Anritsu

MS9710C

Optical Spectrum Analyzer

Remote Control

Operation Manual

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

Anritsu

MS9710C Optical Spectrum Analyzer Remote Control Operation Manual

Anritsu

ANRITSU CORPORATION 5-10-27, Minamiazabu, Minato-ku, Tokyo 106-8570 Japan / Phone: 81-3-3446-1111

Document No.: M–W1580AE
Printed in Japan

MS9710C
Optical Spectrum Analyzer
Remote Control
Operation Manual

Third Edition

To ensure that the MS9710C Optical Spectrum Analyzer is used safely, read the safety information in the MS9710C Optical Spectrum Analyzer Operation Manual first. Keep this manual with the Optical Spectrum Analyzer.

Measuring Instruments Division
Measurement Group
ANRITSU CORPORATION

Document No.: M-W1580AE-3.0

MS9710C
Optical Spectrum Analyzer Remote Control
Operation Manual

14 February 1999 (First Edition)
21 April 2000 (Third Edition)

About This Manual

This manual explains remote control of the MS9710C optical spectrum analyzer. You can control the MS9710C and transfer measurement results into the computer connected to the GPIB/RS-232C interface port of the MS9710C.

Table of Contents

Section 1 Introduction

This section outlines the remote control functions of the MS9710C optical spectrum analyzer.

1.1 Overview

The MS9710C can make measurements automatically in combination with an external controller (host computer, personal computer, etc.) To connect an external controller, the MS9710C has GPIB interface bus (IEEE Standard 488.2-1987) and RS-232C interface ports.

1.2 MS9710C Remove Control Functions

The MS9710C supports the following functions.

(1) Control of almost all functions except some functions such as a POWER switch and LOCAL key

(2) Read of all setting conditions

(3) Setting of the GPIB address from panel

(4) Interrupt function and serial polling (GPIB)

(5) Setting of RS-232C interface conditions from panel

(6) Selection of an interface port application from panel

(7) Configuration of an automatic measurement system by combining the MS9710C with a personal computer and another measurement instrument

(8) Tracking measurement with the tunable laser source

1.3 Interface Port Application Selection Function

The MS9710C comes standard with a GPIB interface bus and an RS-232C interface. Application of these interface ports can be selected from the panel.

External controller connection port:	Select GPIB or RS-232C.
Printer connection port:	GPIB

The above two ports cannot be used at the same time.

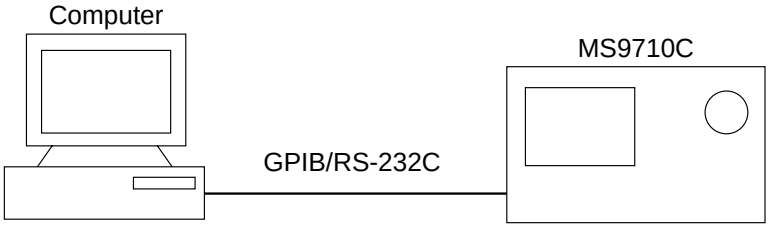
1.4 Examples of Setups Using GPIB/RS-232C

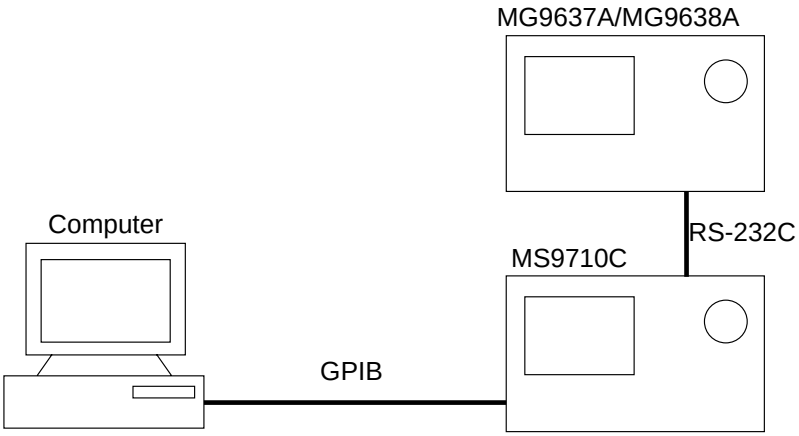
(1) Standalone type

Waveforms measured with the MS9710C are output to the printer.

(2) Control by host computer

The MS9710C is controlled by a computer automatically/remotely.





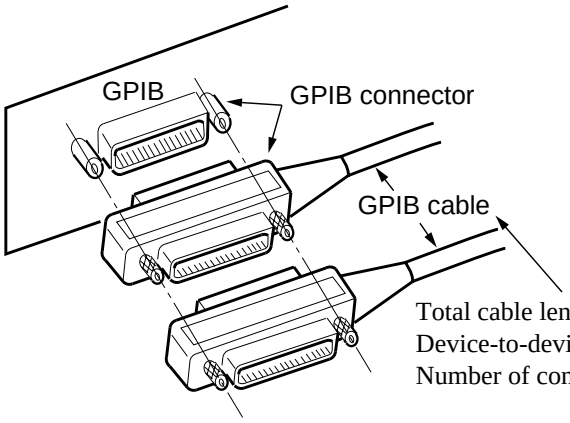
Section 2 How to Connect

This section explains how to connect GPIB and RS-232C cables between the MS9710C and external devices such as a host computer, personal computer, and printer. This section also explains how to set the interfaces of the MS9710C.

2.1 Connecting Devices Using GPIB Cables

The MS9710C has a GPIB cable connection connector on the back panel. Be sure to connect GPIB cables before turning on the power.

A maximum of 15 devices, including a controller, can be connected. Connection conditions are given shown below.



Total cable length	≤ 20 m
Device-to-device cable length	≤ 4 m
Number of connectable devices	< 15

2.1.1 Setting interface conditions for the connection port

When controlling the MS9710C automatically/remotely from a computer, set interface conditions for the connection Press the RS-232C Prmtr function key on the “Others” card to select “GPIB” for “Interface.”

=== RS232C Parameter ==========

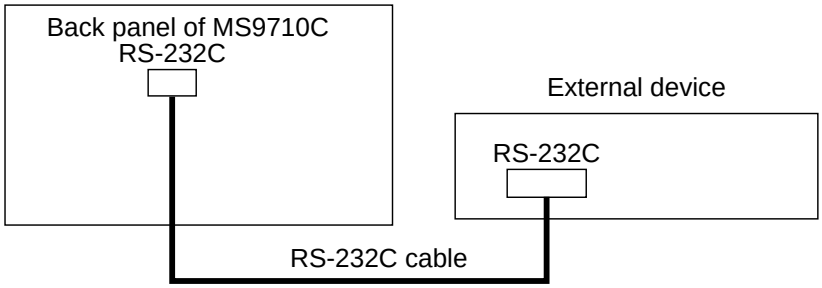
▶Interface	GPIB	RS232C			
Speed(bps)	9600	4800	2400	1200	600
Parity	None	Even	Odd		
Character Length	7Bit	8Bit			
Stop Bit	1Bit	2Bit			

2.1.2 Confirming and Setting the Address

Be sure to set the MS9710C’s GPIB address after turning on the power. The factory-set address “08” is battery-backed up. If you use this address, the address need not be set again. If you want to change the address, place the MS9710C in the local mode, press the GPIB Address function key on the “Others” card, then enter a new address with keyboard keys or an encoder. Immediately after the power is turned on, the devices on the GPIB automatically enters the local mode.

2.2 Connecting a Device Using an RS-232C Cable

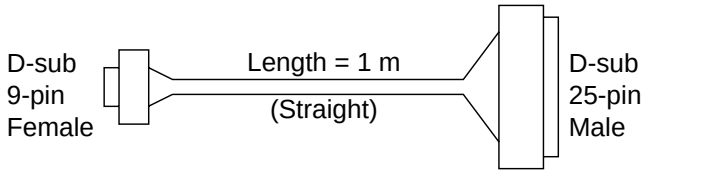
Connect the RS-232C connector (D-sub, 9-pin, male) and the RS-232C connector with an RS-232C cable.



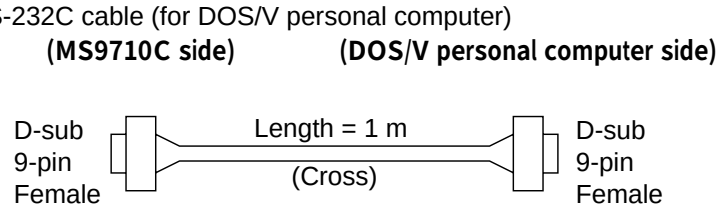
Note:

RS-232C connectors are available in 9-pin and 25-pin types. Before purchasing an RS-232C cable, check the number of pins of the RS-232C connector on the external device. The following two types of RS-232C cables are available as application parts for this analyzer.

•RS-232C cable (for PC98 personal computer)



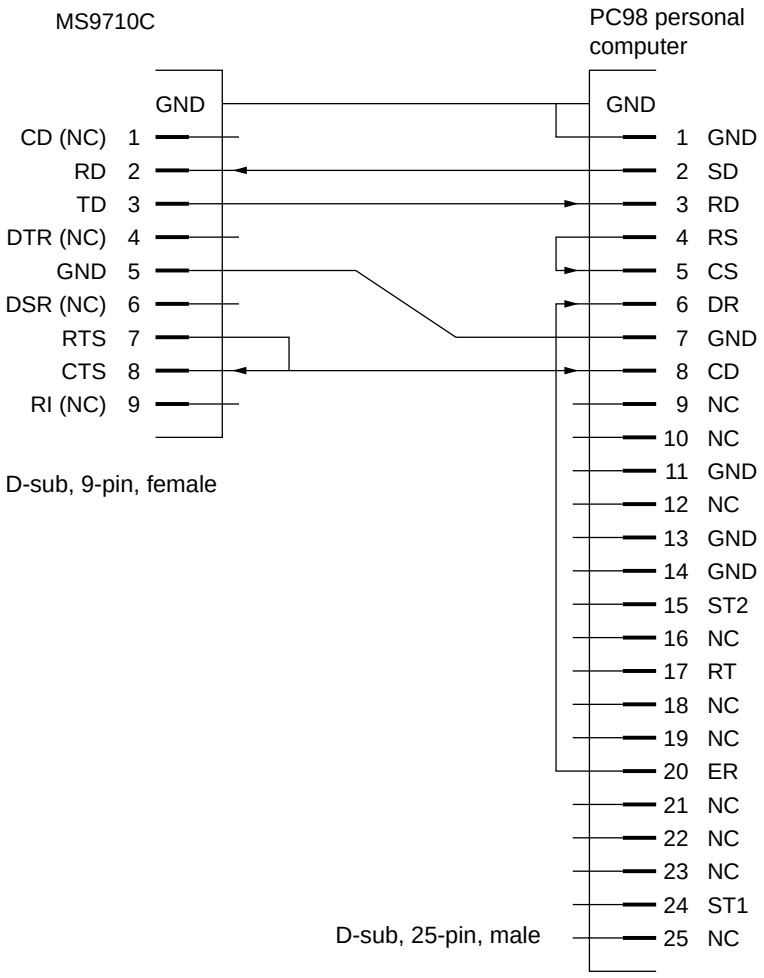
•RS-232C cable (for DOS/V personal computer)



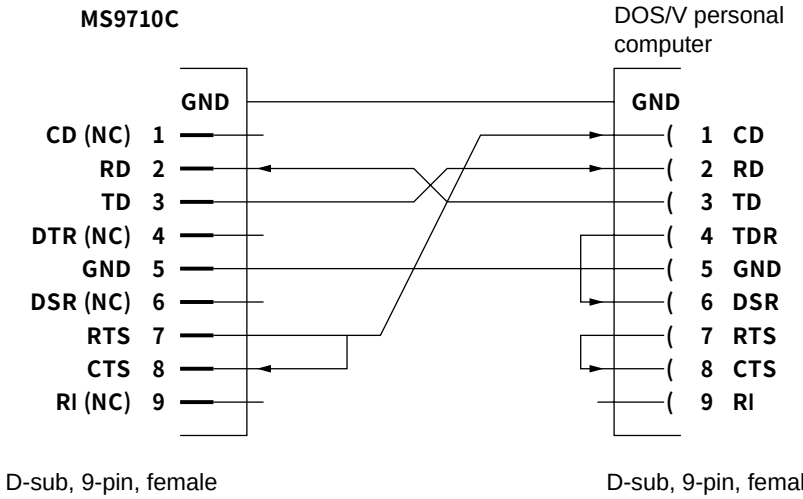
2.2.1 RS-232C interface signal connection diagrams

The following diagrams show connections of RS-232C interface signals between the MS9710C and two types of personal computers.

- Connection with PC98 personal computer



- Connection with DOS/V personal computer



2.2.2 Setting interface conditions for the connection port

When controlling the MS9710C automatically/remotely from a computer, set interface conditions for the connection port.
Press the RS-232C Prmtr function key on the “Others” card and select “RS232C” for “Interface.”

2.2.3 Setting RS-232C interface conditions

Set interface conditions for the RS-232C port of this analyzer so that they match the interface conditions of the connected external device.

Pressing the RS-232C Prmntr function key on the “Others” card will bring up the following screen.

```
=== RS232C Parameter ====================================
▶Interface ···············  GPIB   RS232C
 Speed(bps) ··············  9600   4800   2400   1200   600
 Parity ··················  None   Even   Odd
 Character Length ·········  7Bit  8Bit
 Stop Bit ·················  1Bit   2Bit
```

Using ↑ and ↓ function keys, move the cursor to the item you want to change.

Item	Meaning of setting
Speed	Select a communication speed among 600, 1200, 2400, 4800, and 9600 bps.
Parity	Select a parity bit type. None… No parity bit is added. Even… An even parity bit is added. Odd… An odd parity bit is added.
Stop Bit	Select a stop bit type. 1…… 1 stop bit is added. 2…… 2 stop bits is added.
Character Length	Select a character length. 7…… 7 bits 8…… 8 bits

Section 3 Standards

This section explains the MS9710C's GPIB standard, RS-232C standard, and device message list.

3.1 GPIB Standard

The standard for the GPIB of the MS9710C is summarized below.

Item	Standard value and description
Function	Conforms to IEEE 488.2. MS9710C can be controlled from an external controller. MS9710C can control a printer.
Interface functions	SH1: All of source handshake functions are supported. Data send timing is controlled. AH1: All of acceptor handshake functions are supported. Data receive timing is controlled. T6: Basic talker functions are supported. A serial port function is supported. A talk-only function is not supported. The function of releasing the talker with MLA is supported. L4: Basic listener functions are supported. A listen-only function is not supported. The function of releasing the listener by MTA is supported. SR1: All of service request/status byte functions are supported. RL1: All of remote/local functions are supported. A local lockout function is supported. PP0: A parallel poll function is not supported. DC1: All of device clear functions are supported. DT0: A disk trigger function is not supported. C0: A controller function is not supported. A controller function is performed during external plot output.

3.2 RS-232C Standard

The standard for the RS-232C of the MS9710C is summarized below.

Item	Standard value
Function	Control from external controller
Communication method	Asynchronous (start-stop), half-duplex
Communication control method	No flow control
Baud rate	600, 1200, 2400, 4800, 9600 bps
Data bits	7 bits, 8 bits
Parity	Odd parity (ODD), even parity (EVEN), non-parity (NON)
Start bits	1 bit
Stop bits	1 bit, 2 bits
Connector	D-sub 9-pin connector, female

3.3 Device Message List

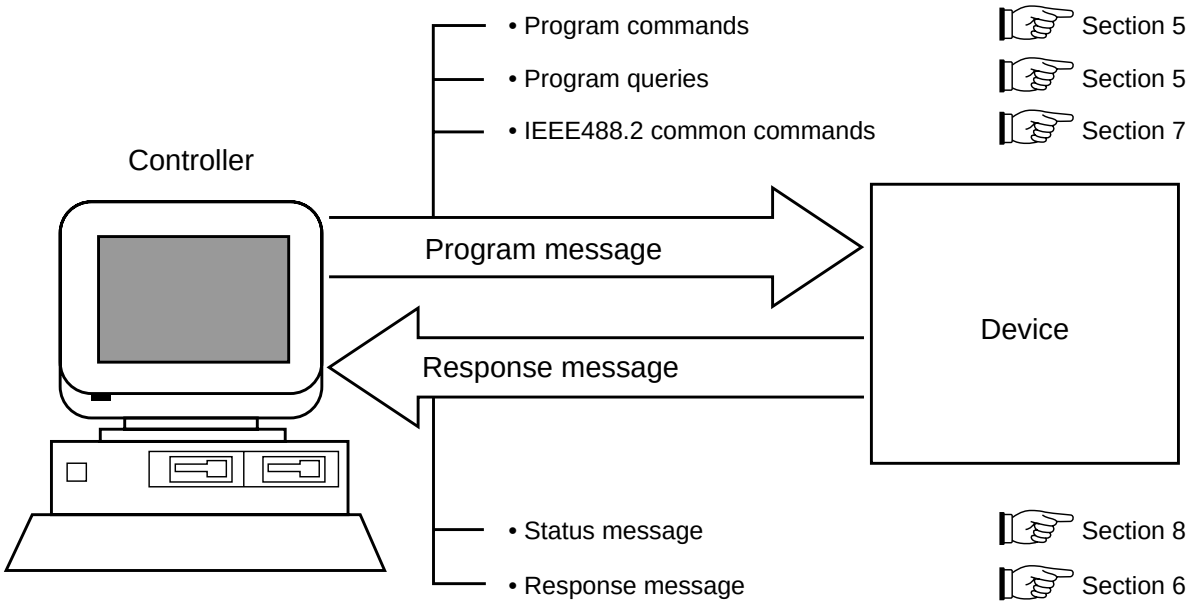
Device messages are data messages which are transferred between a controller and devices. They are classified into program messages and response messages.

Program messages are ASCII messages transferred from a controller to devices. Program messages are further classified into program commands and program queries. These two types of commands are explained on the following pages.

Program commands include device-dependent commands which are exclusively used for controlling the MS9710C and IEEE 488.2 common commands. IEEE 488.2 common commands are program commands which are commonly applicable to other IEEE 488.2-ready measuring instruments (including the MS9710C) on the GPIB interface bus.

Program queries are commands used to get response messages from devices. Program queries must be transferred from a controller to a device in advance so that the controller can receive response messages from the device later.

Response messages are ASCII data messages which are transferred from a device to a controller. Among response messages, status messages and response messages corresponding to program queries are listed on the following pages.



In program and response messages, numeric data may end with a suffix (unit).

The above messages are transferred through the device input/output buffer. The output buffer is also called an output queue. A brief description of the output buffer is given below.

Input buffer

A FIFO (first in first out) type memory area that stores DABs (program and query messages) temporarily before analysis of syntax and execution.
The input buffer size of the MS9710C is 256 bytes.

Output queue

An FIFO-type queue memory area. All DABs (response messages) output from a device to a controller are stored in this memory until they have been read by the controller.
The output queue size of the MS9710C is 256 bytes.

3.3.1 IEEE 488.2 common commands and the commands supported by the MS9710C

The table below lists 39 common commands specified by IEEE 488.2. Among these commands, the commands supported by the MS9710C are marked with .

Mnemonic	Fully spelled out command name	Standardized by IEEE 488.2	Supported by MS9710C
∗ADD	Accept Address Command	Optional	
∗CAL	Calibration Query	Optional	
∗CLS	Clear Status Command	Required	√
∗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	Required	√
∗ESE?	Standard Event Status Enable Query	Required	√
∗ESR?	Standard Event Status Register Query	Required	√
∗GMC?	Get Macro contents Query	Optional	
∗IDN?	Identification Query	Required	√
∗IST?	Individual Status Query	Optional	
∗LMC?	Learn Macro Query	Optional	
∗LRN?	Learn Device Setup Query	Optional	
∗OPC	Operation Complete Command	Required	√
∗OPC?	Operation Complete Query	Required	√
∗OPT?	Option Identification Query	Optional	√
∗PCB	Pass Control Back Command	Other than C0: Required	
∗PMC	Purge Macro Command	Optional	
∗PRE	Parallel Poll Register Enable Command	Optional	
∗PRE?	Parallel Poll Register Enable Query	Optional	
∗PSC	Power On Status Clear Command	Optional	
∗PSC?	Power On Status Clear Query	Optional	
∗PUD	Protected User Data Command	Optional	
∗PUD?	Protected User Data Query	Optional	
∗RCL	Recall Command	Optional	
∗RDT	Resource Description Transfer Command	Optional	
∗RDT?	Resource Description Transfer Query	Optional	
∗RST	Reset Command	Required	√
∗SAV	Save Command	Optional	
∗SRE	Service Request Enable Command	Required	√
∗SRE?	Service Request Enable Query	Required	√
∗STB?	Read Status Byte Query	Required	√
∗TRG	Trigger Command	DT1: Required	
∗TST?	Self Test Query	Required	√
∗WAI	Wait to Continue Command	Required	√

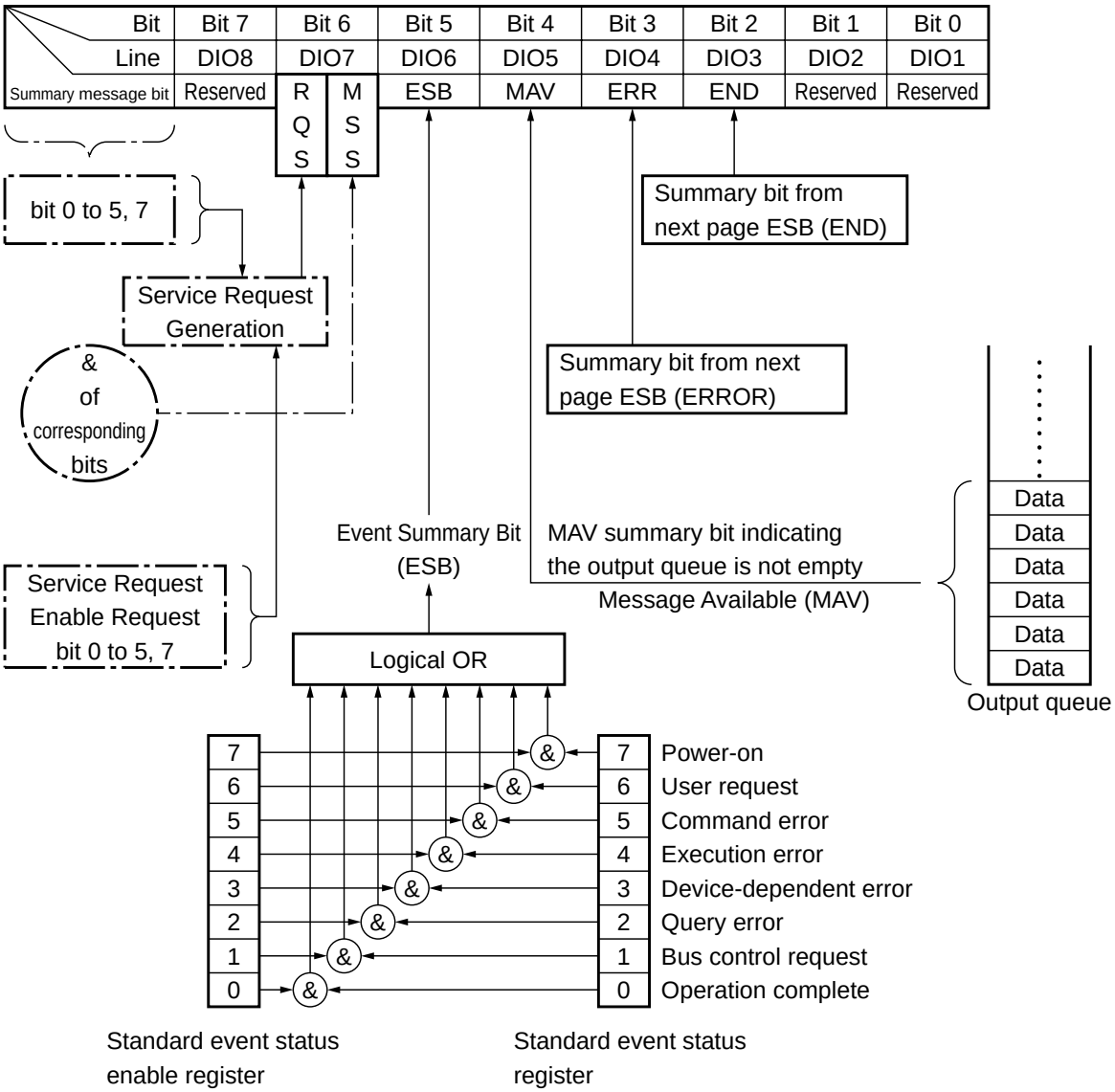
Note:

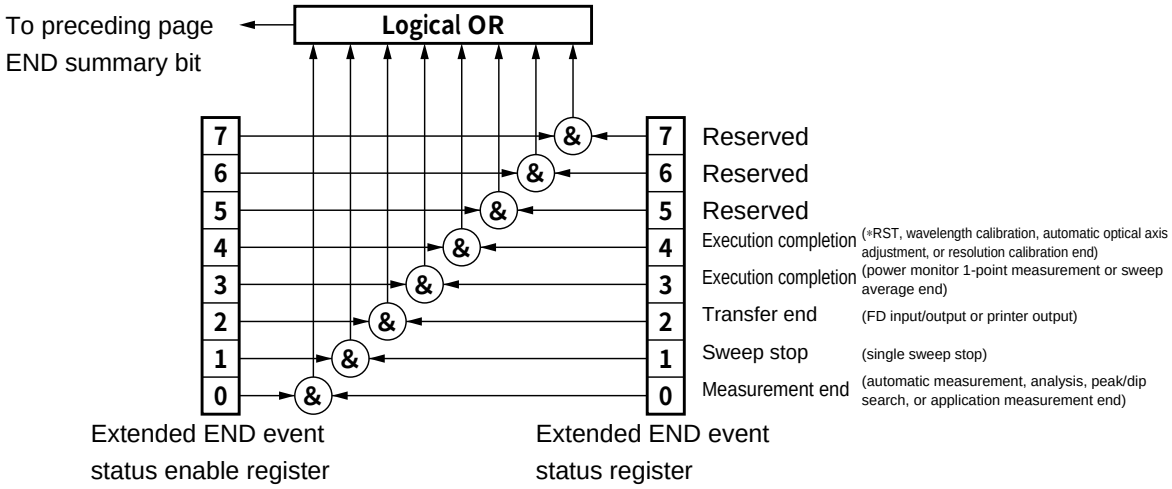
IEEE 488.2 commands begin with ∗. For more details, see Section 7.

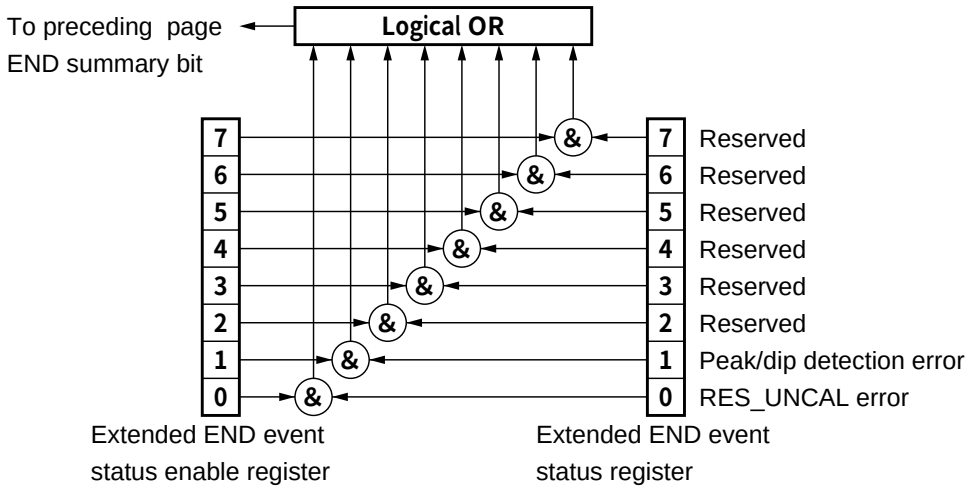
3.3.2 Status Messages

Shown below is the structure of the service request summary message set in the status byte register of the MS9710C.

Summary Bit Configuration of Status Byte Register




To preceding page
END summary bit
Logical OR
7
6
5
4
3
2
1
0
&
Reserved
Reserved
Reserved
Execution completion (*RST, wavelength calibration, automatic optical axis adjustment, or resolution calibration end)
Execution completion (power monitor 1-point measurement or sweep average end)
Transfer end (FD input/output or printer output)
Sweep stop (single sweep stop)
Measurement end (automatic measurement, analysis, peak/dip search, or application measurement end)
Extended END event status enable register
Extended END event status register


To preceding page
END summary bit
Logical OR
7
6
5
4
3
2
1
0
&
Reserved
Reserved
Reserved
Reserved
Reserved
Reserved
Peak/dip detection error
RES_UNCAL error
Extended END event status enable register
Extended END event status register

3.3.3 MS9710C device message list

A list of MS9710C-dependent program commands, program queries, and response messages is shown on the following pages.

MS9710C Device Message List (1/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Wave-length	Center	CNT λ λ = xxxx.xx	CNT?	λ λ = xxxx.xx	λ : Wavelength (nm)	9.14
	Span	SPN λ λ = xxxx.x	SPN?	λ λ = xxxx.x	λ : Wavelength (nm)	9.74
	Start	STA λ λ = xxxx.x	STA?	λ λ = xxxx.x	λ : Wavelength (nm)	9.78
	Stop	STO λ λ = xxxx.x	STO?	λ λ = xxxx.x	λ : Wavelength (nm)	9.79
	Marker Value Wl/Freq	MKV s	MKV?	s s = WL = FREQ	WL : Wavelength FREQ : Frequency	9.56
	Value in Vacuum/Air	WDP s s = VACUUM = AIR	WDP?	s s = VACUUM = AIR	s : Wavelength display mode	9.93
Level	Level Scale		LVS?	s s = LOG LIN	 Log scale Linear scale	9.50
	Log (/div)	LOG l l = xx.x	LOG?	l l = xx.x	Unit : dB/div 0.1 to 10.0	9.49
	Ref Level	RLV l l = ±xx.x	RLV?	l l = ±xx.x	A, B, A&B : dBm (unit) 1 = +30 to –90 A-B, B-A, normalize : dB (unit) 1 = +100 to –100	9.70
	Linear	LLV l	LLV?	l	1 : 1 pW to 1 W (level) 1 to 200 % When unit is omitted : mW, %	9.47
	Opt Att	ATT s s = ON, OFF	ATT?	s s = ON, OFF		9.8

MS9710C Device Message List (2/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Resolu -tion	Res	RES n n : Value shown on the right	RES?	n n = 0.05, 0.07, 0.1, 0.2, 0.5, 1	n : Resolution (nm)	9.69
	Actual Res Off/On	ARES s s = ON, OFF	ARES?	s s = ON, OFF		9.7
	Actual Res Value		ARED?	Δλ Δλ = x.xxx	Δλ (nm)	9.6
VBW		VBW s s : Resolution (nm)	VBW?	s s = 1 MHz, 100 kHz, 10 kHz, 1 kHz, 100 Hz, 10 Hz	s : VBW value When unit is omitted : Hz	9.91
Average	Point Average	AVT n n = 2 to 1000 OFF	AVT?	n n = 2 to 1000, OFF	n : Number of times	9.11
	Sweep Average	AVS n n = 2 to 1000 OFF	AVS?	n n = 2 to 1000, OFF	n : Number of times	9.10
	Smooth	SMT n n : Value shown on the right	SMT?	n n = 3, 5, 7, 9, 11, OFF	n : Number of points	9.72
Sampling Points		MPT n n : Value shown on the right	MPT?	n n = 51, 101, 251, 501, 1001, 2001, 5001	n : Number of points	9.58
Peak Serch		PKS s s = PEAK, NEXT, LAST, LEFT, RIGHT	PKS?	m m = PEAK, NEXT, LAST, LEFT, RIGHT, ERR	ERR : State other than peak search	9.64
Dip Search		DPS s s = DIP, NEXT, LAST, LEFT, RIGHT	DPS?	m m = DIP, NEXT, LAST, LEFT, RIGHT, ERR	ERR : State other than dip search	9.28

MS9710C Device Message List (3/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Analysis	Envelope	ANA ENV, r r = 0.1 to 20.0	ANA?	ENV, r r = 0.1 to 20.0	r : Cut level (dB)	9.2
	RMS	ANA RMS, r, k r = 0.1 to 30.0 k = 1, 2, 2.35, 3	ANA?	RMS, r, k r = 0.1 to 30.0 k = 1, 2, 2.35, 3	r : Slice level (dB) k : Constant of kσ	
	ndB-Loss	ANA ndB, r r = 0.1 to 50.0	ANA?	NDB, r r = 0.1 to 50.0	r : Attenuation (dB)	
	Threshold	ANA THR, r r = 0.1 to 50.0	ANA?	THR, r r = 0.1 to 50.0	r : Cut level (dB)	
	SMSR	ANA SMSR, s s = 2NDPEAK = LEFT = RIGHT	ANA?	SMSR, s s = 2NDPEAK = LEFT = RIGHT		
	Spectrum Power	ANA PWR	ANA?	PWR		
	Analysis Off	ANA OFF	ANA?	OFF		
Analysis Result	Envelope Threshold RMS		ANAR?	λ c, Δ λ λ c = xxxx.xxx Δ λ = xxx.xxx	 λ c (nm) or (THz) Δ λ (nm) or (THz)	9.3
	ndB-Loss		ANAR?	λ c, Δ λ, n λ c = xxxx.xxx Δ λ = xxx.xxx n : Integer	 λ c (nm) or (THz) Δ λ (nm) or (THz) n : Number of axial modes	
	SMSR		ANAR?	Δ λ, Δ l Δ λ = xxx.xxx Δ l = xx.xx (dB)	 Δ λ (nm) or (THz) Δ l (dB)	
	Spectrum Power		ANAR?	p, λ c p = xx.xx λ c = xxxx.xxx	 P : Power (dBm) λ c (nm) or (THz)	
Memory Select		MSL s s = A, B	MSL?	s s = A, B		9.59
Trace Select		TSL s s = A, B, AB, A_B, B_A	TSL?	s s = A, B, AB, A_B, B_A		9.89

MS9710C Device Message List (4/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Save/ Recall	Format	FMT				9.41
	File Delete	DEL n n : File name			n : File name	9.23
	File Option	FOPT a, b, c a = NONE = BMP = TXT = BMP&TXT b = NUMBER = NAME c = 1.44 M = 1.2 M	FOPT?	a, b, c a = NONE = BMP = TXT = BMP&TXT b = NUMBER = NAME c = 1.44 M = 1.2 M	a : Option file specification b : File specification method c : FDD mode c : Omissible	9.42
	Save	SAV n n : File name			n : File name	9.71
	Recall	RCL n n : File name			n : File name	9.68
Graph	Normal	DMD NRM	DMD?	NRM		9.26
	3D	DMD 3, m, n m, n : Values shown on the right	DMD?	3, m, n m= 1, 2, 3 n = 30, 45, 60, 90	m: Type n : Angle	
	Normalize	DMD NRMZ	DMD?	NRMZ		
	Overlap	DMD OVL	DMD?	OVL		
	Max Hold	DMD MHL	DMD?	MHL		
	Graph Clear	GCL				9.43

MS9710C Device Message List (5/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Applica -tion	DFB-LD	AP DFB, s, n s = 2NDPEAK = LEFT = RIGHT n = 1 to 50	AP?	DFB, s, n s = 2NDPEAK = LEFT = RIGHT n = 1 to 50	n : n of "ndB Width"	9.4
	FP-LD	AP FP, n n = 1 to 50	AP?	FP, n n = 1 to 50	n : Axial mode cut level	
	LED	AP LED, n, p n = 1 to 50 p = –10.00 to +10.00	AP?	LED, n, p n = 1 to 50 p = –10.00 to +10.00	n : n of "ndB Width" p : Calibration value (dB) of total power	
	PMD	AP PMD, n n = 0.01 to 1.00	AP?	PMD, n, m n = 0.01 to 1.00 m= 0 : AUTO, 1 : MANUAL	n : Mode coupling factor AUTO/MANUAL mode	
	Opt Amp	AP AMP	AP?	AMP	O.AMP mode setting	
	Opt Amp Memory Select	AP AMP, MSL, s s = PIN = POUT	AP? AMP, MSL	AMP, MSL, s s = PIN = POUT	PN : Pin memory POUT : Pout memory	
	O.Amp Res Cal	AP AMP, CAL, n n = 0 : RES CAL INITIAL 1 : RES CAL	AP? AMP, CAL	AMP, CAL, m m=0 : RES calibration complete 1 : Insufficient optical level 2 : Other faults		

MS9710C Device Message List (6/13)

Function		Device message			Remarks	
		Command	Data request	Response		
Applica -tion	O.Amp Parameter	AP AMP, PRM, a, b, c, d, e, f, g, h, i, j, k	AP? AMP, PRM	AMP, PRM, a, b, c, d, e, f, g, h, i, j, k		9.4
		a = 0 : S-ASE, 1 : Total		a : NF Calculation		
		b = 0 : SpectDiv Off, 1 : On 2 : PlznNull Method 3 : Pulse Method 4 : WDM Measure		b : NF Measurement method		
		c = 0 : Gauss, 1 : Mean		c : Fitting Method		
		d : Fit Span		d : Fitting Span	: 0.10 to 100.00 nm	
		e : Mask Span		e : Masked Span	: 0.10 to 100.00 nm	
		f : Pin Loss		f : Pin Loss	: –10.00 to 10.00 dB	
		g : Pout Loss		g : Pout Loss	: –10.00 to 10.00 dB	
		h : NF Cal		h : NF Cal	: 0.100 to 10.000	
		i : O.BPF LCal		i : O.BPF Level CaL	: 0.00 to 30.00 dB	
		j : O.BPF BW		j : O.BPF BW	: 0.01 to 999.99 nm	
		k : Pol Loss		k : Pol Loss	: –10.00 to 10.00 dB	
	Pout→Pase	AP AMP, ASE				
	Off	AP OFF	AP?	OFF		
	WDM	AP WDM	AP?	AP WDM,m m= "MPK", "SNR", "REL"	Display Mode is previous condition, MPK, SNR, REL mean Multi Peak, SNR, and Relative displays, respectively.	
	WDM S. Level	AP WDM, SLV, s s = 1 to 50	AP? WDM, SLV	AP WDM, SLV, s s = 1 to 50	S is splice level.	
	WDM Multi Peak	AP WDM, MPK	AP? WDM, MPK	(AP, WDM, MPK)	In this case only, same result at AP?	
	WDM SNR	AP WDM, SNR, d, Δ λ, S d = "HIGHER", "LEFT", "RIGHT" = AVERAGE Δ λ = 0.01 to 20.00, "OFF" s = ON = OFF	AP? WDM, SNR	AP WDM, SNR, d, Δ λ d = "HIGHER", "LEFT", "RIGHT" = AVERAGE Δ λ = 0.01 to 20.00, "OFF" s = ON OFF	d is the Dip detection direction, Δ λ is the detection position in 0.01 nm steps; at Δ λ = OFF or 0, detects dip in direction set at d. s is ON/OFF of normalizing noise with the effective resolution.	
	WDM Relative	AP WDM, REL, r r = 1 to 50	AP? WDM, REL	AP WDM, REL, r r = 1 to 50	r is the reference peak number	

MS9710C Device Message List (7/13)

Function		Device message			Remarks	
		Command	Data request	Response		
Applica-tion	WDM Table	AP WDM, TBL, d, Δλ, s d = HIGHER = LEFT = RIGHT = AVERAGE Δλ= 0.01 to 20.00, OFF s = ON = OFF	AP? WDM, TBL	WDM, TBL, d Δλ, s s = HIGHER = LEFT = RIGHT = AVERAGE Δλ = 0.01 to 20.00, OFF s = ON = OFF	d is the Dip detection direction. Δλ is the detection position in 0.01 nm steps; at Δλ = OFF or 0, detects Dip in the specified direction. s is ON/OFF of normalizing noise with the effective resolution.	9.4
	WDM Peak Type	AP WDM, PKT, t t = PEAK = THRESHOLD	AP? WDM, PKT	WDM, PKT, t t = PEAK =THRESHOLD		
	WDM Threshold Cue Level	AP WDM, TCL, u u = 0.1 to 50.0	AP? WDM, TCL	WDM, TCL, u u = 0.1 to 50.0 SMSR, BW ndb,	u: Cut level (dB)	
Applica -tion Result	DFB-LD		APR?	λ p, L p, λ sm, L sm, MOFS, STBW, CNTOFS FWHM, λ m,	Wavelength ****.*** nm Level **.** dBm (dB)	9.5
	FP-LD		APR?	λ p, L p, MODO, MSPC, POW λ fwhm, λ ndb,		
	LED		APR?	FWHM, BW ndb, λ p, L p, PK dens, POW MPKC, d		
	Peak Count		APR? MPKC	d = 0 to 50 Δ t, λ 1st, λ last,	d : Number of multipeaks	
	PMD		APR?	PK count G, NF, λ sig,	Δ t : (fs) Two digits below decimal point	
	O.AMP		APR?	Lase, RES	NF : **.** dB	

MS9710C Device Message List (8/13)

Function		Device message			Remarks	
		Command	Data request	Response		
Applica -tion Result	WDM		APR? (analysis results-link obtained)	n, λ 1, L1, λ 2, L2...	n is the number of peaks, λ x is the wavelength of peak x, Lx is the level of peak x. When there is no peak, n = 0.	9.5
				n, λ 1, L1, S1, d1, λ 2, L2, S2, d2... dx = "LEFT", "RIGHT"	n is the number of peaks, λ x is the wavelength of peak x, Lx is the level of peak x, Sx is the SNR value for peak x, dx indicates whether the dip of peak x is left or right.	
				n, Rn, λ 1, SP1, R λ 1, L1, RL1, λ 2, SP2, R λ 2, L2, RL2...	n is the number of peaks, Rn is the reference peak number, λ x is the wavelength, SPx is the spacing of peak x, Rλx is the relative wavelength of peak x, Lx is the level of peak x, RLx is relative level of peak x. When there is no peak, n = 0.	
				n, λ 1, f1, L1, S1, d1, SP1, SPf1, λ 2, f2, L2, S2, d2, SP2, SPf2...	n is the number of peaks, λx is the wavelength of the peak x, fx is the frequency of the peak x, Lx is the wavelength of the peak x, Sx is the SNR value for the peak x, dx is the spacing of the peak x, and Spfx is the spacing frequency of the peak x.	

