ V
None: dBm

■ Initial setting -10DBM

■ Example OLM_0DBM

OLR Output-Level Resolution

■ Function Sets the output level resolution.

Header	Program command	Query	Response
OLR	OLR_a	OLR?	OLR_a

■ Value of a

0.1DB: 0.1 dB
1DB: 1 dB
10DB: 10 dB
R: Shifts the output level resolution to right digit
L: Shifts the output level resolution to left digit

■ Suffix code None

■ Initial setting 0.1DB

■ Example OLR_1DB

OLS Output-Level-Incremental-Step Up / Down

■ Function Increases (or decreases) the output level by output level incremental step value.

Header	Program command	Query	Response
OLS	OLS_a	—	—

■ Value of a UP: Up
DN: Down

■ Suffix code None

■ Example OLS_UP

OLV Output-Level Unit to volt

■ Function Changes output level units to volt.

Header	Program command	Query	Response
OLV	OLV	—	—

■ Suffix code None

■ Example OLV

O

OLVL Output Level

■ Function Sets the output level.

Header	Program command	Query	Response
OLVL	OLVL_a	OLVL?	OLVL_a (a:Transfer the unit which is represented in the current output level units.)

■ Value of a

When unit is DBM,DM: -143 to 13 dBm

When unit is DBU,DU: -30 to 126 dB μ ---open voltage
-36 to 120 dB μ ---terminated voltage

When unit is V,MV,UV: 0.032 μ V to 2 V ---open voltage
0.016 μ V to 1 V ---terminated voltage

■ Suffix code

DBM, DM: dBm
DBU, DU: dB μ
V: V
MV: mV
UV: μ V
None: dBm

■ Initial setting -30DBM

■ Example OLVL_a-60DBM

OOF Output-Level-Offset On / Off

■ Function Turns the output level offset mode on or off

Header	Program command	Query	Response
OOF	OOF_a	OOF?	OOF_a

■ Value of a

ON: On
OFF: Off

■ Suffix code None

■ Initial setting OFF

■ Example OOF_aON

OOS Output-Level-Offset Value

■ Function Sets the output level offset value

Header	Program command	Query	Response
OOS	OOS_a	OOS?	OOS_a (a = -55.0 to 55.0 dB)

■ Value of a - 55 to 55 dB (0.1dB step)

■ Suffix code DB: dB
None: dB

■ Initial setting 0dB

■ Example OOS_3DB

ORL Output-Level-Relative On / Off

■ Function Turns the relative output level mode on or off.

Header	Program command	Query	Response
ORL	ORL_a	ORL?	ORL_a

■ Value of a ON: On
OFF: Off

■ Suffix code None

■ Initial setting OFF

■ Example ORL_ON

O.

ORLR? Output-Level-Relative Reference Value

■ **Function** Reads out the reference output level of relative output level mode.

Header	Program command	Query	Response
ORLR?	_____	ORLR?	ORLR_ <u>a</u> (a: Transfer the unit which is represented in the current output level units.)

■ **Value of a**

When unit is DBM,DM:	- 143 to 13 dBm
When unit is DBU,DU:	- 30 to 126 dB μ ---open voltage - 36 to 120 dB μ ---terminated voltage
When unit is V,MV,UV:	0.032 μ V to 2 V ---open voltage 0.016 μ V to 1 V ---terminated voltage

■ **Suffix code**

DBM: dBm
 DBU: dB μ
 V: V
 MV: mV
 UV: μ V

■ **Example** ORLR?

ORLV? Output-Level-Relative Value

■ **Function** Reads out the relative output level.

Header	Program command	Query	Response
ORLV?	_____	ORLV?	ORLV_ <u>a</u>

■ **Value of a** - 156 to 156 dB (0.1dB step)

■ **Suffix code** DB: dB

■ **Example** ORLV?

OTR I, Q Output Trim On / Off

■ **Function** Turns on or off the offset and balance adjustment function of I,Q output signals.

Header	Program command	Query	Response
OTR	OTR_a	OTR?	OTR_a

■ **Value of a** ON: On
OFF: Off

■ **Suffix code** None

■ **Initial setting** OFF

■ **Example** OTR_ON

P

PM Pulse-Modulation

- **Function** Sets the pulse modulation signal to either internal (Int) or external (Ext).