MS9710C Device Message List (9/13)

Function		Device message			Remarks	
		Command	Data request	Response		
Applica -tion Result	WDM Peak Count		APR? MPKC	MPKC, n n = 0 to 50	n is the number of peaks	9.5
	WDM Multi Peak		APR? WDM, MPK, x x=1 to 50	WDM, MPK, λ, L λ = xxxx.xxx, L = xxxxx.xx	Finds the wavelength and level of peak x. When there is no data for peak number x, λ = –1 and L = –999.99	
	WDM SNR		APR? WDM, SNR, x x=1 to 50	WDM, SNR, λ, L, S, d λ = xxxx.xxx, L = xxxxx.xx, S = xxx.xx, d = "LEFT", "RIGHT", "ERR"	Peak No. x λ is wavelength, L is level, S is SNR, and d is Dip level detection direction. ERR is displayed when there is no peak. When there is no data for peak number x, λ = –1 and L = –999.99	
	WDM Gain Varia-tion		APR? WDM, SNR, GAV	t	dB units, –999.00 when no peak	
	WDM Relative		APR? WDM, REL, x x=1 to 50	WDM, REL, λ, SP, R λ, L, RL	Peak No. x SP is spacing, λ is wavelength, R λ is relative wave-length, L is level, RL is relative level. When there is no data for peak number x, λ = –1 and L =–999.99	
	WDM Table		APR? WDM, TBL, x	WDM, TBL, λ, f, L, S, d, SP, SPf f = xxx.xxxx SPf = xxx.x	f of Peak No.x is fre-quency. SPf is spacing frequency. Others are the same as SNR. f unit is (THz). SPf unit is (GHz).	

MS9710C Device Message List (10/13)

Function		Device message			Remarks	
		Command	Data request	Response		
Measure Mode	D.range Normal/ High	DRG s s = NORMAL = HIGH	DRG?	s s = NORMAL = HIGH		9.31
	Interval Time	ITM s s = 0 to 99MIN 0 to 99SEC	ITM?	s s = 0 to 99MIN 0 to 99SEC	s : Time When omitted : sec	9.45
	Moduration Mode	MDM s s = NORMAL = TRIGGER	MDM?	s s = NORMAL = TRIGGER	Normal EXT trigger	9.51
TLS Tracking	External Trigger Delay Time	TDL n n = 0 to 5000000	TDL?	n n = 0 to 5000000	n : Time (ms)	9.80
	TLS Tracking	TLST s s = On/Off	TLST?	s s = On/Off		9.85
	Adjust to TLS	TLSA n n = 0 : Stop calibration, 1 : Calibrate	TLSA?	n = 0 : Calibration finished, 1 : Calibrating, 2 : Calibration abnormal, 3 : Uncalibrated		9.84
	Power Monitor Power	PWR λ	PWR?	λ λ = 632.8,850.0 1300.0,1550.0	λ : Wavelength (nm)	9.65
	Monitor Result		PWRR?	P1 P1 = ±xx.xx	P1 : Power value (dBm)	9.67
	Spectrum Mode Set	SPC				9.73
Title	Title	TTL 'character string'	TTL?	Character string	30 characters	9.90
	Title Erase	TER				9.82

MS9710C Device Message List (11/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Calibration	Wl-Offset	WOFS n	WOFS?	n n = ±xx.x	Offset wavelength (nm)	9.94
	Wl-Calibration	WCAL n n = 0: W-CAL INITIAL 1: W-CAL1 2: W-CAL2 3: Forced end	WCAL?	m m = 0: Calibration complete 1: During calibration 2: Insufficient optical level 3: Other faults	W-CAL1;EXT.LIGHT W-CAL2;REF.LIGHT	9.92
	Auto Alignment	ALIN n n = 0: ALIGN INITIAL 1: ALIGN 2: Forced end	ALIN?	m m = 0: Calibration complete 1: During processing 2: Insufficient optical level 3: Other faults		9.1
	Lvl-Offset	LOFS n n = ±xx.x x	LOFS?	n n = ±xx.x x	n: Offset value (dB) –30 to +30	9.48
	Res Cal	R CAL n n = 0: INITIAL = 1: EXCUTE	RCAL?	m m = 0: Initial 1: Ended nomally 2: Excuting 3: Ended abnormally		9.66
Condition	Save	CSAV n n = 1 to 5			n: Save memory No.	9.17
	Recall	CRCL n n = 0 to 5		yy, mm, dd	n: Recall memory No. 0: Init	9.16
Time & Date Set	Date	DATE yy, mm, dd	DATE?	hh, mi s	yy: 00 to 99 mm: 01 to 12 dd: 01 to 31 hh: 00 to 23 mi: 00 to 59	9.18
	Time	TIME hh, mi	TIME?	s = ON, OFF		9.83
	Time & Date On/Off	TDSP s s = ON, OFF	TDSP?	R, G, B		9.81
Display Color		LCD P, R, G, B	LCD?P	n n = 1 to 20, 0	P: Screen No. (0 to 10) RGB: 0 to 7	9.46
Auto Backlight		BKL n n = 1 to 20, 0	BKL?	s s = ON, OFF	n: Time (min) until light is turned off	9.12
Buzzer		BUZ s s = ON, OFF	BUZ?			9.13

MS9710C Device Message List (12/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Marker	Trace Marker	TMK λ λ = xxxx.xxxx	TMK?	λ, l λ = xxxx.xxxx λ = xx.xx (dBm , dB) = Four signifi cant digits (*W, %)	λ : Wavelength (nm) or (THz) l : Level dBm, dB, *W, %	9.87
	ΔMarker	DMK λ λ = xxxx.xxxx	DMK?	Δλ, Δl Δλ = xxxx.xxxx Δl = xx.xx (dB) = xxx.xxx	Δλ : Difference in wavelength (nm) or (THz) Δl : Difference in level Log dB, no linear unit	9.27
	Wl Marker A B	MKA λ MKB λ λ = xxxx.xxxx	MKA? MKB?	λ λ λ = xxxx.xxxx	λ Wavelength (nm) or (THz)	9.52 9.53
	Lvl Marker C D	MKC l MKD l l : Valueshown on the right	MKC? MKD?	l l l = xx.xxx (dBm, dB) = Seven significant digits (*W, %)	l : Level dBm, dB, *W, %	9.54 9.55
	Marker Off	EMK				9.32
Zone Marker	Zone Marker	ZMK WL, λc, λs λc = xxxx.xxx λs = xxxx.xxx	ZMK? WL	WL, λc, λs λc = xxxx.xxx λs = xxxx.xxx	λc, λs : Unit (nm) λc : Zone center λs : Zone span	9.96
	Zone→Span	ZMK SPN				
	Zoom In/Out	ZMK ZOOM, s s = IN = OUT	ZMK? ZOOM	ZOOM, s s = IN = OUT		
	Zone Marker Erase	ZMK ERS				
Sweep	Single	SSI				9.76
	Repeat	SRT				9.75
	Stop	SST				9.59
Auto Measurement		AUT	AUT?	n n = 0: Measurement end = 1: During measurement		9.9
Light Output		OPT s s = ON, OFF	OPT?	s s = ON, OFF	Option	9.60

MS9710C Device Message List (13/13)

Item		Device message			Remarks	
		Command	Data request	Response		
Peak→Center		PKC				9.62
TMkr→Center		TMC				9.86
Peak→Level		PKL				9.63
Internal Printer	Copy	CPY				9.15
	Feed	FED n n = 0 to 25			n = Number of character lines	9.40
Memory Data	Data	d+Terminator	DMA? (memory A) DMB? (memory B)	Log scale ±xxx.xx	Log: Unit (dBm)	9.24 9.25
		d+Separator	DQA? (memory A) DQB? (memory B)	Liner scale x.xxxxE ±x	Linear: Unit (mW)	9.29 9.30
		Binary	DBA? (memory A) DBB? (memory B)	LOG: 2 bytes /1 data Linear: 4 bytes/1 data	LOG: ×0.01 (dBm) Linear: ×0.0001 (mW)	9.19 9.20
	Data Condition		DCA? (memory A) DCB? (memory B)	λ1, λ, 2, n λ1 = xxxx.xx λ2 = xxxx.xx n = 251 to 5001	λ1, λ2: Unit (mn) λ1: Start wavelength λ2: Stop wavelength n: Measuring point	9.21 9.22
Measurement Status			MOD?	n n = 0: No measurement of spectrum n = 1: Single sweep of spectrum n = 2: Repeat sweep of spectrum n = 3: Power monitor		9.57
Terminater		TRM 0 TRM 1	TRM?	0 1	0 = LF, EOI 1 = CR, LF, EOI	9.88
Header		HEAD ON HEAD OFF				9.44
Error			ERR?	n n = xxx	n: Error No.	9.33
Extended Event Status Register			ESR1? ESR2? ESR3?	n n n	n: Register value 0 to 255	9.37 9.38 9.39
Extended Event Status Enable Register		ESE1 n ESE2 n ESE3 n	ESE1? ESE2? ESE3?	n n n	n: Register value 0 to 255	9.34 9.35 9.36

Section 4 Initial Setting

The GPIB interface system is initialized at three levels. At level 1, “bus initialization” is performed to place the system bus in the idle state. At level 2, “message exchange initialization” is performed to enable devices to receive program messages. At level 3, “device initialization” is performed to initialize device-dependent functions.
At these three initialization levels, preparations are made for starting devices.

E 488.1 defined the following two levels of GPIB system initialization.

Initialization of bus:
Interface functions of all devices connected to the bus are initialized by an IFC message from the controller.

Initialization of devices:
All devices on the GPIB are initialized with a GPIB bus command “DCL”, or only the specified devices are initialized to their specified states with a GPIB bus command “SDC.”

IEEE 488.2 defines three levels. At level 1, “bus initialization” is performed. This is the highest level. “Device initialization” is divided into “message exchange initialization” (level 2) and “device initialization” (level 3). IEEE 488.2 also defines the device power-on status.

The following table provides a summary of the above explanation.

Level	Initialization type	Overview	Combination and priority of levels
1	Bus initialization	Interface functions of all devices connected to the bus are initialized by an IFC message from a controller.	This level may be combined with other levels. However, initialization at level 1 must be performed before initialization at other levels.
2	Message exchange initialization	Message exchange is initialized and the function of reporting completion of operation to the controller is disabled. This initialization can be ferformed either for all devices on the GPIB using GPIB bus command DCL, or only for the specified devices using a GPIB bus command SDC.	This level may be combined withother levels. However, initialization at level 2 must be performed before initialization at level 3.
3	Device initialization	Only the specified devices on the GPIB are initialized to the known states with an ∗RST command irrespective of the past use state.	This level may be combined with other levels. However, initialization at level 3 must be performed after initialization at levels 1 and 3.

When controlled from a controller via the RS-232C interface port, the MS9710C can use the “device initialization” function (level 3). However, it cannot use “bus initialization” (level 1) and “message exchange initialization” (level 2) functions. When controlled from a controller via a GPIB interface bus, the MS9710C can use all the above initialization functions (levels 1 to 3).

Let’s take a look at the commands for performing initialization at levels 1 to 3 and the items to be initialized as well as the known states set at power-on.

4.1 Initialization of Bus by IFC Statement

■ Format

IFC∆@select-code

■ Application example

IFC @1

■ Explanation

This function can be used when the MS9710C is controlled from a controller via a GPIB interface bus.

On the GPIB corresponding to the specified select code, the IFC line is activated for about 100 µs (electrically set at the low level). When IFC@ is executed, interface functions of all devices connected to the GPIB bus line corresponding to the specified select code are initialized. Only the system controller can send this command.

"Initialization of interface functions" refers to the processing in which controller-set device interface functions (talker, listener, etc.) are reset to their initial states. Functions marked with √ in the following table are initialized. The function marked with ∆ is initialized partially.

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	√

If the IFC statement is true (the IFC line is set at the low level through execution of the IFC@ statement), initialization is not performed at levels 2 and 3. That is, device operating states are not affected.

Let's take a look at some device states set by the IFC statement.

(1) Talker/listener:

All talkers and listeners are set in the idle state (TIDS, LIDS) within 100 μs.

(2) Controller:

If the controller is not active (SACS: System control Active State), it enters the idle state "CIDS" (Controller IDle State) within 100 μs.

(3) Return of control right:

If the system controller (the first device on the GPIB which is used as a controller) has granted the control right to another device when IFC@ is executed, the control right is returned to the system controller. Generally, pressing the [RESET] key on the system controller allows an IFC message to be output from the system controller.

(4) Devices issuing service request:

The state in which an SRQ message is issued by a device (the SRQ line is set at the low level by the device) is not canceled, but the state in which all devices on the system bus are placed in the serial poll mode by the controller is canceled.

(5) Devices in remote state:

For the devices currently in the remote state, the remote state is not canceled by the IFC message.

4.2 Initialization of Message Exchange by DCL and SDC Bus Commands

■ Format

DCL∆@select-code[primary-address][secondary-address]

■ Application example

DCL@1 Initializes message exchange for all devices on the bus. (Issue of DCL)

DCL@103 Initializes message exchange only for the device at address 3. (Issue of SDC)

■ Explanation

This function can be used when the MS9710C is controlled by a controller via the GPIB interface bus.
This statement initializes message exchange for all device on the GPIB corresponding to the specified select code or only for the specified devices.

The purpose of message exchange is to allow the controller to send new commands when the controller cannot control message-exchange-related parts inside the devices due to execution of programs although it is not necessary to change the panel settings.

■ When only a select code is specified

Message exchange is initialized for all the devices on the GPIB corresponding to the specified select code. DCL@ issues a DCL (Device Clear) bus command to the GPIB.

■ When an address is also specified

Message exchange is initialized only for the specified device. Listeners on the GPIB corresponding to the specified select code are canceled, only the specified device is set as a listener, and an SDC (Selected Device Clear) bus command is issued.

■ Items subject to initialization of message exchange

(1) Input buffer and output queue:

Cleared.

(2) Syntax analysis, execution control, and response generation parts:

Reset

(3) Device commands including ∗RST:

All commands interfering with execution of these commands are cleared.

(4) Paired parameter/program message:

All commands and queries whose execution has been suspended due to paired parameters are discarded.

(5) ∗OPC command processing:

The specified device is set in the OCIS (Operating Complete Command Idle State). The operation complete bit cannot be set in the standard event status register.
(Section 7)

(6) ∗OPC? query processing:

The specified device is set in the OCIS (Operating Complete Command Idle State). The operation complete bit cannot be set in the output queue. The MAV bit is cleared.
(Section 7)

(7) Automatic system configuration:

∗ADD and ∗DFL common commands are invalidated. (The MS9710C does not support these commands.)

(8) Device function:

All parts related to message exchange are set in the idle state. The device waits for a message from the controller.

The following operations are prohibited.

(1) Changing the current device settings and stored data

(2) Interrupting front panel I/O

(3) Changing status bits other than the MAV bit when clearing the output queue

(4) Affecting or interrupting the device operation currently being performed

■ Orders of issue of GPIB bus commands using DCL@ statements

Orders of issue of GPIB bus commands using DCL@ statements are summarized below.

Statement	Bus command issue order (ATN line: Low level)	Data (ATN line: High level)
DCL @ select-code	UNL, DCL	———
DCL @ device-number	UNL, LISTEN address, [secondary-address], SDC	———

4.3 Initialization of Devices by ∗RST Command

■ Format

∗RST

■ Application example

WRITE @103:"∗RST"

Only the device at address 3 is initialized at level 3.

■ Explanation

The ∗RST(Reset) command, an IEEE 488.2 common command, is used to reset a specified device at level 3.

Generally, devices are set in various states using device-dependent commands (device messages). Among these commands, the ∗RST command is used to reproduce a known state of a device. Completion of device operation is invalidated like level 2.

■ Specification of device number in WRITE @ statement

The device at the specified address is initialized at level 3.

■ Items subject to device initialization

(1) Device-dependent functions and states:

The specified device is set in a known state irrespective of its history. (See the lists on the following pages.)

(2) ∗OPC command processing:

The specified device is set in the OCIS (Operation Complete Command Idle State). The operation complete bit cannot be set in the standard event status register. (Section 7)

(3) ∗OPC? query processing:

The specified device is set in the OCIS (Operating Complete Command Idle State). The operation complete bit cannot be set in the output queue. The MAV bit is cleared. (Section 7)

(4) Macro command:

Macro operation is disabled, setting the state in which macro commands cannot be accepted. The designer can show macro definitions.

Notes:

∗RST command does not affect the following items.

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 standard
8. RS-232C interface condition

Table 4-1 lists MS9710C-dependent initial settings.
The “Set condition” column lists device’s initial states set by the ∗RST command. In the “Battery backup” column, items battery-backed-up after power-off are marked with √.

Table 4-1 MS9710C-dependent initial settings (1/3)

Item group	Item	Set condition	Battery backup
Wavelength	Center	1350 nm	√
	Span	500 nm	√
	Start	1100 nm	√
	Stop	1600 nm	√
	Mkr Value	Wl	√
	Value in	Air	√
Level Scale	Scale	Log	√
	Log/div	10 dB/div	√
	Reference Level	+20 dBm	√
	Linear Level	100 mW	√
	Att On/Off	Off	√
Res/VBW/Avg	Res	1.0 nm	√
	VBW	1 kHz	√
	Point Avg	Off	√
	Sweep Avg	Off	√
	Smooth	Off	√
	Sampling Points	501	√
	Act Res	Off	√
Peak/Dip Search	Status	Off	√
Analysis	Status	Off	√
	Threshold	Cut Lvl: 3 dB	√
	ndB Lossnd	B: 3 dB	√
	SMSR	Side Mode: 2nd Peak	√
	Envelope	Cut Lvl: 3 dB	√
	RMS	k: 2.35, S.Level: 20 dB	√
Save/Recall	File Option	File Option: None	√
		File ID: Number	√
		FDD Mode: 1.44 M	√
Graph	Status	Normal	√
	3D	Type: 1, Angle: 45 deg	√

Table 4-1 MS9710C-dependent initial settings (2/3)

Item group	Item	Set condition	Battery backup
Application	Status	Off	√
	DFB-LD	ndB Width: 20 dB	√
		Side Mode: 2nd Peak	√
	FP-LD	Mode Cut Lvl: 3 dB	√
	LED	ndB Width: 3 dB	√
		Power Cal: 0 dB	√
	PMD	Auto/Manual: Auto	√
		Mode Cpl Factor: 1	√
		Peak Count: 2	√
	O.Amp	NF Select: S-ASE	√
		Spect Div: On	√
		ASE Fitting: Gauss Fit	√
		Fitting Span: 5 nm	√
		Masked Span: 2 nm	√
		Pin Loss: 0 dB	√
		Pout Loss: 0 dB	√
		NF Cal: 1	√
		O.BPF Lvl Cal: 0 dB	√
		O.BPF BW: 3 nm	√
		Pol Loss: 0 dB	√
	WDM	Display Mode: Multi Peak	√
		Peak S.Level: 30 dB	√
		Dip Prmtr: Higher	√
		Δ l : Off	√
		Ref No.: 1	
		Page Top No.: 1	
Measure Mode	D.range	Normal	√
	Peak Hold	Off	√
		Gate Time: 1 msec	√
	Ext Trigger	Off	√
		Delay Time: 0 µsec	√
	Interval Time	0sec	√
	TLS Tracking	Off	√
	Power Monitor	Off	
		Wavelength: 1550 nm	√
Title			√
CaL	Wl Offset	0 nm	√
	Level Offset	0 dB	√
Others	Printer Prmtr	Device Type: Int	√
		Device Address: 17	√
	Back Light	On	√
		Time: 10 min	√

Table 4-1 MS9710C-dependent initial settings (3/3)

Item group	Item	Set condition	Battery backup
Status Register	Service request enable register	0 (All inhibited)	
	Standard event status enable register	0 (All inhibited)	
	Extended event status enable register	0 (All inhibited)	

4.4 Device States at Power-on

When the power is turned on.

(1) The MS9710C is restored to the last power-off state.

(2) The input buffer and output queue are cleared.

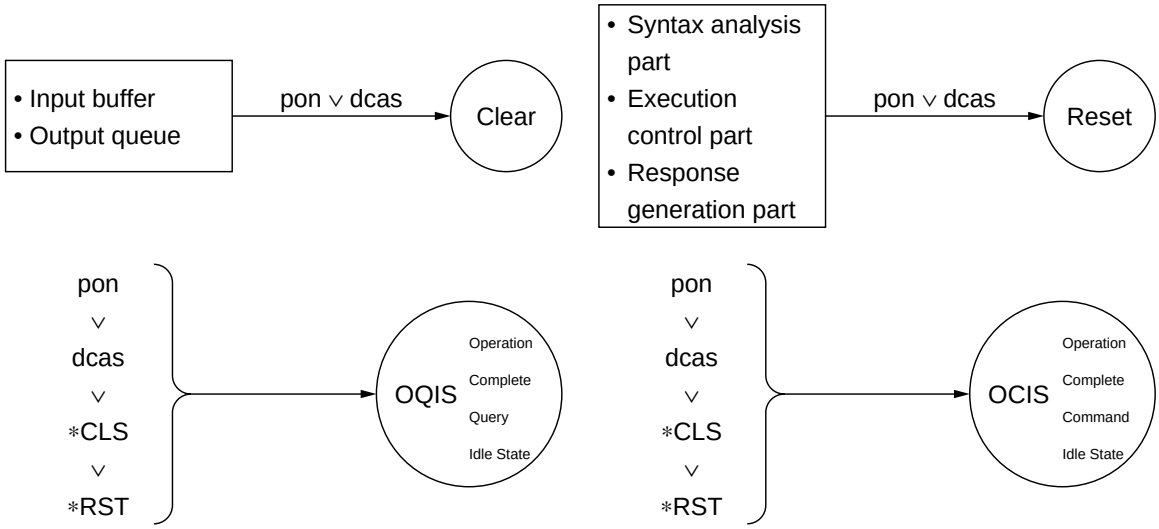
(3) Syntax analysis, execution control, and response parts are reset.

(4) The device is set in the OCIS (Operation Complete Command Idle State).

(5) The device is set in the OQIS (Operation Complete Query Command Idle State).

(6) The MS9710C does not support a ∗PSC command. So the standard event status register and standard event status enable register are cleared. Events are recorded after being cleared.

States (2) to (5) are set except when the power is turned on. The state diagram is shown below.



■ Items not changes at power-on

(1) Address

(2) Associated calibration data

(3) Data and states that change with the responses to the following common query commands

∗IDN?	( Section 7)
∗OPT?	( Section 7)
∗PSC?	(Not supported by the MS9710C)
∗PUD?	(Not supported by the MS9710C)
∗RDT?	(Not supported by the MS9710C)

■ Items related to power-on status clear (PSC) flag

When the PSC flag is false, the service request enable register ( Section 8.3), standard event status enable register ( Section 8.4), and parallel poll enable register are not affected.

When the PSC flag is true or the ∗PSC command has not been executed, the above registers are not cleared.

( The PSC command is not supported by the MS9710C)

■ Items that change at power-on

(1) Current device function test

(2) Status information

(3) ∗SAV/∗RCLregister (Not supported by the MS9710C)

(4) Macro definition made with a ∗DDT command
(Not supported by the MS9710C)

(5) Macro definition made with a ∗DMC command
(Not supported by the MS9710C)

(6) Macro definition made with an ∗EMC command
(Not supported by the MS9710C)

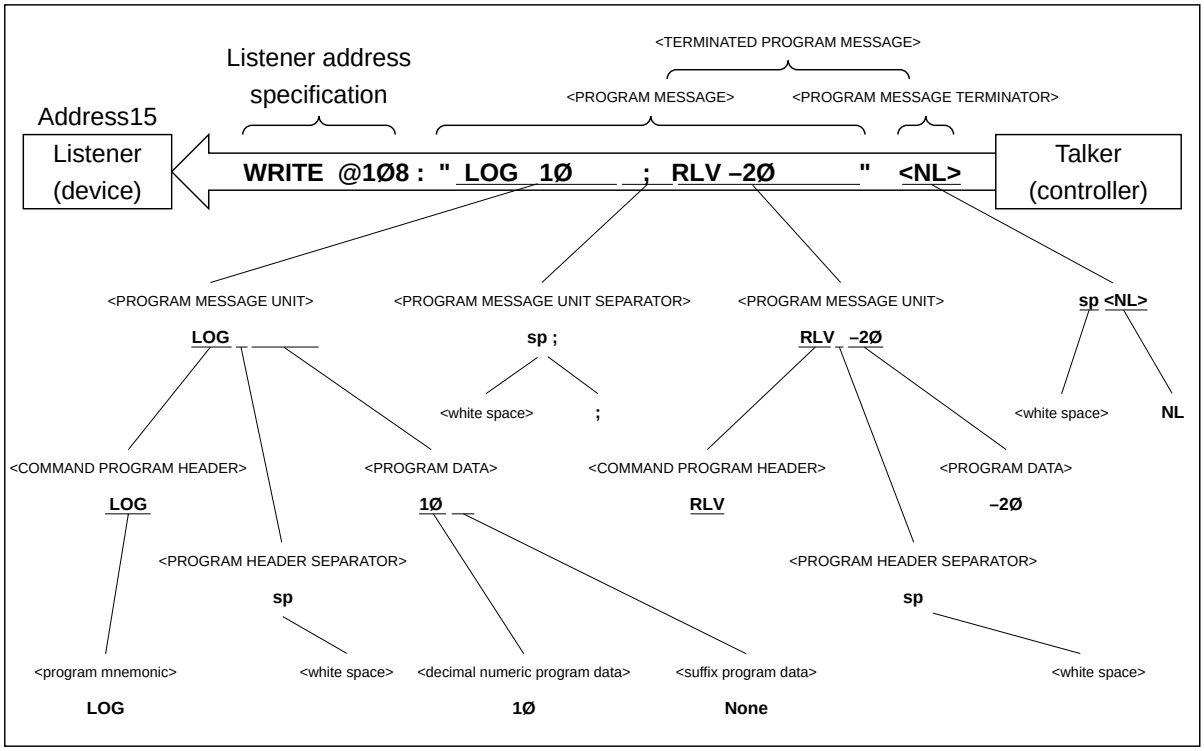
(7) Address received with a ∗PCB command (Not supported by the MS9710C)

Section 5 Listner Input Formats

Device messages transferred between the controller and devices are classified into program messages and response messages. This section explains the formats of the program messages received by listeners.

A program message is a sequence of program message units. Each unit is a program command or query.

The following figure shows that a program message made by connecting two program messages LOG 10 and RLV –20 with a program message unit separator is sent from a controller to a device to set the log scale to 10 dB/div and the reference level to –20 dBm.



A program message is a sequence of functional elements, the minimum units that can represent functions. In the above figure, functional elements are indicated by capital characters with them enclosed in < >. Functional elements are further classified into coding elements which are indicated by lowercase characters with them enclosed in < >.

The chart indicating the route of selection of functional elements is called a functional syntactical chart. The chart indicating the route of selection of coding elements is called a coding syntactical chart. On the following pages, program message formats are explained using these functional and coding syntactical charts.

Coding elements indicate coding of the actual bus which is required to send functional element data byte to a device. Upon receipt of a functional element data byte, the listener checks whether individual elements follow the coding syntax rules. If they do not follow the rules, the listener causes a command error without regarding the elements as functional elements.

5.1 Summary of Listener Input Program Message Syntactical Notation

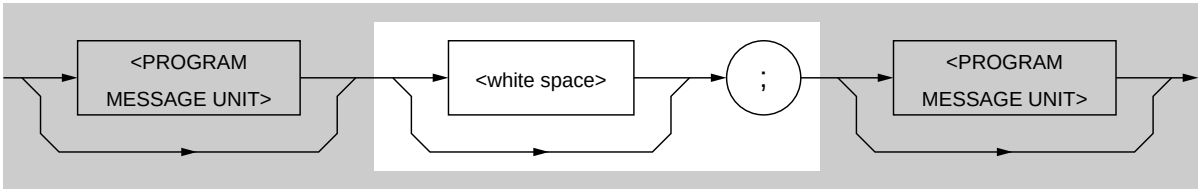
This section gives a general description of program messages functional units (Section 5.2) and program data formats (Section 5.3). (Compound commands and common commands are excluded.)

5.1.1 Separator, terminator, and space before header

(1) PROGRAM MESSAGE UNIT SEPARATOR

Link two or more program message units using zero or more spaces and a semicolon.

<Example 1> General format for linking two program message units



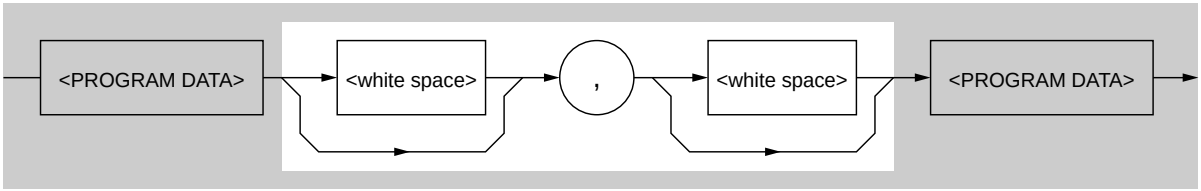
<Example 2> One space + Semicolon

LOGΔ10Δ ;RLVΔ–20 LOG 10 ;RLV –20: Set the log scale to 10 dB/div and the reference level to –20 dBm.

(2) PROGRAM DATA SEPARATOR

When there are two or more pieces of program data, separate two contiguous pieces of program data using zero or more spaces, a command, and zero or more spaces.

<Example 1> General format for separating two pieces of program data



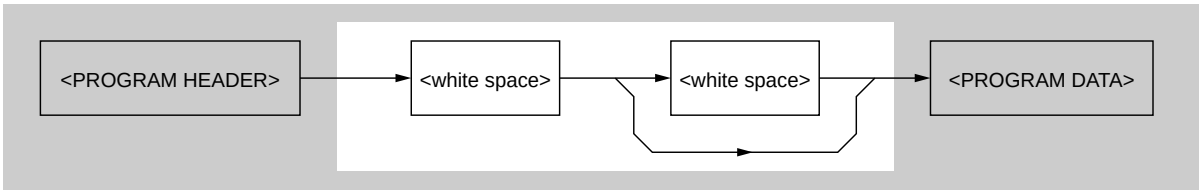
<Example 2> Comma only **<Example 3> Comma + One space**

TIMEΔ10, 15 TIME Δ10, Δ15 Set the times to 10:15.

(3) PROGRAM HEADER SEPARATOR

Separate a program header and program data using one space and zero or more spaces.

<Example 1> General format of simple command program header



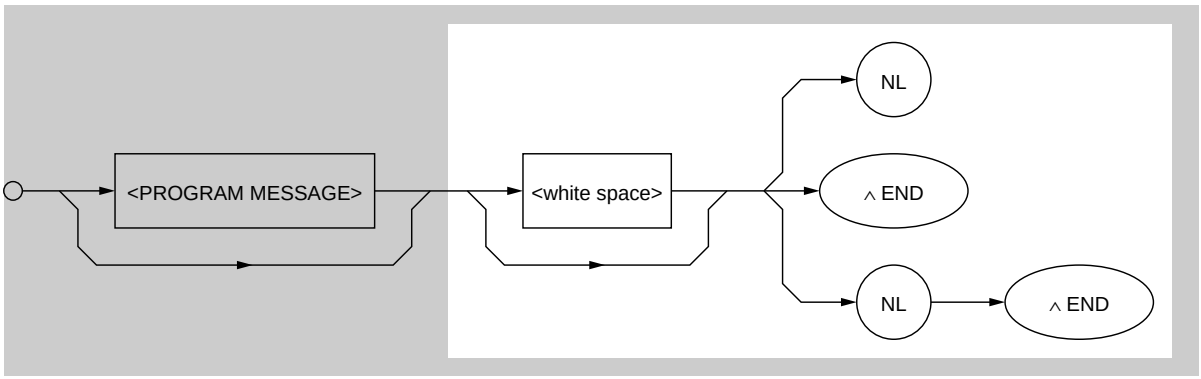
<Example 2> One space

LLV Δ10mW

(4) PROGRAM MESSAGE TERMINATOR

Add zero or more spaces and one of NL, EOI and a combination of NL and EOI at the end of a program message.

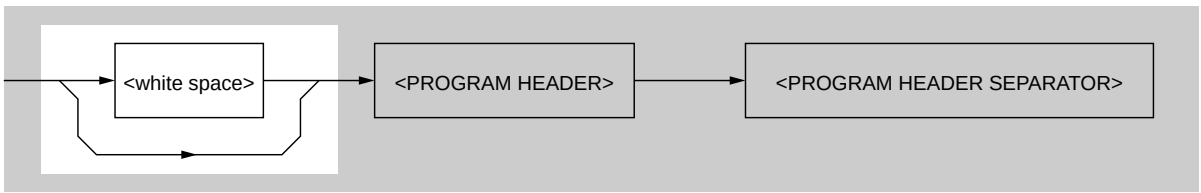
<General format>



(5) Space before header

Zero or more spaces can precede a program header.

<General format>

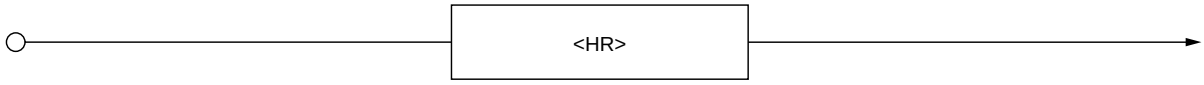


<Example> One space before second program header RLV

LOG Δ10 ;Δ RLV Δ –20 Set the log scale to 10 dB/div and the reference level to –20 dBm.

5.1.2 General format of program command message

(1) Message without data specification

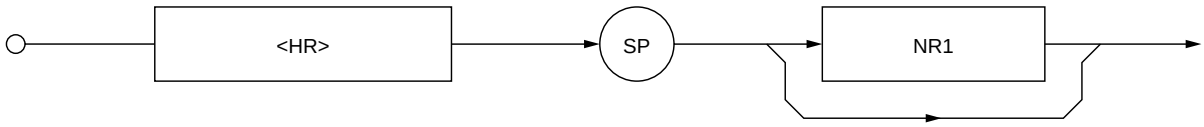


HR: COMMAND PROGRAM HEADER

<Examples>

AUT	Automatic setting
SSI	Single sweep start

(2) Message with integer data

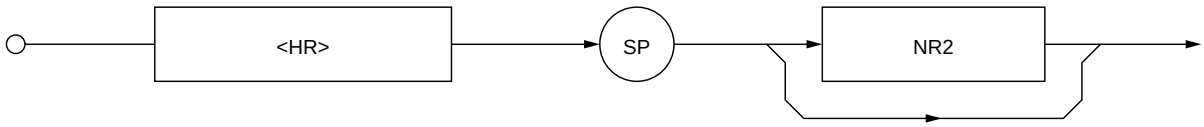


NR1: Integer

<Example>

AVT Δ 500	Set the point average count to 500.

(3) Message with real number

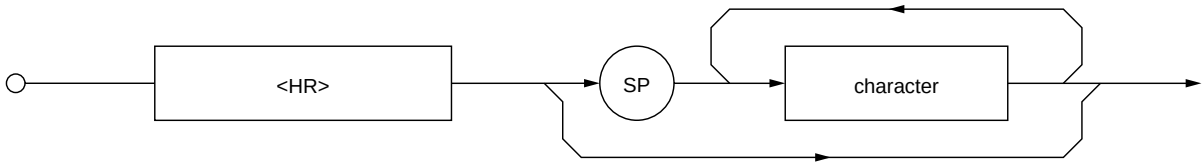


NR2: Real number

<Example>

CNT Δ 1305.8	Set the center wavelength to 1305.8 nm.

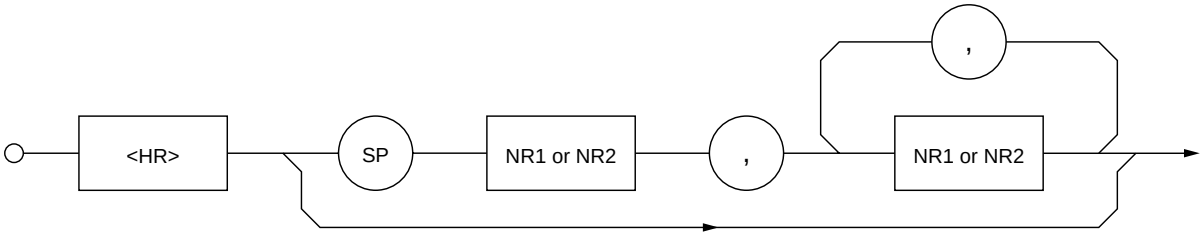
(4) Message with fixed or arbitrary character string data (data length ≤12 characters)



<Example>

DMD Δ NRM	Set the measurement mode to NORMAL.
MSL Δ A	Select memory A.

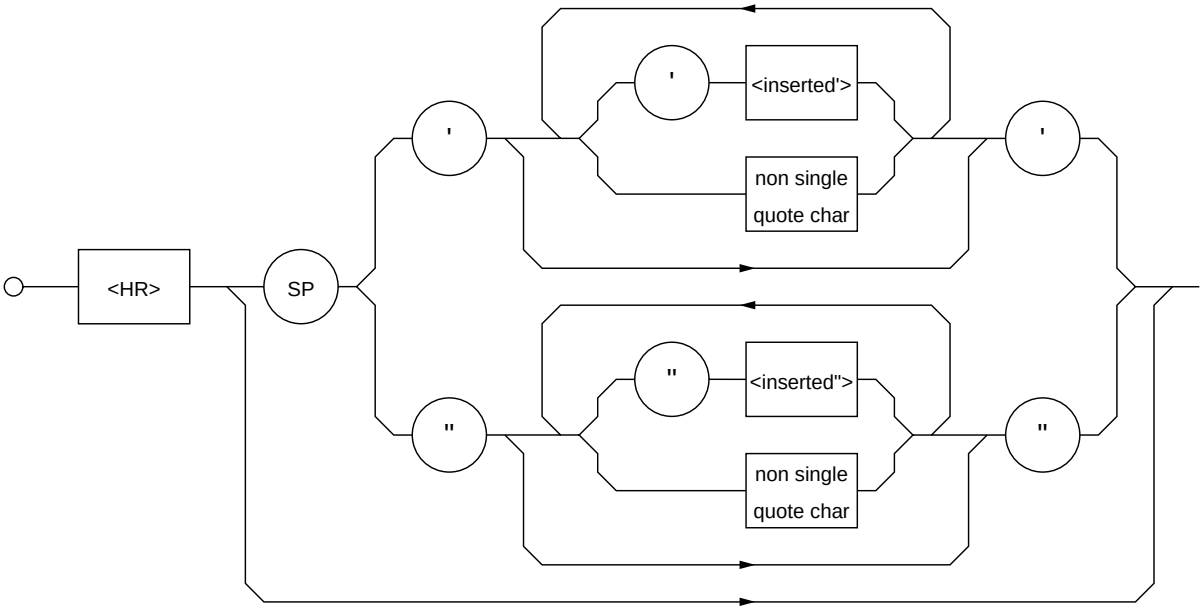
(5) Message with multiple pieces of program data (first: NR1)



<Example>

DATE Δ 96, 10 Set the date to Oct. 10, 1996.

(6) Character-only message that can use all seven ASCII bits



<inserted'>: A single ASCII code representing a value 27
non-single quote char: A single ASCII code representing a value other than 27
<inserted">: A single ASCII code representing a value 22
non-single quote char: A single ASCII code representing a value other than 22

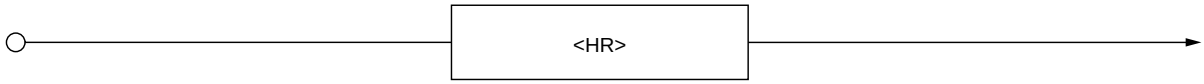
<Example>

TTL Δ "ABC" Set a title "ABC."

5.1.3 General format of query message

Add ? at the end of a query program header.

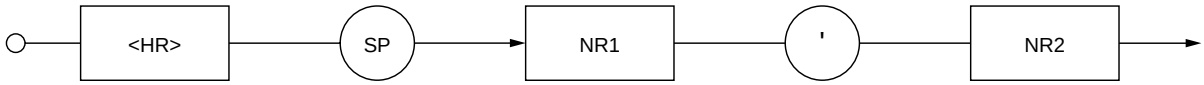
(1) Message without query data specification



<Example>

CNT? Request output of a center wavelength value.

(2) Message with query data specification



<Example>

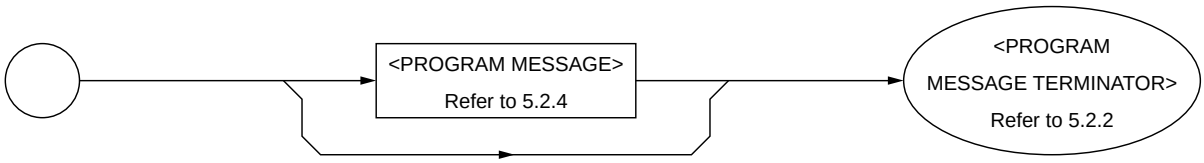
LCD? Δ 1 Inquire about the display color of display screen No.1.

5.2 Program Message Functional Elements

A device accepts a program message by detecting the terminator added at the end of the program message. Functional elements of the program message is described below.

5.2.1 <TERMINATED PROGRAM MESSAGE>

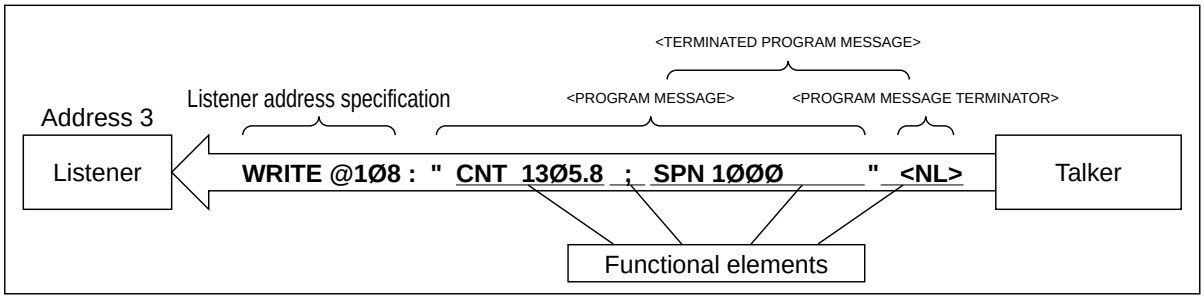
<TERMINATED PROGRAM MESSAGE> is defined as follows:



<TERMINATED PROGRAM MESSAGE> is a data message having all the necessary functional elements to be sent from a controller to a device.

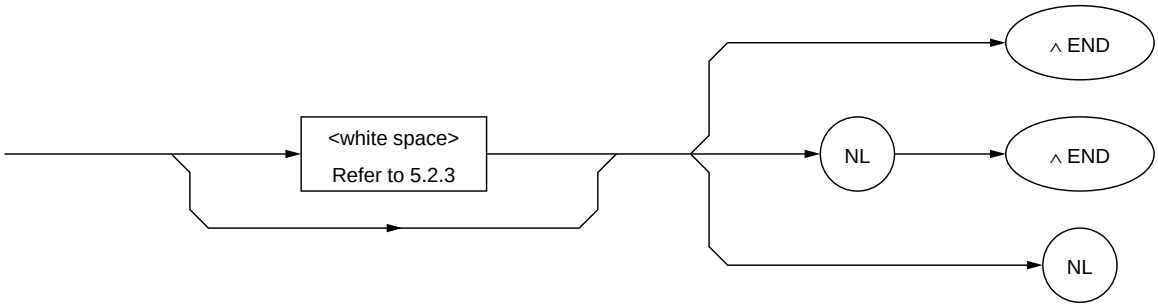
To complete transfer of <PROGRAM MESSAGE>, <PROGRAM MESSAGE TERMINATOR> is added at the end of <PROGRAM MESSAGE>.