Header	Program command	Query	Response
PM	PM_a	PM?	PM_a

- **Value of a** INT: Internal pulse modulation signal
EXT: External pulse modulation signal
- **Suffix code** None
- **Initial setting** INT
- **Example** PM_EXT

PMP Pulse-Modulation Polarity

- **Function** Sets the pulse modulation signal polarity.

Header	Program command	Query	Response
PMP	PMP_a	PMP?	PMP_a

- **Value of a** POS: Positive
NEG: Negative
- **Suffix code** None
- **Initial setting** POS
- **Example** PMP_POS

PRCL Parameter-Memory Recall

- Function Recalls parameter from parameter memory.

Header	Program command	Query	Response
PRCL	PRCL_a	_____	_____

- Value of a 0 to 99
- Suffix code None
- Example PRCL_15

PRE Preset

- Function Initializes parameters.

Header	Program command	Query	Response
PRE	PRE	_____	_____

- Suffix code None
- Example PRE

P

PRMTR Parameter

- **Function** Switches to parameter setting screen.

Header	Program command	Query	Response
PRMTR	PRMTR	—	—

- **Suffix code** None
- **Example** PRMTR

PSAV Parameter-Memory Save

- **Function** Saves parameter to parameter memory.

Header	Program command	Query	Response
PSAV	PSAV_a[, b]	—	—

- **Value of a** 0 to 99
- **Value of b** Character string within 8 characters enclosed by double quotation marks or single quotation marks.
Saves the title which sets currently if b is omitted.
Saves the title which is the first eight characters if b is over eight characters.
- **Suffix code** None
- **Example** PSAV_5, "MG3670A"

PSYNC Pattern Sync Output

- **Function** Selects the pattern sync output signal.
(When the modulation unit and the burst function unit are supported, this function is available)

Header	Program command	Query	Response
PSYNC	PSYNC_a	PSYNC?	PSYNC_a

- **Value of a** PNCLK: PN Clock
PNGAT: PN Gate
RFGAT: RF Gate
- **Suffix code** None
- **Initial setting** PNCLK
- **Example** PSYNC_PNGAT

R

REF Reference Frequency

■ Function Selects the reference frequency.

Header	Program command	Query	Response
REF	REF_a	REF?	REF_a

■ Value of a 10MHZ: 10 MHz
13MHZ: 13 MHz

■ Suffix code None

■ Initial setting 10MHZ

■ Example REF_13MHZ

RS RPP Reset

■ Function Reset the RPP(Revers Power Protection) circuit to return to the original state.

Header	Program command	Query	Response
RS	RS	—	—

■ Suffix code None

■ Example RS

TRM Terminator

■ **Function** Sets the terminator of the response data sent out from GPIB.

Header	Program command	Query	Response
TERM	TERM_a	TERM?	TERM_a

- **Value of a** 0: LF
 1: CR/LF
- **Suffix code** None
- **Initial setting** 0 (but not to be initialized)
- **Example** TERM_1

V

VDSPL Voltage Display

■ **Function** Selects the output level display for open voltage or terminated voltage .

Header	Program command	Query	Response
VDSPL	VDSPL_a	VDSPL?	VDSPL_a

■ **Value of a** EMF: Electro-Motive Force (open voltage display)
TERM: Terminated (terminated voltage display)

■ **Suffix code** None

■ **Initial setting** EMF

■ **Example** VDSPL_TERM

8.2 Control by MG3633A Device Messages

This paragraph describes MG3633A Device Messages, which can be used for MG3670B, calssified by the function type.

Note: The query message has a trailing "OA" that is always added at the end of the program header. There must be a space between program header and program data.

8.2.1 Frequency

- FR Frequency
- FC Frequency Center
- CF Center Frequency
- Function Sets the frequency.

Header	Program command	Query	Response
FR	FR_a	FROA	FR_a
FC	FC_a	FCOA	FR_a
CF	CF_a	CFOA	FR_a
			(a = 0 to 2250 000 000 Hz)

- Value of a 0 Hz to 2.25 GHz
- Suffix code
 - HZ: Hz
 - KHZ, KZ: kHz
 - MHZ, MZ: MHz
 - GHZ, GZ: GHz
 - None: Hz
- Initial setting 10MHZ
- Example FR_123MHZ

FIS Frequency Incremental Step Value

■ **Function** Sets the frequency incremental step value.