<Example> <TERMINATED PROGRAM MESSAGE> for sending two pieces of commands with a WRITE statement



5.2.2 <PROGRAM MESSAGE TERMINATOR>

<PROGRAM MESSAGE TERMINATOR> is defined as follows.



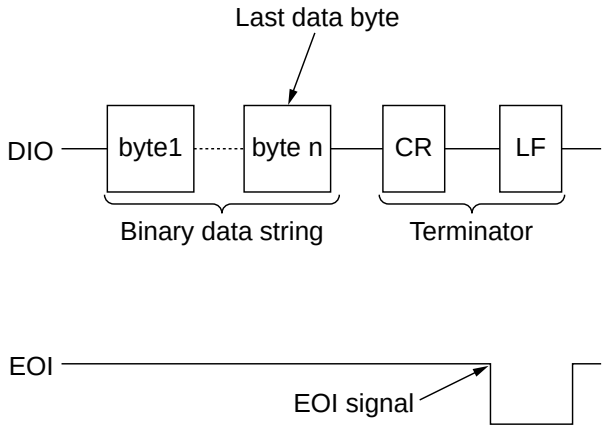
<PROGRAM MESSAGE TERMINATOR> terminates a sequence of one or more fixed-length <PROGRAM MESSAGE UNIT> elements.

NL: Defined as a single ASCII code byte 0A (decimal 10). That is, it is an ASCII control character LF (Line Feed) that moves the printing position down one line. As printing starts at a new line, it is also called NL (New Line). When sending <PROGRAM MESSAGE> with a WRITE@ statement, the WRITE@ statement automatically issues CR/LF. So the CR/LF codes need not be written in the program. To generate only the LF code, the following statement must be executed at the beginning of the program.

```
TERM IS CHR $ (10)
```

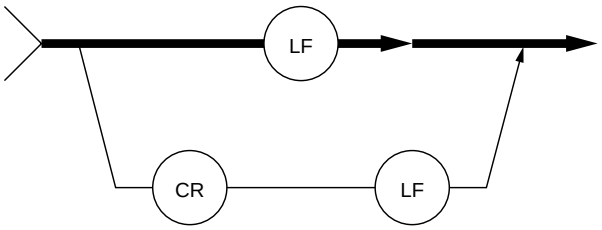
END: Sets the EOI line, one of GPIB control buses, at the LOW level (TRUE), generating an EOI signal.

An EOI ON/OFF statement can be used to control the EOI line. EIO OFF is the default (the EOI line is not controlled). If the EOI ON statement is executed in advance, an EOI signal is issued along with the terminator LF when the last byte of the WRITE@ statement is issued. It is also possible to terminate <PROGRAM MESSAGE> using only an END signal without generating an LF code.



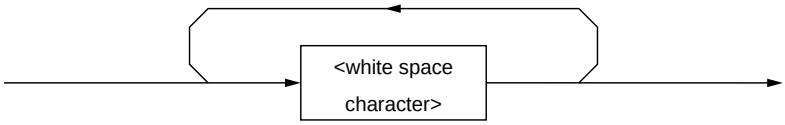
Note:

The CR code is used to return the printing position to the first character position on the same line; however, most listeners ignore it. Some products available on the market uses CR-LF code, so most controllers are so designed that CR and LF codes are issued in succession.



5.2.3 <white space>

<white space> is defined as follows.

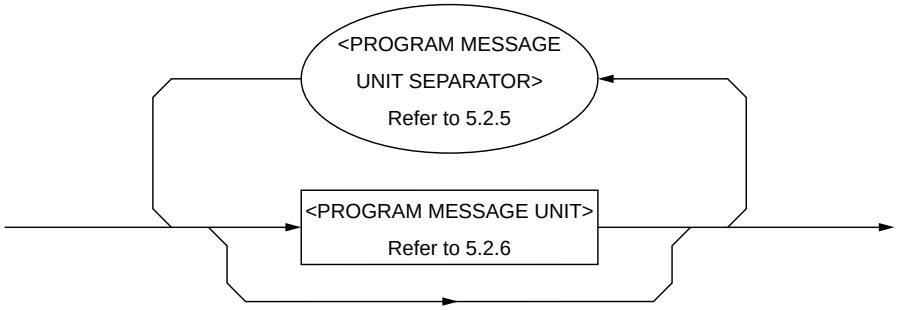


<white space character> is one of ASCII code bytes 00 to 09 and 0B to 20 (decimal values 0 to 9 and 11 to 32).

This range includes ASCII control codes and space signals (except NL). The device does not regard these codes as ASCII control codes, but it regards them as spaces or skips them.

5.2.4 <PROGRAM MESSAGE>

<PROGRAM MESSAGE> is defined as follows.

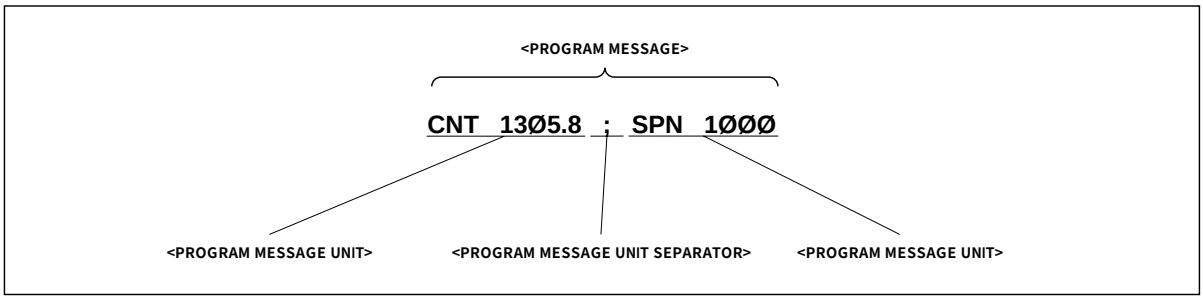


<PROGRAM MESSAGE> is zero, a <PROGRAM MESSAGE UNIT> element, or a sequence of <PROGRAM MESSAGE UNIT> elements. A <PROGRAM MESSAGE UNIT> element is a programming command or data which is sent from a controller to a device.
A <PROGRAM MESSAGE UNIT SEPARATOR> element is used to separate two or more <PROGRAM MESSAGE UNIT> elements.

<Example 1> Program message for setting the center wavelength to 1.3058 μm

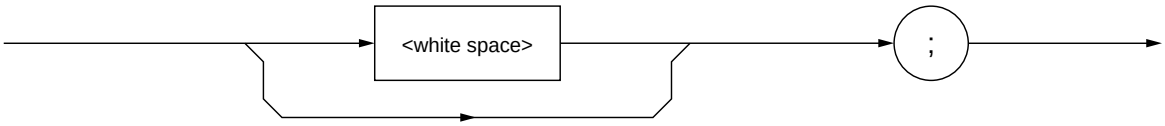
CNT 1305.8

<Example 2> Program message for setting the span to 1000 nm

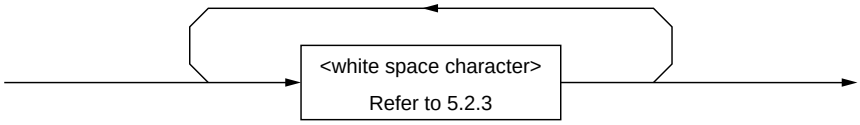


5.2.5 <PROGRAM MESSAGE UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT SEPARATOR> is defined as follows.



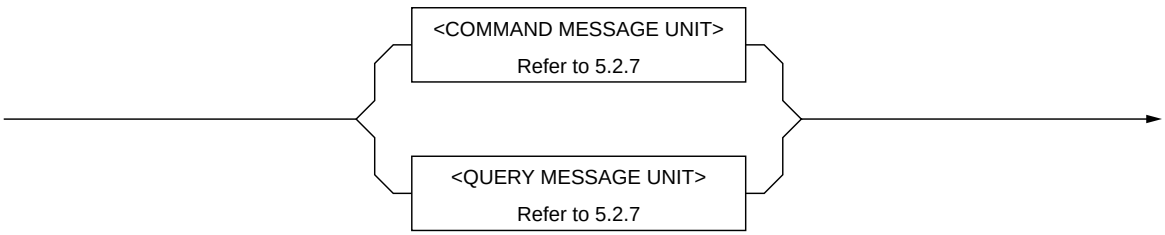
<white space> is defined as follows.



<PROGRAM MESSAGE UNIT SEPARATOR> divides a sequence of <PROGRAM MESSAGE UNIT> elements within the <PROGRAM MESSAGE> range.
A device interprets a semicolon (;) as the separator between <PROGRAM MESSAGE UNIT> elements. Accordingly, <white space character> before and after the semicolon are ignored. It should be noted that <white space character> improves program readability. <white space> following a semicolon is also used as a <white space> for the next program header. (See <Example 2> on the Section 5.2.4 or Section 5.2.8.)

5.2.6 <PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT> is defined as follows:

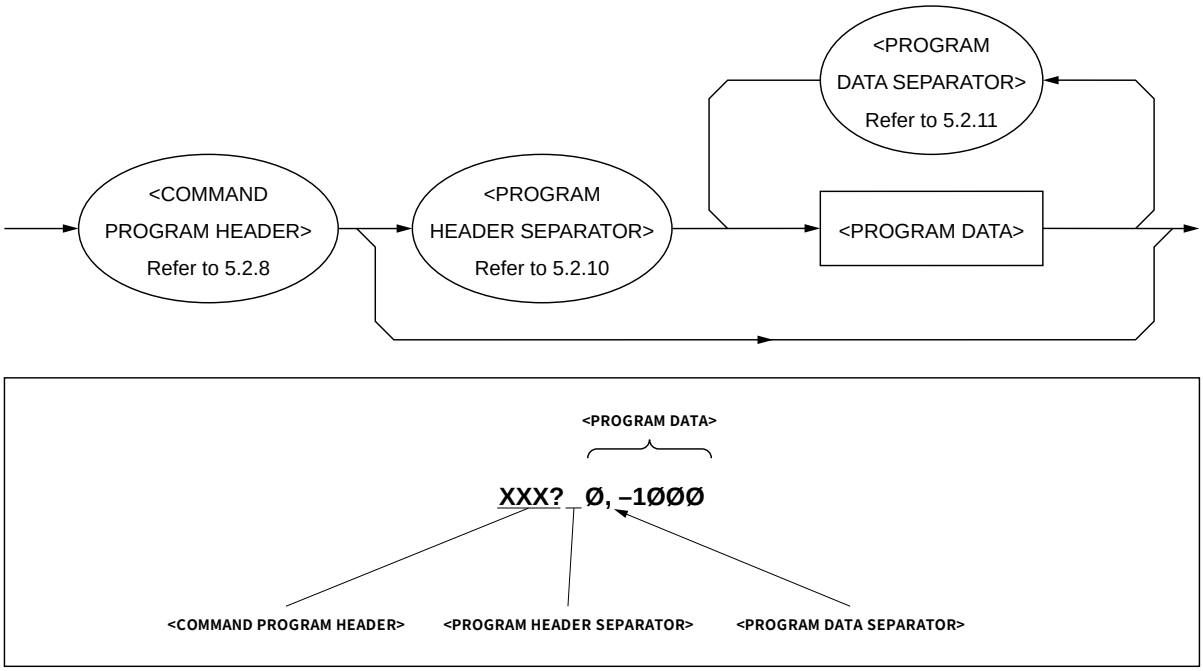


<PROGRAM MESSAGE UNIT> is a single command message received by a device.
It consists of <COMMAND MESSAGE UNIT> or <QUERY MESSAGE UNIT>, a single query message.

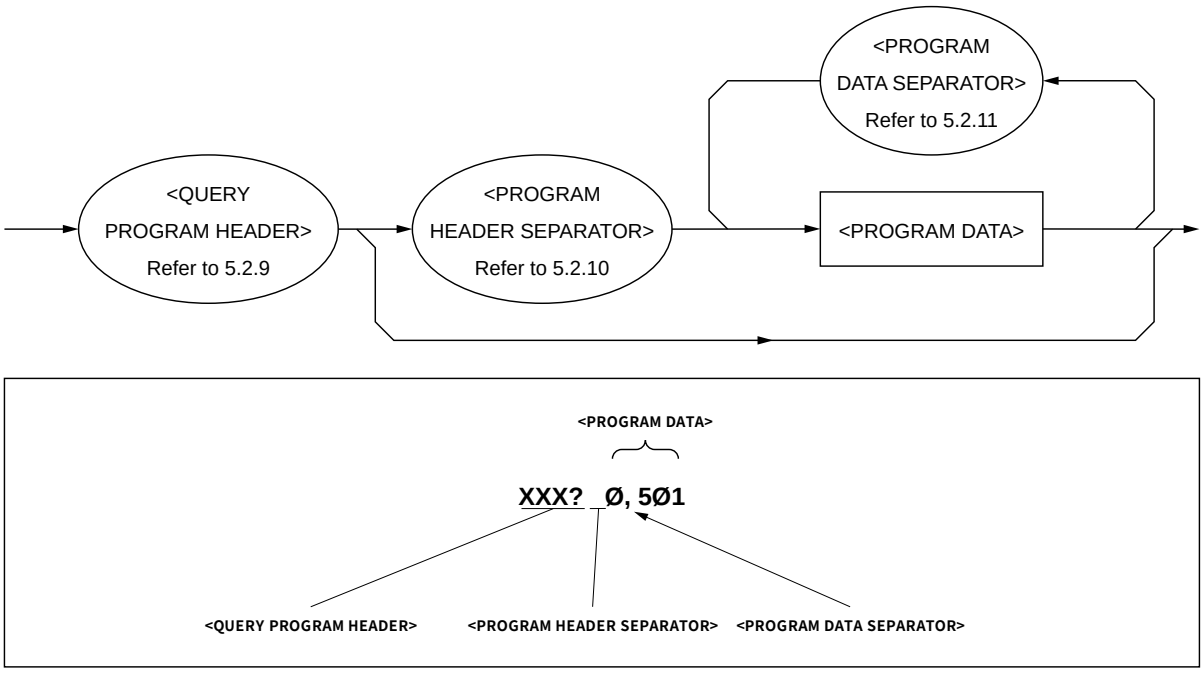
For details on <COMMAND MESSAGE UNIT> and <QUERY MESSAGE UNIT>, see the next page.

5.2.7 <COMMAND MESSAGE UNIT>/<QUERY MESSAGE UNIT>

(1) <COMMAND MESSAGE UNIT> is defined as follows.



(2) <QUERY MESSAGE UNIT> is defined as follows.

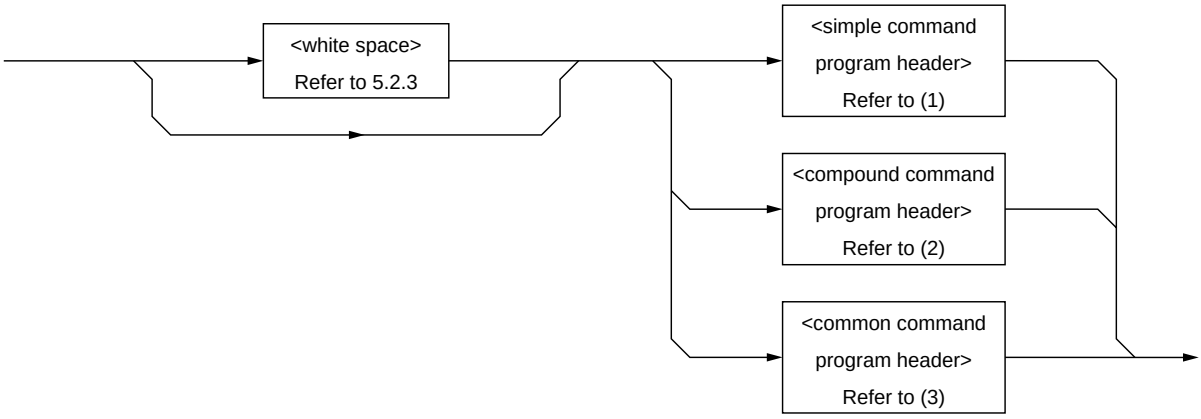


When a program header <COMMAND MESSAGE UNIT> or <QUERY MESSAGE UNIT> is followed by program data, a space is inserted between them. A program header indicates the application, function, and operation of the program. If a program header is not followed by program data, the program header solely indicates the application, function, and operation to be performed in the device.

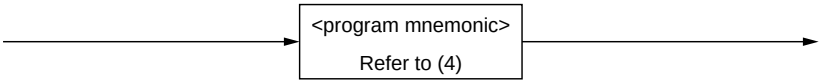
Among program headers, <COMMAND PROGRAM HEADER> is a control command issued from a controller to a device and <QUERY PROGRAM HEADER> is a query command that is issued from a controller to a device in advance so that the controller can receive responses from the device. These headers always end with a query indicator "?".

5.2.8 <COMMAND PROGRAM HEADER>

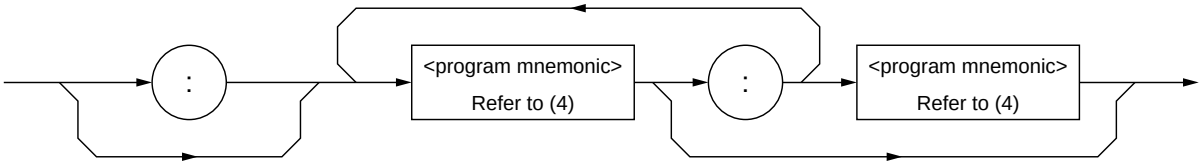
<COMMAND PROGRAM HEADER> is defined below.
Each header can be followed by <white space>.



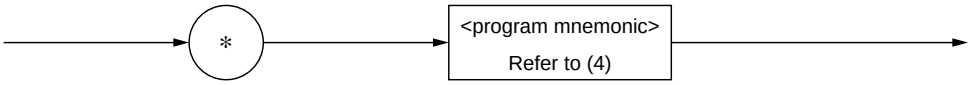
(1) <simple command program header> is defined as follows.



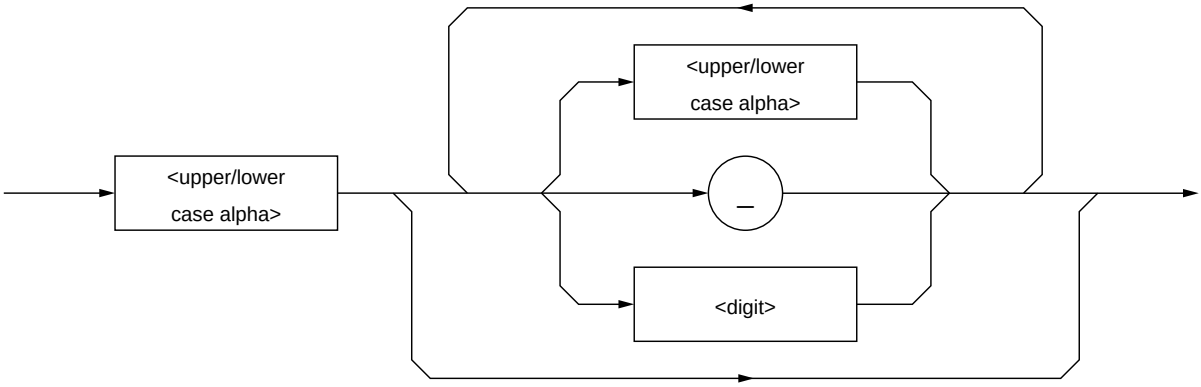
(2) <compound command program header> is defined as follows.



(3) <common command program header> is defined as follows.



(4) <program mnemonic> is defined as follows.



■ <COMMAND PROGRAM HEADER>

This element indicates the application, function, and operation of the program data to be executed by the device. When it is not followed by program data, the header solely indicates the application, function, and operation to be performed in the device.

The meanings of an application, function, or operation is represented by <program mnemonic> which is widely called a mnemonic. Mnemonics and the command program headers defined in (1) to (3) above are explained below.

■ <program mnemonic>

A mnemonic begins with an uppercase or lowercase character, which is followed by an arbitrary combination of characters such as uppercase characters (A to Z) or lowercase characters (a to z), underline (_), and numeric characters (0 to 9). A mnemonic can contain a maximum of 12 characters; however, most mnemonics contain 3 to 4 characters. (No space is inserted between characters.)

- <upper/lower case alpha> One of ASCII code bytes 41 to 5A and 61 to 7A (decimal values 65 to 90 and 97 to 122 = uppercase characters A to Z and lowercase characters a to z). The device can accept a header irrespective of whether it is represented by uppercase or lowercase characters.

- <digit> One of ASCII code bytes 30 to 39 (decimal values 48 to 57 = characters 0 to 9).

- (_) An ASCII code byte, i.e., ASCII code byte 5F (decimal value 95 = underline).

■ <simple command program header>

The above rules for <program mnemonic> applies. For example, the MS9710C uses “SSI” as a mnemonic indicating “sweep.” It is also used as a “simple command program header” which means execution of sweep without program data. “CNT” is a mnemonic which means a center wavelength; however, it can be used as a “simple command program header” to set a center wavelength only when it is provided with the program data indicating a center waveform.

■ <compound command program header>

<compound command program header> is a command program header that executes a compound function. <program mnemonic> is always preceded by a colon (:) to separate it from <compound command program header>. When only one <compound command program header> is used, the succeeding colon (:) may be omitted.
The MS9710C does not support this compound command program header. However, it is explained here taking into account future extension.

● Function

On a complex device, a device command set is organized logically by providing a compound function instead of limiting the number of unique headers. A hierarchical command structure can be handled effectively.

● <Example 1>

To allow the MS9710C to use all device commands of another model (e.g., model MSXXXX), the compound program header would be

:MSXXXX

● <Example 2>

To allow the MS9710C to use a WXYZ device command of another model (e.g., model MSXXXX), the compound program header would be

MSXXXX:WXYZ or :MSXXXX:WXYZ

● <Example 3>

The name of a white buck rabbit living in a FOREST is WHITE.

The name of a white doe rabbit living in a GROVE is WHITE, too. If only WHITE is used as a command, we cannot distinguish between the above rabbits.

FOREST:WHITE or :FOREST:WHITE ..... White buck rabbit
GROVE:WHITE or :GROVE:WHITE..... White doe rabbit

■ <common command program header>

An asterisk (∗) is always added before <program mnemonic> of <common command program header>. “Common” means that this command is a program command which commonly used for other IEEE 488.2-ready measuring instruments connected to the bus.

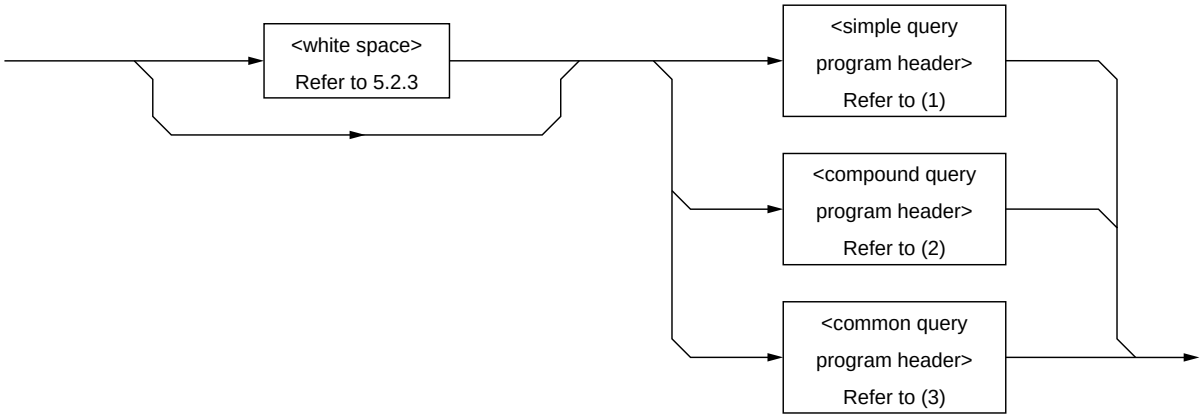
● <Example>

To idle completion of operation of the device at address 8, which is connected to the GPIB interface corresponding to select code 1, and restore devices to their initial states, the following common command is used:

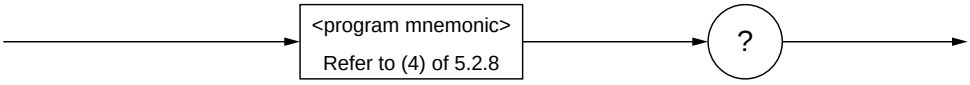
WRITE @108:"∗RST" ..... The character string enclosed with quotation marks (" ") is an IEEE 488.2 common command ∗RST for executing the above processing.

5.2.9 <QUERY PROGRAM HEADER>

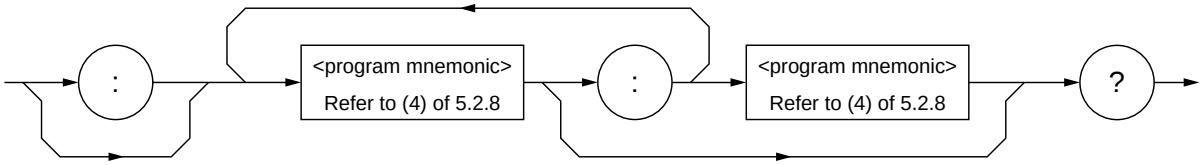
<QUERY PROGRAM HEADER> is defined as follows:
<white space> may be written before each header.



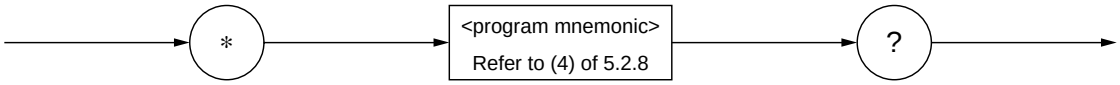
(1) <simple query program header> is defined as follows:



(2) <compound query program header> is defined as follows:



(3) <common query program header> is defined as follows:



■ <QUERY PROGRAM HEADER>

<QUERY PROGRAM HEADER> is a query command which is sent from a controller to a device in advance so that the controller can receive response messages from the device. This header always ends with a query indicator "?". It is explained below using examples of programs.

☞ The format of <QUERY PROGRAM HEADER> is the same as that of <COMMAND PROGRAM HEADER> with the exception that a query indicator "?" is added at the end. See Section 5.2.8.

● <Example 1> Setting and reading a center wavelength

```
10  WRITE @108:"CNT 1000"
20  WRITE @108:"CNT?"! ..... Query message CNT?
30  READ  @108:A
40  PRINT A;"nm"
```

Line 10: A command header CNT for setting a center wavelength and a program message consisting of program data 1. 1000 nm is set for the device.

Line 20: A program message that requires the device to send the set 1000 nm to the controller. A query header "CNT?" is used.

Line 30: The listener device MS9710C that received the query header "CNT?" from the controller becomes a talker. The device is a controller that has become a listener, and it sends a response message 1000 in response to CNT?. The listener reads the response message into the numeric variable A.

Line 40: The wavelength "1000 nm" is displayed on the CRT. However, if HEAD ON is specified with a HEAD command, "CNT 1000" is sent.

● <Example2> Reading measurement data on 501 measuring points from memory A and printing the measurement data

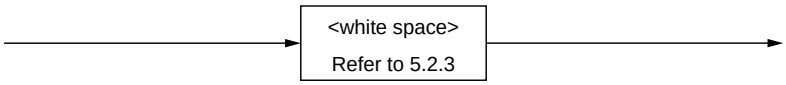
```
100  WRITE @108:"DMA?"
110  FOR K=0 TO 500
120  READ @108:DT(K)
130  PRINT DT(K);"dBm"
140  NEXT
150  END
```

Line 100: A query message "DMA?" is sent to the listener to store 501 pieces of data, starting at address 0.

Line 120: Line 100 causes the device to reply, response messages at points 0 to 500 are sent to the controller, and they are read into a numeric array variable DT (K).

5.2.10 <PROGRAM HEADER SEPARATOR>

<PROGRAM HEADER SEPARATOR> is defined as follows.



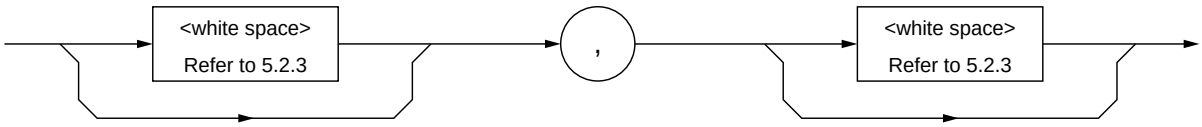
<PROGRAM HEADER SEPARATOR> is used as the separator between <COMMAND PROGRAM HEADER> (or <QUERY PROGRAM HEADER>) and <PROGRAM DATA>.

When there are two or more <white space character> elements between the program header and the program data, the first <white space character> is interpreted as a separator and the remaining <white space character> is ignored. It should be noted that <white space character> improves program readability.

That is, at least one header separator must exist between the header and the data. It indicates both the end of the program header and the beginning of the program data.

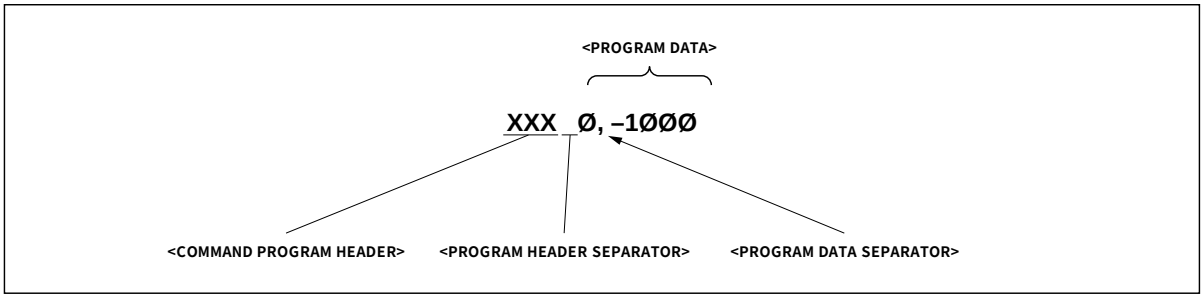
5.2.11 <PROGRAM DATA SEPARATOR>

<PROGRAM DATA SEPARATOR> is defined as follows.



When <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> has many parameters, <PROGRAM DATA SEPARATOR> is used to separate them.

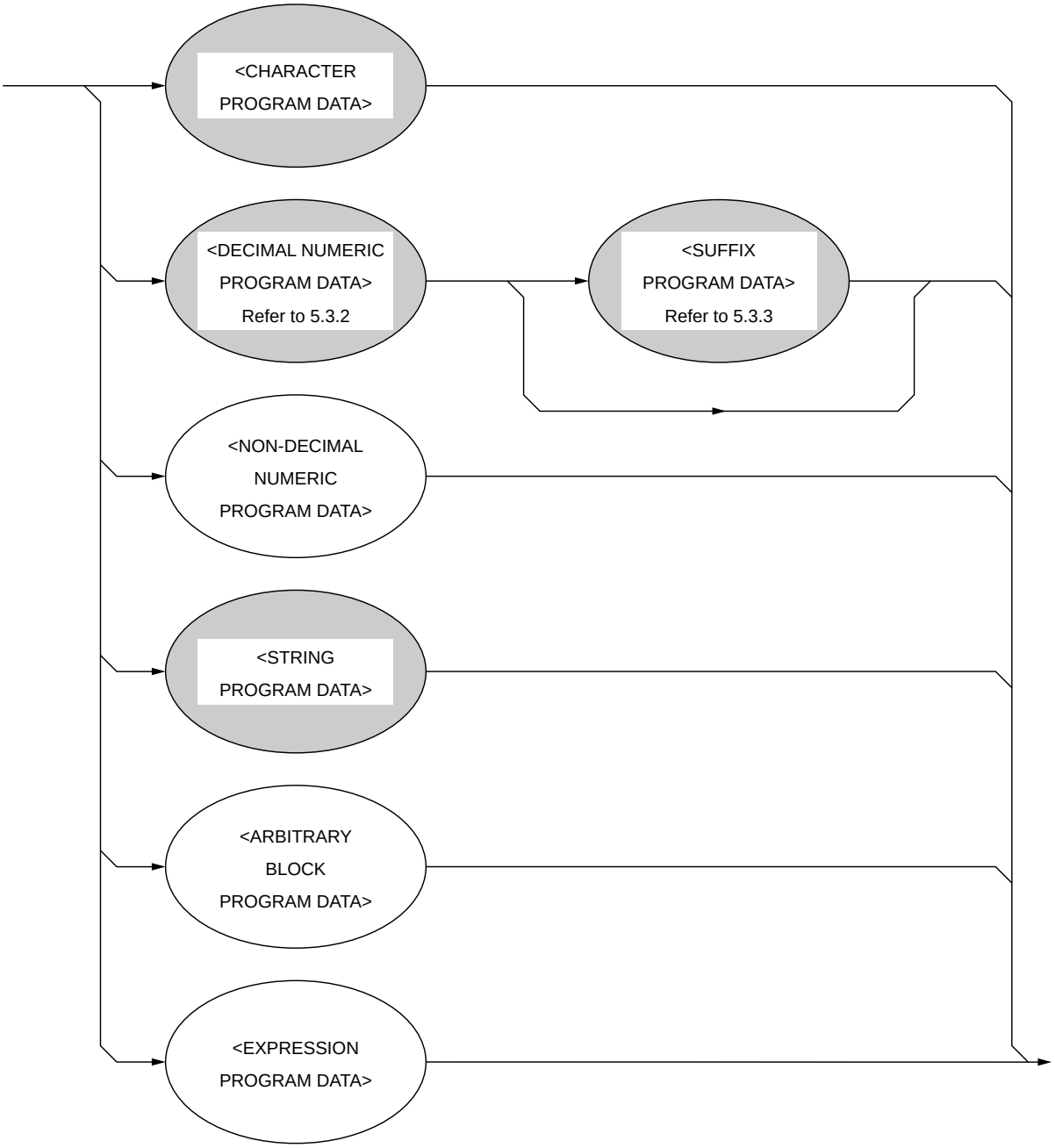
When this data separator is used, a comma is mandatory but <white space character> is omissible. The <white space character> before a comma and the <white space character> after a comma are ignored. It should be noted that <white space character> improves program readability.



5.3 Program Data Format

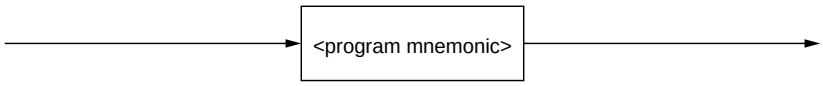
This section explains the format of the <PROGRAM DATA> shown in the functional syntactical charts (Section 5.2.7), which is one of terminated pr`ogram message formats.

The functional element <PROGRAM DATA> is used to transfer various types of parameters related to the program header. Program data types are shown below. The MS9710C accepts the program data shown in the hollow squares surrounded by a shade. For the program data not supported by the MS9710C, read this section just for reference.

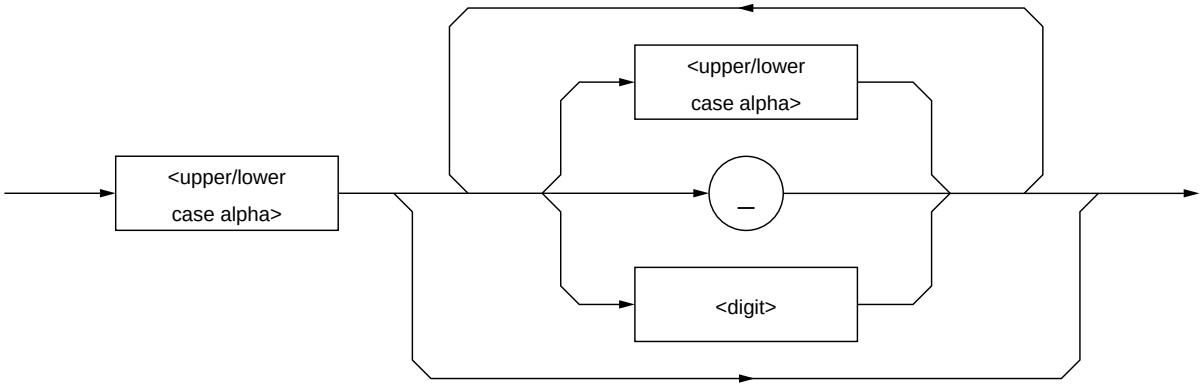


5.3.1 <CHARACTER PROGRAM DATA>

The functional element <CHARACTER PROGRAM DATA> is used to perform remote control by transferring short alphabetic or alphanumeric data. It is defined as follows.



Details on character data are the same as those on program mnemonics. So far, we discussed control data focusing on numeric data. However, program data can also be used to perform control. A coding syntactical chart is as follows.



Data always begins with an uppercase or lowercase character, which is followed by an arbitrary combination of characters such as uppercase characters (A to Z) or lowercase characters (a to z), underline (_), and numeric characters (0 to 9). Since combinations of alphanumeric characters are used as mnemonic-like symbols, the maximum data length is 12 characters.

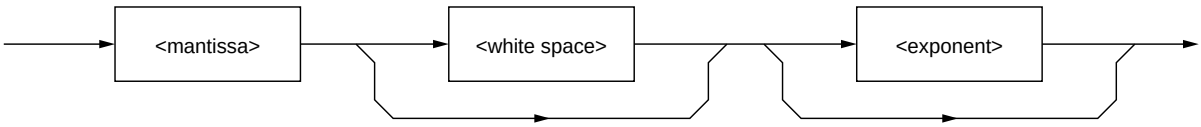
- <upper/lower case alpha> One of ASCII code bytes 41 to 5A and 61 to 7A (decimal values 65 to 90 and 97 to 122 = uppercase characters A to Z and lowercase characters a to z). The device can accept a header irrespective of whether it is represented by uppercase or lowercase characters.
- <digit> One of ASCII code bytes 30 to 39 (decimal values 48 to 57 = characters 0 to 9).
- (_) A single ASCII code byte, i.e., ASCII code byte 5F (decimal value 95 = underline).

Therefore, <CHARACTER PROGRAM DATA> is program data used to transfer relatively short mnemonic-type alphanumeric codes.

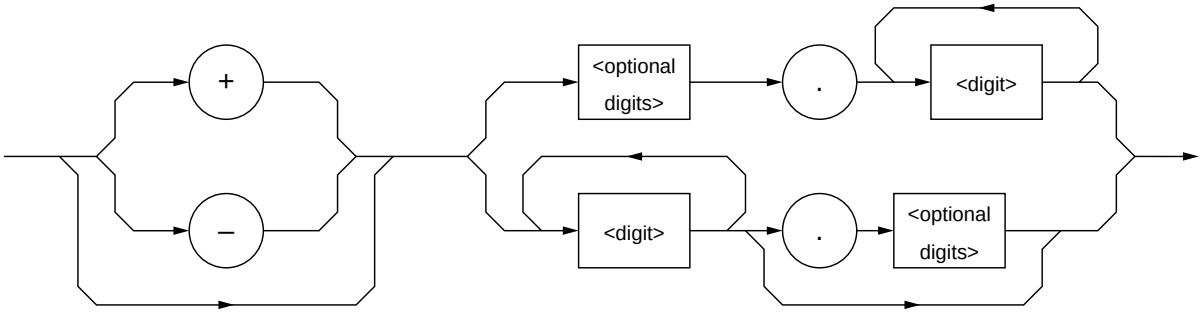
5.3.2 <DECIMAL NUMERIC PROGRAM DATA>

<DECIMAL NUMERIC PROGRAM DATA> is program data used to transfer numeric constants represented in decimal notation. There are three types of decimal numeric representation: integer, fixed- point, and floating-point.

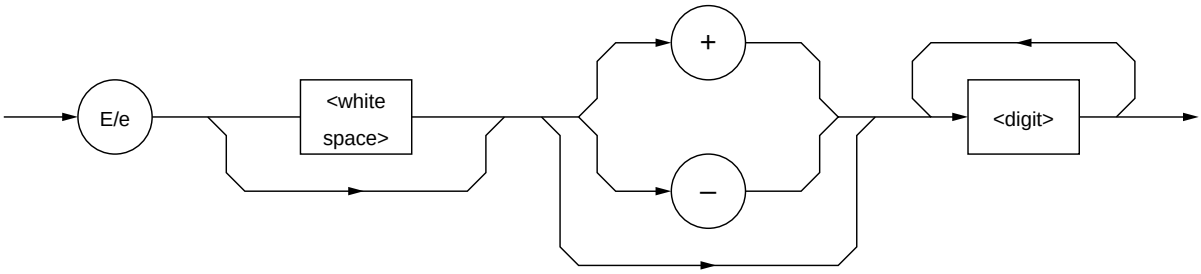
These three types of numerics represent decimal numeric program data, which can contain spaces, flexibly (NRF: flexible numeric representation), so they are defined as follows.



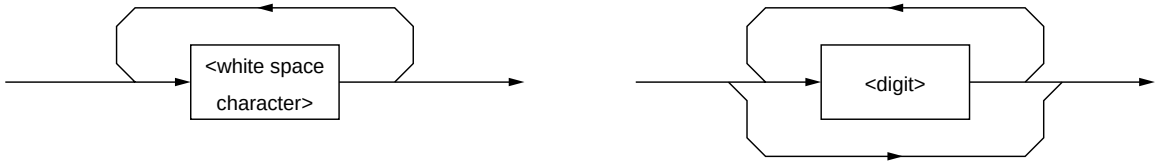
<mantissa> is defined as follows.



<exponenet> is defined as follows.



<white space> and <optional digits> are defined as follows.



☞ For <white space>, see Section 5.2.3. For <digit>, see Section5.3.1.

Let's take a look at coding syntactical charts of decimal numeric program data with respect to integer, fixed-point, and floating- point notations respectively.

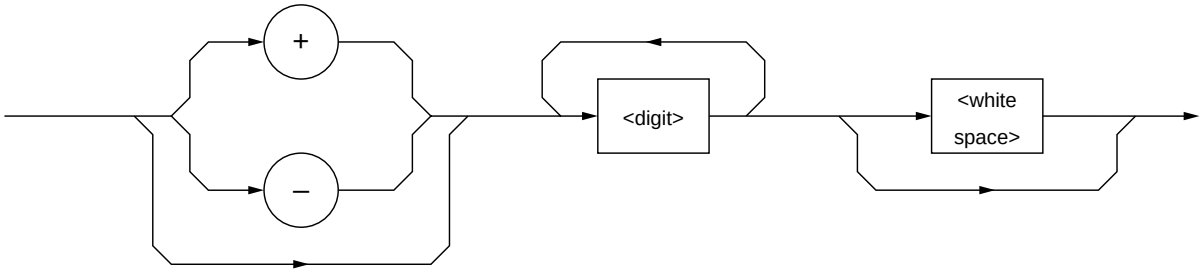
Note that the following processing is performed during transfer of any type of numeric representation.