Header	Program command	Query	Response
FIS	FIS_a	FISOA	FIS_a (a = 1 to 2250 000 000 Hz)

■ **Value of a** 1 Hz to 2.25 GHz

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

■ **Initial setting** 1MHZ

■ **Example** FIS_12.5KHZ

UFR **Frequency Incremental Step Up**
DFR **Frequency Incremental Step Down**

■ **Function** Increases (or decreases) the frequency by frequency incremental step value.
 UFR: Increase the frequency.
 DFR: Decrease the frequency.

Header	Program command	Query	Response
UFR	UFR	_____	_____
DFR	DFR	_____	_____

■ **Suffix code** None
 ■ **Example** DFR

TFR

Frequency Rotary Knob Up

EFR

Frequency Rotary Knob Down

■ Function

Increases (or decreases) the frequency by frequency Resolution.

TFR: Increase the frequency.

EFR: Decrease the frequency.

Header	Program command	Query	Response
TFR	TFR	—	—
EFR	EFR	—	—

■ Suffix code None

■ Example EFR

- R2 Frequency Resolution 1 Hz
- R3 Frequency Resolution 10 Hz
- R4 Frequency Resolution 100 Hz
- R5 Frequency Resolution 1 kHz
- R6 Frequency Resolution 10 kHz
- R7 Frequency Resolution 100 kHz
- R8 Frequency Resolution 1 MHz
- R9 Frequency Resolution 10 MHz
- FSR Frequency Resolution to Right
- FSL Frequency Resolution to Left

■ **Function** Sets the frequency resolution.
 R2: 1 Hz
 R3: 10 MHz
 R4: 100 Hz
 R5: 1 kHz
 R6: 10 kHz
 R7: 100 kHz
 R8: 1 MHz
 R9: 10 MHz
 FSR: Shifts the frequency resolution to right digit.
 FSL: Shifts the frequency resolution to left digit.

Header	Program command	Query	Response
R2	R2	_____	_____
R3	R3	_____	_____
R4	R4	_____	_____
R5	R5	_____	_____
R6	R6	_____	_____
R7	R7	_____	_____
R8	R8	_____	_____
R9	R9	_____	_____
FSR	FSR	_____	_____
FSL	FSL	_____	_____

- **Suffix code** None
- **Initial setting** R2
- **Example** R3

REFOA Frequency Relative Reference Value

■ **Function** Reads out the reference frequency of relative frequency mode.

Header	Program command	Query	Response
REFOA	_____	REFOA	REF_a (a = 0 to 2250 000 000 Hz)

■ **Value of a** 0 Hz to 2.25 GHz

■ **Suffix code** HZ: Hz

■ **Example** REFOA

RLFOA **Frequency Relative Value**

■ **Function** Reads out the relative frequency.

Header	Program command	Query	Response
RLFOA	_____	RLFOA	RLF_a (a = -2250 000 000 to 2250 000 000 Hz)

■ **Value of a** -2.25 to 2.25 GHz

■ **Suffix code** HZ: Hz

■ **Example** RLFOA

FO **Frequency Relative On**
FF **Frequency Relative Off**

■ **Function** Turns the relative frequency mode on or off.
 FO: On
 FF: Off

Header	Program command	Query	Response
FO	FO	_____	_____
FF	FF	_____	_____

■ **Suffix code** None
■ **Initial setting** FF
■ **Example** FO

FOS Frequency Offset Value

■ **Function** Sets the frequency offset value.

Header	Program command	Query	Response
FOS	FOS_a	FOSOA	FOS_a (a = -2250 000 000 to 2250 000 000 Hz)

■ **Value of a** - 2.25 to 2.25 GHz

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

■ **Initial setting** 0HZ

■ **Example** FOS_25KHZ

SP12 Frequency Offset On
 SP11 Frequency Offset Off

- Function Turns the frequency offset mode on or off.
 SP12: On
 SP11: Off

Header	Program command	Query	Response
SP12	SP12	_____	_____
SP11	SP11	_____	_____

- Suffix code None
- Initial setting SP11
- Example SP12

8.2.2 Output level

OIS Output Level Incremental Step Value

■ **Function** Sets the output level incremental step value.

Header	Program command	Query	Response
OIS	OIS_a	OISOA	OIS_a (a=0.1 to 156 dB)

■ **Value of a** 0.1 to 156 dB (0.1 dB step)

■ **Suffix code** DB: dB
None: dB

■ **Initial setting** 1DB

■ **Example** OIS_60DB

UOL Output Level Incremental Step Up
 DOL Output Level Incremental Step Down

■ Function Increases (or decreases) the output level by output level incremental step value.
 UOL: Up
 DOL: Down

Header	Program command	Query	Response
UOL	UOL	_____	_____
DOL	DOL	_____	_____

■ Suffix code None

■ Example DOL

TOL Output Level Rotary Knob Up
 EOL Output Level Rotary Knob Down

■ **Function** Increases (or decreases) the output level by output level Resolution.
 TOL: Up
 EOL: Down

Header	Program command	Query	Response
TOL	TOL	_____	_____
EOL	EOL	_____	_____

■ **Suffix code** None

■ **Example** EOL

L0 Output Level Resolution 0.1 dB
 L1 Output Level Resolution 1 dB
 L2 Output Level Resolution 10 dB
 OSR Output Level Resolution to Right
 OSL Output Level Resolution to Left

■ Function Sets the output Level resolution.
 L0: 0.1 dB
 L1: 1 dB
 L2: 10 dB
 OSR: Shifts the output level resolution to right digit.
 OSL: Shifts the output level resolution to left digit.

Header	Program command	Query	Response
L0	L0	_____	_____
L1	L1	_____	_____
L2	L2	_____	_____
OSR	OSR	_____	_____
OSL	OSL	_____	_____

■ Suffix code None
 ■ Initial setting L0
 ■ Example L2

LC **Output Level Continuous Mode On**
LN **Output Level Continuous Mode Off**

■ **Function** Turns the output level continuous mode On or Off.
 LC: On
 LN: Off

Header	Program command	Query	Response
LC	LC	_____	_____
LN	LN	_____	_____

- **Suffix code** None
- **Initial setting** LN
- **Example** LC

RE00A Output Level Relative Reference Value

- **Function** Reads out the reference output level of relative output level mode.

Header	Program command	Query	Response
RE00A	_____	RE00A	RE0_a (a: Transfer the unit which is represented in the current output level units.)

- **Value of a**
- When unit is DBM : -143 to 13 dBm
- When unit is DBU : -30 to 126 dB μ ---open voltage
-36 to 120 dB μ ---terminated voltage
- When unit is V, MV, UV : 0.032 μ V to 2 V ---open voltage
0.016 μ V to 1 V ---terminated voltage
- **Suffix code**
- DBM: dBm
DBU: dB μ
V: V
MV: mV
UV: μ V
- **Example** RE00A

RLOOA **Output Level Relative Value**

■ **Function** Reads out the relative output level.

Header	Program command	Query	Response
RLOOA	_____	RLOOA	RLO_a

■ **Value of a** -156 to 156 dB (0.1 dB step)

■ **Suffix code** DB: dB

■ **Example** RLOOA

LO Output Level Relative On
 LF Output Level Relative Off

■ Function Turns the relative output level mode on or off.
 LO: On
 LF: Off

Header	Program command	Query	Response
LO	LO	—	—
LF	LF	—	—

■ Suffix code None
 ■ Initial setting LF
 ■ Example LO

OOS **Output Level Offset Value**

■ **Function** Sets the output level offset value.

Header	Program command	Query	Response
OOS	OOS_a	OOS0A	OOS_a (a = -55.0 to 55.0 dB)

■ **Value of a** -55 to 55 dB (0.1 dB step)

■ **Suffix code** DB: dB
None: dB

■ **Initial setting** 0dB

■ **Example** OOS_3DB

SP08 Output Level Offset On
 SP07 Output Level Offset Off

■ Function Turns the output level offset mode on or off.
 SP08: On
 SP07: Off

Header	Program command	Query	Response
SP08	SP08	—	—
SP07	SP07	—	—

■ Suffix code None
 ■ Initial setting SP07
 ■ Example SP08

RO Output Level On
 RF Output Level Off

■ Function Turns the output level on or off.
 RO: On
 RF: Off

Header	Program command	Query	Response
RO	RO	—	—
RF	RF	—	—

- Suffix code None
- Initial setting RO
- Example RF

SP03 **EMF: Electro-Motive Force (open voltage display)**

SP04 **TERM: Terminated (terminated voltage display)**

- **Function** Selects the output level display for open voltage or terminated voltage.
 SP03: EMF
 SP04: TERM

Header	Program command	Query	Response
SP03	SP03	_____	_____
SP04	SP04	_____	_____

- **Suffix code** None
- **Initial setting** SP03
- **Example** SP04

OLDBM Output Level Unit to dBm

OLDM

APDBM

APDM

■ **Function** Changes output level units to dBm.