- Rounding of numeric element: When a device receives a <DECIMAL NUMERIC PROGRAM DATA> element having too many digits to handle, it ignores the sign of the element value and rounds it off.

- Data outside the range: If the <DECIMAL NUMERIC PROGRAM DATA> element value is outside the range permitted in relation to the program header, an execution error is reported.

(1) Integer NR1 transfer

A decimal value not including a decimal point and exponent, i.e., an integer (NR1) in a real number, is transferred.

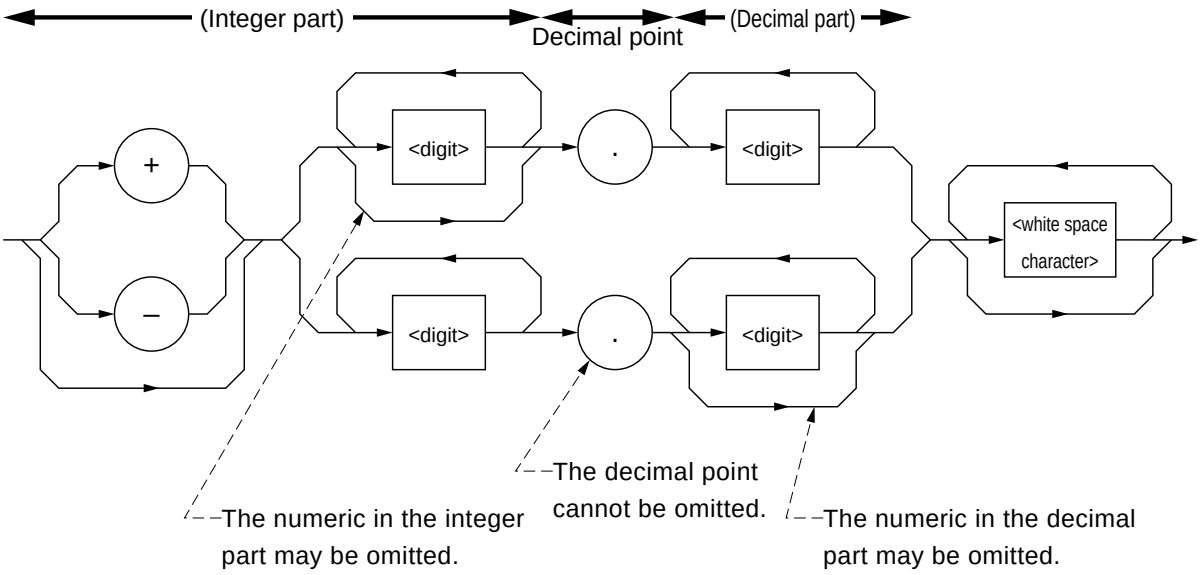


- 0 (s) may be added at the beginning. → 005, + 000045
- A space (+ or –) must not be inserted between a sign and a numeric. → + 5, + Δ5 (×)
- Spaces may be added after a numeric. → + 5ΔΔΔ
- The + sign may be omitted. → + 5, 5
- Commas must not be used to indicate decimal places. → 1,234,567 (×)

(2) Fixed-point NR2 transfer

A decimal number having digits below the decimal point, i.e., an integer and a real number (NR2) except an exponent, is transferred.

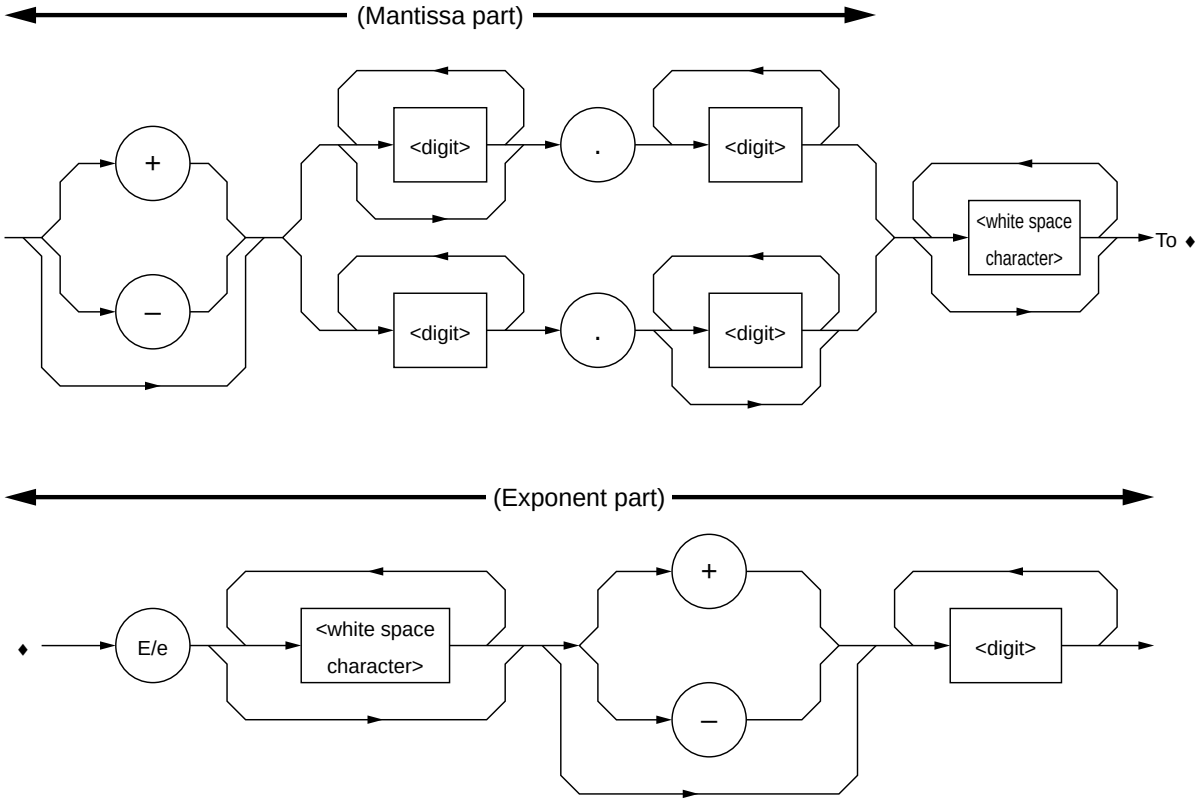
The syntactical chart shows an integer part and a decimal point (and a decimal part).



- An integer representation is applied to the integer part.
- A space must not be inserted between a numeric and a decimal point. → +753Δ.123 (×)
- Spaces may be added after the numeric in the decimal part. → + 753.123 ΔΔΔΔ
- The decimal point need not follow a numeric. → .05
- A sign may be written before a decimal point. → + .05, –.05
- A numeric may end with a decimal point. → 12.

(3) Floating-point NR3 transfer

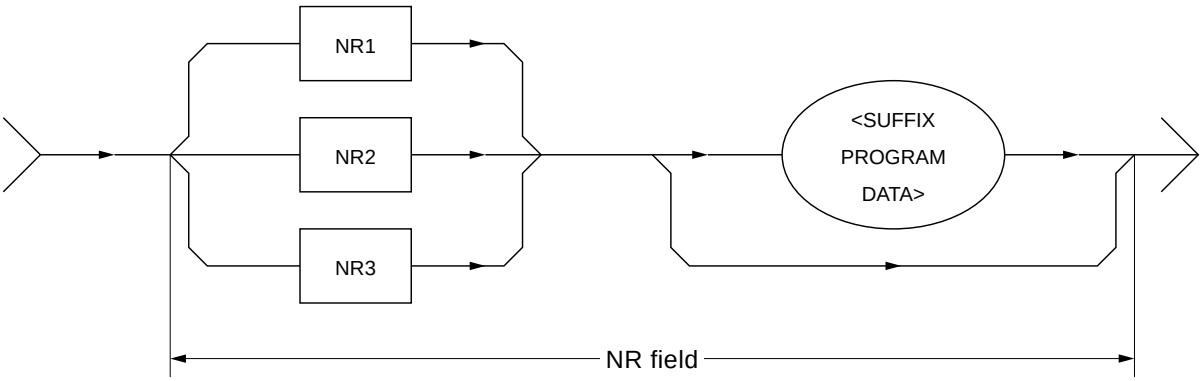
A decimal numeric having an exponent, i.e., a real number (NR3) represented in floating-point notation, is transferred. The syntactical chart consists of a mantissa part and an exponent part. The exponent part is represented in integer and floating-point notation to indicate precision of the numeric. The exponent part begins with E. On the right of E is a number to the power of 10.



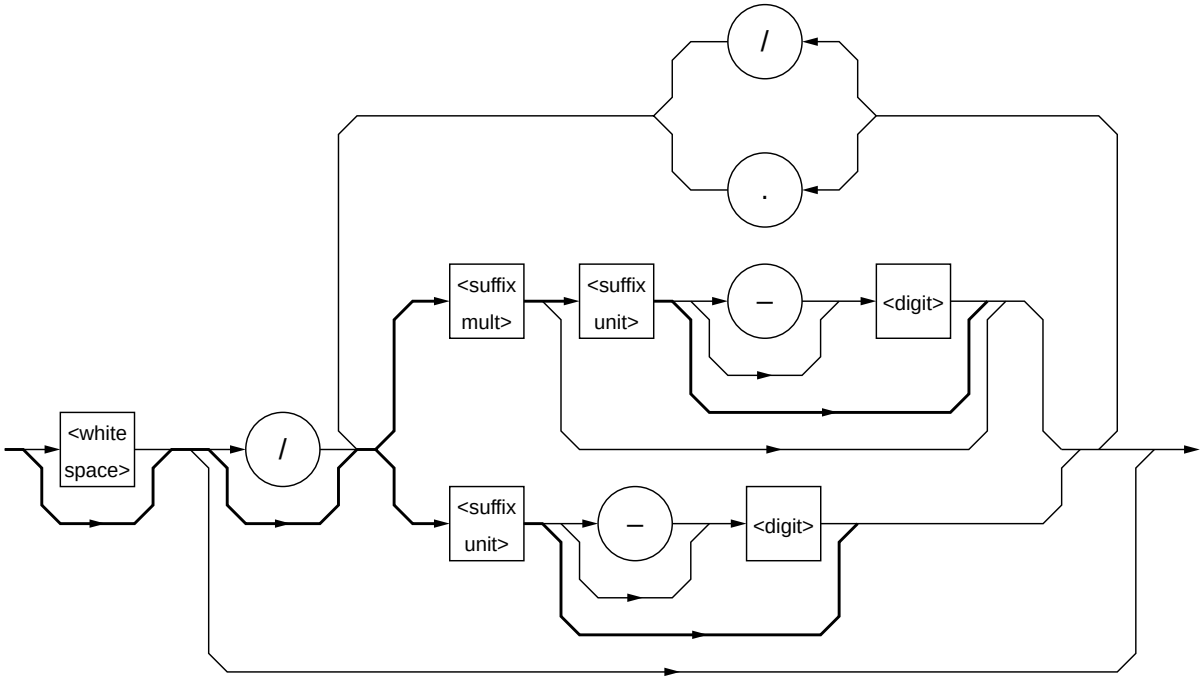
- E indicates power of 10. It indicates the beginning of the exponent part.
- E may be either an uppercase or lowercase character. → 1.234E + 12, 1.234e + 12
- A space may be written before or after E/e. → 1.234 ΔEΔ + 12
- If the sign is +, it may be omitted in mantissa and exponent parts. → + 1.234E + 4, 1.234E4
- The numeric in the exponent part cannot be omitted. → –1E2, –E2 (×), –.E2 (×)

5.3.3 <SUFFIX PROGRAM DATA>

<SUFFIX PROGRAM DATA> follows <DECIMAL NUMERIC PROGRAM DATA> (integer NR1, fixed-point NR2, or floating-point NR3). The NR1, NR2, and NR3 may be followed by a suffix.



A suffix is added at the end of decimal numeric program data only when the data requires a unit of measure. It is a combination of a suffix unit and a suffix multiplier. The syntactical chart is shown below. Bold-line routes are used frequently.



- A suffix multiplier is represented by an uppercase or lowercase character.
  For example, 1E3 Hz is represented by 1 kHz assuming 1E3 = k.
- A suffix unit is represented by an uppercase or lowercase character.
- Placing E at the beginning of <SUFFIX PROGRAM DATA> is prohibited because it may be confused with the E used for floating-point decimal numerics.

Suffix multipliers and units are listed in the table below.

(1) Suffix multipliers

Table 5-1 Suffix multipliers

Multiplier	Mnemonic	Name
1E18	EX	EXA
1E15	PE	PETA
1E12	T	TERA
1E9	G	GIGA
1E6	MA (NOTE)	MEGA
1E3	K	KILO
1E-3	M (NOTE)	MILLI
1E-6	U	MICRO
1E-9	N	NANO
1E-12	P	PICO
1E-15	F	FEMTO
1E-18	A	ATTO

Note:

According to convention, Hz to the sixth power of 10 is MHz (mega-hertz) and OHM to the six power of 10 is MOHM (megaohm). These are not listed in the above table, but they are listed in Table 5-2, “Suffix units.”

(2) Relative units (dB)

- Decibel relative to 1 μV ..... DBUV
- Decibel relative to 1 μW ..... DBUW
- Decibel relative to 1 mW ..... DBMW

(3) Suffix units

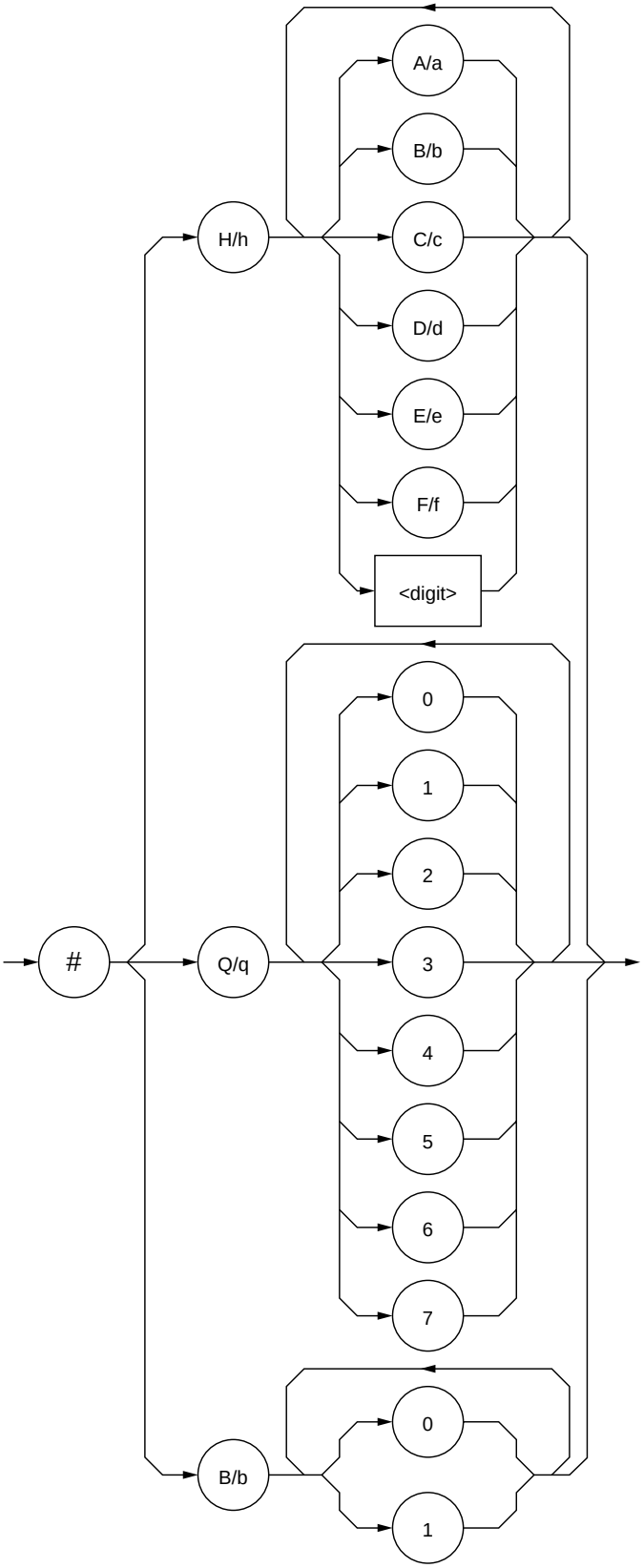
Table 5-2 Suffix units

Item	Recommended mnemonic of unit	Quasi recommended mnemonic of unit	Name
Current	A		Ampere
Atmospheric pressure	ATM		Atmosphere
Charge	C		Coulomb
Luminance	CD		Candela
Decibel	DB		Decibel
Power	DBM		Decibel milliwatt
Capacitance	F		Farad
Mass		G	Gram
Inductance	H		Henry
Frequency (hertz)	HZ		Hertz
Mercury column	INHG		Inches of mercury
Joule	J		Joule
Temperature	K	 CEL FAR	Degree Kelvin Degree Celsius Degree Fahrenheit
Volume	L		Liter
Luminance	LM		Lumen
Luminance	LX		Lux
Length (meter)	M	 FT IN	Meter Feet Inch
Frequency (1E3 Hz)		MHZ	Megahertz
Resistance		MOHM	Megaohm
Force	N		Newton
Resistance	OHM		Ohm
Pressure	PAL		Pascal
Ratio (percent)	PCT		Percent
Angle (radian)	RAD		Radian
Angle (degree)		DEG MNT	Degree Minute (of arc)
Time (second)	S	SEC	Second
Conductance	SIE		Siemens
Automatic speed	T		Tesla
Pressure	TORR		Torr
Voltage	V		Volt
Power (watt)	W		Watt
Speed/hour	WB		Weber
Luminance	LM		Lumen

5.3.4 <NON-DECIMAL NUMERIC PROGRAM DATA>

<NON-DECIMAL NUMERIC PROGRAM DATA> is program data used to transfer decimal, octal, and binary numeric data as non-decimal numeric values. Non-decimal data always begins with #. It is defined as shown in the coding syntactical chart below.

When an unspecified character string is sent, a command error occurs.



The character string following #H or #h is accepted by the device as a hexadecimal number.
The character strings in parentheses are decimal numbers.

#Habc1230	(11,256,099D)
#hAbC123	
#H2DC3	(11,715D)
#h2dc3	
#H8301	(33,537D)
#h8301	

The character string following #Q or #q is accepted by the device as an octal number.

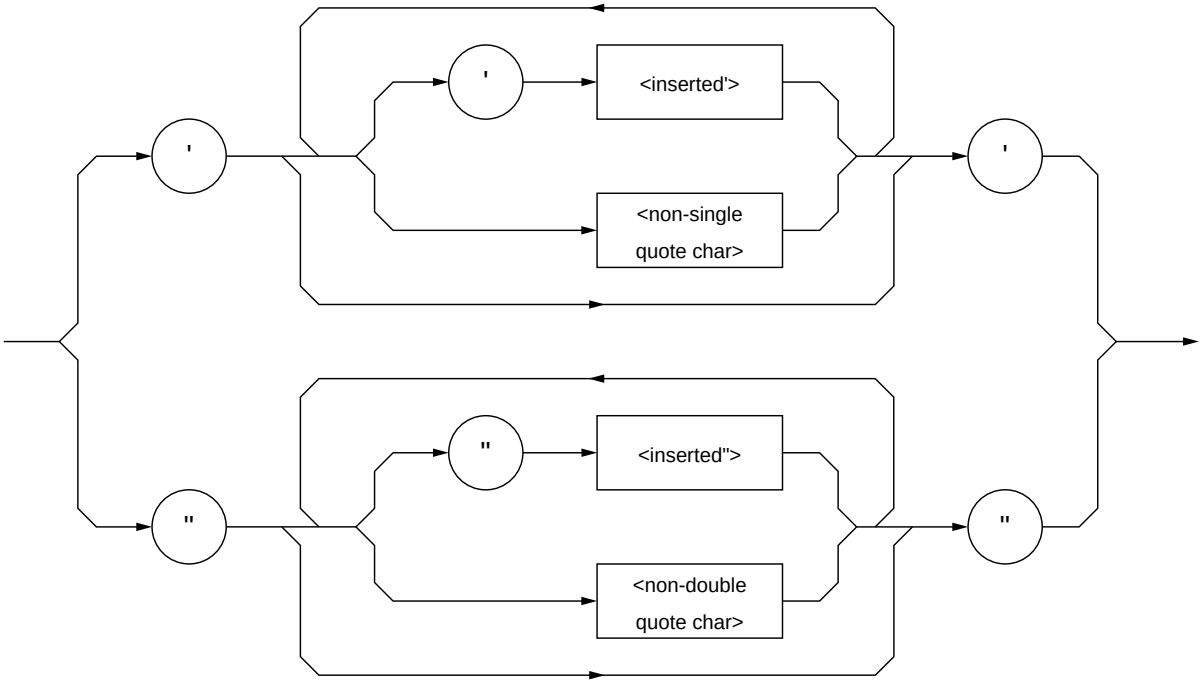
#Q37	(31D)
#q37	
#Q26703	(11,715D)
#q26703	

The character string following #B or #b is accepted by the device as a binary number.

#B10101011110000010010011	(11,256,099D)
#b0010110111000011	(11,715D)

5.3.5 <STRING PROGRAM DATA>

<STRING PROGRAM DATA> is program data consisting of only character strings. All ASCII 7-bit codes can be used. When a character string includes single or double quotation marks, two identical quotation marks must be written in succession per quotation mark.



- A character string must be enclosed with single or double quotation marks irrespective of whether the character string contains any quotation mark. For example,

  It's a nice day. → "It's a nice day."
  → :'It' 's a nice day.'

- When a character string is enclosed with single quotation marks, each single quotation mark contained in the character string must be doubled. Other characters, including double quotation marks, must be written as they are. For example,

  "I shouted` 'Shame'." → '"I shouted`' 'Shame' '." '

- When a character string is enclosed with double quotation marks, these double quotation marks must be doubled. Other characters, including single quotation marks, must be written as they are. For example,

  "I shouted` 'Shame'." → """I shouted` 'Shame'."""

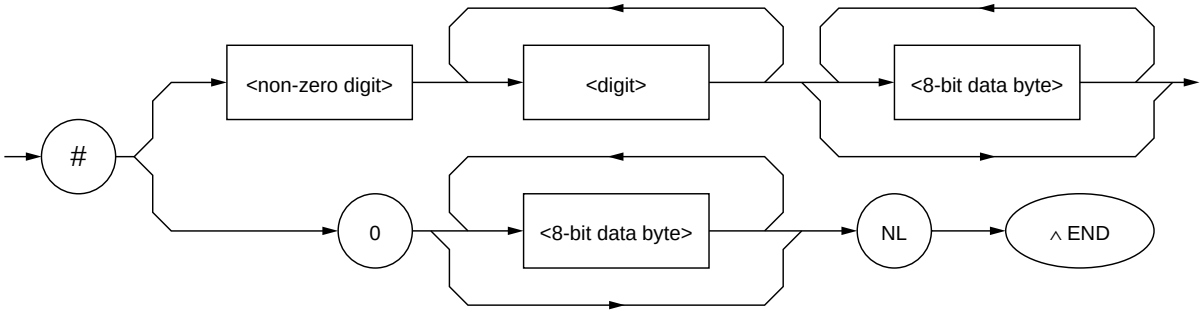
- <inserted '> is an single ASCII code set in ASCII code byte 27 (decimal 39 = symbol '). <inserted "> is a single ASCII code set in ASCII code byte 22 (decimal 34 = symbol "). <non-single quote char> and <non-double quote char> are single ASCII codes other than single and double quotation marks.

5.3.6 <ARBITRARY BLOCK PROGRAM DATA>

<ARBITRARY BLOCK PROGRAM DATA> is non-decimal program data starting with #. Binary data is transferred directly in 1-byte (8-bit) blocks. Differences from the non-decimal numeric program data (<NON-DECIMAL NUMERIC PROGRAM DATA>) mentioned on Section 5.3.4 are as follows:

- Data is not limited to numeric data, but character string data and numeric data can be handled.
- The number of data bytes to be transferred can be written between # and the first data.

The non-decimal data is program data that can specify the data bytes to be transferred.



- <digit> One of ASCII code bytes 30 to 39 (decimal values 48 to 57 = characters 0 to 9).
- <non-zero digit> One of ASCII code bytes 31 to 39 (decimal values 49 to 57 = characters 1 to 9).
- <8-bit data byte> An 8-bit byte within the range from 00 to FF (decimal values 0 to 255).

(1) When the number of data bytes to be transferred is known

The upper-right route in the above syntactical chart is applied.
Specify the number of <8-bit data byte> bytes to be transferred at the <digit> position, i.e., just before writing data. Write the number of digits of the specified number of bytes between # and <non-zero digit>. For example, to send four data bytes (DABs), write <ARBITRARY BLOCK PROGRAM DATA> as follows:

To send four bytes, specify 4 at the <digit> position.

```
      ↓
#14<DAB><DAB><DAB><DAB>
      ↑
```

The number of digits of the value 4 at the <digit> position is 4. So specify 1 at the <non-zero digit> position.

To send four bytes, specify 4 at the <digit> position. Leading 0s may be specified.

```
      ↓
#3004<DAB><DAB><DAB><DAB>
      ↑
```

The number of digits of the value 4 at the <digit> position is 3. Specify 3 at the <non-zero digit> position.

(2) When the number of data bytes to be transferred is unknown

The lower-right route in the syntactical chart on page 5-31 is applied. Write #0 before the first data and write NL^END after the last data, causing exitless termination.

```
#0<DAB><DAB><DAB><DAB>   <DAB>NL^END
```

If the following statements are specified for NL and ^END at the beginning of the program, then an EOI signal (END signal) is issued along with the terminator LF when the last byte has been transferred. (See Section 5.2.2.)

- For NL, `TERM IS CHR $ (10)`
- For END, `EOI ON`

(3) Handling integer-precision binary data

Integer-precision binary data is used as <ARBITRARY BLOCK>-type transfer data, whether it is program data or response data, and has the specifications summarized below. Negative values are processed as two's complements.

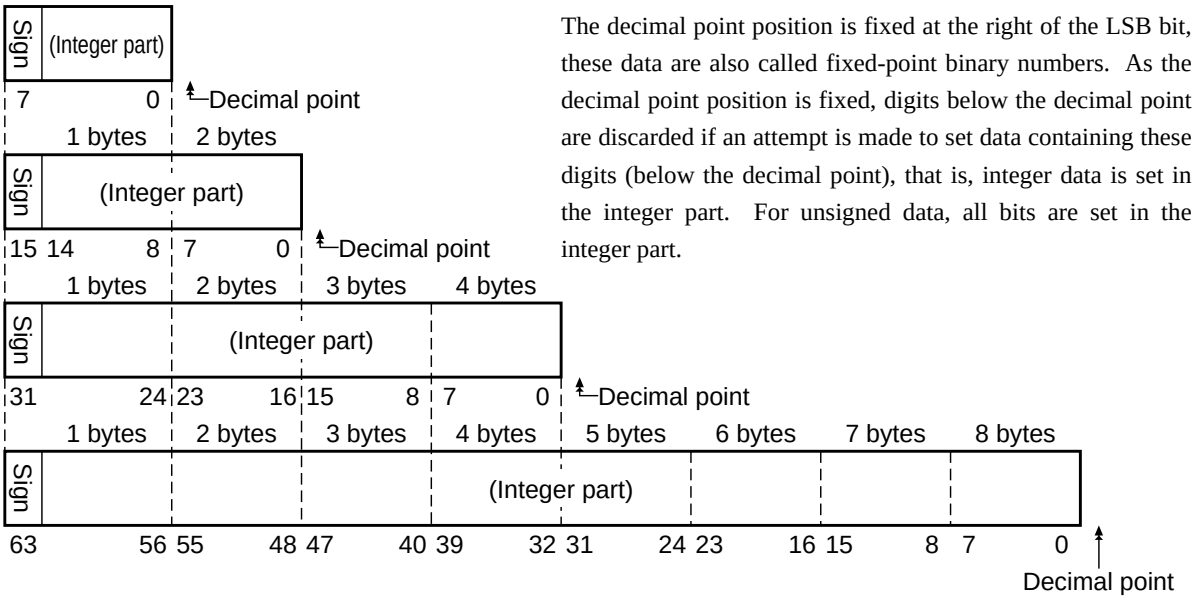
Number of transfer bytes	1, 2, 4, or 8 bytes
Byte transfer order	Bytes are transferred sequentially, starting at the most significant byte.
Signed binary code	LSD ····· Right-justify MSB ····· Sign bit When the data length is shorter than the field length, pad the remaining field with MSBs.
Unsigned binary code	LSD ····· Right-justify MSB ····· Not a sign bit Pad unused high-order bits with 0s.

Ranges of signed and unsigned 1-byte (8-bit) and 2-byte (16-bit) integer data are shown below.

8-Bit Binary	With Sign	No Sign
10000000	–128	128
10000001	–172	129
10000010	–126	130
11111101	–3	253
11111110	–2	254
11111111	–1	255
00000000	0	0
00000001	1	1
00000010	2	2
00000011	3	3
01111101	125	125
01111110	126	126
01111111	127	127

16-Bit Binary	With Sign	No Sign
1000000000000000	–32768	32768
1000000000000001	–32767	32769
1000000000000010	–32766	32770
1111111111111101	–3	65533
1111111111111110	–2	65534
1111111111111111	–1	65535
0000000000000000	0	0
0000000000000001	1	1
0000000000000010	2	2
0000000000000011	3	3
0111111111111101	32765	32765
0111111111111110	32766	37266
0111111111111111	32767	32767

Internal representations of signed 1-, 2-, 3-, 4-, and 8-byte integer data are shown below. When the sign bit is 0, it indicates positive data. When a sign bit is 1, it indicates negative data.



The decimal point position is fixed at the right of the LSB bit, these data are also called fixed-point binary numbers. As the decimal point position is fixed, digits below the decimal point are discarded if an attempt is made to set data containing these digits (below the decimal point), that is, integer data is set in the integer part. For unsigned data, all bits are set in the integer part.

(4) Floating-point binary data

Floating-point binary data, whether it is program data or response data, is used as <ARBITRARY BLOCK>-type transfer data. Our products do not support floating-point binary data; however, general specifications are explained below.

Floating-point binary data must consists of the following three fields:

(a) Sign field (sign bit)
(b) Exponent field (exponent bit)
(c) Mantissa field (mantissa bit)

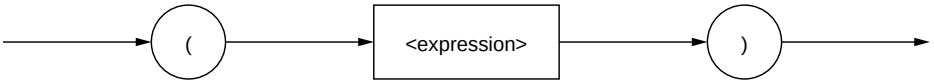
Numeric data having a decimal point is handled here. It has two types of precision: single precision and double precision. Field structures and transfer orders are shown below. Meanings of symbols are as follows:

S: Sign bit
EM: Most significant exponent bit
EL: Least significant exponent bit
FM: Most significant mantissa bit
FL: Least significant mantissa bit

<table>
<tr><th>Precision</th><th>Number of transfer bytes</th><th>Field structure and transfer order</th></tr>
<tr><td>Single precision</td><td>4 bytes</td><td>
<table>
<tr><th rowspan="2">Transfer byte</th><th colspan="8">DIO line</th></tr>
<tr><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th></tr>
<tr><td>1st byte</td><td>S</td><td>EM</td><td>E</td><td>E</td><td>E</td><td>E</td><td>E</td><td>E</td></tr>
<tr><td>2nd byte</td><td>EL</td><td>FM</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td></tr>
<tr><td>3rd byte</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td></tr>
<tr><td>4th byte</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>FL</td></tr>
</table>
Sign bit : 1 bit
Exponnent bit : 8 bits (+127 to –126)
Mantissa bit : 23 bits
</td></tr>
<tr><td>Double precision</td><td>8 bytes</td><td>
<table>
<tr><th rowspan="2">Transfer byte</th><th colspan="8">DIO line</th></tr>
<tr><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th></tr>
<tr><td>1st byte</td><td>S</td><td>EM</td><td>E</td><td>E</td><td>E</td><td>E</td><td>E</td><td>E</td></tr>
<tr><td>2nd byte</td><td>E</td><td>E</td><td>E</td><td>EL</td><td>FM</td><td>F</td><td>F</td><td>F</td></tr>
<tr><td>3rd to 7th byte</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td></tr>
<tr><td>8th byte</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>FL</td></tr>
</table>
Sign bit : 1 bit
Exponnent bit : 11 bits (+1023 to –1022)
Mantissa bit : 52 bits
</td></tr>
</table>

5.3.7 <EXPRESSION PROGRAM DATA>

The <EXPRESSION PROGRAM DATA> element sends the expression for obtaining a scalar, vector, matrix, or string value to a device, allowing the device to calculate a value in place of the controller. Its coding syntactical chart is as follows:



- <expression>: A sequence of ASCII characters represented by ASCII code bytes 20 to 7E (decimal values = 32 to 126), excluding the following six characters in [ ]:

  [" # ' ( ) ;]

  That is, a double quotation mark, number code (sharp), single quotation mark, left parenthesis, right parenthesis, and semicolon are excluded.

  If a+b+c is written as <expression>, then the above syntactical chart will be expressed as

  (a + b + c)

  To transfer this to a device, program data discussed on pages 5-20 to 5-34 can be used with the exception of the <INDEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA>. Upon receipt of (<expression>), the device obtains the solution to this expression.

Note:

The MS9710C does not support the <expression> function. If calculation of an expression is required, the solution to the expression must be obtained by the controller and the resultant numeric data must be transferred to the device as program data.

Section 6 Talker Output Format

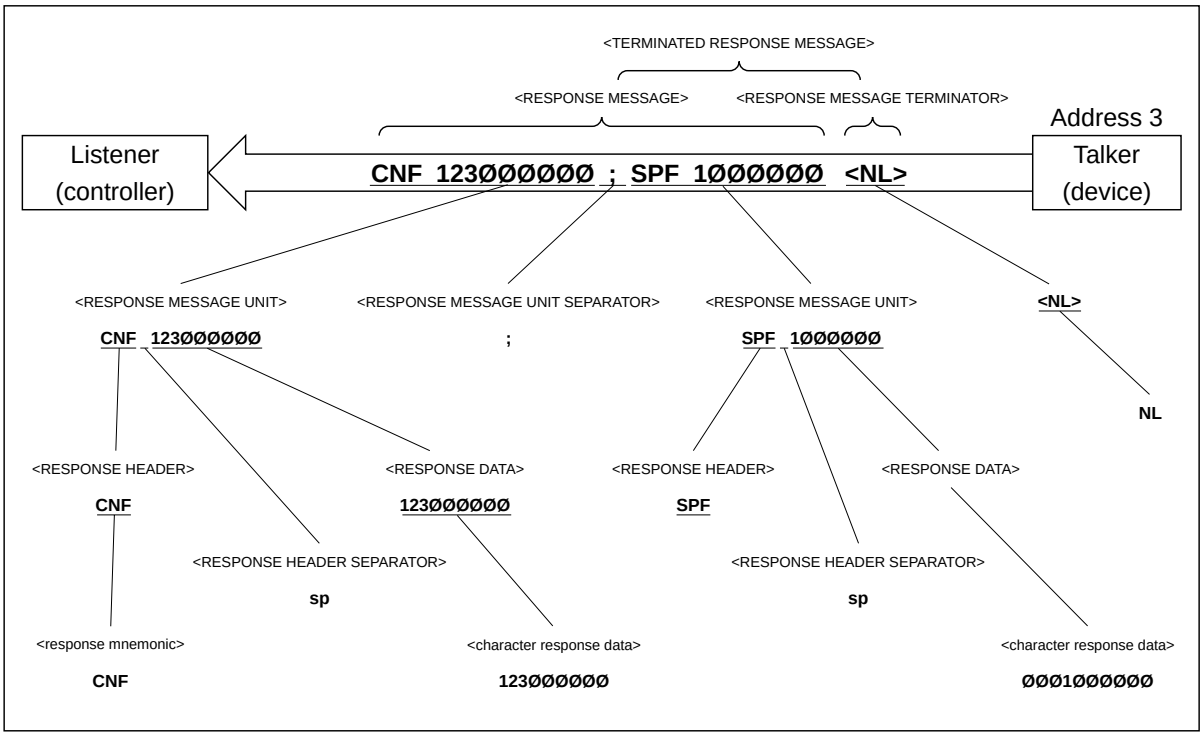
Device messages transferred between the controller and devices are classified into program messages and response messages. This section explains the formats of the program messages sent from a talker to a listener.

Note:

In this section, CNF? and SPF? are used to explain talker output formats. The MS9710C does not support these commands.

Typical response messages are: measurement result, setting, and status information. Response messages are classified into those with header and those without header.

The following figure shows that messages, ASCII character strings with header, are sent from a device to a controller in response to a center frequency message unit CNF? and a span frequency response message unit SPF?.



Only the operation-related parts is programmed as follows:

```
100 WRITE @103: "CNF? ; SPF?"!   Center and span frequency query message
110 READ @103:A$!     ←          When a terminator NL is detected, a response message "CNF
                                 123000000; SPF 1000000" is read into A$.
```

A response message is a sequence of functional elements, the minimum units that can represent functions, as is the case with the program message. In the above figure, functional elements are indicated by uppercase characters with them enclosed in < >. Functional elements are further classified into coding elements which are indicated by lowercase characters with them enclosed in < >.

Let's take a look at talker output formats focusing on the differences from listener input formats.

6.1 Differences in Syntax between Listener Input Formats and Talker Output formats

Significant differences in syntax between the listener and the talker are as follows:

- Listener format: Program can be written flexibly so that devices can accept program messages from the controller. If a program message involves some description errors, it can execute its function normally. For example, you can joint as many <white space> elements as you want to make an easy-to-read program.

- Talker format: Messages are output following strictly defined syntactical rules to allow the controller to accept the response messages from the device. Therefore, the syntax of response messages permits only one notation for a function.

The table below summarizes the differences in output format between the listener and the talker. In this table, "0/1 or more spaces" means <white space>.

Item	Listener input program message syntax	Talker output response message syntax
Characteristic	(Flexible)	(Strict)
Alphabetic characters	No difference between uppercase	Uppercase characters only and lowercase charactersr
Character before and after NR3 exponent part E	0 or more spaces + E/e + 0 or more spaces	Uppercase character E only
+ sign of NR3 exponent part	Omissible	Required
<white space>	Two or more white spaces can be written before/after a separator or before a terminator.	Not used
Message unit	(a) Header with program data (b) Header without program data	(a) Data with header (b) Data without header
Unit separator	0 or more spaces + Semicolon	Semicolon only
Space before header	0 or more spaces + Header	Header only
Header separator	Header + 1 or more spaces	Header + One $20^{\dagger 1}$
Data separator	0 or more spaces + Comma + 0 or more spaces	Comma only
Terminator	0 or more spaces + One of NL, EOI, and NL+EIO	NL+EOI

†1:

ASCII code byte 20 (decimal value 32 = ASCII character SP, space)

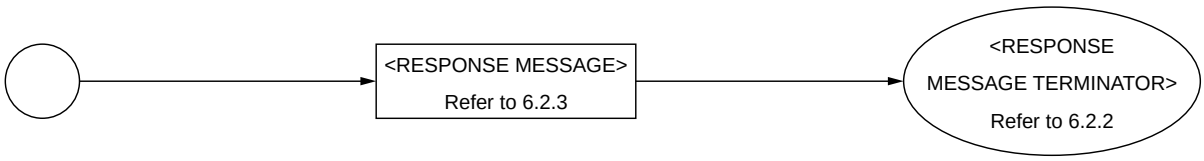
6.2 Response Message Functional Elements

Response messages output from a talker are terminated with an NL^END signal, allowing the controller to accept them. Functional elements of these response messages are explained here.

Rules for syntactical chart notation are the same as those for program messages, so see section 5. Functional and coding elements which are the same as those of program messages are not explained in this section, so see section 5 for them.

6.2.1 <TERMINATED RESPONSE MESSAGE>

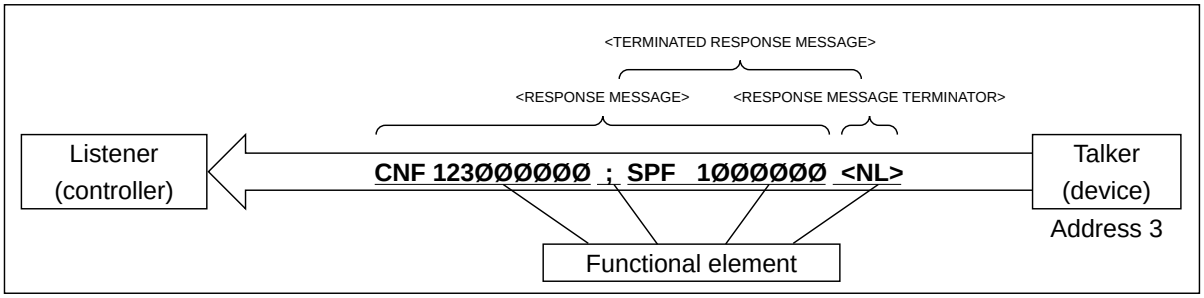
<TERMINATED RESPONSE MESSAGE> is defined as follows:



<TERMINATED RESPONSE MESSAGE> is a data message having all the necessary functional elements to be sent from a talker to a device.

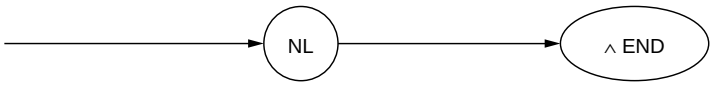
To complete transfer of <RESPONSE MESSAGE>, <RESPONSE MESSAGE TERMINATOR> is added at the end of <RESPONSE MESSAGE>.

<Example> <TERMINATED RESPONSE MESSAGE> in which two message units are connected



6.2.2 <RESPONSE MESSAGE TERMINATOR>

<RESPONSE MESSAGE TERMINATOR> is defined as follows:



<RESPONSE MESSAGE TERMINATOR> is placed after the last <RESPONSE MESSAGE UNIT> to terminate the sequence of one or more fixed-length <RESPONSE MESSAGE UNIT> elements.
If the following statements are specified for NL and ^END at the beginning of the program, then an EOI signal (END signal) is issued along with the terminator LF when the last data byte has been transferred. (☞ See Section 5.2.2.)