Header	Program command	Query	Response
OLDBM	OLDBM	_____	_____
OLDM	OLDM	_____	_____
APDBM	APDBM	_____	_____
APDM	APDM	_____	_____

■ **Suffix code** None

■ **Example** OLDm

OLDBU Output Level Unit to dB μ
 OLDU
 APDBU
 APDU

■ Function Changes output level units to dB μ .

Header	Program command	Query	Response
OLDBU	OLDBU	_____	_____
OLDU	OLDU	_____	_____
APDBU	APDBU	_____	_____
APDU	APDU	_____	_____

■ Suffix code None

■ Example OLDU

OLV **Output Level Unit to volt**

APV

■ **Function** Changes output level units to volt.

Header	Program command	Query	Response
OLV	OLV	_____	_____
APV	APV	_____	_____

■ **Suffix code** None

■ **Example** APV

OLM Output Level Limit Value

■ **Function** Sets the output level limit value.

Header	Program command	Query	Response
OLM	OLM_a	OLMOA	OLM_a (a = Transfer the unit which is represented in the current output level units.)

■ **Value of a**

When unit is DBM, DM :	-143 to 13 dBm
When unit is DBU, DU :	-30 to 126 dB μ ---open voltage -36 to 120 dB μ ---terminated voltage
When unit is V, MV, UV :	0.032 μ V to 2 V ---open voltage 0.016 μ V to 1 V ---terminated voltage

■ **Suffix code**

DBM, DM:	dBm
DBU, DU:	dB μ
V:	V
MV:	mV
UV:	μ V
None:	dBm

■ **Initial setting** -10DBM

■ **Example** OLM_0DBM

SP06 Output Level Limit On
 SP05 Output Level Limit Off

- Function Turns the output level limit mode on or off.
 SP06: On
 SP05: Off

Header	Program command	Query	Response
SP06	SP06	—	—
SP05	SP05	—	—

- Suffix code None
- Initial setting SP05
- Example SP06

8.2.3 Save/recall

FQaST Frequency Memory Save

■ **Function** Saves frequency to frequency memory.

Header	Program command	Query	Response
FQaST	FQaST	—	—

■ **Value of a** 0 to 999

■ **Suffix code** None

■ **Example** FQ15ST

FQaRC**Frequency Memory Recall**■ **Function**

Recalls frequency from frequency memory.

Header	Program command	Query	Response
FQaRC	FQaRC	—	—

- Value of a 0 to 999
- Suffix code None
- Example FQ15RC

FNaST**Parameter Memory Save**

■ **Function** Saves parameter to parameter memory.

Header	Program command	Query	Response
FNaST	FNaST	—	—

■ **Value of a** 0 to 99

■ **Suffix code** None

■ **Example** FN5ST

FNaRC Parameter Memory Recall

■ **Function** Recalls parameter from parameter memory.

Header	Program command	Query	Response
FNaRC	FNaRC	—	—

- **Value of a** 0 to 99
- **Suffix code** None
- **Example** FN15RC

8.2.4 Others

RS**RPP Reset**

■ **Function** Resets the RPP (Revers Power Protection) circuit to return to the original state.

Header	Program command	Query	Response
RS	RS	—	—

■ **Suffix code** None

■ **Example** RS

SP00**Preset****■ Function** Initializes parameters.

Header	Program command	Query	Response
SP00	SP00	—	—

■ Suffix code None**■ Example** SP00

SP02

Buzzer On

SP01

Buzzer Off

- Function Turns the buzzer switch on or off.
SP02: On
SP01: Off

Header	Program command	Query	Response
SP02	SP02	—	—
SP01	SP01	—	—

- Suffix code None
- Initial setting SP02
- Example SP01

SP60 Header On
 SP61 Header Off

- **Function** Selects whether or not to add header to response message.
 SP60: On
 SP61: Off

Header	Program command	Query	Response
SP60	SP60	—	—
SP61	SP61	—	—

- **Suffix code** None
- **Initial setting** SP61
- **Example** SP60

SECTION 9

SAMPLE PROGRAMS

This section describes some sample programs with the common commands and the program commands, queries, and response messages specific to the MG3670B.

TABLE OF CONTENTS

9.1	Precautions on Creating the GPIB Program	9-3
9.2	Sample Programs	9-4
9.2.1	Initializing MG3670B	9-4
9.2.2	Setting frequency and output level	9-5
9.2.3	Setting frequency using increment step frequency	9-6
9.2.4	Calibrating by external I, Q signal input	9-7

(Blank)

SECTION 9 SAMPLE PROGRAMS

9.1 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"> ① Initializing the interface functions (ISET IFC) ② Initializing message exchange functions of each device (WBYTE &H3F, &H14) ③ Initializing the functions proper to each device (PRE or *RST)
2	Turn the device to the remote state of RWLS (Remote With Lockout State).	<p>In a simple remote state, pressing the Local key 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.)</p>
3	Do not send any command (related to the device) other than the INPUT @ statement immediately after sending a query.	<p>If MLA is received when a command other than the INPUT @ statement is sent to the controller before the response to a query is read, the output buffer is cleared, and the response message disappears. For this reason, write the INPUT @ statement in immediate succession to a query.</p>
4	Create a program that avoids an exception processing of the protocol	<p>Avoid stoppage of execution (caused by an error) by means of providing a program with exception-processing section against exceptions that can be foreseen.</p>
5	Confirm the interface functions of each device (subset).	<p>Execution of program does not advance if necessary subset (s) has (have) not been prepared in the device. Be sure to confirm the subset (s) of each device. Also confirm that each device complies with IEEE488.2.</p>

9.2 Sample Programs

9.2.1 Initializing MG3670B

< Example 1 > Initializes MG3670B.

```

10 '-----
20 ' MG3670B 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 DSG=1 ..... Assigns MG3670B address to variable DSG
90 PRINT @DSG;"PRE" ..... Initializes MG3670B
100 END

```

There is a '*RST' command in another GPIB command for executing initialization.

For general usage of PRE and *RST, first initialize the MG3670B device functions with the PRE or *RST command, then use the program commands to set only the functions to be changed. This prevents the MG3670B from being controlled while unnecessary functions are set.

9.2.2 Setting frequency and output level

<Example 2> Sets the frequency to 100 MHz and output level to 0 dBm, and then performs serial polling to confirm the end of setting parameter.

```

10  '-----
20  ' MG3670B GPIB SAMPLE PROGRAM
30  ' FREQUENCY & LEVEL SET
40  '-----
50  ISET IFC
60  ISET REN
70  CMD DELIM=0
80  LET DSG=1
90  PRINT @DSG;"PRE"
100 SRQ ON ..... Permits the receipt of SRQ
110 PRINT @DSG;"*CLS" ..... Clears Status Byte Register
120 PRINT @DSG;"*SRE 4" ..... Masks bits other than SRE bit 2
130 PRINT @DSG;"*ESE2 4" ..... Masks bits other than ESE2 bit 2
140 PRINT @DSG;"FREQ 100MHZ" ..... Sets the frequency to 100MHz
150 PRINT @DSG;"OLVL 0DBM" ..... Sets the output level 0dBm
160 POLL DSG,N ..... Performs serial polling
170 IF N<>68 THEN GOTO 160 ..... Terminates this program when STB ESB(bit 2) is 1
180 END

```

Executing the SRQ command at line 100 permits the receipt of SRQ .

Lines 110 to 130 clear and mask the Status Byte Register.

Lines 140 and 150 set the frequency and the output level.

The serial polling at lines 160 and 170 is executed to check the end of setting parameter.