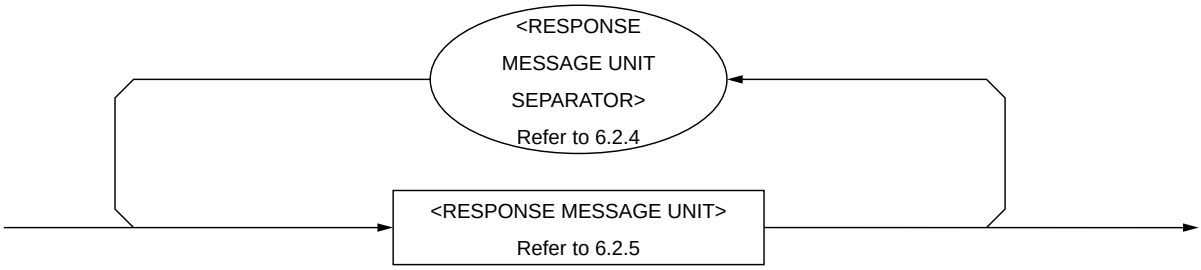
- For NL, TERM IS CHR$ (10)
- For END, EOI ON

<Example> Reading the currently set center frequency

```
10 LET ADR=101
20 TERM IS CHR$ (10) ! ..... Specify LF (New Line) as a terminator code.
30 EOI ON ! ..... Output a EOI signal for making the EOI line true when the last data byte
                  has been transferred.
40 WRITE @ADR : "CNT?" ! .... Center wavelength read query
50 READ @ADR : A$ ! ..... Terminate response data read with an EOI signal.
60 PRINT A$
70 END
```

6.2.3 <RESPONSE MESSAGE>

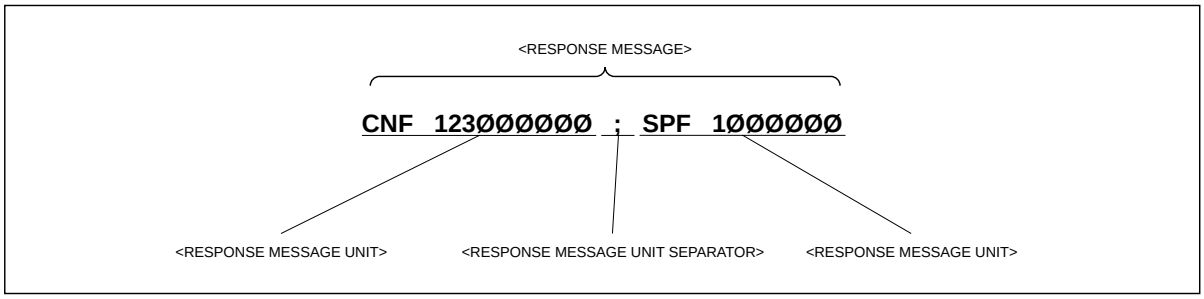
<RESPONSE MESSAGE> is defined as follows:



<RESPONSE MESSAGE> is a sequence of one or more <RESPONSE MESSAGE UNIT> elements.

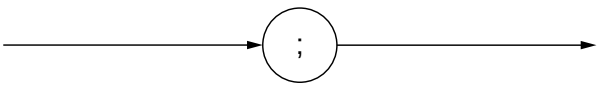
The <RESPONSE MESSAGE UNIT> element is a single message sent from a device to a controller. A <RESPONSE MESSAGE UNIT SEPARATOR> is used as a separator for separating multiple <RESPONSE MESSAGE UNIT> elements.

<Example> Adding CNF to the center frequency, adding SPF to the response data, and transferring them using a 1-character fixed format



6.2.4 <RESPONSE MESSAGE UNIT SEPARATOR>

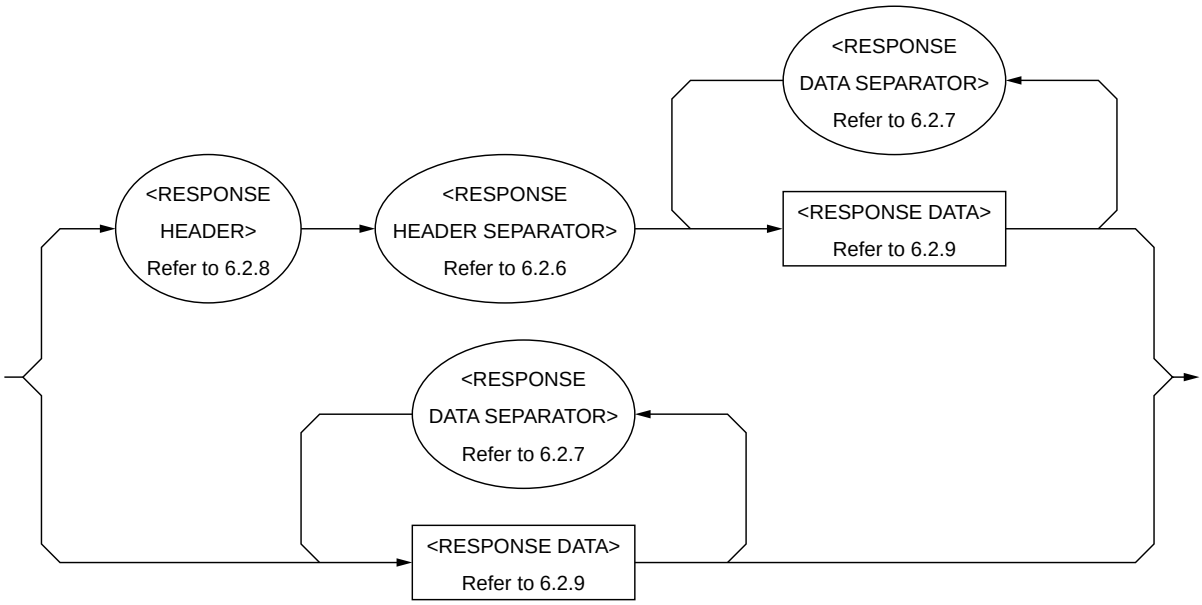
<RESPONSE MESSAGE UNIT SEPARATOR> is defined as follows:



<RESPONSE MESSAGE UNIT SEPARATOR> is used to separate <RESPONSE MESSAGE UNIT> elements with a <UNIT SEPARATOR> (semicolon (;)) when outputting a sequence of multiple <RESPONSE MESSAGE UNIT> elements as one <RESPONSE MESSAGE>.

6.2.5 <RESPONSE MESSAGE UNIT>

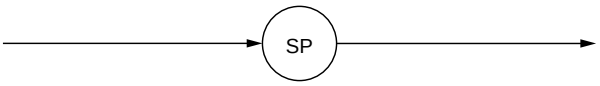
<RESPONSE MESSAGE UNIT> is defined as follows:



One is a response message unit with header, which returns the result of processing the program-message-set information accurately. The other is a response message unit without header, which returns only the measurement result.

6.2.6 <RESPONSE HEADER SEPARATOR>

<RESPONSE HEADER SEPARATOR> is defined as follows:



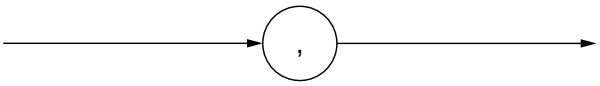
<RESPONSE HEADER SEPARATOR> is a space written after <RESPONSE HEADER> to be separated from <RESPONSE DATA>.

The space SP corresponds to ASCII code byte 20 (decimal 32).

In a response message with header, a space must always exist between the header and the data as a response header separator. It indicates the end of the header and the beginning of response data at the same time.

6.2.7 <RESPONSE DATA SEPARATOR>

<RESPONSE DATA SEPARATOR> is defined as follows:



When multiple <RESPONSE DATA> elements are to be output, <RESPONSE DATA SEPARATOR> must be placed between them.

6.2.8 <RESPONSE HEADER>

The format of <RESPONSE HEADER> is the same as that of <COMMAND PROGRAM HEADER> stated on Section 5.2.8 with the exception of the following three points:

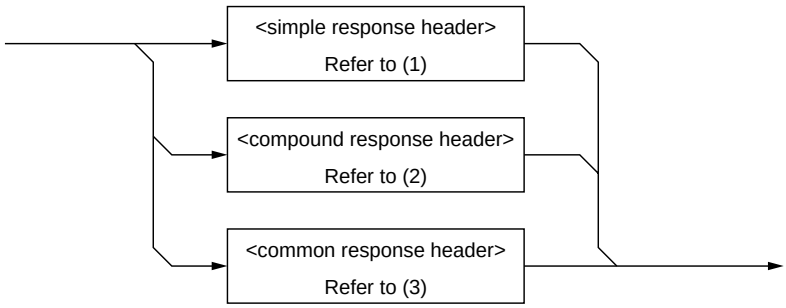
(1) Characters that can be used in <response mnemonic> are specified. For alphanumeric characters, only uppercase characters must be used. Other points are the same as those of <program mnemonic>.

(2) A space cannot be written before a response header while it can be written before a program header.

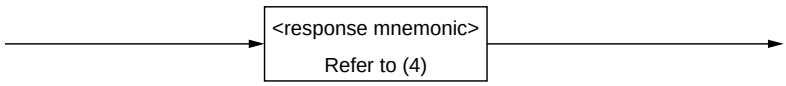
(3) Only one space can be written before a response header while two or more spaces can be written before a program header.

On the next page, the response header is explained up to <response mnemonic>.
(☞ It should be noted that only uppercase characters must be used in <response mnemonic>. Other points are the same as those of <program mnemonic> discussed on Section 5.2.8.)

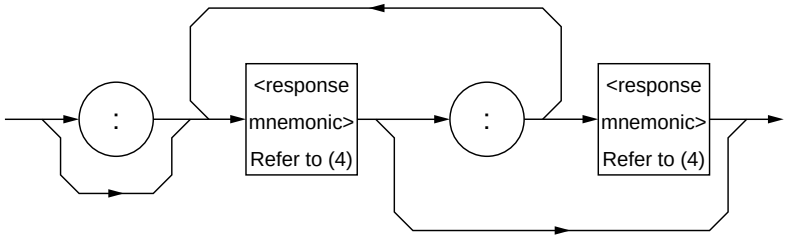
Item	Function
RESPONSE HEADER	A header indicates a function of <RESPONSE DATA>. It explains the function with a 12-character-long character-long character string or a <response mnemonic> element that consists of uppercase characters, numeric characters, and/or underline.



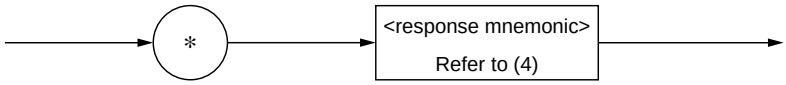
(1) <simple response header> is defined as follows.



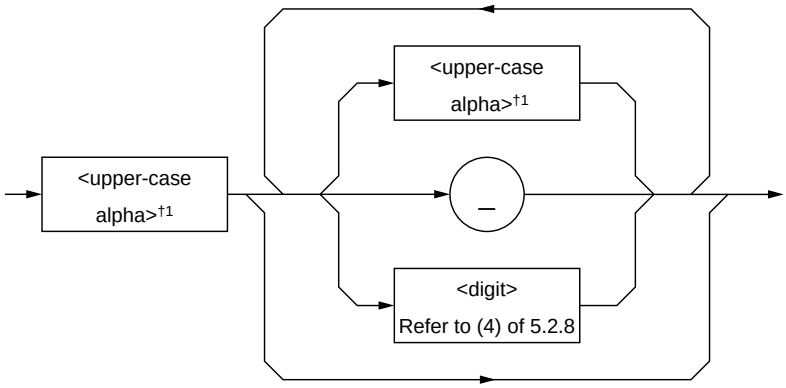
(2) <compound response header> is defined as follows.



(3) <common response header> is defined as follows.



(4) <response mnemonic> is defined as follows.

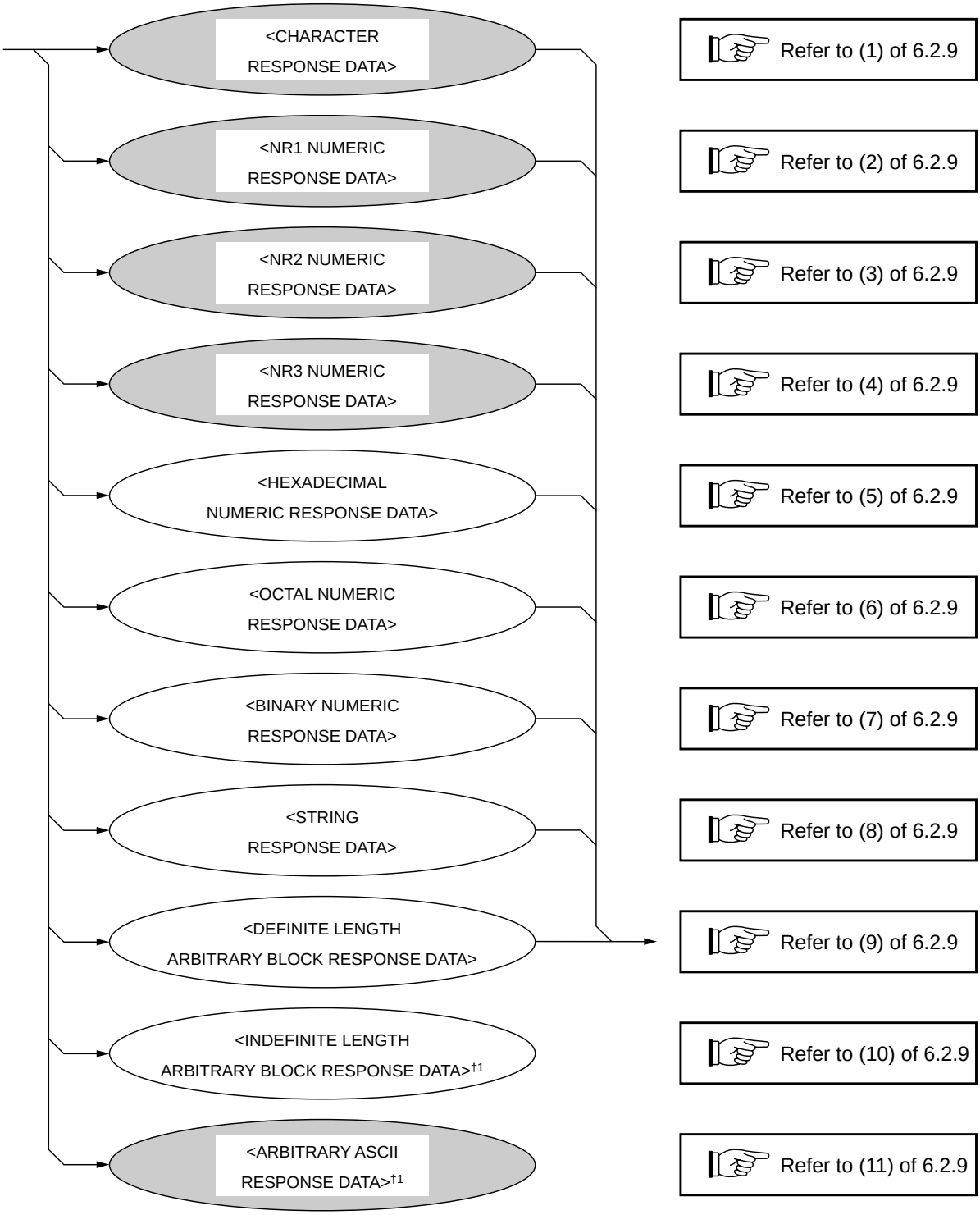


NOTE †1:

<upper-case alpha> ASCII code bytes 41 to 5A
(decimal values 65 to 90 = uppercase characters A to Z)

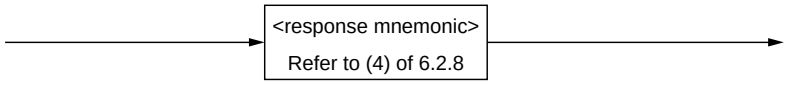
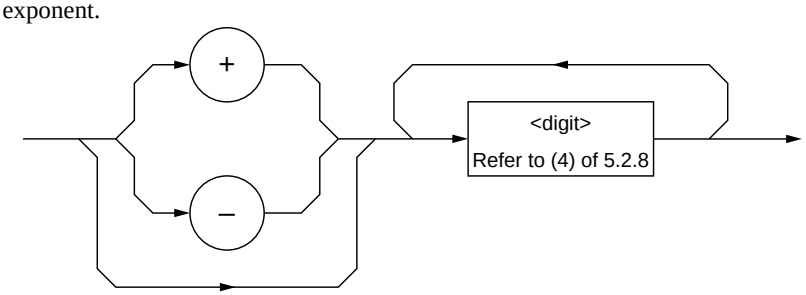
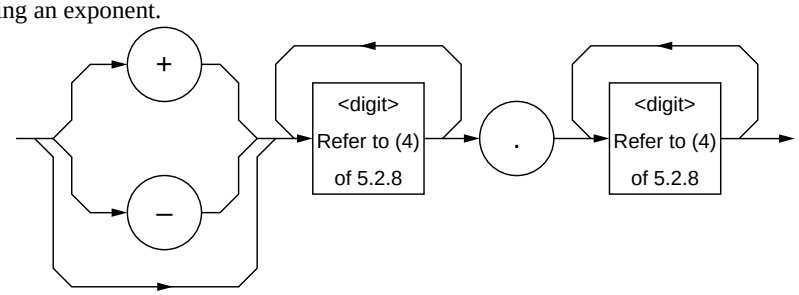
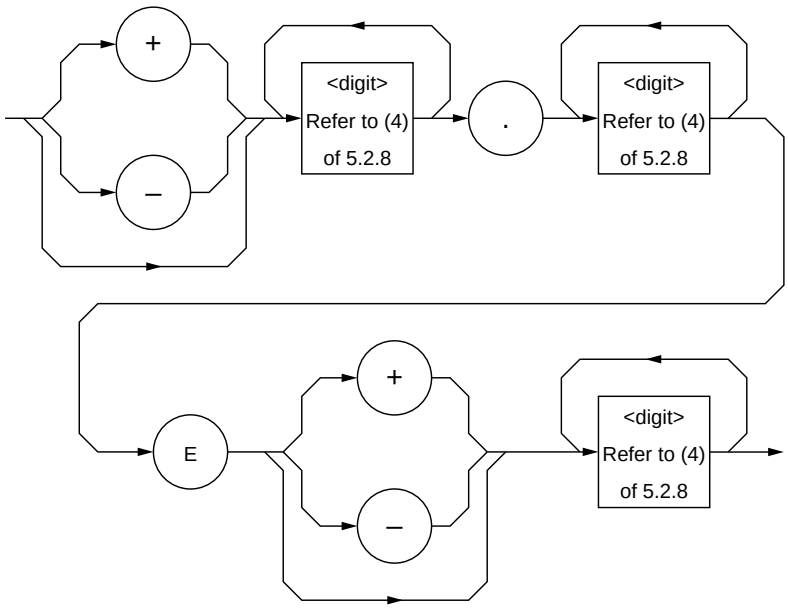
6.2.9 <RESPONSE DATA>

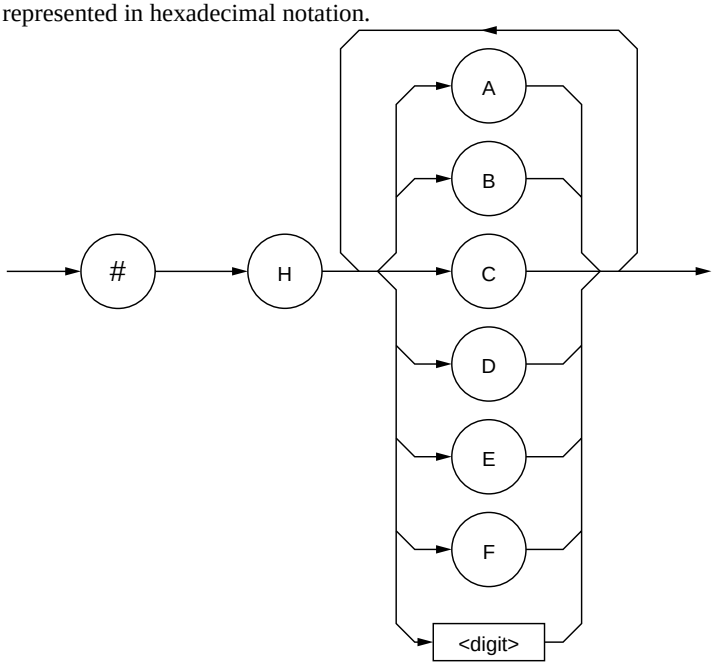
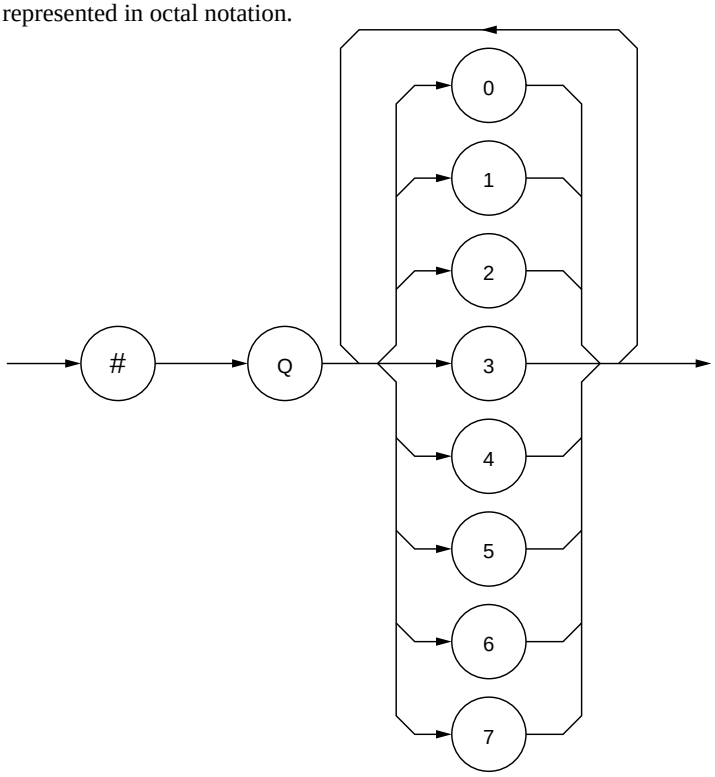
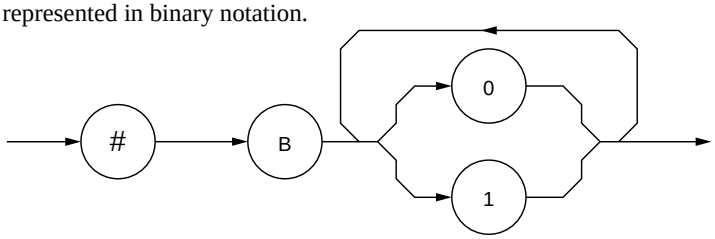
There are 11 types of <RESPONSE DATA> elements. Among them, the MS9710C transfers the response data shown in the hollow squares surrounded by a shade. The response data to be returned depends on the query message.

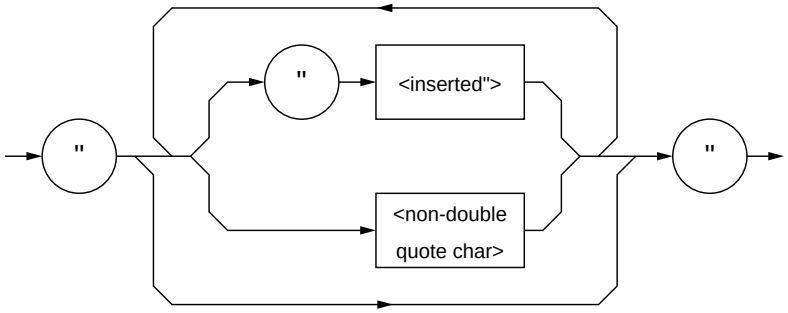
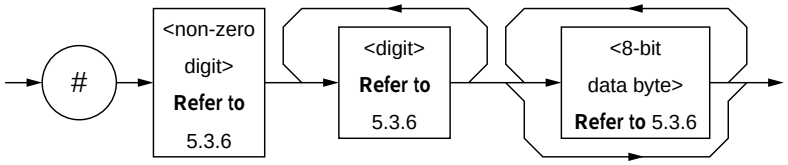
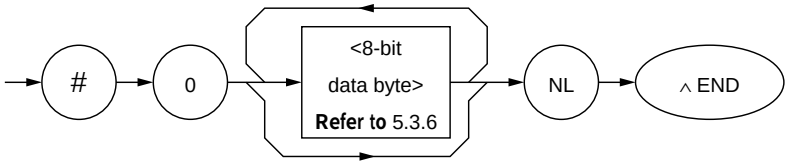
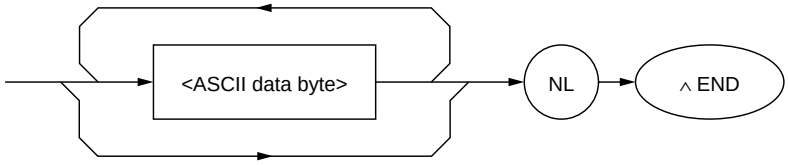


† 1:

<INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> and <ARBITRARY ASCII RESPONSE DATA> is terminated with NL ^ END after the last byte has been transferred.

Item	Function
(1) CHARACTER RESPONSE DATA	Data consisting of the same character string as that of <response mnemonic>. Accordingly, the character string always begins with an uppercase character and its length is less than 12 characters. Numeric parameters must not be used. 
(2) NR1 NUMERIC RESPONSE DATA <Example> 123 +123 –1234	Integer data, i.e., a decimal value of an integer that has neither decimal point nor exponent. 
(3) NR2 NUMERIC RESPONSE DATA <Example> 12.3 +12.34 –12.345	Fixed-point data, i.e., a decimal value other than integers or a decimal value having an exponent. 
(4) NR3 NUMERIC RESPONSE DATA <Example> 1.23E+4 +12.34E–5 –12.345E+6 • Lowercase characters cannot be used for E. • E must not be preceded and followed by a space. • + in the exponent part is mandatory. • + in the mantissa part is mandatory.	Fixed-point data, i.e., a decimal value having an exponent. 

Item	Function
(5) HEXADECIMAL NUMERIC RESPONSE DATA <Example> #HABC123 #H2DC3 #H8301	Data represented in hexadecimal notation. 
(6) OCTAL NUMERIC RESPONSE DATA <Example> #Q37 #Q26703 #Q30562	Data represented in octal notation. 
(7) BINARY NUMERIC RESPONSE DATA <Example> #B011101 #B1011 #B1011	Data represented in binary notation. 

Item	Function
(8) STRING RESPONSE DATA <Example> "This is a text" "Say,""Hello""."	Any ASCII 7-bit code can be used. The character string must be enclosed with double quotation marks ("). When a character string contains double quotation marks, two identical quotation marks must be written in succession per quotation mark. Since a CR, LF, of space can be used, this element is suitable for outputting a text to the printer or CRT. 
(9) DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA <Example> Transferring 11256099D in a 4-byte blocks ↓ #1400ABC123	Fixed-point 8-bit binary block data. It is suitable for transferring large-volume data, 8-bit extended ASCII code, and non-display data. Refer to the Section 5.3.6 "<ARBITRARY BLOCK PROGRAM DATA>" for more details on individual elements. 
(10) INDEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA <Example> Indefinite-length –250, –50, 120, ... are transferred ↓ #0FF06FFCE0078	Indefinite-length 8-bit binary block data. #0 must be written before the first data. The last data must be followed by NL∧END for termination. 
(11) ARBITRARY ASCII RESPONSE DATA <Example 1> <ASCII Byte><ASCII Byte>NL∧END <Example 2> NL∧END	ASCII data bytes except NL character transferred in succession. The last data must be followed by NL∧END for termination. 

Section 7 Common Commands

This section explains common commands and common query commands specified by IEEE 488.2. These common commands are not bus commands which are used as interface messages. Like device messages, the common commands are data messages used when the bus data mode (or the ATN line) is false. They can be applied to all measuring instruments, including those of other companies, that comply with IEEE 488.2. IEEE 488.2 common commands always begin with ∗.

7.1 Classification of MS9710C-Supported Common Commands by Group Function

The table below shows classification of MS9710C-supported IEEE 488.2 common commands by group function. Common commands to be supported are explained in an alphabetical order on the following pages.

7.2 Classification of Supported Commands and References

MS9710C-supported commands discussed previously are classified by function group as shown below. Details on these commands are given in alphabetical order on the next and subsequent pages.

Group	Function by group	Mnemonic
System data	Information about device connected to the system (e.g., manufacturer name, type name, and serial number) is returned.	∗IDN? ∗OPT?
Internal operation	Control inside the device: (a) Resetting of device at level 3 (b) Self-test and error detection inside the device	∗RST ∗TST?
Synchronization	A device is synchronized with the controller by: (a) Service request wait (b) Device output queue wait (c) Forced sequential execution	∗OPC ∗OPC? ∗WAI
Status and event	A status byte consists of a status summary message. Summary bits of the status summary message are set by a standard event register, output queue, and extended event register (or an extended queue). Three commands and four queries are provided to set, clear, validate, and invalidate the data in these registers and queues and to know the register settings using queries.	∗CLS ∗ESE ∗ESE? ∗ESR? ∗SRE ∗SRE? ∗STB?

∗CLS Clear Status Command

(Clears status byte registers)

■ Format

∗CLS

■ Application example

WRITE @108 : "∗CLS"
WRITE @108 : "CNT 1305.8 ; SPN 1000 ; ∗CLS"

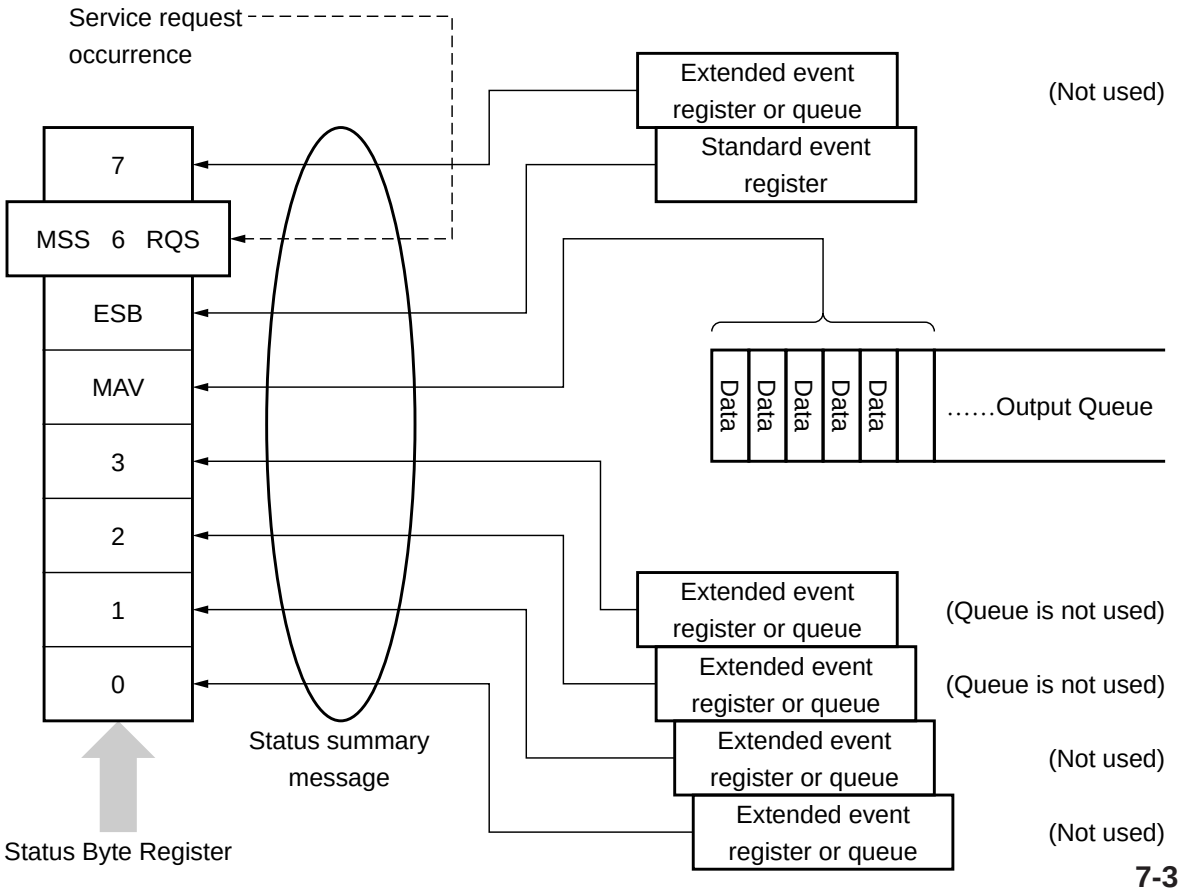
■ Explanation

The ∗CLS command clears all status structures (i.e., event registers and queues) except an output queue and its MAV summary messages, thus clearing the corresponding summary messages.

The output queue and its MAV summary messages are also cleared in the following case:

30 WRITE @108 : "CNT 1305.8 ; SPN 1000"
40 WRITE @108 : "∗CLS ; CNT?"

Issuing a ∗CLS command after <PROGRAM MESSAGE TERMINATOR> or before <QUERY MESSAGE UNIT> will clear all status bytes. With this method, all unread messages in the output queue will also be cleared. Values set in enable registers are not changed by the ∗CLS command.



∗ESE Standard Event Status Enable Command

(Sets or clears the standard event status enable register)

■ Format

∗ESE<HEADER SEPARATOR> <DECIMAL NUMERIC PROGRAM DATA>

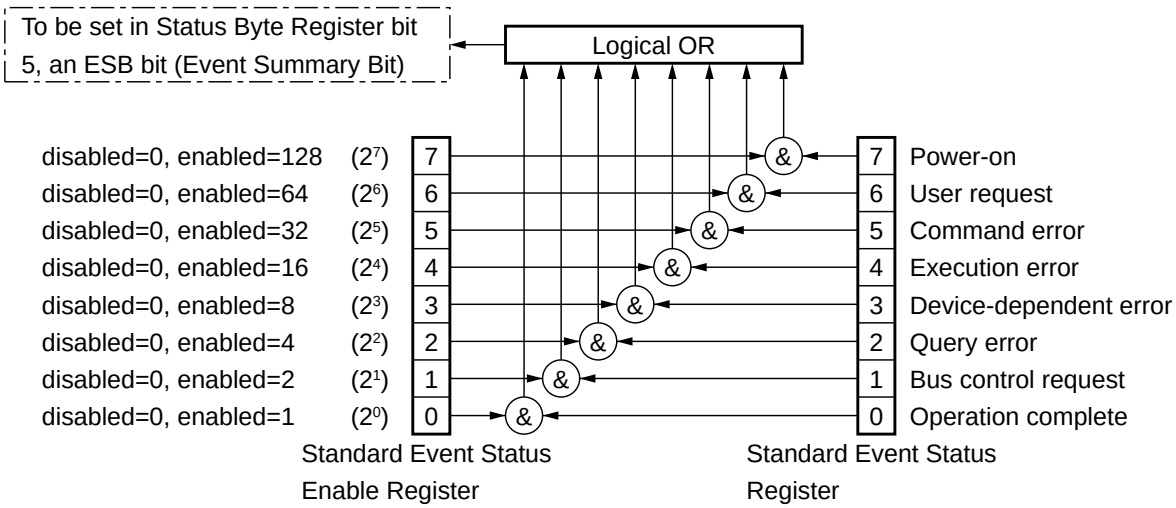
<DECIMAL NUMERIC PROGRAM DATA>: A value rounded to an integer, 0 to 255 (base is 2 and binary weights are assigned).

■ Application example

WRITE "108 : "∗ESE 20"!
Sets enable register bits 2 and 4.

■ Explanation

The total of values (2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32, 2^6 = 64, and/or 2^7 = 128) corresponding to the standard event status enable register bits 1, 2, 3, 4, 5, 6, and/or 7 that are to be enabled becomes program data.
The value of the bit to be disabled is 0.



∗ESE? Standard Event Status Enable Quer

(Returns the current value of the standard event status enable register)

■ Format

∗ESE?

■ Application example

Issuing ∗ESE? after executing ∗ESE 20 will return 20.

■ Explanation

The value (NR1) of the standard event status enable register is returned.

■ Response message

NR1 = 0 to 255

*ESR? Standard Event Status Register Query

(Returns the current value of the standard event status register)

■ Format

*ESR?

■ Application example

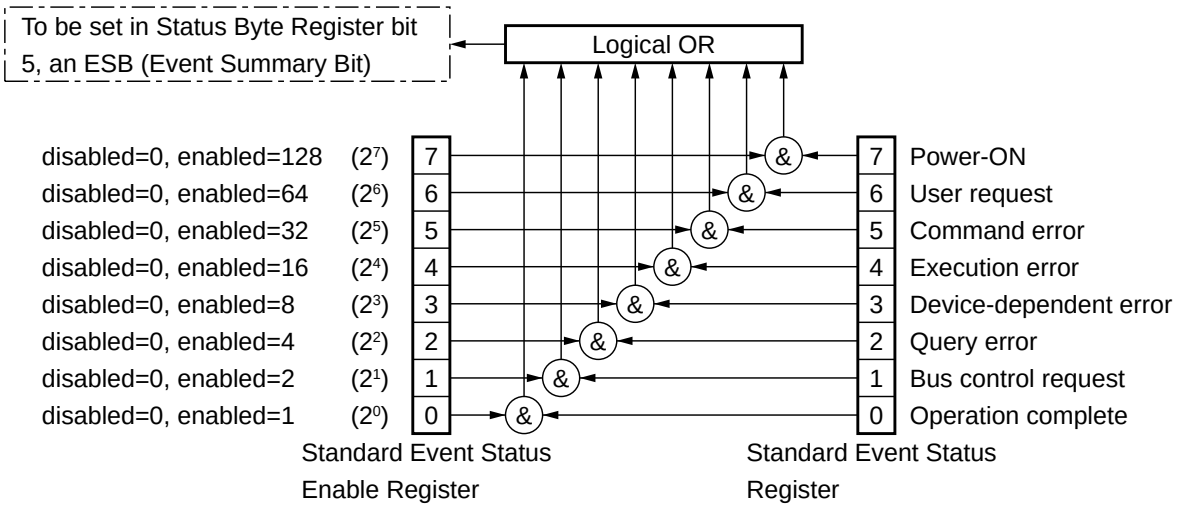
```
30  WRITE  @108 : "*ESR?"
40  READ  @108 : STEVET!   A command error occurs if the variable value is 32.
50  PRINT  STEVET
```

■ Response message NR1

NR1 = 0 to 255

■ Explanation

The current value NR1 of the standard event status register is returned. The total of values (2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32, 2^6 = 64, and/or 2^7 = 128) corresponding to the standard event status enable register bits 1, 2, 3, 4, 5, 6, and/or 7 that are enabled becomes NR1. When the response has been read (e.g., line 40), this register is cleared.



∗IDN? Identification Query

(Returns the manufacturer name, type name, serial number, and firmware level of the product)

■ Format

∗IDN?

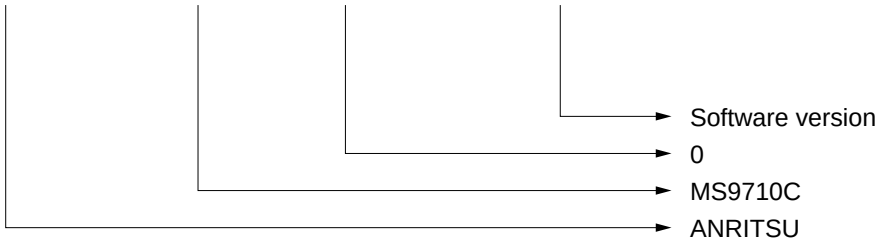
■ Application example

```
30 WRITE  @108 : "∗IDN?"
40 READ  @108 : IDEN$!
```

Stores the manufacturer name, type name, serial number, and irmware level.

■ Explanation

A manufacturer name, type name, serial number, and firmware level are returned.



When the manufacturer of the product whose type name, serial number, and software/hardware version number are Anritsu, 0, and 1 respectively, sending a common query ∗IDN? to a device will return a response message consisting of the above four fields.

Field 1: ..... Product manufacturer (e.g., ANRITSU)
Field 2: ..... Type name
Field 3: ..... Serial number (e.g., 0)
Field 4: ..... Firmware version No. (control software version and optical software version)

If you don't want to return a serial number and firmware version in fields 3 and 4, you can return ASCII character "0."

■ Response message

A response message which consists of the above four fields separated by commas is sent as <ARBITRARY ASCII RESPONSE DATA>.

Overall length of the response message comprising fields 1 to 4 ≤ 72 characters

∗OPC Operation Complete Command

(Sets bit 0 of the standard event status register when device operations have been completed)

■ Format

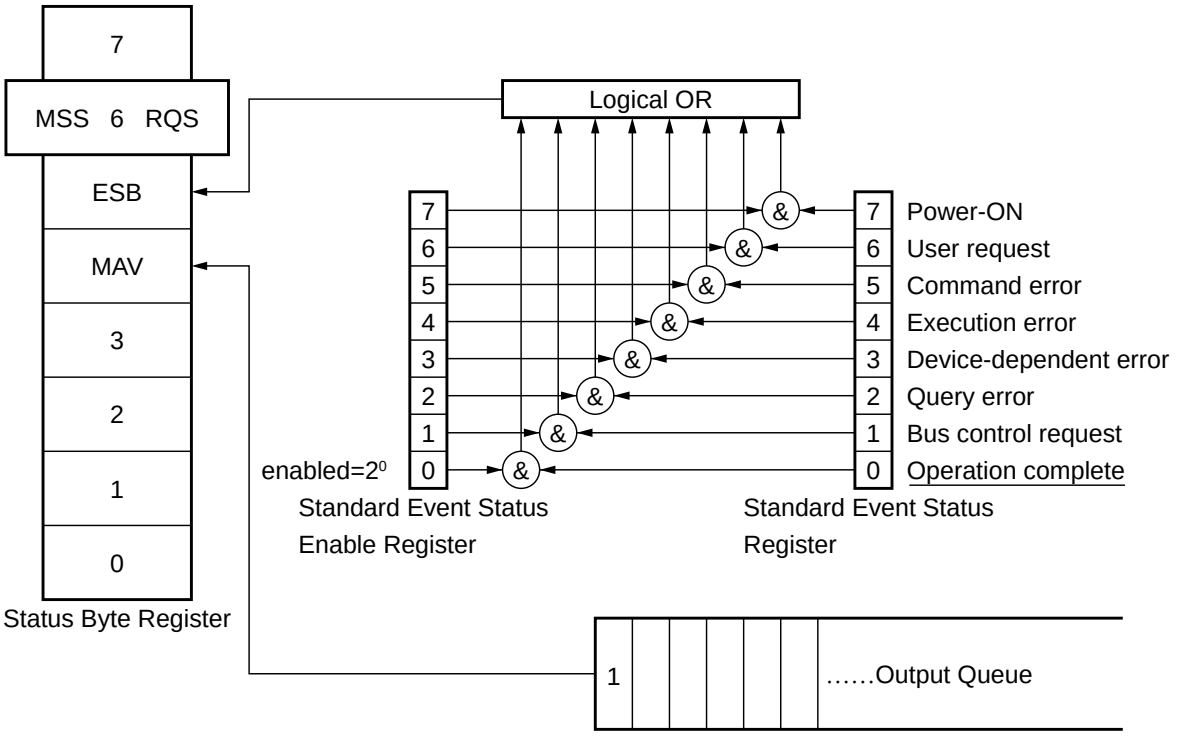
∗OPC

■ Application example

WRITE @108 : "∗OPC"

■ Explanation

When all the pending device operations have been completed, standard event status register bit 0 (i.e., operation complete bit) is set. However, since the MS9710C does not have an overlap command, the ∗OPC command counts for nothing.