9.2.3 Setting frequency using increment step frequency

<Example 2> Sets the frequency to 100MHz, increment step frequency to 250kHz and then increases the frequency in 250kHz steps from 100 MHz to 200MHz.

```

10  '-----
20  ' MG3670B GPIB SAMPLE PROGRAM
30  ' FREQUENCY INCREMENTAL STEP SET
40  '-----
50  ISET IFC
60  ISET REN
70  CMD DELIM=0
80  LET DSG=1
90  PRINT @DSG;"PRE"
100 SRQ ON
110 PRINT @DSG;"*CLS
120 PRINT @DSG;"*SRE 4" ..... Masks bits other than SRE bit 2
130 PRINT @DSG;"ESE2 1" ..... Masks bits other than ESE2 bit 2
140 PRINT @DSG;"FREQ 100MHZ" ..... Sets the frequency to 100MHz
150 PRINT @DSG;"FIS 250KHZ" ..... Sets the incremental step frequency to 250kHz
160 FOR I=1 TO 400 ..... Repeats the FOR-NEXT loop 400 times until the frequency becomes 200 MHz
170 PRINT @DSG;"FRS UP" ..... Increase the frequency by 250kHz
180 POLL DSG,N
190 IF N<>68 THEN GOTO 180
200 NEXT I
210 END

```

Lines 140 and 150 set the frequency (100MHz) and incremental step frequency.

Lines 160 to 200 increase the frequency in 250kHz steps from 100 MHz to 200MHz.

Lines 180 to 190 executes waiting until setting frequency parameter is completed.

9.2.4 Calibrating by external I,Q signal input

<Example 2> Sets the frequency to 100MHz, output level to 0dBm, I,Q signals source to external input, and then calibrates the output level.

```

10 '-----
20 ' MG3670B GPIB SAMPLE PROGRAM
30 ' CALIBRATION
40 '-----
50 ISET IFC
60 ISET REN
70 CMD DELIM=0
80 LET DSG=1
90 PRINT @DSG;"PRE"
100 SRQ ON
110 PRINT @DSG;"*CLS"
120 PRINT @DSG;"*SRE 4" ..... Masks bits other than SRE bit 2
130 PRINT @DSG;"*ESE2 2" ..... Masks bits other than ESE2 bit 1
140 PRINT @DSG;"FREQ 100MHZ" ..... Sets the frequency to 100MHz
150 PRINT @DSG;"OLVL 0DBM" ..... Sets the output level to 0dBm
160 PRINT @DSG;"MOD ON" ..... Turns modulation on
170 PRINT @DSG;"MODE EXT" ..... Sets the I,Q signals source to external input
180 PRINT @DSG;"CAL" ..... Calibrates the output level
190 POLL DSG,N
200 IF N<>68 THEN GOTO 190
210 END

```

Lines 140 and 150 set the frequency and the output level.

The modulation is turned on and I,Q signals source is set to external input at lines 160 and 170, respectively.

Line 180 calibrates the output level.

Lines 190 and 200 check that calibration execution is terminated.

(Blank)

APPENDICES

TABLE OF CONTENTS

APPENDIX A	ASCII*CODE TABLE	A-1
APPENDIX B	COMPARISON TABLE OF CONTROLLER'S GPIB INSTRUCTIONS	B-1

(Blank)

A ASCII CODE TABLE

BITS B7 B6 B5 B4 B3 B2 B1	0 0	0 1	0 1	0 1	1 0	1 0	1 1	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 ENQ	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	30 CAN	50 (70 8	110 H	130 X	150 h	170 x
1 0 0 1	9 HT	31 EM	51)	71 9	111 I	131 Y	151 i	171 y
1 0 1 0	10 LF	32 SUB	52 *	72 :	112 J	132 Z	152 j	172 z
1 0 1 1	11 VT	33 ESC	53 ÷	73 ;	113 K	133 [153 k	173 {
1 1 0 0	12 FF	34 FS	54 ' <	74 <	114 L	134 \	154 l	174 ;
1 1 0 1	13 CR	35 GS	55 - =	75 =	115 M	135]	155 m	175 }
1 1 1 0	14 SO	36 RS	56 . >	76 >	116 N	136 ^	156 n	176 ~
1 1 1 1	15 SI	37 US	57 / ?	77 ? UNL	117 O	137 _ UNT	157 o	177 RUBOUT (DEL)
	Address command	Universal command	Listen address	Talk address	Secondary address or command			

KEY octal 25 PPU GPIB code
 NAK ASCII character
 hex 15 21 decimal

† USA Standard Code for Information Interchange

(Blank)

B 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()	