∗OPC? Operation Complete Query

(When device operations have been completed, sets 1 in the output queue to generate an MAV summary message)

■ Format

∗OPC?

■ Application example

WRITE @108 : "∗OPC?"

■ Explanation

When all the pending device operations have been completed, 1 is set in the output queue, waiting for an MAV summary message to occur.

■ Response message

1 is returned as <NR1 NUMERIC RESPONSE DATA>.

*OPT? Option Identification Query

(Reports an installed option list)

■ Format

*OPT?

■ Application example

```
30 WRITE @103 "*OPT?"
40 READ @103:OPTI$!
```

Stores information about installed options.

■ Explanation

States of installed options are returned using 1 or 0.

	Option type	Option state
OPT01	Not used	" 1 "
OPT02	White light source option	0, Not installed; 1, Installed
OPT03	Reference light source plus SLD option	0, Not installed; 1, Installed
OPT04	SLD	0, Not installed; 1, Installed
OPT05	Reference light option	0, Not installed; 1, Installed
OPT06	Not used	" 0 "
OPT07	Not used	" 1 "

■ Response message

A response message which consists of the above three fields separated by commas is sent as <ARBITRARY ASCII RESPONSE DATA>.

<OPT01 option state><OPT02 option state><OPT03 option state>
<OPT04 option state><OPT05 option state><OPT06 option state>
<OPT07 option state>

∗RST Reset Command

(Resets a device at level 3)

■ Format

∗RST

■ Application example

WRITE @108 : "∗RST" Initializes only the device at address 3.

■ Explanation

The ∗RST (Reset) command resets a device at level 3 (P. 4-3).
At level 3, the following items are initialized:

(1) Device-dependent functions and states are restored to known states irrespective of the device history.

(2) The ∗DDT-command-defined macro is restored to the device-defined state.

(3) A mode in which macro operation is disabled and macros are not accepted, is set. Macro definitions are restored to the designer-specified states.

(4) The specified device is set in the OCIS (Operation Complete Command Idle State). The operation complete bit cannot be set in the standard event status register. (Section 8.12)

(5) The specified device is set in the OQIS (Operating Complete Query Idle State). The operation complete bit cannot be set in the output queue. The MAV bit is cleared.

The ∗RST command does not affect the following:

(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 standard

(8) RS-232C interface condition

∗SRE Service Request Enable Comman

(Sets a service request enable register bit)

■ Format

∗SRE<HEADER SEPARATOR><DECIMAL NUMERIC PROGRAM DATA>

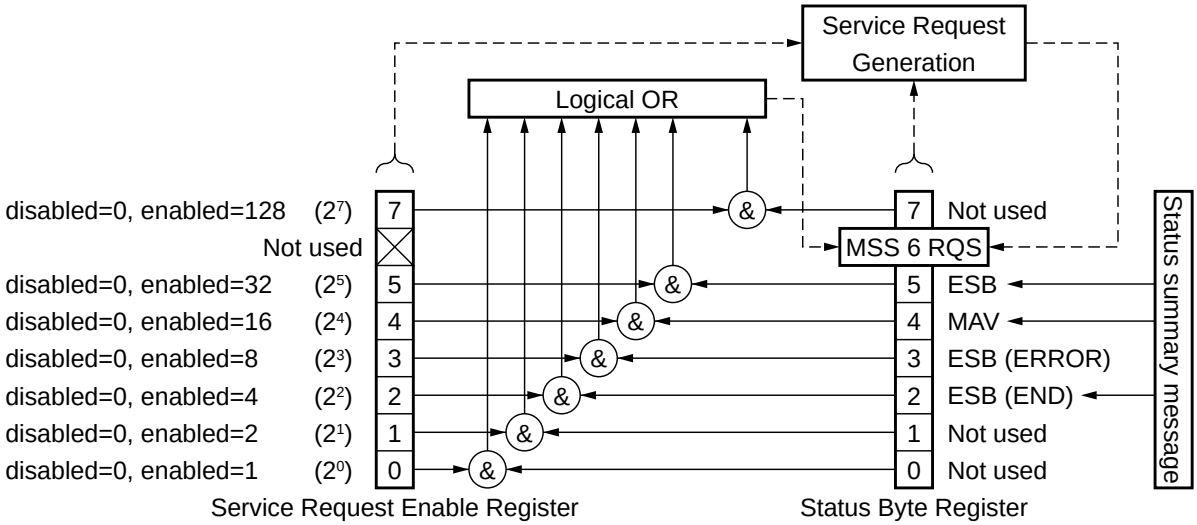
<DECIMAL NUMERIC PROGRAM DATA>:
A value rounded to an integer, 0 to 255 (base is 2 and binary weights are assigned).

■ Application example

WRITE @108 : "∗SRE 16"! Sets enable register bit 4.

■ Explanation

The total of values (2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32, and/or 2^7 = 128) corresponding to the service request enable register bits 1, 2, 3, 4, 5, 6, and/or 7 that are to be enabled becomes NR1. The value of the bit to be disabled is 0.



∗SRE? Service Request Enable Query

(Returns the current value of the service request enable register)

■ Format

∗SRE?

■ Application example

Issuing an ∗SRE? command after executing an ∗SRE16 returns 16.

■ Explanation

The value NR1 of the service request enable register is returned.

■ Response message NR1

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

∗STB? Read Status Byte Command

(Returns the current value of the status byte including the MSS bit)

■ Format

∗STB?

■ Application example

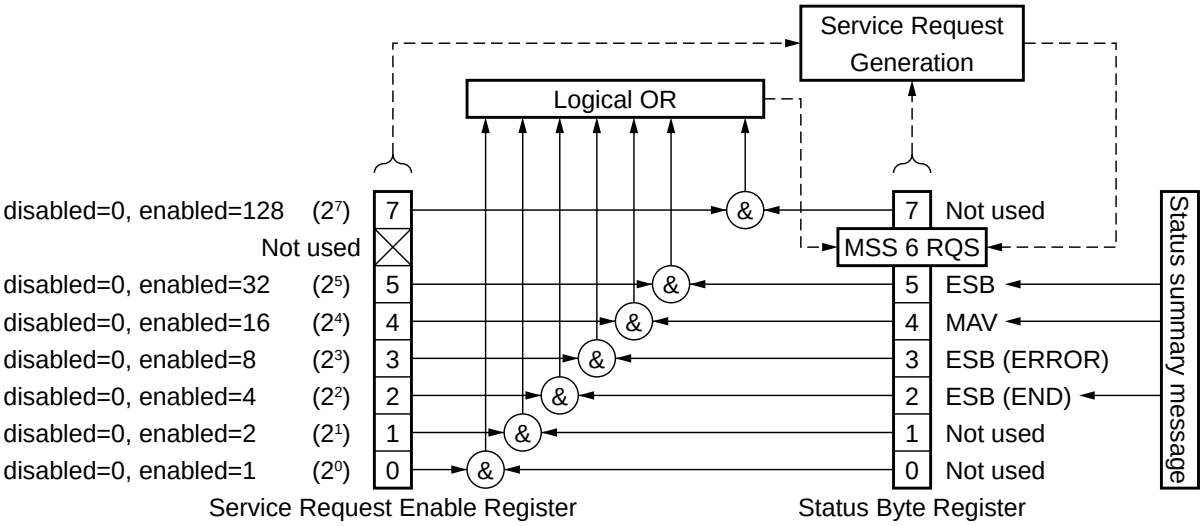
```
30  WRITE  @108 : "∗STB?"
40  READ  @108 : STBV
50  PRINT  STBV
```

■ Explanation

The ∗STB? command returns the total of the status register value assigned binary weights and MSS summary message value as <NR1 NUMERIC RESPONSE DATA>.

■ Response message

A response message (<NR1 NUMERIC RESPONSE DATA>) is an integer ranging from 0 to 255. It is the total of status byte register bit values. Status byte register bits 0 to 5 and 7 is assigned weights 1, 2, 4, 8, 16, 32, and 128 respectively, and the MSS (Master Summary Status) bit is assigned weight 64. The MSS indicates that there is at least one reason for requesting a service. MS9710C's status byte register conditions are summarized in the table below.



Bit	Bit weights	Bit name	Status byte register conditions	
7	128	————	0 = Not used.	
6	64	MSS	0 = No service is not requested.	1 = A service is requested.
5	32	ESB	0 = No event service has occurred.	1 = An event status has occurred.
4	16	MAV	0 = Data does not exist in the output queue.	1 = Data exists in the output queue.
3	8	ESB (ERROR)	0 = No event status has occurred.	1 = An event status has occurred.
2	4	ESB (END)	0 = No event status has occurred.	1 = An event status has occurred.
1	2	————	0 = Not used.	
0	1	————	0 = Not used.	

∗TST? Self - Test Query

(Conducts an internal self-test and indicates whether any error has occurred)

■ Format

∗TST?

■ Application example

```
30  WRITE  @108 : "∗TST?"
40  READ  @108 : TEST
50  PRINT  TEST
```

■ Explanation

The ∗TST? command conducts a self-test inside the device. The test result is set in the output queue. The data in the output queue indicates that the test has been completed without causing any error. The self-test does not require operator intervention.
The MS9710C conducts a self-test on the optical unit.

■ Response message

A response message is sent as <NR1 NUMBER RESPONSE DATA>.
Data range = –32767 to 32767

NR1 = –: The test has been completed without causing any error.
NR1 = 1: The test has not been conducted or any error occurred during the test.

∗WAI Wait - to - Continue Command

(Causes the next command to wait until the current command has been executed by the device)

■ Format

∗WAI

■ Application example

WRITE @108 : "∗WAI"

■ Explanation

The ∗WAI command executes overlap commands as sequential commands.

If the device can start executing the next command while processing a command or query from the controller, the command or query is called an overlap command.

If a ∗WAI command is executed after an overlap command, the next command must wait for the ∗WAIT common command to end. This also applies to sequential commands.

However, since the MS9710C does not support overlap commands. The ∗WAI command counts for nothing.

Section 8 Status Structure

This section explains the device status data specified by IEEE 488.2, the status data structure, and the technique of synchronization between a device and a controller.

IEEE 488.2 additionally provides common commands and common queries to get more detailed information compared with IEEE 488.1. For details on these commands and queries, see Section 7.

The status byte (STB) sent to the controller is specified by IEEE 488.1. The bits of the status byte represent a status summary message, providing a summary of the current contents of the data stored in a register or queue.

The following sections explain the status summary message bits, the status data structure for generating these status summary message bits, and the technique of synchronizing a device with the controller using the status messages.

These functions are used to control devices from an external controller via the GPIB interface. These functions, except a few, can also be used to control devices from an external controller via the RS 232C interface.

8.1 IEEE 488.2 Standard Status Model

Shown below is the standard model of the status data structure specified by IEEE 488.2

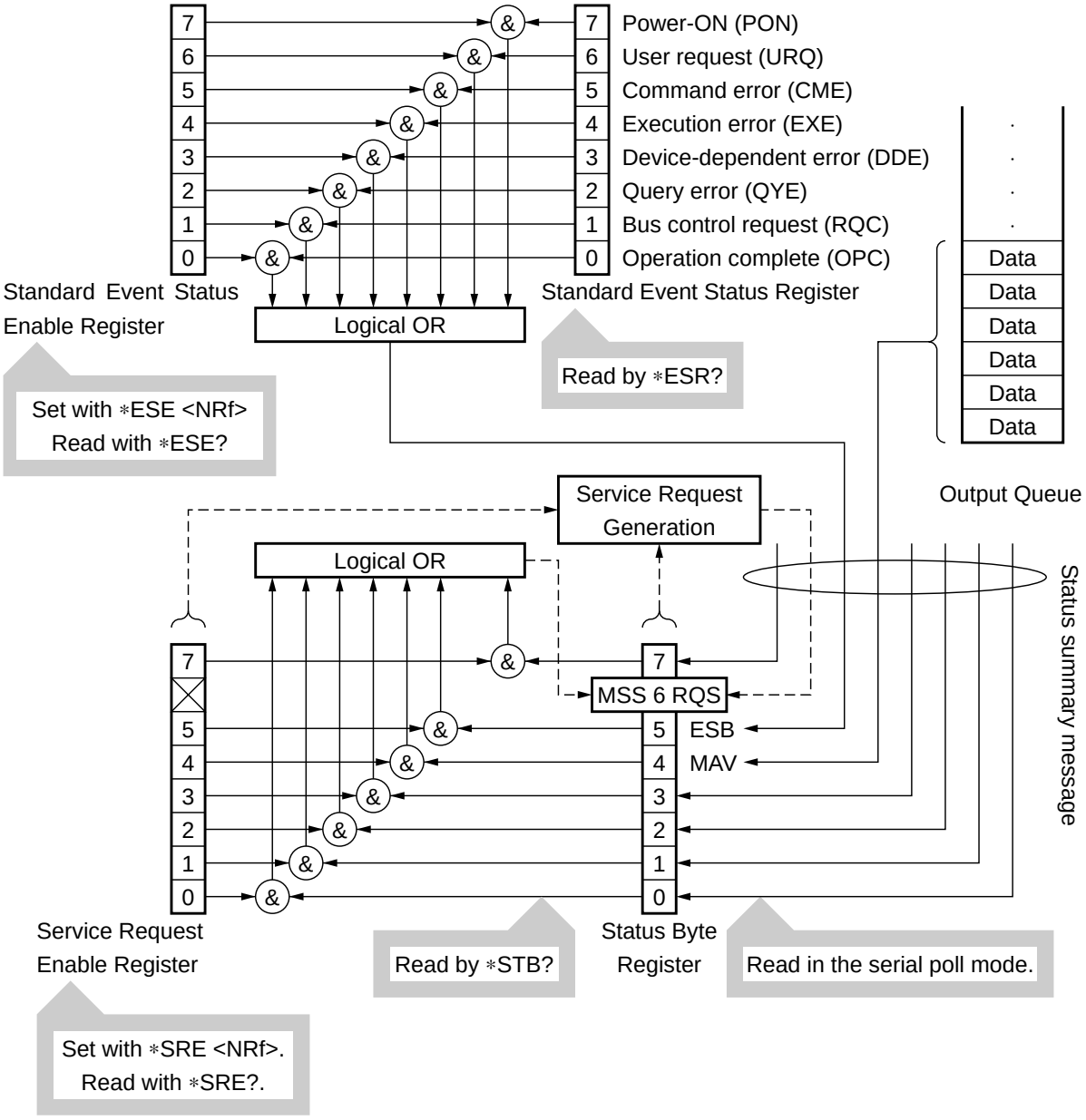


Fig. 8-1 Standard status model

The status model uses an IEEE 488.1 status byte. This status byte consists of seven summary message bits provided by the status data structure. To generate these summary message bits, the status data structure is comprised of two models: a register model and a queue model.

Register model

A pair of registers used to record an event that a device has encountered and a condition. It consists of an event status register and an event status enable register. When the results of ANDing the values of bits of these registers is not 0, the corresponding status register bits are set to 1s. In other cases, the corresponding status register bits are set to 0s. If the result of ORing the values of status register bits is 1, the summary message bit is set to 1. If the result of ORing these bits is 0, the summary message bit is set to 0.

Queue model

A data structure in which status values or information are removed in the same order they were entered. Only when the queue structure contains data, the corresponding bit is set to 1. If it is empty, the corresponding bit is set to 0.

Based on the concept of the above register model and queue model, the IEEE 488.2 standard status model is constructed from two types of register models and a queue model.

(a) Standard event status register and standard event status enable register

(b) Status byte register and service request enable register

(c) Output queue

Standard Event Status Register

This register has the register model structure mentioned above. It has eight bits corresponding to eight standard events encountered by the device: (1) power on, (2) user request, (3) command error, (4) execution error, (5) device dependent error, (6) query error, (7) bus control request, (8) operation complete. The result of logical OR is output to the status byte register bit 5 (DIO 4) as an event status bit (ESB) summary message.

Status Byte Register

The status byte register consists of an RQS bit and seven summary message bits for setting status summary messages from the status data structure. It is used in combination with a service request enable register. When the result of ORing the values of these two registers is 0, the SRQ is set ON. In this case, the status byte register bit "DIO 7" is reserved by the system as an RSQ bit, so this bit indicates to an external controller that a service request exists. The function of the SRQ conforms to IEEE 488.1.

Output Queue

This queue has the queue model structure mentioned above. Its contents are summarized and transferred to the status byte register bit 4 (DIO 5) as a message available (MAV) summary message.

8.2 Status Byte (STB) Register

The STB register consists of device STB and RQS (or MSS) messages. IEEE 488.1 defines the method of reporting STB and RQS messages, but it does not define the setting and clearing protocols and STB meaning. IEEE 488.2 defines device status summary messages and the master summary status (MSS) transferred to bit 6 along with an STB in response to the ∗STB? common query.

8.2.1 ESB and MAV summary message

Let's take a look at an ESB summary message and an MAV summary message.

(1) ESB summary message

The ESB (event summary bit) summary message is defined by IEEE 488.2. It appears in STB register bit 5. This bit indicates whether one or more IEEE 488.2 defined events have occurred, with the service request enable register set to allow events to occur, after the standard event status register was read or cleared last. The ESB summary message bit becomes true when at least one event registered in the standard event status register becomes true with event occurrence enabled. Conversely, the ESB summary bit becomes false when none of the registered events has occurred even if event occurrence is enabled.

(2) MAV summary message

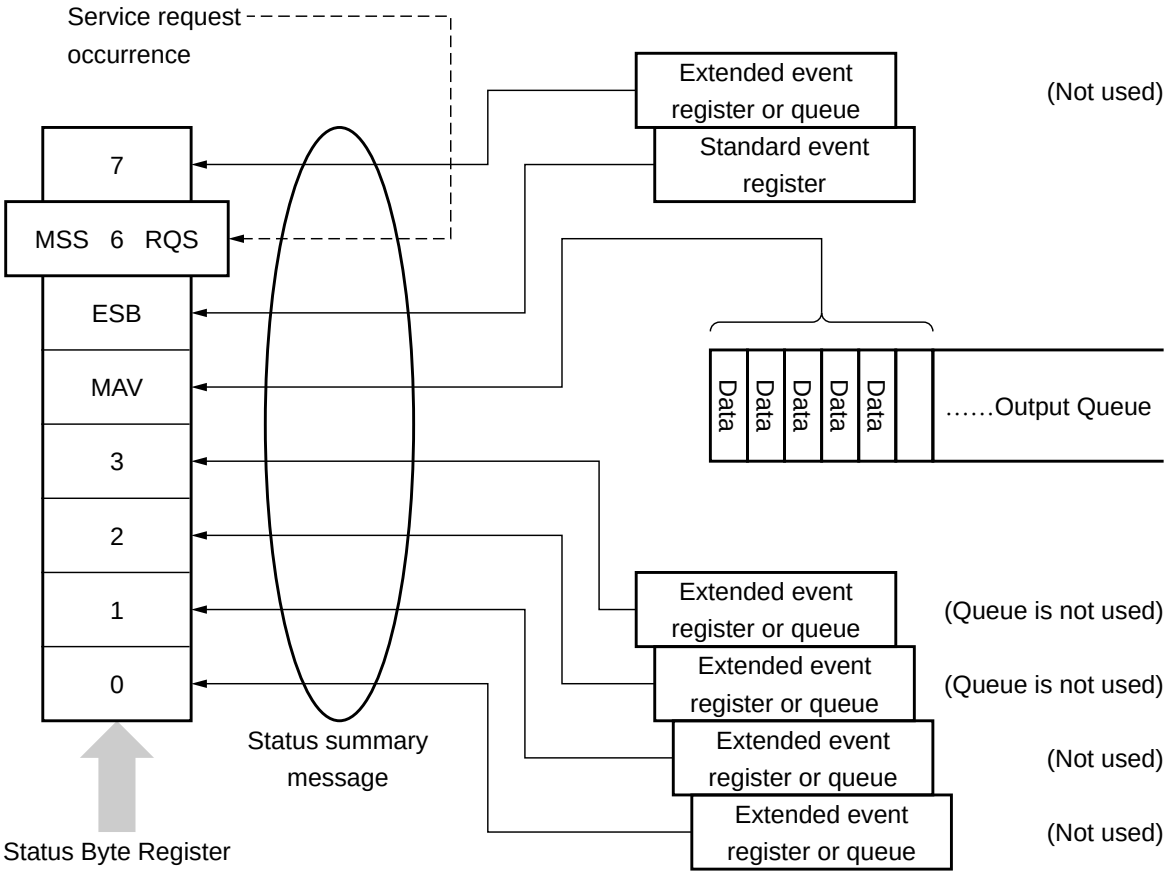
The MAV (message available) summary message is defined by IEEE 488.2. It appears in STB register bit 4. This bit indicates whether the output queue is empty. When a device is ready for accepting response messages from the controller, the MAV summary message bit becomes 1 (true). When the output queue is empty, this bit becomes 0 (false). This message is used to synchronize information exchange with the controller. For example, the controller can send a query message to the device and wait for the MAV to become true. The controller can perform another processing while waiting for a response from the device. If the controller has started reading the output queue without checking the MAV, all system bus operations are suspended until a response is received from the device.

8.2.2 Device dependent summary message

IEEE 488.2 does not define whether status register bit 7 (DIO 8) and bits 3 (DIO 4) to 0 (DIO 1) are used as status register summary bits or the bits indicating existence of data in the queue. Accordingly, these bits can be used as device dependent summary message bits.

Device dependent summary messages have a register model or queue model status data structure. This status register is a pair of registers used to report events and states in parallel or a queue used to report states and information sequentially. The summary bit provides a summary of the current status of the corresponding status data structure. For the register model, the summary message bit becomes true when one or more events have become true with occurrence of events enabled. For the queue model, the summary message bit becomes true when the queue is not empty.

The MS9710C does not use bits 7, 1, and 0 and uses bits 2 and 3 as status register summary bits. So the register model has four types of status data structures (two extended status data structures), and the queue model has an output queue (no extended status data structure).



8.2.3 Reading and Clearing the STB register

STB register contents can be read using serial polling or an ∗STB? common inquiry. IEEE 488.1 defined STB messages can be read by either method, but the value transferred to bit 6 (position) varies depending on the method.
STB register contents can be cleared using a ∗CLS command.

(1) Reading the STB register using serial polling (only when a GPIB interface bus is used)

When IEEE 488.1 defined serial polling is carried out, the device must return a 7-bit status byte and IEEE 488.1 defined RQS message bit. According to IEEE 488.1, the RQS message indicates whether the device has issued SRQs in the true state. The status byte value is not affected by serial polling. Immediately after being polled, the device must set the rsv message in the false state. If the device is polled again before a cause of issuing a new service request occurs, the RQS message has already been set in the false state.

(2) Reading the STB register using an ∗STB? common query

The ∗STB? common query causes the device to output STB register contents and one <NR1 NUMERIC RESPONSE DATA> from the MSS (master summary status) summary message. The response is the total of the status register value assigned binary weights and MSS summary message value. STB register bits 0 to 5 and 7 are assigned weighs 1, 2, 4, 8, 16, 32, and 128 respectively, and the MSS is assigned weights 64. The response to the ∗STB? is the same as that to serial polling with the exception that an MSS summary message appears in bit 6 instead of an RQS message.

(3) Definition of MSS (Master Summary Status)

The MSS indicates that the device has at least one cause of issuing a service request. In the device's response to the ∗STB? query, the MSS message appears in bit 6. However, it does not appear in the response to serial polling. It must not be regarded as part of the IEEE 488.1 defined status byte. The MSS is the result of ORing the values of STB register and SRQ enable (SRE) register bits totally. Specifically, the MSS is defined as follows:

(STB Register bit 0 AND SRE Register bit 0)
OR
(STB Register bit 1 AND SRE Register bit 1)
OR
:
:
(STB Register bit 5 AND SRE Register bit 5)
OR
(STB Register bit 7 AND SRE Register bit 7)

In the definition of the MSS, the values of bits 6 of the STB register and SQR enable register are ignored. Accordingly, when calculating the MSS value, the status byte may be handled assuming that it is represented by 8 bits and bit 6 is always 0.

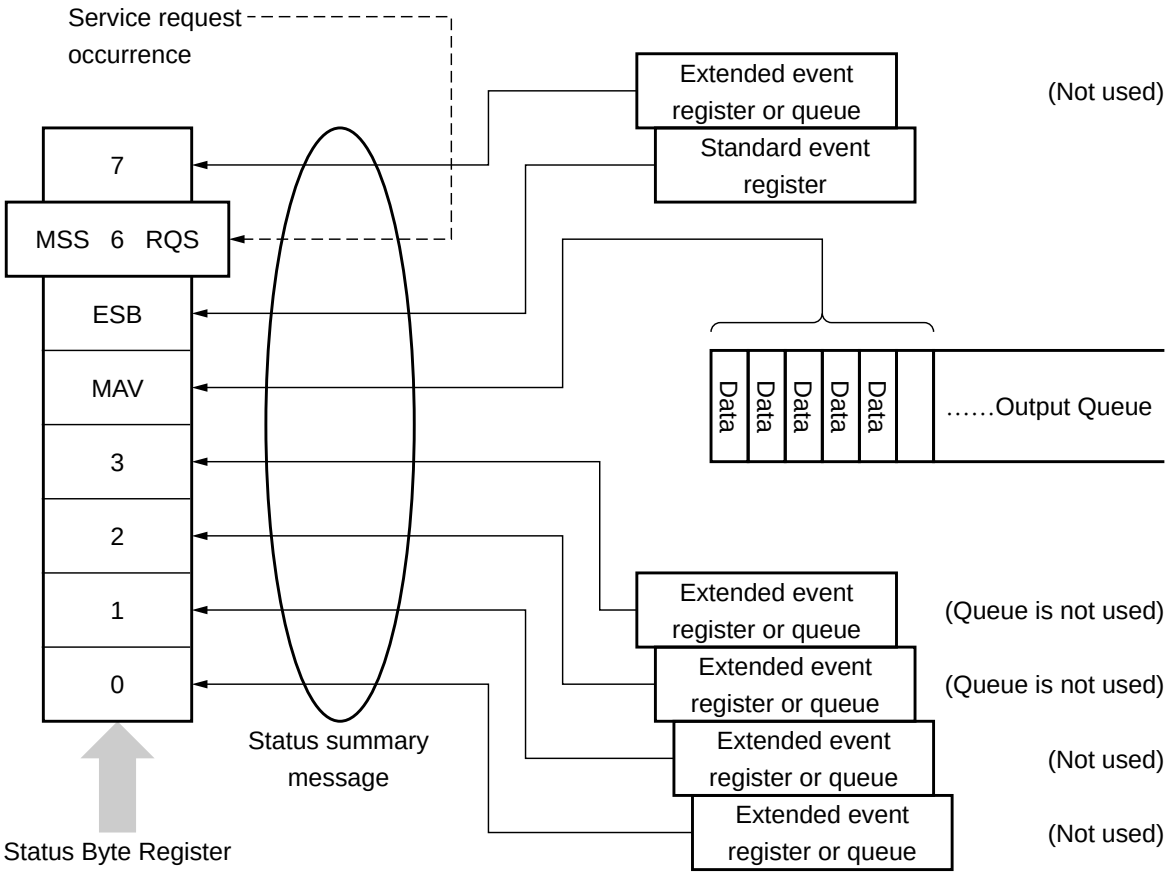
(4) Clearing the STB register using a ∗CLS common command

The ∗CLS common command clears all status structures, except the output queue and MAV summary message (i.e., event registers and queues), and the corresponding summary messages.

The output queue and the MAV summary message are also cleared in the following case:

```
30  WRITE  @ADR : "CNT  1305.8 ; SPN  1000"
40  WRITE  @ADR : "∗CLS ; CNT?"
```

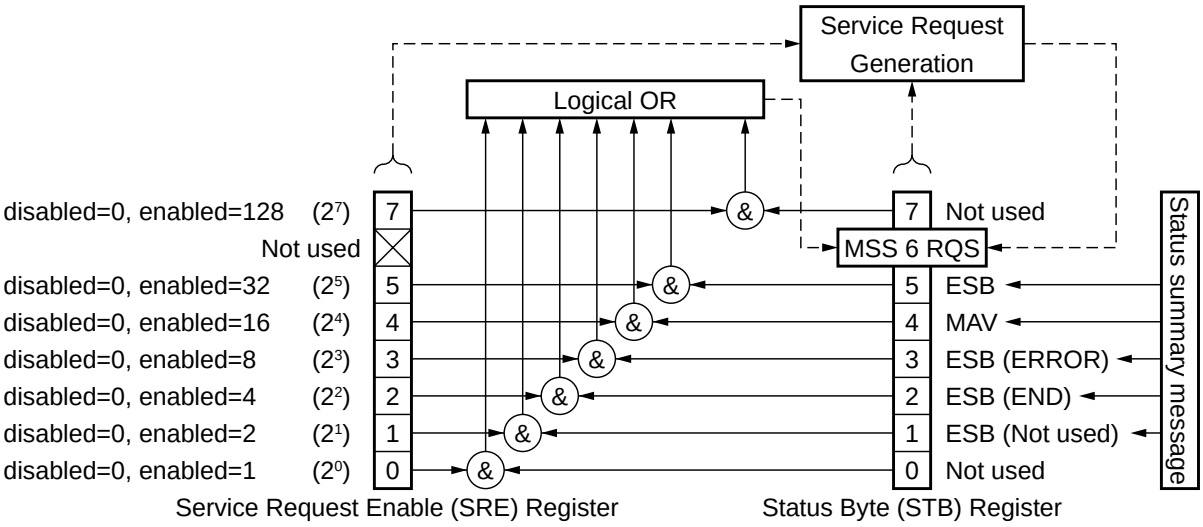
That is, issuing a ∗CLS command after the <PROGRAM MESSAGE TERMINATOR> element or before the <Query MESSAGE UNIT> element clears all status bytes. With this method, all unread messages in the output queue are cleared and the MAV message becomes false. When replying to the ∗STB?, the MSS message becomes false, too. Values of enable registers are not affected by ∗CLS.



8.3 Enabling the SRQ

Enabling the SRQ allows a summary message in the STB register to be selected in response to a service request. The service request enable (SRE) register shown below can be used to select a summary message.

Bits of the service request enable register correspond to the bits of the status byte register. When 1 is set in a status byte bit corresponding to a significant bit of the service request enable register, the devices sets the RQS bit to 1 and issues a service request to the controller. For example, when bit 4 of the service request enable register is set (enabled) in advance, a service request can be issued to the controller each time the MAV bit is set to 1 (if the output queue has data).



(1) Reading the SRE register

SRE register contents can be read using an ∗SRE? common inquiry. The response message to this query is <NR1 NUMERIC RESPONSE DATA>, an integer ranging from 0 to 255. It is a total of values of the service request enable register. Service request enable register bits 0 to 5 and 7 are assigned weights 1, 2, 4, 8, 16, 32, and 128, respectively. Unused bit 6 must always be 0.

(2) Updating the SRE register

The SRE register is written using an ∗SRE common command. The ∗SRE common instruction is followed by a <DECIMAL NUMERIC PROGRAM DATA> element. <DECIMAL NUMERIC PROGRAM DATA> is rounded to an integer. It is represented in binary notation using a base 2, indicating the total of values of SRE register bits (weight value). When the value of this bit is 1, it indicates the enabled state. When the value of this bit is 0, it indicates the disabled state. The value of bit 6 must always be ignored.

(3) Clearing the SRE register

The SRE register can be cleared by executing an ∗SRE common command or turning on the power.

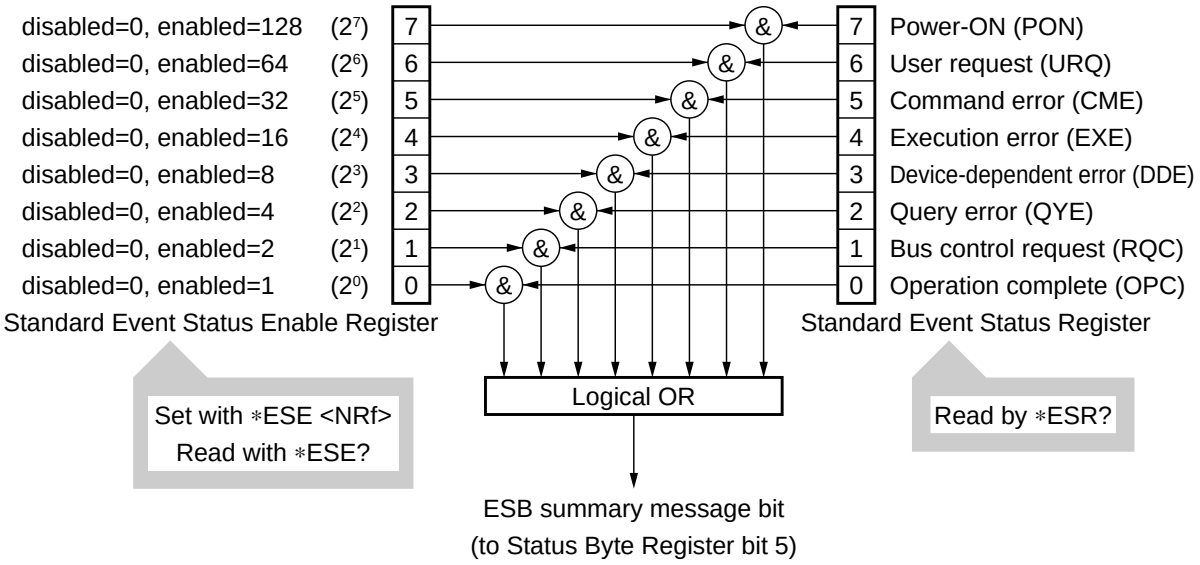
When an ∗SRE common command is used, the SRE register can be cleared by bringing the <DECIMAL NUMERIC PROGRAM DATA> element value to 0. Clearing the SRE register disables the status information to generate an rsv local message, suppressing issue of a service request.

When the power is turned on, the SRE register is cleared if the power-on status clear flag is true and the ∗PSC command for disabling clearing of this register is not supported.

8.4 Standard Event Status Register

8.4.1 Definition of standard event status register bits

Any device conforming to IEEE 488.2 must have the standard event register. Operation of the standard event register model is shown below. As it has already been explained, here we will explain the meaning of standard event status register bits given in IEEE 488.2.



Bit	Event name	Description
7	Power-ON (PON)	The power has been turned on.
6	User request (URQ)	Local control is requested. This bit is set irrespective of the remote/local state of the device. Since this bit is not supported by MS9710C, it is always 0.
5	Command error (CME)	A program message including a syntax error or a misspelled command has been received or a GET command has been received in a program message.
4	Execution error (EXE)	A program message which is syntactically correct but cannot be executed has been received.
3	Device-dependent error (DDE)	An error other than CME, EXE, and QYE has occurred.
2	Query error (QYE)	An attempt was made to read data from the output queue while it has no data, or the data in the output queue has been lost due to overflow, etc.
1	Request control (RQC)	The device is required to be an active controller. Since this bit is not used by MS9710C, it is always 0.
0	Operation complete (OPC)	The device has completed the specified pending operation and ready for receiving a new instruction. This bit responds only to the ∗OPC command and sets the operation complete bit.

8.4.2 Details on query errors

No.	Item	Description
1	Incomplete program message	When a device receives an MTA from the controller before receiving a program message terminator, it discards the incomplete message which has been received so far and waits for the next program message. To discard the incomplete program message, the device clears the input/output buffer, reports a query error to the status report part, and sets the standard status register bit 2 (query error bit).
2	Interruption of response message output	When a device receives an MLA from the controller before completing output of a response message terminator, it automatically interrupts output of the response message and waits for a next program message. To interrupt output of the response message, the device clears the input/output buffer, reports a query error to the status report part, and sets the standard status register bit 2 (query error bit).
3	When the next program message is sent without reading a response message	When the device cannot output a response message because the controller has output a program message (including a query message) and the next program message in succession, the device discards the response message and waits for the next program message. A query error is reported to the status report part like item No. 2.
4	Output queue overflow	When a program message containing many query messages is executed one after another, too many response messages to be stored in the output queue (256 bytes) may be generated. If more query messages are input and the response messages to queries must be output, the output queue overflows. When this happens, the device clears the output queue and resets the response message generation part. The device also sets the standard event status register bit 2 (query error bit) in the status report part.

8.4.3 Reading, writing, and clearing the standard event status register

Read	This register is read destructively in response to the ∗ESR? common command. In other words, this register is cleared after being read. The event bit assigned binary weights and converted to a decimal value <NR1> is the response message.
Write	This register cannot be written externally; however, it can be cleared.
Clearing	This register is cleared in the following cases: (1) A ∗CLS command is received. (2) The power is turned on if the Power-ON status clear flag is True. The device executing a Power-ON sequence first clears the standard event status register, then records the events that have occurred in this sequence (e.g., PON event bit setting). (3) An event is read in response to an ∗ESR? query command.

8.4.4 Reading, writing, and clearing the standard event status enable register

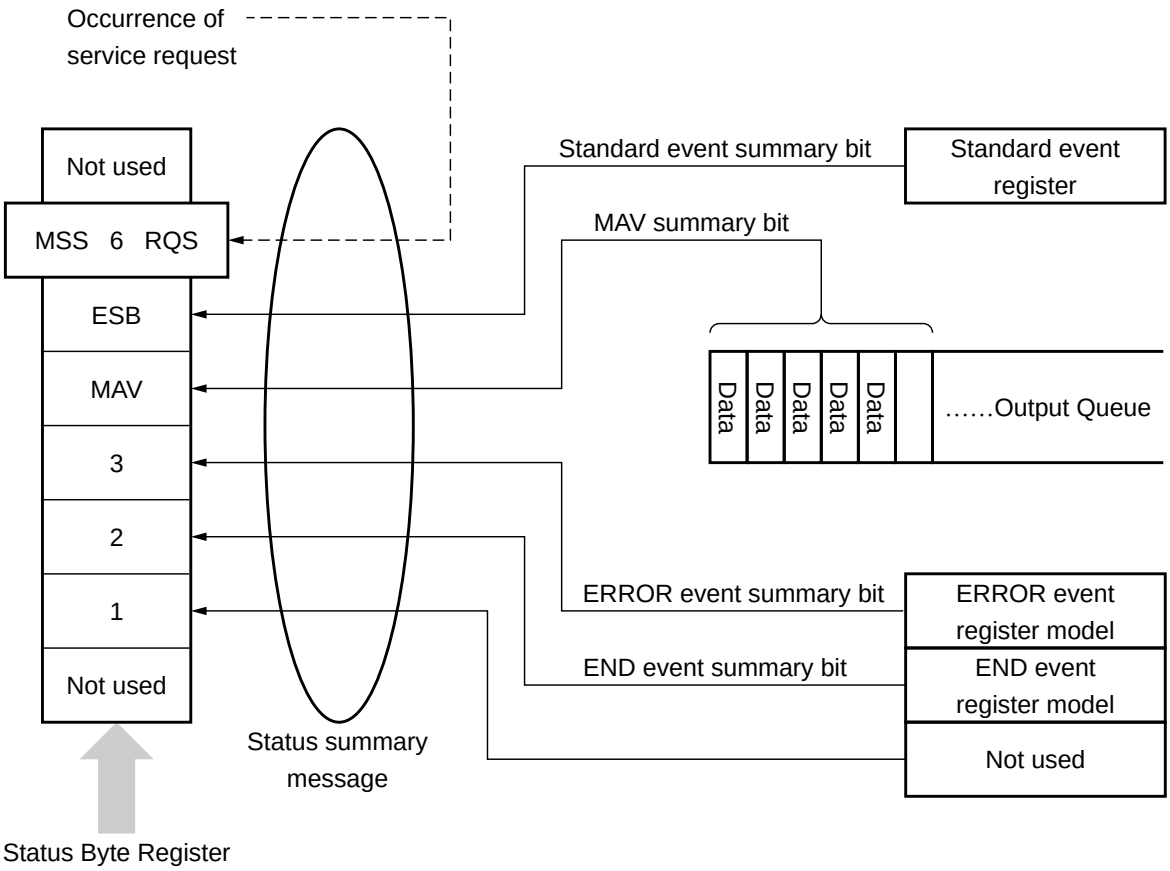
Read	This register is read non-destructively in response to the ∗ESR? common command. In other words, this register is not cleared after being read. The response message is assigned binary weights, converted from a binary value to a decimal value <NR1>, and returned.
Write	This register is written using an ∗ESS common command. Register bits 0 to 8 are assigned weights 1, 2, 4, 8, 16, 32, 64, and 128 respectively, so a total of values of the desired write data bits is sent as <DECIMAL NUMERIC PROGRAM DATA>.
Clearing	This register is cleared in the following cases: (1) An ∗ESE command with its data value being 0 is received. (2) The power is turned on with the Power-ON status clear flag in the True state or the power is turned on when a ∗PSC command is not supported. The standard event status register is not affected by the following: (1) Change in the state of the IEEE 488.1-defined device clear function (2) Reception of an ∗RST common command (3) Reception of a ∗CLS common command

8.5 Extended Event Status Register

Devices conforming to IEEE 488.2 require register models for status byte and standard event status registers including an enable register.

IEEE 488.2 assigns status byte register bit 7 (DIO 8) and bits 3 (DIO 4) to 0 (DIO 1) to the status summary bits transferred from an extended register model and extended queue model.

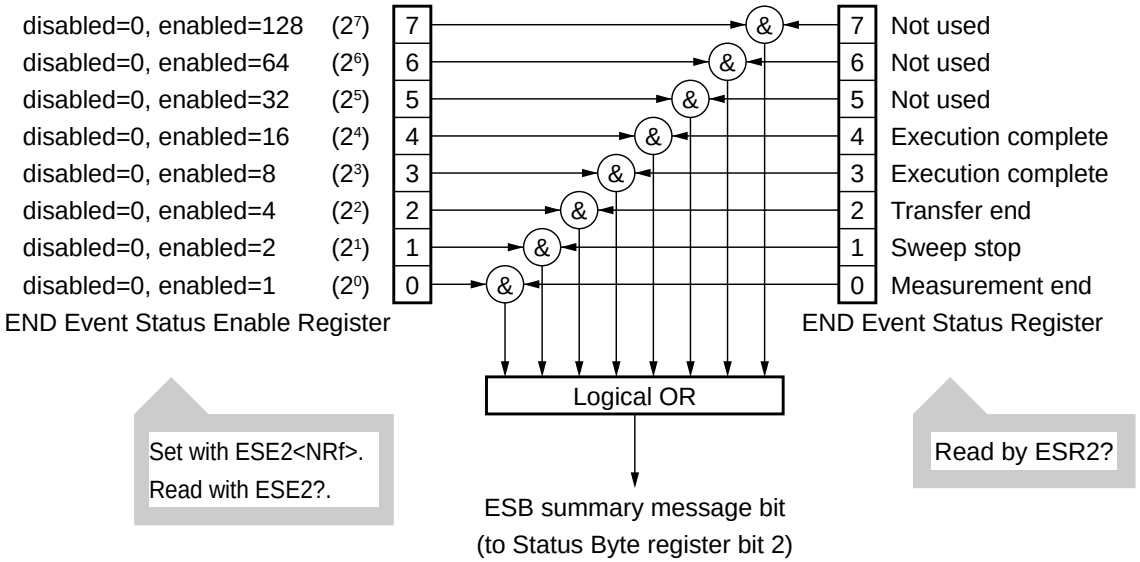
As shown below, the MS9710C does not use bits 7, 1, and 0. It assigns bits 3 and 2 to END and ERROR summary bits for status summary bits transferred from the extended register model.



Let’s take a look at definition, read, write, and clearing of END and ERROR extended event register model bits.

8.5.1 Definition of END event status register bits

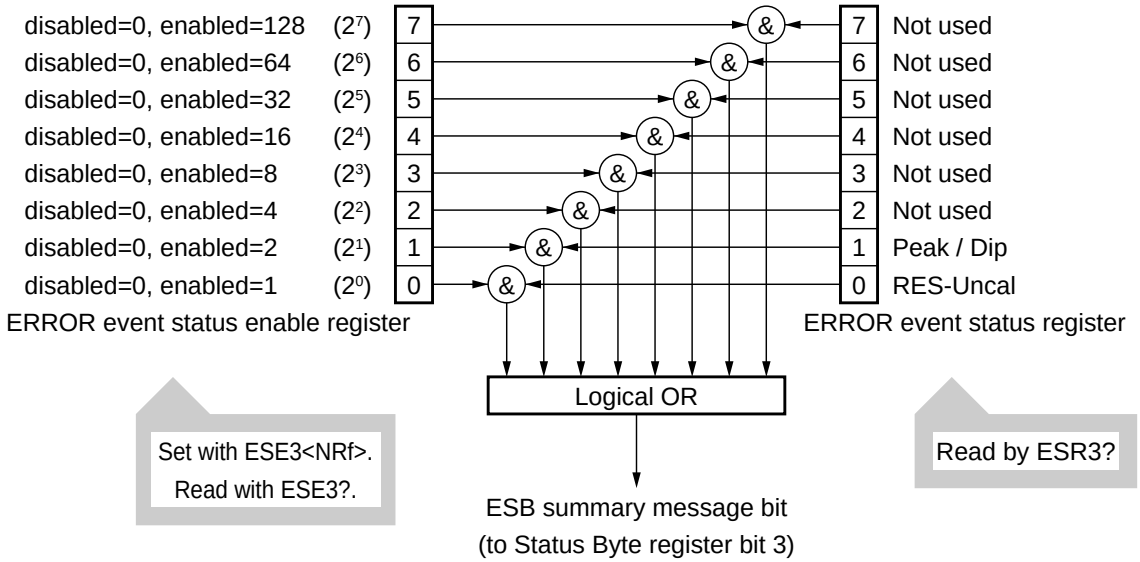
This section explains END event status register model operation and names and meanings of events.



Bit	Event name	Discription
7	Not used	
6	Not used	
5	Not used	
4	Execution complete	Completion of ∗RST, wavelength calibration, automatic axis alignment, or resolution calibration
3	Execution complete	Completion of power monitor 1-point measurement or sweep averaging
2	Transfer end	Completion of transfer to FD or printer output
1	Sweep stop	Single sweep stop
0	Measurement end	Completion of automatic measurement, analysis, peak/dip search, or application measurement

8.5.2 Definition of ERROR event status register bits

This section explains ERROR event status register model operation and names and meanings of event bits.



Bit	Event name	Description
7	Not used	
6	Not used	
5	Not used	
4	Not used	
3	Not used	
2	Not used	
1	Peak/Dip Error	Occurrence of peak/dip detection error
0	RES-Uncal	Occurrence of RES-Uncal error

8.5.3 Reading, writing, and clearing the extended event status register

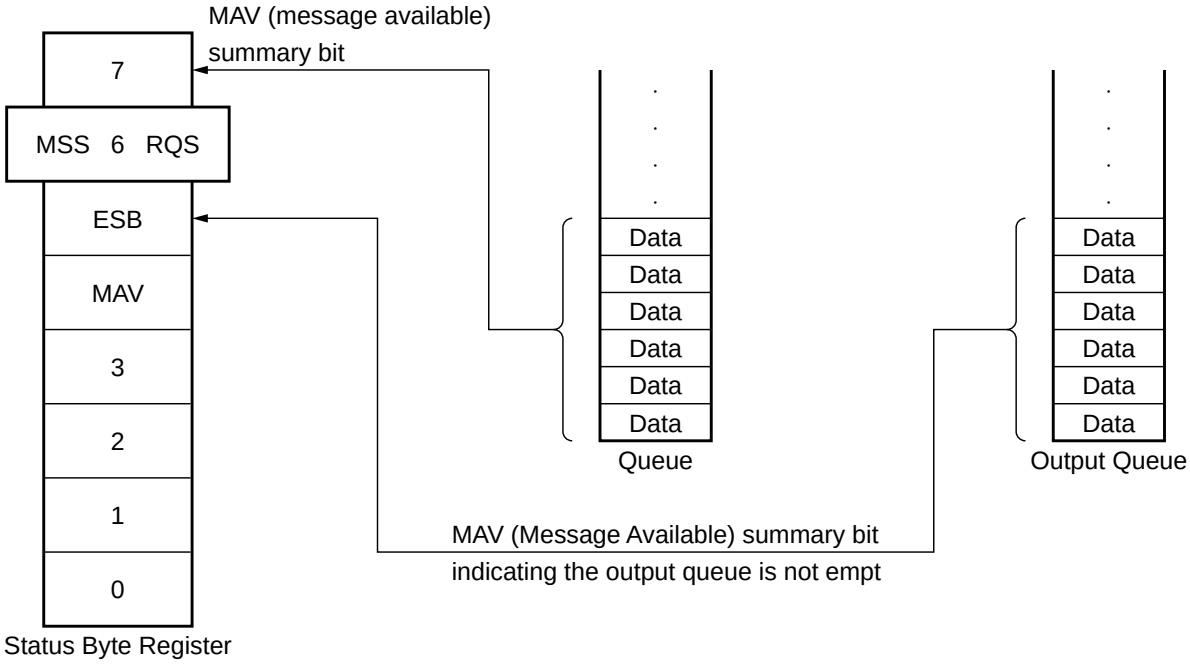
Read	This register is read destructively in response to the ∗ESR? common command. In other words, this register is cleared after being read. The event bit assigned binary weights and converted to a decimal value <NR1> is the response message.
Write	This register cannot be written externally; however, it can be cleared.
Clearing	This register is cleared in the following cases: (1) A ∗CLS command is received. (2) The power is turned on if the Power-ON status clear flag is True. The device executing a Power-ON sequence first clears the standard event status register, then records the events that have occurred in this sequence (e.g., PON event bit setting). (3) An event is read in response to an ∗ESR? query command.

8.5.4 Reading, writing, and clearing the extended event status enable register

Read	This register is read non-destructively in response to the ∗ESR? common command. In other words, this register is not cleared after being read. The response message is assigned binary weights, converted from a binary value to a decimal value <NR1>, and returned.
Write	This register is written using an ∗ESS common command. Register bits 0 to 8 are assigned weights 1, 2, 4, 8, 16, 32, 64, and 128 respectively, so a total of values of the desired write data bits is sent as <DECIMAL NUMERIC PROGRAM DATA>.
Clearing	This register is cleared in the following cases: (1) An ∗ESE command with its data value being 0 is received. (2) The power is turned on with the Power-ON status clear flag in the True state or the power is turned on when a ∗PSC command is not supported. The standard event status register is not affected by the following: (1) Change in the state of the IEEE 488.1-defined device clear function (2) Reception of an ∗RST common command (3) Reception of a ∗CLS common command

8.6 Queue Model

The right-hand illustration shown below is a queue model having a status data structure. A queue is a data structure in which data is arranged sequentially, providing information such as sequential status. A summary message indicates that such information exists in the queue. Queue contents are read by an handshake when the device is in the talker active state (TACS).



The queue that outputs an MAV summary bit to status byte register bit 4 is called an “output queue.” This queue is mandatory. The queue that can output an MAC summary message to one of status byte register bits 0 to 3 and 7 is simply called a “queue.” It is optional. A summary message from the register model can also be output to status byte register bits 0 to 3 and 7, so the summary message type depends on the device type.

We use status byte register bit 7 for the summary message bit transferred from the “queue.” However, we do not use this bit if only the “output queue” suffices and therefore the “queue” need not be used.

The table on the next page provides a comparison of the “output queue” to general queues.

Table Comparsion of Output Queue to General Queues

Item	Output queue	Queue
Data input/output type	FIFO type	Not necessary to be FIFO type
Read	Response message units are read using only an IEEE 488.2 message exchange protocol. The type of these response message units depends on the query type.	Response message units are read with device-dependent query commands. These response message units must be of the same type.
Write	Program message elements are not written directly. This queue communicates with the system interface using only an IEEE 488.2 message exchange protocol.	Program message elements are not written directly. Coded device information is indicated.
Summary message	When the output queue is not empty, the summary message bit becomes True (1). When it is empty, the summary message bit becomes False (0). The MAV summary message is used to synchronize information exchange between a device and the controller.	When the queue is not empty, the summary message bit becomes True (1). When it is empty, the summary message bit becomes False (0).
Clearing	This queue is cleared in the following cases: (a) All items in the queue are read. (b) A DCL bus command is received for message exchange. (c) The PON bit becomes True at Power-ON. (d) Operation is unterminated or interrupted.	This queue is cleared in the following cases: (a) All items in the queue are read. (b) A *CLS command is received. (c) Other device-dependent means

Section 9 Details on Device Messages

9.1 ALIN [Auto Alignment]

■ Function

With the measuring light radiated, executes auto alignment to create alignment position data. When auto alignment is complete, bit 4 (execution complete bit) of the extended event register (ESR2) is set to 1.

Header	Program	Query	Response
ALIN	ALIN n	ALIN ?	m

■ • Value of n

n = 0: Specify the alignment position data as the default.
= 1: Execute auto alignment to create alignment position data.
= 2: Terminate auto alignment forcibly.

■ • Value of m

m = 0: Normal termination of waveform calibration
= 1: Wavelength calibration is in process.
= 2: Wavelength calibration has been interrupted due to an insufficient light level.
= 3: Wavelength calibration has been interrupted due to any other fault.

9.2 ANA [Analysis]

ANA ENV [Spectrum Analysis (Envelope)]

■ Function

Carries out a spectrum analysis using an envelope method.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
ANA	ANA ENV, r	ANA ?	ENV, r

■ Value of r

r is a cut level. The unit is always dB.
Data range: $0.1 \leq r \leq 20.0$

■ Initial setting value

r is a backed up value.

■ Default

r = 3 (dB)

ANA RMS [Spectrum Analysis (RMS)]

■ Function

Carries out a spectrum analysis using an RMS method.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
ANA	ANA RMS, r, k	ANA ?	RMS, r, k

■ • Value of r

r is a slice level. The unit is always dB.
Data range: $0.1 \leq r \leq 30.0$

• Value of k

k is a coefficient (k σ). Input one of the following values
1, 2, 2.35, 3

■ Initial setting value

r and k are backed up values.

■ Default

r = 0 (dB), k = 2.35

ANA ndB [Spectrum Analysis (ndB - Loss)]

■ Function

Carries out a spectrum analysis using the ndB-Loss method.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
ANA	ANA NDB, r	ANA ?	NDB, r

■ Value of r

r is an attenuation value. The unit is always dB.
Data range: $0.1 \leq r \leq 50.0$

■ Initial setting value

r is a backed up value.

■ Default

r = 20 (dB)

ANA THR [Spectrum Analysis (Threshold)]

■ Function

Carries out a spectrum analysis using the threshold method.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
ANA	ANA THR, r	ANA ?	THR, r

■ Value of r

r is a cut level. The unit is always dB.
Data range: $0.1 \leq r \leq 50.0$

■ Initial setting value

r is a backed up value.

■ Default

r = 20 (dB)

ANA SMSR [Spectrum Analysis (SMSR)]

■ Function

Carries out an SMSR analysis.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
ANA	ANA SMSR, s	ANA ?	SMSR, s

■ Value of s

s = 2NDPEAK: Carry out an analysis with respect to the side mode at second highest level.
= LEFT: Carry out an analysis with respect to the side mode on the left (shorter wavelength side) of the peak wave.
= RIGHT: Carry out an analysis with respect to the side mode on the right (shorter wavelength side) of the peak wave.

■ Initial setting value

s is a backed up value.

■ Default

s = 2NDPEAK

ANA PWR [Spectrum Analysis (Spectrum Power)]

■ Function

Carries out power integration.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
ANA	ANA PWR	ANA ?	PWR

ANA OFF [Spectrum Analysis Off]

■ Function

Turns off the analysis mode.

Header	Program	Query	Response
ANA	ANA OFF	ANA ?	OFF

9.3 ANAR [Spectrum Analysis Result]

■ Function

Reads the result of the analysis made with ANA command.
Response data is the result of the last analysis made by executing an AND command.
There are four types of response data formats:

Header	Program	Query	Response
ANAR	None	ANAR ?	λ c , hw : Type 1 λ c , hw, n : Type 2 Δ λ , Δ l : Type 3 p, λ c : Type 4

■ Response data

Type 1: Envelope, Threshold, RMS
Type 2: ndB - Loss
Type 3: SMSR
Type 4: Spectrum - Power

• Value of λ c

λ c is the center wavelength or frequency obtained through an analysis. The unit is nm or THz, and a value is output down to the fourth decimal place.
The value is tail-zero-suppressed.
If analysis is impossible, –1 is output.

• Value of hw

hw is the spectrum width obtained through an analysis. The unit is always nm, and a value is output down to the third decimal place.
The value is tail-zero-suppressed.
If analysis is impossible, –1 is output.

• Value of n

n is the number of axis modes obtained through the analysis carried out using an ndB-Loss method. A positive integer is output.
If analysis is impossible, –1 is output.

• Value of Δ λ

Δ λ is the difference in waveform length between the main peak and the side mode obtained through the SMSR analysis. The unit is always nm, and a value is output down to the third decimal place.
The value is tail-zero-suppressed.
If analysis is impossible, –1 is output.

• Value of Δ l

Δ l is the difference in level between the main peak and the side mode obtained through the SMSR analysis. The unit is dB, and a value is output down to the second decimal place.
If analysis is impossible, –999.99 is output.

• Value of p

p is a power integration result obtained through the spectrum power analysis. The unit is always dBm, and a value is output down to the second decimal place.

9.4 AP [Application]

AP DFB [Application (DFB - LD)]

■ Function

Measures DFB-LD.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
AP	AP DFB, s , n	AP ?	DFB, s , n

■ • Value of s

s = 2NDPEAK: Carry out an analysis with respect to the side mode at second highest level.
= LEFT: Carry out an analysis with respect to the side mode on the left (shorter wavelength side) of the peak wave.
= RIGHT: Carry out an analysis with respect to the side mode on the right (shorter wavelength side) of the peak wave.

• Value of n

n is the value used for ndB-width measurement (wavelength reduced by n dB). The unit is dB. Input an integer.
Data range: $1 \leq n \leq 50$

■ Initial setting value

s and n are backed up values.

■ Default

s = 2NDPEAK
n = 20 (dB)

AP FP [Application (FP - LD)]

■ Function

Measures FP-LD.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
AP	AP FP, n	AP ?	FP, n

■ Value of n

n is the axis mode cut level (difference between the peak level and the cut level) used for measuring the number of axis modes. The unit is dB. Input an integer.
Data range: $1 \leq n \leq 50$

■ Initial setting value

n is a backed up value.

■ Default

n = 20 (dB)

AP LED [Application (LED)]

■ Function

Measures LED.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
AP	AP LED, n , p	AP ?	LED, n , p

■ • Value of n

n is the value used for ndB-width measurement (ndB-down wave width). The unit is dB. Input an integer.
Data range: $1 \leq n \leq 50$

• Value of p

p is a total power compensation value. The unit is dB. Input a value is input down to the second decimal place.
Data range: $-10.00 \leq n \leq +10.00$

■ Initial setting value

n and p are backed up values.

■ Default

n = 3 (dB)
p = 0 (dB)

AP PMD [Application (PMD)]

■ Function

Measures PMD (Polarization Mode Dispersion). (Auto measurement)
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
AP	AP PMD, n	AP ?	PMD, n , m

■ • Value of n

n is a mode coupling coefficient. Input a value down to the second decimal place.
Data range: $0.01 \leq n \leq 1.00$

• Value of m

m indicates a measurement mode.
m = 0: Auto measurement
= 1: Manual measurement

■ Initial setting value

n , P Is a backed up value.

■ Default

n = 1
m = 0 (Auto)

AP AMP [Application (Optical AMP)]

■ Function

Sets/reads the optical AMP measurement mode.

Header	Program	Query	Response
AP	AP AMP	AP ?	AMP

AP AMP MSL [Application (Optical AMP Memory Select)]

■ Function

Selects a measurement memory for optical AMP measurement.
Setting/read is enabled only in the optical AMP measurement mode.

Header	Program	Query	Response
AP	AP AMP, MSL,s	AP ? AMP,MSL	AMP,MSL,s

■ Value of s

s indicates a measurement memory.

s = PIN: Pin memory used to measure the light input to the optical amplifier.
= POUT: Pout memory used to measure the light output from the optical amplifier.

■ Initial setting value

s = PIN

AP AMP CAL [Application (Optical AMP Resolution Calibration)]

■ Function

Calibrates the resolution of the spectrum analyzer for optical AMP measurement.
When the calibration is complete, bit 4 (execution complete bit) of the extended event status register (ESR2) is set to 1.
Setting/read is enabled only in the optical AMP measurement mode.

Header	Program	Query	Response
AP	AP AMP,CAL,n	AP ? AMP,CAL	AMP,CAL,m

■ Value of n

○ Program

Make the following setting with n.

n = 0: Specify the current resolution calibration data as the default.
= 1: Carry out resolution calibration to obtain resolution calibration data.

○ Response

m indicates a resolution calibration state.

m = 0: Resolution calibration has been terminated normally.
= 1: Resolution calibration has been interrupted due to an insufficient light level.
= 2: Resolution calibration has been interrupted due to any other fault.

AP AMP PRM [Application (Optical AMP Parameter)]

■ Function

Sets parameters used for optical AMP measurement.
Setting/read is enabled only in the optical AMP measurement mode.

Header	Program	Query	Response
AP	AP AMP,PRM, a,b,c,d,e,f,g,h,i,j,k	AP ? AMP,PRM	AMP,PRM, a,b,c,d,e,f,g,h,i,j,k

■ • Value of a [NF (S-ASE)/NF (Total)]

Select an NF calculation mode between NF (S-ASE) and NF (TOTAL).
a = 0: NF (S-ASE)
= 1: NF (Total)

• Value of b [Method]

Select an NF measurement method.
b = 0: NF measurement made without using a spectrum division method
= 1: NF measurement made using a spectrum division method
= 2: NF measurement made using a polarization nullification method.
= 3: NF measurement made using a pulse method
= 4: WDM measurement

• Value of c [Gauss/Mean]

Select a fitting method used for obtaining an ASE level (P ASE).
c = 0: Gauss fitting
= 1: Mean

• Value of d [Fitting span]

Specify a span, which represents the shortest and longest wavelengths subject to the fitting carried out to obtain an ASE level (P ASE), with the signal light centered.
The unit is nm. Input a value down to the second decimal place.
Data range: 0.10 nm ≤ d ≤ 100.00 nm

• Value of e [Masked Span]

Specify a span, which represents the wavelength range (near the signal light) where fitting is carried out to obtain an ASE level (P ASE), with the signal light centered.
Fitting is carried out in the d-specified range excluding the e-specified range.
The unit is always nm. Input a value down to the second decimal place.
Data range: 0.10 nm ≤ d ≤ 100.00 nm

Note:

If the d- and e-specified wavelength area for fitting exceeds the measurement range, the wavelength data in this area is excluded from fitting.

- **Value of f [Pin Loss]**

  Specify the difference between the level of the signal light input to the optical spectrum analyzer and the level of the signal light input to the optical amplifier.
  The unit is dB. Input a value down to the second decimal place.
  Data range: –10.00 dB ≤ f ≤ + 10.00 dB

- **Value of g [Pout Loss]**

  Specify the difference between the level of the optical amplifier output signal input to the optical spectrum analyzer and the actual level of the optical amplifier output signal.
  The unit is dB. Input a value down to the second decimal place.
  Data range: –10.00 dB ≤ g ≤ + 10.00 dB

- **Value of h [NF Cal]**

  Specify a value for calibrating the result of NF calculation.
  Input a value down to the third decimal place.
  Data range: 0.100 ≤ h ≤ 10.000

- **Value of i [O.BPFL-Cal]**

  Specify the difference in level between the pass and block ranges of the optical band pass filter inserted between the optical amplifier and the optical spectrum analyzer.inserted.
  The unit is dB. Input a value down to the second decimal place.
  Data range: 0.00 dB ≤ i ≤ 30.00 dB

- **Value of j [O.BPF BW]**

  Specify an effective optical filter width (total ASE bandwidth) used for NF (total) calculation.
  The unit is always nm. Input a value down to the second decimal place.
  Data range: 0.01 nm ≤ j ≤ 999.99 nm

- **Value of k [Pol Loss]**

  Specify a level loss at the polarization control nulling stage used for polarization nullification.
  The unit is dB. Input a value down to the second decimal place.
  Data range: –10.00 dB ≤ k ≤ 10.00 dB

■ Initial setting value

a to k are backed up values.

■ Default

a =	0	(NF (S-ASE))
b =	0	(Spect Div On)
c =	0	(Gauss)
d =	5.0	(nm)
e =	2.0	(nm)
f =	0.00	(dB)
g =	0.00	(dB)
h =	1.000	
i =	0.00	(dB)
j =	3.00	(nm)
k =	0	(dB)

AP AMP ASE [Application (Pout → Pase)]

■ Function

During the measurement made using a polarization nullification method, copies the spectrum temporarily written into the Pout memory to the internal memory Pase as an ASE.
Setting is enabled only in the optical AMP measurement mode.

Header	Program	Query	Response
AP	AP AMP, ASE		

AP WDM [Application (WDM)]

■ Function

Sets to WDM Application
The display mode at this time uses the previously backed-up value.

Header	Program	Query	Response
AP	AP WDM	AP? WDM	WDM, m
	AP WDM, SLV, n	AP? WDM, SLV	WEM, SLV, n
	AP WDM, MPK	AP? WDM, MPK	WDM, MPK
	AP WDM, SNR, d, Δ λ, s	AP? WEM, SNR	WDM, SNE, d, Δ λ, s
	AP WDM, REL, r	AP? WDM, REL	WDM, REL, r
	AP WDM, TBL, d, Δ λ, s	AP? WDM, TBL	WDM, TBL, d, Δ λ, s

■ Send Data

• Value of n

Sets splice level at splice level setting command (SLV)
n = 1 to 50 [dB]

• Value of d

Sets noise detection direction at SNR display
d = LEFT: Left side
= RIGHT: Right side
= HIGHER: Side with highest level
= AVERAGE: Average of LEFT and RIGHT

• Value of Δ λ

Setting for detection position using wavelength difference at SNR display. Evaluates SNR of point at Δλ from peak in direction specified at s.
Δλ = 0 [nm] or OFF : Auto-detects Dip in specified direction
Δλ = 0 to +20 [nm] (0.01 step)

• Value of r

Sets reference peak to r number at REL (Relative) display

r = 1 to 128

• Value of s

Sets whether the detected noise value should be normalized with the effective resolution.

s = "ON"

= "OFF"

■ Response Data

m indicates the display mode as follows:

m = MPK: Multi Peak display

= SNR: SNR display

= REL: Relative display

Refer to the send data items for n, s, Δλ, and r.

AP WDM, PKT, t [Application WDM Peak Type]

■ Function

Sets the method for detecting the signal wavelength in the WDM application.

Header	Program	Query	Response
AP	AP WDM, PKT, t	AP ? WDM, PKT	WDM, PKT, t

■ Parameters

• Value of t

Selects a method for detecting the signal wavelength in the WDM application.

○ MAX ..... The wavelength at the maximum point shall be the signal wavelength.

○ THRESHOLD .. The Threshold analysis central wavelength shall be the signal wavelength.

■ Initial setting value

t is a backed up value.

■ Default

t = MAX

AP WDM, TCL, u [Application WDM Threshold Cue Level]

■ Function

Sets the cut level for calculating the signal wavelength by using the Threshold analysis in the WDM application.

Header	Program	Query	Response
AP	AP WDM, TCL, u	AP ? WDM, TCL	WDM, TCL, u

■ Value of u

Sets the cut level for calculating the signal wavelength by using the Threshold analysis in the WDM application.

$0.1 \leq u \leq 50.0$

0.1 Step

■ Initial setting value

u is a backed up value.

■ Default

u = 3.0 dB

AP OFF [Application Off]

■ Function

Terminates application measurement.

Header	Program	Query	Response
AP	AP OFF	AP ?	OFF

9.5 APR [Application Result]

■ Function

Reads the result of analysis carried out with an AP command.
Response data is the result of the last analysis made by executing an AP command.

Header	Program	Query	Response
APR	None	APR ?	SMSR, BWndb, λ p, Lp, λ sm, Lsm, MOFS, STBW, CNTOFS :Type 1 FWHM, λ m, λ p, Lp, MODE, MSPC, POW :Type 2 λ fwhm, λ ndb, FWHM, BWndb, λ p, Lp, PKdens, POW :Type 3 Δ t, λ 1st, λ last, PKcount :Type 4 G,NF ,λ sig, Lase, RES :Type 5

■ Response data

Values are tail-zero-suppressed.

Type 1: DFB-LD measurement

• Value of SMSR

SMSR indicates a side mode suppression ratio
The unit is dB. A value is output down to the second decimal place.
When analysis is impossible, –999.99 is output.

• Value of BWndb

BWndb indicates an wave width (ndB-down width).
The unit is nm. A value is output down to the third decimal place.
When analysis is impossible, –1 is output.

• Value of λp

λp indicates a main peak wavelength.
The unit is nm. A value is output down to the third decimal place.
When analysis is impossible, –1 is output.

• Value of Lp

Lp indicates a main peak level.
The unit is nm. A value is output down to the second decimal place.
When analysis is impossible, –999.99 is output.

- **Value of λ sm**

  λ sm indicates a side move wavelength.
  The unit is nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of L sm**

  L sm indicates a side mode level.
  The unit is dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of MOFS**

  MOFS indicates a mode offset (difference in wavelength between the main peak and side mode).
  The unit is nm. A value is output down to the third decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of STBW**

  STBW indicates a stop band (interval between wavelengths of both side modes of main peak).
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of CNTOFS**

  CNTOFS indicates a center offset (difference between main peak wavelength and center wavelength of both side modes).
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –999.99 is output.

Type 2: FP-LD measurement

- **Value of FWHM**

  FWHM indicates the half-magnitude full width obtained through the RMS analysis (k = 2.35).
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of λ m**

  λ m indicates the center wavelength obtained through the RMS analysis (k = 2.35).
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of λp**

  λp indicates a main peak wavelength.
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of Lp**

  Lp indicates a main peak level.
  The unit is dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of MODE**

  MODE indicates the number of axis modes used in the ndB-Loss analysis. An integer is output.
  When analysis is impossible, –1 is output.

- **Value of MSPC**

  MSPC indicates a mode spacing (axis mode interval)
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of POW**

  POW indicates a power integration value.
  The unit is always dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

Type 3: LED measurement

- **Value of λ fwhm**

  λ fwhm indicates the center wavelength obtained through the RMS analysis (k = 2.35).
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of λ ndb**

  λ ndb indicates the center wavelength obtained through the threshold analysis (ndB).
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of FWHM**

  FWHM indicates the half-magnitude full width obtained through the RMS analysis (k = 2.35)
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of BWndb**

  BWndb indicates the wave width obtained through the threshold analysis (ndB).
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of λ p**

  λ p indicates a peak wavelength.
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of Lp**

  Lp indicates a peak level.
  The unit is dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of PKdens**

  PKdens indicates a peak power per mm.
  The unit is always dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of POW**

  POW indicates a power integration value.
  The unit is always dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

Type 4: PMD measurement

- **Value of Δ t**

  Δ t indicates a differential group delay time.
  The unit is always fs. A value is output down to the second decimal place.
  When analysis is impossible, –1 is output.

- **Value of λ 1st**

  λ 1st indicates the first peak wavelength.
  The unit is always nm. A value is output down to the fourth decimal place.
  When analysis is impossible, –1 is output.

- **Value of λ last**

  λ last indicates the last peak wavelength.
  The unit is always nm. A value is output down to the fourth decimal place.
  When analysis is impossible, –1 is output.

- **Value of PKcount**

  PKcount indicates a number of peaks. A positive integer is output.
  When analysis is impossible, –1 is output.
  Type 5: Optical AMP measurement

Type 5: Optical AMP measurement

- **Value of G**

  G indicates the gain of the optical amplifier.
  The unit is always dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of NF**

  NF indicates a noise figure.
  The unit is always dB. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of λ sig**

  λ sig indicates a signal light wavelength.
  The unit is always nm. A value is output down to the third decimal place.
  When analysis is impossible, –1 is output.

- **Value of L ase**

  L ase indicates the ASE level per resolution.
  The unit is always dBm. A value is output down to the second decimal place.
  When analysis is impossible, –999.99 is output.

- **Value of RES**

  RES indicates the resolution data used in the NF calculation.
  The unit is always nm. A value is output down to the third decimal place.

APR? MPKC [ApplicationResult (Multi Peak Counter)]

■ Function

Reads the number of peaks according to the multipeak detection result.

Header	Program	Query	Response
APR	None	APR ? MPKC	MPKC, d

■ Response data

d indicates a number of peaks.
The unit is not used.
Data range: $0 \leq d \leq 300$

APR? WDM [Application Result (WDM)]

■ Function

Queries WDM Application analysis results

Header	Program	Query	Response
APR	None	APR? (analysis result link obtained)	n, λ1, L1, λ2, L2...
			n, λ1, L1, S1, d1, l2, L2, S2, d2...
			n, Rn, λ1, SP1, Rλ 1, L1, RL1, λ2, SP2, Rλ2, L2, RL2...
			n, λ1, f1, L1, S1, d1, SP1, SPf1, λ2, f2, L2, S2, d2, SP2, SPf2...
	None	APR? MPKC	MPKC, n
	None	APR? WDM, MPK, m	WDM, MPK, λ, L
	None	APR? WDM, SNR, m	WDM, SNR, λ, L, S, d
	None	APR? WDM, SNR, GAV	t
	None	APR? WDM, REL, m	WDM, REL, λ, SP, Rλ, L, RL
	None	APR? WDM, TBL, m	WDM, TBL, λ, f, L, S, d, SP, SPf

■ Send Data

• Value of m

m indicates the query peak number.
m = 1 to 300

■ Response

• Value of n

n is the peak number.
n = 0 to 300

• Value of λk (λ1, λ2, λ3,...)

λk indicates the wavelength of peak No. k.
λk = xxxx.xx in nm units to two decimal places

• Value of fk (f1, f2, f3,...)

fk indicates the value of the frequency of peak No. k.
f = xxx.xxxx: Outputs values in fixed THz units to four decimal places.

• Value of Lk (L1, L2, L3,...)

Lk indicates the level of peak No. k.
Lk = xxxx.xx in dBm unit to two decimal places.

- **Value of Sk (S1, S2, S3,...)**

  Sk indicates the SNR of peak No. k.
  Sk = xxx.xx in dB units to two decimal places.
  The first Spacing S1 is 0.

- **Value of dk (RL1, RLS2, RL3,...)**

  dk is the noise detection direction for peak No.k.
  dk = LEFT, RIGHT, or ERR (when Dip not detected)

- **Value of Rn**

  Rn indicates the reference peak No.
  Rn = 1 to 128

- **Value of SPk (SP1, SP2, SP3,..)**

  SPk indicates the spacing of peak No.k.
  SPk = xxxx.xxx in nm units to three decimal places
  However, since there is no SP1 spacing data, SP1 = 0.

- **Value of SPfk(SPf1, SPf2, SPf3...)**

  SPfk indicates the value of the spacing frequency of peak k.
  SPfk = xxxx.xx: Outputs values in fixed GHz units to two decimal places. The spacing data for SPf, however, does not exist, thus, outputs Spf1 = 0.

- **Rλ (Rλ1, Rλ2, Rλ3,...)**

  Rλk indicates the relative wavelength of peak No. k.
  Rλk = xxx.xxx in nm units to three decimal places

- **Value of RLk (RL1, RL2, RL3,...)**

  RLk indicates the relative level of peak No. k.
  RLk = xxx.xx in dB units to two decimal places

- **Value of λ**

  λ indicates the wavelength of the specified peak
  λ = xxxx.xx in nm to two decimal units

- **Value of f**

  f indicates the value of the frequency of the specified peak.
  f=xxx.xxxx: Outputs values in fixed THz units to two decimal places.

- **Value of L**

  L indicates the level of the specified peak.
  L = xxxx.xx in dBm units to two decimal places

- **Value of S**

  S indicates the SNR of the specified peak
  S = xxx.xx in dB units to two decimal places

- **Value of t**

t indicates the gain tilt.
t = xx.xx in dB units to two decimal places

- **Value of SP**

SP indicates the spacing of the specified peak.
SP = xxxx.xxx in nm units to three decimal places

- **Value of Spf**

SPf indicates the value of the spacing frequency of the specified peak.
SPf = xxxx.xx: Outputs values in fixed GHz units to two decimal places.
The spacing data for m = 1, however, does not exist, thus, outputs SPf = 0.

- **Value of Rλ**

Rλ indicates the relative wavelength of the specified peak.
Rλ = xxxx.xxx in nm units to three decimal places.

- **Value of RL**

RL indicates the relative level of the specified peak.
RLk = xxxx.xx in dB units to two decimal places.

When there is no data each of the above returns wavelength l = –1, and Level L = –999.99.

9.6 ARED [Acutual Resolution Data]

■ Function

Reads the actual resolution value.

Header	Program	Query	Response
ARED	None	ARED ?	n

■ Value of n

The unit is always nm. A value is output down to the third decimal place.

9.7 ARES [Actual Resolution]

■ Function

Determines whether the actual resolution is to be displayed.

Header	Program	Query	Response
ARES	ARES s	ARES ?	s

■ Value of s

s = ON: Display the actual resolution.
= OFF: Do not display the actual resolution.

■ Initial setting value

s is a backed up value.

■ Default

s = OFF

9.8 ATT [Optical Attenuater]

■ Function

Turns on/off the internal optical attenuator.

Header	Program	Query	Response
ATT	ATT s	ATT ?	s

■ Value of s

s = ON: Turns on the optical attenuator.
= OFF: Turns off the optical attenuator.

9.9 AUT [Auto Measure]

■ Function

Carries out auto measurement. The wavelength and resolution are automatically set for the incident light spectrum.
When the measurement is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
AUT	AUT	AUT ?	n

■ Value of n

n indicates a measurement state. 0 or 1 is output.
0: Measurement has been completed.
1: Measurement is being carried out.

9.10 AVS [Sweep Average]

■ Function

Sets the number of sweep operations for sweep averaging.

Header	Program	Query	Response
AVS	AVS n	AVS ?	n

■ Value of n

The unit is number of times. Input an integer comprising 1 to 3 digits or OFF.
Data range: $2 \leq n \leq 1000$, or OFF

■ Initial setting value

n is a backed up value.

■ Default

n = OFF

9.11 AVT [Point Average]

■ Function

Sets the number of points for point averaging.

Header	Program	Query	Response
AVT	AVT n	AVT ?	n

■ Value of n

The unit is number of points. Input an integer comprising 1 to 3 digits or OFF.
Data range: $2 \leq n \leq 1000$, or OFF

■ Initial setting value

n is a backed up value.

■ Default

n = OFF

9.12 BKL [Back Light]

■ Function

Sets the time until the backlight turns off.

Header	Program	Query	Response
BKL	BKL n	BKL ?	n

■ Value of n

n is the time until the backlight is turned off. The unit is always min. Input a value within the range of 0 to 20.
If 0 is input, the backlight will not turned off.

■ Initial setting value

n = 10 (min)

9.13 BUZ [Buzzer ON/OFF]

■ Function

Turns on or off the buzzer.

Header	Program	Query	Response
BUZ	BUZ s	BUZ ?	s

■ Value of s

ON: Turn on the buzzer.
OFF: Turn off the buzzer.

■ Initial setting value

s is a backed up value.

■ Default

s = ON

9.14 CNT [Center Wavelength]

■ Function

Sets a center wavelength.

Header	Program	Query	Response
CNT	CNT λ	CNT ?	λ

\■ Value of l

The unit is always nm. Input a value down to the second decimal place.
Data range: 600.00 ≤ l ≤ 1750.00

■ Initial setting value

λ is a backed up value.

■ Default

λ = 1350 (nm)

9.15 CPY [Copy]

■ Function

Copies the screen to the internal printer.
When transfer to the printer is complete, bit 2 (transfer end bit) of the extended event status register (ESR2) is set to 1.
Execute a CPY command with the printer set Int state.

Header	Program	Query	Response
CPY	CPY	None	None

9.16 CRCL [Condition Recall]

■ Function

Recalls the condition of the internal backup RAM as the current condition.

Header	Program	Query	Response
CRCL	CRCL n	None	None

■ Value of n

n is the number of the memory whose condition is to be recalled. Input an integer between 0 and 5.

9.17 CSAV [Condition Save]

■ Function

Saves the current condition in the internal backup RAM.

Header	Program	Query	Response
CSAV	CSAV n	None	None

■ Value of n

n is the number of the memory, 1 to 5, into which the current condition is to be saved.

9.18 DATE [Date Set]

■ Function

Sets the year, month, and day.

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

■ • Value of yy

Input the last two digits (00 to 99) of Gregorian year.

• Value of mm

Input two digits (01 to 12) of month.

• Value of dd

Input two digits (01 to 31) of day.

9.19 DBA [Memory Data A]

■ Function

Outputs binary measurement data equivalent to the number of sampling points from memory A.
The count of data in memory A is equal to the number of sampling points.

Header	Program	Query	Response
DBA	None	DBA ?	d (equivalent to the number of sampling points; binary data)

■ Value of d

d indicates measurement data. The data format depends on the scale.
See Appendix B, “Binary Data Transfer Format.”

Note:

In the 3-dimensional or overlapping display mode, the last measurement data is output.
In the normalize display mode, an absolute level is output.

9.20 DBB [Memory Data B]

■ Function

Outputs binary measurement data equivalent to the number of sampling points from memory B.
The count of data in memory B is equal to the number of sampling points.

Header	Program	Query	Response
DBB	None	DBB ?	d (equivalent to the number of sampling points; binary data)

■ Value of d

d indicates measurement data. The data format depends on the scale.
See Appendix B, “Binary Data Transfer Format.”

Note:

In the 3-dimensional or overlapping display mode, the last measurement data is output.
In the normalize display mode, an absolute level is output.

9.21 DCA [Data Condition Memory A]

■ Function

Reads data measurement conditions from memory A.

Header	Program	Query	Response
DCA	None	DCA ?	λ 1, λ 2, n

■ • Value of λ 1

λ 1 indicates a start wavelength. The unit is always nm. A value is output down to the second decimal place.

• Value of λ 2

λ 2 indicates a start wavelength. The unit is always nm. A value is output down to the second decimal place.

• Value of n

n indicates a sampling point, 251, 501, 1001, 2001, or 5001.

9.22 DCB [Data Condition Memory B]

■ Function

Reads data measurement conditions from memory B.

Header	Program	Query	Response
DCB	None	DCB ?	λ 1, λ 2, n

■ • Value of λ 1

λ 1 indicates a start wavelength. The unit is always nm. A value is output down to the second decimal place.

• Value of λ 2

λ 2 indicates a start wavelength. The unit is always nm. A value is output down to the second decimal place.

• Value of n

n indicates a sampling point, 251, 501, 1001, 2001, or 5001.

9.23 DEL [FD File Delete]

■ Function

Deletes the specified file from the FD.
When file deletion is complete, bit 2 (transfer end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
DEL	DEL n	None	None

■ • Value of n

Input a DOS-recognizable file name comprising eight or fewer characters in the following format:
xxxxxxxx.dat
The file extension ".dat" may be omitted.

9.24 DMA [Memory Data A]

■ Function

Outputs ASCII measurement data equivalent to the number of sampling points from memory A.
The count of data in memory A is equal to the number of sampling points.

Header	Program	Query	Response
DMA	None	DMA ?	d + Terminator (equivalent to the number of sampling points)

■ Value of d

d indicates measurement data. The data format depends on the scale.
The response data does not have a header, and x's are zero-suppressed.

○ Log scale ..... d =± xxx.xx (Unit: dBm)
(–120 dBm ≤ d ≤ +30 dBm)

○ Linear scale ..... d = xx.xxxxE ± xx (Unit: mW)
(0.1000 < Mantissa part ≤ 1.0000, –8 ≤ Exponent ≤ +3)

Note:

In the 3-dimensional or overlapping display mode, the last measurement data is output.
In the normalize display mode, an absolute level is output.

9.25 DMB [Memory Data B]

■ Function

Outputs ASCII measurement data equivalent to the number of sampling points from memory B.
The count of data in memory B is equal to the number of sampling points.

Header	Program	Query	Response
DMB	None	DMB ?	d + Terminator (equivalent to the number of sampling points)

■ Value of d

d indicates measurement data. The data format depends on the scale.
The response data does not have a header, and x's are zero-suppressed.

○ Log scale ..... d =± xxx.xx (Unit: dBm)
(–120 dBm ≤ d ≤ +30 dBm)

○ Linear scale ..... d = xx.xxxxE ± xx (Unit: mW)
(0.1000 < Mantissa part ≤ 1.0000, –8 ≤ Exponent ≤ +3)

Note:

In the 3-dimensional or overlapping display mode, the last measurement data is output.
In the normalize display mode, an absolute level is output.

9.26 DMD [Display Mode]

■ Function

Sets a display mode.

Header	Program	Query	Response
DMD	DMD s DMD 3, m, n	DMD ?	s 3, m, n

■ • Value of s

s = NRM: Specify a normal display mode.
= NRMZ: Specify a normalize display mode.
= OVL: Specify an overlapping display mode.
= MHL: Specify a max. hold display mode.
= 3: Specify a 3-dimensional display mode.

• Values of m and n

m is a display type of the 3-dimensional display mode. Input 1, 2, or 3.
m = 1 : Type 1
2 : Type 2
3 : Type 3
n is a display angle of the 3-dimensional display mode. Input, 30, 45, 60, or 90.
(When 3 is input for the display type, only 45 can be input.)

■ Initial setting value

s = NRM
m, n is a backed up value.

■ Default

x = NRM (normal)
m = 1 (type), n = 45 (angle)

9.27 DMK [Δ Marker]

■ Function

Sets the Δ marker with a wavelength and reads the differences in wavelength and level between the Δ marker and the trace marker.

Header	Program	Query	Response
DMK	DMK λ	DMK ?	Δλ , Δl Δf , Δl

■ • Value of λ

λ is a wavelength value.
The unit is nm or THz. Input a wavelength down to the fourth decimal place and input a frequency down to the fifth decimal place.
Data range: Start wavelength ≤ λ ≤ Stop wavelength

• Value of Δλ

Δλ is the difference in wavelength between the Δ marker and the trace marker.
The unit is always nm. A value is output down to the fourth decimal place.

• Value of Δf

Δf is the value of the frequency of Δλ.

• Value of Δl

Δl is the difference in level between the Δ marker and the trace marker. The level difference andthe unit on the specified scale are output.
When the log scale is selected, a value is output down to the second decimal place. When the linear scale is selected, a value is output down to the third decimal place.
If analysis is impossible, –1 is output. (Linear scale only)

■ Suffix

Δl
Log scale: DB
Linear scale: No unit

9.28 DPS [Dip Search]

■ Function

Detects a spectrum dip and moves the trace marker there.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.
To move the dip, specify DIP, NEXT, LAST, LEFT, or RIGHT.
To read the dip wavelength and level, use a TMK? command.

Header	Program	Query	Response
PTS	PTS s	PTS ?	PTS m

■ Value of s

s = DIP: Detects the main dip whose level is highest, and move the trace marker there.
= NEXT: Detects the dip whose level is the second highest compared with the current dip, and moves the trace marker there.
= LAST: Detects the dip whose level is the second lowest compared with the current dip, and moves the trace marker there.
= LEFT: Detects the dip whose wavelength is the second shortest compared with the current dip, and moves the trace marker there.
= RIGHT: Detects the dip whose wavelength is the second longest compared with the current dip, and moves the trace marker there.

■ Value of m

m = DIP, NEXT, LAST, LEFT, RIGHT
= ERR (state other than dip search)

■ Initial setting value

s = DIP

9.29 DQA [Memory Data A]

■ Function

Outputs ASCII measurement data equivalent to the number of sampling points from memory A.
The count of data in memory A is equal to the number of sampling points.

Header	Program	Query	Response
DQA	None	DQA ?	d + Separator (equivalent to the number of sampling points)

■ Value of d

d indicates measurement data. The data format depends on the scale.
The response data does not have a header, and x's are zero-suppressed.

○ Log scale ..... d = ± xxx.xx (Unit: dBm)
(–120 dBm ≤ d ≤ +30 dBm)

○ Linear scale ..... d = x.xxxxE ± xx (Unit: mW)
(0.1000 < Mantissa part ≤ 1.0000, –8 ≤ Exponent ≤ +3)

Note:

In the 3-dimensional or overlapping display mode, the last measurement data is output.
In the normalize display mode, an absolute level is output.

9.30 DQB [Memory Data B]

■ Function

Outputs ASCII measurement data equivalent to the number of sampling points from memory B.
The count of data in memory B is equal to the number of sampling points.

Header	Program	Query	Response
DQB	None	DQB ?	d + Separator (equivalent to the number of sampling points)

■ Value of d

d indicates measurement data. The data format depends on the scale.
The response data does not have a header, and x's are zero-suppressed.

○ Log scale ..... d = ± xxx.xx (Unit: dBm)
(–120 dBm ≤ d ≤ +30 dBm)

○ Linear scale ..... d = x.xxxxE ± xx (Unit: mW)
(0.1000 < Mantissa part ≤ 1.0000, –8 ≤ Exponent ≤ +3)

Note:

In the 3-dimensional or overlapping display mode, the last measurement data is output.
In the normalize display mode, an absolute level is output.

9.31 DRG [Dynamicrange Mode]

■ Function

Sets a high or normal dynamic range.

Header	Program	Query	Response
DRG	DRG s	DRG ?	s

■ Value of s

s indicates a selected dynamic range.
High dynamic range ..... HIGH
Normal dynamic range ..... NORMAL

■ Initial setting value

s is a backed up value.

■ Default

s = NORMAL

9.32 EMK [Marker Off]

■ Function

Deletes all markers

Header	Program	Query	Response
EMK	EMK	None	None

9.33 ERR [Error]

■ Function

Reads the number of the error occurred during GPIB operation.

Header	Program	Query	Response
ERR	None	ERR ?	n

■ Value of n

n indicates an error code number, a 3-digit integer.

An error number is set when the ESB bit (bit 5) of the status byte register (STB) is on and the command error bit (bit 5), execution error bit (bit 4), or device dependent error bit (bit 3) of the standard event status register (ESR) is on.

9.34 ESE1 [Extended Event Status Enable Register1]

■ Function

Sets/reads the extended event status register 1 enable register.

Header	Program	Query	Response
ESE1	ESE1 n	ESE1 ?	n

■ Value of n

n is a positive integer, 0 to 255. When n = 0, the register is disabled.

■ Initial setting value

n = 0

9.35 ESE2 [Extended Event Status Enable Register2]

■ Function

Sets/reads the extended event status register 2 enable register.

Header	Program	Query	Response
ESE2	ESE2 n	ESE2 ?	n

■ Value of n

n is a positive integer, 0 to 255. When n = 0, the register is disabled.

■ Initial setting value

n = 0

9.36 ESE3 [Extended Event Status Enable Register3]

■ Function

Sets/reads the extended event status register 3 enable register.

Header	Program	Query	Response
ESE3	ESE3 n	ESE3 ?	n

■ Value of n

n is a positive integer, 0 to 255. When n = 0, the register is disabled.

■ Initial setting value

n = 0

9.37 ESR1 [Extended Event Status Register1]

■ Function

Reads the extended event status register 1 information generated by GPIB operation.

Header	Program	Query	Response
ESR1	None	ESR1 ?	n

■ Value of n

n is not used, so it is always 0.

9.38 ESR2 [Extended Event Status Register2]

■ Function

Reads the extended event status register 2 information generated by GPIB operation.

Header	Program	Query	Response
ESR2	None	ESR2 ?	n

■ Value of n

n is an integer, 0 to 255.

9.39 ESR3 [Extended Event Status Register3]

■ Function

Reads the extended event status register 3 information generated by GPIB operation.

Header	Program	Query	Response
ESR3	None	ESR3 ?	n

■ Value of n

n is an integer, 0 to 255.

9.40 FED [Feed]

■ Function

Causes the internal printer to feed lines.

Header	Program	Query	Response
FED	FED n	None	None

■ Value of n

n is a number of lined to be fed, 0 to 255.

9.41 FMT [FD Format]

■ Function

Formats the FD.
When formatting is complete, bit 2 (transfer end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
FMT	FMT	None	None

9.42 FOPT [FD File Option]

■ Function

Sets FD file options.

Header	Program	Query	Response
FOPT	FOPT a, b, c	FOPT ?	a, b, c

■ • Value of a

Specify an additional file to be saved.
a = NONE: None (No additional file)
= BMP: ∗bimp file output
= TXT: ∗txt (text) file output
= BMP&TXT: bimp and text file output

• Value of b

Specify a file ID.
b = NUMBER: File number input
= NAME: File name input

• Value of c

Specify an FDD mode. c may be omitted.
c = 1.44 M: PC-compatible machine
= 1.2 M: PC98 Series (NEC/EPSON)

Note:
c can be changed by turning off the power and then turning it on.

9.43 GCL [Graph Clear]

■ Function

Clears the waveform.

Header	Program	Query	Response
GCL	GCL	None	None

9.44 HEAD [Header]

■ Function

Determines whether a header is to be added to the data replied to a query command.

Header	Program	Query	Response
HEAD	HEAD s	None	None

■ s Value of

s = ON: Add a header.
= OFF:Do not add a header.

■ Initial setting value

s = OFF

9.45 ITM [Interval Time]

■ Function

Sets a sweep start interval.

Header	Program	Query	Response
ITM	ITM s	ITM ?	s

■ Value of s

s is an interval time. Input a unit and a value (S' = 0 to 99), or input a value with its unit fixed at sec. (When the value is greater than 99 sec, it is displayed in min.)
Unit ..... MIN, SEC
Data range ..... 0 sec ≤ s' ≤ 99 min

■ Initial setting value

s is a backed up value.

■ Default

s = "0 SEC"

9.46 LCD [Display Color]

■ Function

Sets a display color.

Header	Program	Query	Response
LCD	LCD p, r, g, b	LCD ?	r, g, b

■ • Value of p

Specify a screen number with one of the following numbers:

p = 0: Grid
1: Graph-B
2: Graph-A
3: LMkr
4: l Mkr
5: Card
6: Card Shadow
7: Zone
8: Zone Chara.
9: Window
10: Character

• Values of r, g, and b

Specify levels of three colors (r, red; g, green; b, blue) with integers 0 to 7.

9.47 LLV [Linear Scale]

■ Function

Selects a linear scale as a level scale and sets a scale value.

Header	Program	Query	Response
LLV	LLV l	LLV ?	l

■ Value of l

○ When A, B, or A & B is selected by TSL (Trace Select)
l is a scale value. Specify a unit and a value, 1 to 4, or specify a value with its unit fixed at mW.
Unit ..... W, MW, UW, NW, PW
Data range ..... 1PW ≤ 1 ≤ 1W

○ When A-B or B-A is selected by TSL (Trace Select) and a normalize display mode is set
l is a scale value. Specify a unit and a value, 1 to 3. The unit may be omitted.
Unit ..... PCT
Data range ..... 1PCT ≤ 1 ≤ 200 PCT

■ Initial setting value

l is a backed up value.

■ Default

l = 100 (mW)

9.48 LOFS [Level Offset]

■ Function

Sets a level offset value.
Inputting a level offset value will move the displayed spectrum the distance equivalent to the level offset.

Header	Program	Query	Response
LOFS	LOFS n	LOFS ?	n

■ Value of n

n indicates a level offset value.
The unit is dB. Input a value down to the second decimal place.
Data range: –30.00 dB ≤ n ≤ + 30.00 dB

9.49 LOG [Log Scale]

■ Function

Selects a log scale as a level sale and sets a scale value (dB/div).

Header	Program	Query	Response
LOG	LOG l	LOG ?	l

■ Value of l

The unit is dB/div. Input a value down to the first decimal place.
Data range:$0.1 \leq l \leq 10.0$

■ Initial setting value

l is a backed up value.

■ Default

l = 10 (dB/div)

9.50 LVS [Level Scale]

■ Function

Checks whether a log or linear scale is set as a level scale.

Header	Program	Query	Response
LVS	None	LVS ?	s

■ Value of s

s indicates a level sale type.
Log scale ..... LOG
Linear scale ..... LIN

9.51 MDM [Modulation Mode]

■ Function

Sets a modulation measurement mode.

Header	Program	Query	Response
MDM	MDM s	MDM ?	s

■ Value of s

s = NORMAL: Normal measurement
= TRIGGER: EXT trigger measurement

■ Initial setting value

s is a backed up value.

■ Default

s = NORMAL

9.52 MKA [Wavelength Marker A]

■ Function

Sets wavelength A with a wavelength.

Header	Program	Query	Response
MKA	MKA λ	MKA ?	λ

■ Value of λ

The unit is nm or THz. Input a wavelength down to the fourth decimal place and input a frequency down to the fifth decimal place.
Data range: Start wavelength ≤ λ ≤ Stop wavelength

■ Initial setting value

λ = 600.000 (nm)

9.53 MKB [Wavelength Marker B]

■ Function

Sets wavelength A with a wavelength.

Header	Program	Query	Response
MKB	MKB λ	MKB ?	λ

■ Value of λ

The unit is nm or THz. Input a wavelength down to the fourth decimal place and input a frequency down to the fifth decimal place.
Data range: Start wavelength ≤ λ ≤ Stop wavelength

■ Initial setting value

λ = 1800.000 (nm)

9.54 MKC [Level Marker C]

■ Function

Sets level marker C with a level.

Header	Program	Query	Response
MKC	MKC l	MKC ?	l

■ Value of l

λ is a level value including the unit of the trace marker on the set scale.
When the log scale is selected, λ is a value represented down to the third decimal place. When the linear scale is selected, it is represented by a maximum of seven digits.
Data range:

–190DBM ≤ l ≤ +50 DBM ..... Log scale, normal/max. hold/overlap, trace A/B/AB
–160DB ≤ l ≤ +160 DB ..... Log scale, normal/max. hold/overlap, trace A-B/B-A
Log scale, normalize
–200DB ≤ l ≤+120 DB ..... Linear scale, normal/max. hold/overlap, trace A/B/AB
0PW ≤ l ≤ 1.2 W ..... Linear scale, normal/max. hold/overlap, trace A/B/AB
0PCT ≤ l ≤ 240 PCT ..... Linear scale, normal/max. hold/overlap, trace A-B/B-A
Linear scale, normalize

■ Suffixes

DBM: Log scale, normal/max. hold/overlap, trace A/B/AB
DB: Log scale, normal/max. hold/overlap, trace A-B/B-A
Log scale, normalize
W: Linear scale, normal/max. hold/overlap, trace A/B/AB
MW: Linear scale, normal/max. hold/overlap, trace A/B/AB
UW: Linear scale, normal/max. hold/overlap, trace A/B/AB
NW: Linear scale, normal/max. hold/overlap, trace A/B/AB
PW: Linear scale, normal/max. hold/overlap, trace A/B/AB
PCT: Linear scale, normal/max. hold/overlap, trace A-B/B-A
Linear scale, normalize

9.55 MKD [Level Marker D]

■ Function

Sets level marker D with a level.

Header	Program	Query	Response
MKD	MKD l	MKD ?	l

■ Value of l

λ is a level value including the unit of the trace marker on the set scale.
When the log scale is selected, λ is a value represented down to the third decimal place. When the linear scale is selected, it is represented by a maximum of seven digits.
Data range:

–190 DBM ≤ λ ≤ +50 DBM .... Log scale, normal/max. hold/overlap, trace A/B/AB

–160 DB ≤ λ ≤ +160 DB ..... Log scale, normal/max. hold/overlap, trace A-B/B-A
Log scale, normalize

–200 DB ≤ λ ≤ +120 DB ..... Linear scale, normal/max. hold/overlap, trace A/B/AB

0 PW ≤ λ ≤ 1.2 W ..... Linear scale, normal/max. hold/overlap, trace A/B/AB

0 PCT ≤ λ ≤ 240 PCT ..... Linear scale, normal/max. hold/overlap, trace A-B/B-A
Linear scale, normalize

■ Suffixes

Same as level marker C.

9.56 MKV [Marker Value Wavelength/Frequency Select]

■ Function

Converts the trace marker, the delta marker and wavelength values obtained from a part of the analysis into frequencies and then displays them.

Header	Program	Query	Response
MKV	MKV s	MKV ?	s

■ Value of s

s = WL: Wavelength
= FREQ: Frequency

■ Initial setting value

s is a backed up value.

■ Default

s = WL

9.57 MOD [Measure Mode]

■ Function

Outputs a number indicating a measurement mode.

Header	Program	Query	Response
MOD	None	MOD ?	n

■ Value of n

n is output with a number, 0 to 3, indicating the following:
0: A spectrum is not being measured.
1: A spectrum is being measured (single sweep).
2: A spectrum is being measured (repeat sweep).
3: Power monitor

9.58 MPT [Sampling (Measureing) Points]

■ Function

Sets a number of sampling points.

Header	Program	Query	Response
MPT	MPT n	MPT ?	n

■ Value of n

The unit is number of points. Input one of the following values:
51, 101, 251, 501, 1001, 2001, 5001

■ Initial setting value

n is a backed up value.

■ Default

n = 501 (points)

9.59 MSL [Memory Select]

■ Function

Selects memory A or B for saving measurement data.

Header	Program	Query	Response
MSL	MSL s	MSL ?	s

■ Value of s

s = A: Select memory A.
= B: Select memory B.

■ Initial setting value

s = A (memory A)

9.60 OPT [Light Output]

■ Function

Turns on/off the light output from the internal light source (optional white light source or reference light source).

Header	Program	Query	Response
OPT	OPT s	OPT ?	s

■ Value of n

s = ON: Turn on the light output.
= OFF: Turn off the light output.

9.61

(Not used.)

9.62 PKC [Peak → Center]

■ Function

Sets the spectrum peak wavelength as a center wavelength.

Header	Program	Query	Response
PKC	PKC	None	None

9.63 PKL [Peak → Level]

■ Function

Sets the spectrum peak level as a reference level.

Header	Program	Query	Response
PKL	PKL	None	None

9.64 PKS [Peak Search]

■ Function

Detects the spectrum peak and moves the trace marker there.
When the processing is complete, bit 0 (measurement end bit) of the extended event status register (ESR2) is set to 1.
To move the peak, specify PEAK, NEXT, LAST, LEFT, or RIGHT.
To read the peak wavelength and level, use a TMK? command.

Header	Program	Query	Response
PKS	PKS s	PKS ?	m

■ Value of s

s = PEAK: Detects the main peak whose level is highest, and moves the trace marker there.
= NEXT: Detects the peak whose level is the second highest compared with the current peak, and moves the trace marker there.
= LAST: Detects the peak whose level is the second lowest compared with the current peak, and moves the trace marker there.
= LEFT: Detects the peak whose wavelength is the second shortest compared with the current peak, and moves the trace marker there.
= RIGHT: Detects the peak whose wavelength is the second longest compared with the current peak, and moves the trace marker there.

■ Value of m

m = PEAK, NEXT, LAST, LEFT, RIGHT
= ERR (state other than peak search)

■ Initial setting value

s = PEAK

9.65 PWR [Power Monitor]

■ Function

Sets parameters for the power monitor, and goes to the power monitor measurement mode.
Each time it becomes possible to read a power monitor value, bit 3 (execution complete bit 1) of the extended event status register (ESR2) is set to 1.
To go to the spectrum measurement mode, execute an SPC command.

Header	Program	Query	Response
PWR	PWR λ	PWR ?	λ

■ Value of λ

λ is a wavelength range. The unit is nm. Input one of the following values:
632.8, 850.0, 1300.0, 1550.0
If the digits below the decimal point are 0s, they may be omitted.

■ Initial setting value

λ is a backed up value.

■ Default

λ = 1550.0

9.66 RCAL [Resolution Calibration]

■ Function

Calibrates the value of the effective resolution.

Header	Program	Query	Response
RCAL	RCAL n	RCAL ?	m

■ Value of n

n = 0: Initializes the corrected value of the resolution.
= 1: Executes calibration of the resolution to calculate the corrected value of the resolution.

■ Value of m

m = 0: The corrected value of the resolution is the initial value (= 1).
= 1: Calibration of the resolution ended normally.
= 2: Calibration of the resolution is being executed.
= 3: Calibration of the resolution ended abnormally.

9.67 PWRR [Power Monitor Result]

■ Function

Reads the result of measurement made using a power monitor.

Header	Program	Query	Response
PWRR	None	PWRR ?	p

■ Response

• Value of p

p is a measured power value.
The unit is dBm. A value is output down to the second decimal place.
Data is tail-zero-suppressed.

9.68 RCL [FD File Recall]

■ Function

Recalls the specified file on the FD.
When file recall is complete, bit 2 (transfer end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
RCL	RCL n	None	None

■ Value of n

Input a DOS-recognizable file name comprising eight or fewer characters in the following format:
xxxxxxxx.dat
The file extension ".dat" may be omitted.

9.69 RES [Resolution]

■ Function

Sets measurement resolution.

Header	Program	Query	Response
RES	RES n	RES ?	n

■ Value of n

n indicates measurement resolution. The unit is always nm. Input one of the following values:
1.0, 0.5, 0.2, 0.1, 0.07, 0.05

■ Initial setting value

n is a backed up value.

■ Default

n = 1.0 (nm)

9.70 RLV [Reference Level Scale]

■ Function

Sets a reference level when setting a log scale.

Header	Program	Query	Response
RLV	RLV l	RLV ?	l

■ Value of l

Input a value down to the first decimal place.

○ When A, B, or A & B is selected by TSL (Trace Select)
The unit is dBm.
Data range: $-90.0 \leq l \leq +30.0$

○ When A-B or B-A is selected by TSL (Trace Select) and a normalize display mode is set
The unit is dBm.
Data range: $-100.0 \leq l \leq 100.0$

■ Initial setting value

l is a backed up value.

■ Default

l = +20 (dBm)

9.71 SAV [FD File Save]

■ Function

Saves measurement data in the specified file on the FD.
When file saving is complete, bit 2 (transfer end bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
SAV	SAV n	None	None

■ • Value of n

Input a DOS-recognizable file name comprising eight or fewer characters in the following format:
xxxxxxxx.dat
The file extension ".dat" may be omitted.

9.72 SMT [Smooth]

■ Function

Sets a number of smoothing points.

Header	Program	Query	Response
SMT	SMT n	SMT ?	n

■ Value of n

The unit is number of points. Input one of the following values or OFF:
3, 5, 7, 9, 11, or OFF

■ Initial setting value

n is a backed up value.

■ Default

n = OFF

9.73 SPC [Spectrum]

■ Function

Goes to the spectrum measurement mode.

Header	Program	Query	Response
SPC	SPC	None	None

9.74 SPN [Span Wavelength]

■ Function

Sets a span wavelength.

Header	Program	Query	Response
SPN	SPN λ	SPN ?	λ

■ Value of l

The unit is always nm. Input a value down to the first decimal place.
Data range: λ = 0,0.2 ≤ λ ≤ 1200.0

■ Initial setting value

λ is a backed up value.

■ Default

λ = 500 (nm)

9.75 SRT [Repeat Sweep]

■ Function

Starts repeat sweep.

Header	Program	Query	Response
SRT	SRT	None	None

9.76 SSI [Single Sweep]

■ Function

Starts single sweep.
When single sweep is complete, bit 1 (sweep stop bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
SSI	SSI	None	None

9.77 SST [Sweep Stop]

■ Function

Stops sweep.

Header	Program	Query	Response
SST	SST	None	None

9.78 STA [Start Wavelength]

■ Function

Sets a start wavelength.

Header	Program	Query	Response
STA	STA λ	STA ?	λ

■ Value of λ

The unit is always nm. Input a value down to the first decimal place.
Data range: $600.0 \leq \lambda \leq 1750.0$

■ Initial setting value

λ is a backed up value.

■ Default

λ = 1100 (nm)

9.79 STO [Stop Wavelength]

■ Function

Sets a stop wavelength.

Header	Program	Query	Response
STO	STO λ	STO ?	λ

■ Value of λ

The unit is always nm. Input a value down to the first decimal place.
Data range:$600.0 \leq \lambda \leq 1800.0$

■ Initial setting value

λ is a backed up value.

■ Default

λ = 1600 (nm)

9.80 TDL [EXT-trigger Delay Time]

■ Function

Sets a trigger delay used for EXT trigger measurement.

Header	Program	Query	Response
TDL	TDL n	TDL ?	n

■ Value of n

The unit is always m s. Input a value with an integer comprising 1 to 7 digits.
Data range: $0 \leq n \leq 5000000$

■ Initial setting value

n is a backed up value.

■ Default

n = 0

9.81 TDSP [Time & Date Display On/Off]

■ Function

Turns on/off the timer display.

Header	Program	Query	Response
TDSP	TDSP s	TDSP ?	s

■ • Value of s

S indicates whether the timer display is on or off.
s = ON: Turn on the timer display.
= OFF: Turn off the timer display.

9.82 TER [Title Erase]

■ Function

Clears all title characters.

Header	Program	Query	Response
TER	TER	None	None

9.83 TIME [Time Set]

■ Function

Sets hours and minutes.

Header	Program	Query	Response
TIME	TIME hh,mi	TIME ?	hh,mi

■ • Value of hh

Input hours with two digits (00 to 23).

• Value of mi

Input minutes with two digits (00 to 59).

9.84 TLSA [Measure Mode (Adjust to TLS)]

■ Function

Calibrates wavelength of optical spectrum for TLS Tracking measurement
Setting/Reading are only possible in the TLS Tracking measurement mode.

Header	Program	Query	Response
TLSA	TLSA n	TLSA ?	m

■ Values of n and m

n indicates the following conditions.

n = 0: TLS wavelength calibration forcibly quit
= 1: Executing TLS wavelength calibration and finding TLS wavelength calibration data

m indicates the following conditions.

m = 0: TLS wavelength calibration ended normally
= 1: Calibrating TLS wavelength
= 2: TLS wavelength calibration suspended due to abnormality
= 3: Not calibrated

9.85 TLST [Measure Mode (TLS Tracking)]

■ Function

Sets/Reads TLS Tracking measurement mode

Header	Program	Query	Response
TLST	TLST s	TLST ?	s

■ Value of s

s indicates the following conditions

s = ON: TLS Tracking measurement condition or TLS Tracking measurement switched to ON

= OFF: Normal measurement or Power Monitor condition, or TLS Tracking measurement switched to OFF

9.86 TMC [TMKR → Center]

■ Function

Sets a trace marker wavelength as the center wavelength.

Header	Program	Query	Response
TMC	TMC	None	None

9.87 TMK [Trace Marker]

■ Function

Sets the trace marker with a wavelength.

Header	Program	Query	Response
TMK	TMK λ	TMK ?	λ , l

■ • Value of λ

λ is a wavelength value.

The unit is nm or THz. Input a wavelength down to the fourth decimal place and input a frequency down to the fifth decimal place.

Data range: Start wavelength ≤ λ ≤ Stop wavelength

• Value of l

l indicates a level value including the unit of the trace marker on the set scale.

If it is impossible to calculate l on the linear scale, –1 is output.

■ Suffixes

DBM: Log scale, normal/max. hold, trace A/B
DB: Log scale, normal/max. hold, trace A-B/B-A
Log scale, normalize
w: Linear scale, normal/max. hold, trace A/B
MW Linear scale, normal/max. hold, trace A/B
UW: Linear scale, normal/max. hold, trace A/B
NW: Linear scale, normal/max. hold, trace A/B
PW: Linear scale, normal/max. hold, trace A/B
PCT: Linear scale, normal/max. hold, trace A-B/B-A
Linear scale, normalize

9.88 TRM [Terminater]

■ Function

Switches between terminators when the MS9710C is used as a talker (data is sent from the MS9710C).

Header	Program	Query	Response
TRM	TRM n	TRM ?	n

■ Value of n

n = 0: LF ^ EOI
= 1: CR · LF ^ EOI

■ Initial setting value

n = 1

9.89 TSL [Trace Select]

■ Function

Selects a waveform trace mode among A, B, A & B, A-B, and B-A.

Header	Program	Query	Response
TSL	TSL s	TSL ?	s

■ Value of s

s = A: Select trace mode A.
= B: Select trace mode B.
= AB: Select trace mode A & B.
= A-B: Select trace mode A-B.
= B-A: Select trace mode B-A.

■ Initial setting value

s = A (trace A)

9.90 TTL [Title]

■ Function

Writes/reads a title.

Header	Program	Query	Response
TTL	TTL s	TTL ?	s

■ Value of s

s is a title character string enclosed in single quotation marks (‘ ‘).
A title character string can be comprised of 30 or fewer characters.

9.91 VBW [Video Band Width]

■ Function

Sets a video bandwidth.

Header	Program	Query	Response
VBW	VBW s	VBW ?	s

■ Value of s

s is a VBW value. Input one of the following integers including the unit (always Hz):
1 MHz, 100 kHz, 10 kHz, 1 kHz, 100 Hz, or 10 Hz

■ Initial setting value

s is a backed up value.

■ Default

s = 1 kHz

9.92 WCAL [Wavelength Calibration]

■ Function

Carries out wavelength calibration using an external or reference light source to create wavelength calibration data.
When wavelength calibration is complete, bit 4 (execution complete bit) of the extended event status register (ESR2) is set to 1.

Header	Program	Query	Response
WCAL	WCAL n	WCAL ?	m

■ • Value of n

n = 0: Use the default value as the wavelength calibration data.
= 1: Carries out wavelength calibration using an external light source to create wavelength calibration data.
= 2: Carries out wavelength calibration using a reference light source to create wavelength calibration data.
= 3: Terminate wavelength calibration forcibly.

• Value of m

m = 0: Waveform calibration has been terminated normally.
= 1: Waveform calibration is being carried out.
= 2: Waveform calibration has been interrupted due to an insufficient light level.
= 3: Waveform calibration has been interrupted due to any other fault.

9.93 WDP [Vacuum/Air Set]

■ Function

Sets the wavelength display mode to “in Vacuum” or “in Air.”

Header	Program	Query	Response
WDP	WDP s	WDP ?	s

■ Value of s

s = VACUUM: in Vacuum
= AIR: in Air

■ Initial setting value

s is a backed up value.

■ Default

s = AIR

9.94 WOFS [Wavelength Offset]

■ Function

Sets a wavelength offset value.

Header	Program	Query	Response
WOFS	WOFS n	WOFS ?	n

■ Value of n

n is a waveform offset value.
The unit is always nm. Input a value down to the second decimal place.
Data range: $-1.00 \text{ nm} \leq n \leq 1.00 \text{ nm}$

9.95 WSS [Setting Wavelength Start and Stop]

■ Function

Sets the start wavelength and the stop wavelength at the same time.

Header	Program	Query	Response
WSS	WSS λ1, λ2	WSS ?	λ1, λ2

■ • Value of l1

The Value to be set as the start wavelength, which is in nm units and may be set to one decimal place.

• Value of l2

The Value to be set as the stop wavelength, which is in nm units and may be set to one decimal place.

9.96 ZMK [Zone Marker]

ZMK WL [Zone Marker Wavelength]

■ Function

Sets zone markers with its center wavelength and span.

Header	Program	Query	Response
ZMK	ZMK WL, λ c, λs	ZMK ? WL	WL, λc, λs

■ • Value of λ c

λ c is a center wavelength of zone markers.
The unit is always nm. Input a value down to the third decimal place.
Data range: Start wavelength ≤ λ c ≤ Stop wavelength

• Value of λ s

λ s is a span of zone markers.
The unit is always nm. Input a value down to the third decimal place.
A zone marker should not exceed the range between the start wavelength and the stop wavelength.

■ Initial setting value

λ c = Center wavelength
λ s = Span wavelength

ZMK SPN [Zone Marker → Span]

■ Function

Sets the center wavelength and span of zone markers.

Header	Program	Query	Response
ZMK	ZMK SPN	None	None

ZMK ZOOM [Zone Marker → Zoom In/Out]

■ Function

Zooms in or out zone markers.

Header	Program	Query	Response
ZMK	ZMK ZOOM, s	ZMK ? ZOOM	ZOOM, s

■ Value of s

s = IN: Zoom in zone markers.
= OUT: Zoom out zone markers.

■ Initial setting value

s = OUT

ZMK ERS [Zone Marker Erase]

■ Function

Erases zone markers.

Header	Program	Query	Response
ZMK	ZMK ERS	None	None

Section 10 Program Examples

10.1 Precautions on Creating a Program

Precautions on creating a remote control program are as follows.

No.	Precaution	Description
1	Be sure to initialize devices.	Devices may be in various states after they have been operated by their own operator panels and other programs. In many cases, these states may not be proper at the start of use. Therefore, these devices must be initialized so that they can be used under certain conditions. (a) Initialization of interface function (IFC@) (b) Initialization of device message exchange function (DCL@) (c) Initialization of device-dependent functions (*RST) When the RS-232C interface is used, only 3) is required.
2	Set devices in the RWLS (Remote With Lockout State).	If devices are set in a simple remote state, they will enter the local state when the LOCAL key is pressed. If a panel key is pressed with the device in the local state, the device cannot carry out automatic measurement properly and therefore the measurement data becomes inaccurate. Execute an LLO@ statement to lock out devices to prevent devices from returning to the local state.
3	Immediately after sending a query, do not send any device-related command other than READ@	If a command other than a READ@ statement is sent to the controller before reading the query result, the output message is cleared at reception of MLA and therefore the response message disappears. Be sure to write a READ@ statement directly after the query.
4	Avoid exception handling in the protocol.	Expected exceptions must be handled in the exception handling section in the program so that execution does not stop due to errors.
5	Check interface functions (subset) of individual devices (GPIB).	If a created program is executed for a device that does not have a subset, processing will not proceed. Be sure to check subsets of devices. Also check the devices conform to IEEE 488.2.
6	Prevent buffer overflow (RS-232C).	The RS-232C interface of the MS9710C has a 256-byte data area as an internal receive buffer. However, overflow may occur depending on the processing type. To prevent errors from occurring due to overflow, do not send a large volume of data (control commands) at a time when performing remote control using the RS-232C interface. After sending a sequence of commands, you can send an *OPC? command, wait for a response to be received, then send the next command for synchronization.

10.2 Program Examples

(1) Obtaining a peak wavelength and peak level

Shown below is an example of a program that obtains a peak wavelength and peak level through spectrum measurement.

```
 90  WRITE  @108 : “SSI”
100  DO
110       WRITE  @108 : “ESR2?”
120       READ  @108 : ESR2
130       EXIT  IF  BIT (1, ESR2) = 1!                    sweep  end
140  LOOP
150  WRITE  @108 : “PKS”
160  DO
170       WRITE  @108 : “ESR2?”
180       READ  @108 : ESR2
190       EXIT  IF  BIT (0, ESR2) =1!                     Peak  search  end
200  LOOP
210  WRITE  @108 : “TMK?”
220  READ  @108 : WAVE$, LEVEL$
230  PRINT  “Peak  wave  length     ” : WAVE$ : “nm”
240  PRINT  “Peak  level                  ” : LEVEL$
```

Line 90:	Start sweep.
Lines 100 to 140:	Wait for the end of sweep.
Line 150:	Search for a peak.
Lines 160 to 120:	Wait for the end of peak search.
Lines 210 to 220:	Input a peak value.
Lines 230 to 240:	Print the result.

In this example, the end of sweep is waited to obtain a peak from the recognized waveform.
The end of sweep and the ends of peak search are checked by monitoring the extended event register. However, when a GPIB interface bus is used, an SRQ interrupt can be used. When it is used, the bits corresponding to the service request enable register and extended event status register must be set in advance using the ∗SRE and ESE2 commands, respectively.

(2) Obtaining a side mode ratio

Shown below is an example of a program that obtains a spectrum waveform side mode ratio. In this example, the second peak existing on the left of the peak (shorter wavelength side) is subjected to the analysis.

```
 90  WRITE  @108 : “ANA  SMSR, LEFT”
100  DO
110       WRITE  @ 108 : “ESR2 ?”
120       READ  @ 108 ; ESR2
130       EXIT  IF  BIT (0, ESR2) =1 !                analysis  end
140  LOOP
150  WRITE  @108 : “ANAR?”
160  READ  @108 : WAVE, LEVEL
170  IF  WAVE < 0  THEN
180       PRINT  “***  can’t  analysis  ***”
190  ELSE
200       PRINT  “wave  diff  =” : WAVE : “nm”

210       PRINT  “level  diff  =” : LEVEL : “dB”
220  END  IF
```

Line 90:	Carry out side mode analysis.
Lines 100 to 140:	Wait for the end of analysis.
Lines 150 to 160:	Input the analysis result.
Lines 170 to 210:	Print the result.

After execution of analysis, the end of the analysis is waited and the result is input. The result is printed to allow the operator to check whether the second peak exists and whether a normal analysis value is obtained.

(3) Obtaining a power value

Shown below is an example of a program that carries out power monitor measurement to obtain the power value of a specific wavelength.

```
 90  WRITE  @108 : “PWR  632.8”
100  DO
110       WRITE  @108 : “ESR2?”
120       READ  @108 : ESR2
130       EXIT  IF  BIT(3, ESR2) = 1 !              power  monitor  READY
140  LOOP
150  WRITE  @108 : “PWRR?”
160  READ  @108 : POWER
170  PRINT  “power” ; POWER ; “dBm”
```

Line 90:	Start power monitor measurement.
Lines 100 to 140:	Wait for power monitor measurement to be ready.
Lines 150 to 160:	Input a power value.
Line 170:	Print the measurement result.

When carrying out power monitor measurement, it is necessary to enter the power monitor mode and wait for a normal measurement result to be obtained. Therefore, lines 100 to 140 are used to check mode switching.

(4) Reading memory data

Shown below is an example of a program that reads the main data of the measured waveform. In this example, the main data is read from memory A. The program for reading the main data from memory B is similar to this program. In this example, binary data is converted to numeric data with DMA? contrasted with DBA?.

```
 70   DIM  DT  (500), D 0 (i500), D 1(1001), D (500)
 80   DIM  L $ *10
 90  !
100   WRITE  @108 : "DMA ?"
110   FOR  I = 0  TO  500
120    READ  @108 : FDT( I )
130   NEXT  I
140  !
150   WRITE  @108 : "LVS ?"
160   READ  @108 : L $
170   PRINT  "LEVEL  SCALE  =" : L $
180   IF  L $ = "LOG"  THEN
190      WRITE  @108 : "DBA?"
200      MAT  READ  @108  USING  "WH" : D 0
210    !
220      FOR  I = 0  TO  500
230           LET  D ( I ) = D 0 ( i ) /100
240      NEXT  I
250    !
260   ELSE
270      WRITE  @108 : "DBA?"
280      MAT  READ  @108  USING  "WH" : D 1
290    !
300      FOR  I = 0  TO  500
310           LET  D ( I ) = D 1 ( 2*I +1 )/10000*10^D 1 ( 2*I )
320      NEXT  I
330    !
340   END  IF
```

Lines 70 to 80:	Define arrays and character strings.
Lines 100 to 130:	Read data (ASCN data) with a DMA? command.
Lines 150 to 180:	Check the current level scale.
Lines 190 to 200:	Read binary data when a log scale is selected.
Lines 220 to 240:	Convert binary data to numeric data when a log scale is selected.
Lines 270 to 280:	Read binary data when a linear scale is selected.
Lines 300 to 320:	Convert binary data to numeric data when a linear scale is selected.

The array data DT ( ) represented in ASCN numeric format is the same as the array data D ( ) which is the numeric data converted from binary data.

When the log scale is selected, 1-point measurement data is represented by two bytes. When the linear scale is selected, 1-point measurement data is represented by four bytes. For the binary data format, see Appendix B. The time needed to read the whole data on 501 points varies with the data value as follows:

	DMA?	DQA?	DBA?		
			Read	Conversion	Read + Conversion
Log data	7.5 to 8.0	1.7	0.5	0.7	1.2
Linear data	8.0 to 8.5	2.0	0.9	2.3	3.2

Unit: Second

(5) Examining details on errors

When there is information (error occurrence, processing end, etc.) that a device (MS9710C) wants to report to a controller (personal computer) to perform GPIB control, the device reports it to the controller using a status byte. There are two methods to check the status byte: a method by which an SRQ interrupt is issued to the controller to force the controller to check the status byte and a method by which the controller is expected to check the status byte voluntarily.
Let's take a look at the former method.

Shown below is an example of a program that consists of a main processing block (lines 70 to 450) and an interrupt block (lines 500 to 820). Assume that the interrupt processing block is a task that is activated when an SRQ interrupt event occurs.

```
 70   PROCESS  EVENT  SRQ"@01, 02"
 80   START  GPIB
 90   CONNECT  EVENT  SRQ
100   ENABLE  HANDLER  READERR
110  !
120   WRITE  @108 : "*CLS"
130   WRITE  @108 : "*SRE  40" !   Set ESB, ERROR bit to STB
140   WRITE  @108 : "*ESE  60" !  Set QERR, DDE, EXE, CERR bit to SESER
150   WRITE  @108 : "ESE3  1"
160  !
170   ENABLE  TIMEOUT  3
180  !
190  !
200  !
400   HANDLER  READERR
410         PRINT "***  Query  read  error  ***"
420         CONTINUE
430   END  HANDLER
440  !
450   END
500  ! ======================================
510   PARACT  GPIB  URGENCY  80
520         GPINT :   WAIT  EVENT  SRQ
530           WRITE  @108 : "*STB?"
540           READ  @108 : STB
550           IF  BIT (3, STB) = 1  THEN
560             PRINT  "***  RES  Uncal  ***"
570           END  IF
580           IF  BIT (5, STB) = 1  THEN
590                 WRITE  @108 : "*ESR?" !
600                 READ  @108 : ESR
610                 IF  BIT (2, ESR) = 1  THEN
620                 PRINT  "***  Query  error  ***"
```

```
630             END IF
640             IF BIT (3, ESR) = 1 THEN
650                PRINT "*** DDE error ***"
660                WRITE @108 : "ERR?"
670                READ @108 : ERR $
680                PRINT "Error code = " : ERR $
690             END IF
700             IF BIT (4, ESR) = 1 THEN
710                PRINT "*** EXE error ***"
720                WRITE @108 : "ERR?"
730                READ @108 : ERR $
740                PRINT "Error code = " : ERR $
750             END IF
760                IF BIT (5, ESR) = 1 THEN
770                PRINT "*** Command error ***"
780             END IF
790          END IF
800          WRITE @108 : "*CLS"
810          GO TO GPINT
820  END PARACT
```

Lines 70 to 90: Define SRQ and start the interrupt processing block "GPIB."

Line 100: Declare the error handler "READERR."

Lines 120 to 150: Clear and set enable registers. (For more details, see Section 7.)
Enable event status (ESB) and ERROR bits of the status byte register. Enable command error, EXE error, DDE error, and query error bits of the event status byte.
Enable the RES_Uncal bit of the extended event status byte.
As the result, an SRQ interrupt occurs when a command error, EXE error, DDE error, query error, or RES_Uncal error occurs.

Line 170: Set the READ command time-out time to 3 seconds.
When a query error occurs, the READ command following the query command must be executed within the specified time. The reason is that the READ command is not completed for ever when a query command error occurs. That is, the READ instruction must be terminated forcibly to prevent deadlock.
In this program, control is transferred to the error handler "READ ERR" on line 400 when time-out occurs.

Lines 180 to 200: Execute desired commands. In this example, these lines are used to write a comment.

Lines 400 to 430: An error handler processing block. This block is executed when READ command time-out occurs. An error message is printed and processing is continued.

Lines 500 to 820: An SRQ processing block.
When an interrupt occurs, the processing specified on line 530 and later is performed. The status byte and event status byte are read to check the error, and the check result is printed.
When a DDE or EXE error occurs, the error code is checked and printed.

Section 11 LabVIEW Measuring Instrument

This section explains the measuring instrument drivers (MX971003G/S) used to control the MS9710C remotely under LabVIEW.

LabVIEW measuring instrument drivers are modules in which command send and receive functions are incorporated, allowing measuring instruments to be controlled under the U.S. National Instruments Graphic Programming System "LabVIEW." Using these drivers, you can control the MS9710C remotely without remembering control commands.
To use this measuring instrument, a controller in which National Instruments LabVIEW software (Windows version) is installed is required.

The measuring instrument drivers have been created using LabVIEW Ver. 4.0/J (Windows version).
For how to use LabVIEW, see the LabBIEW User's Guide.

About LabVIEW

LabVIEW is a graphical program language suitable for controlling measuring instruments and saving and analyzing data.

LabVIEW allows you to create a program as if you drew a circuit diagram, so you can easily get used to use it compared with text-based program languages. The execution speed is almost the same as the C language.

LabVIEW supports various libraries related to measuring instrument control and data saving, analysis, and display. Using LabVIEW and measuring instrument drivers, you can create a graphical user interface (GUI) program with ease.

11.1 Installation

The floppy disk (MX971003G/S) that come standard with the MS9710A store the following files.

MS9710AG.EXE	(GPIB driver)
MS9710AS.EXE	(RS-232C driver)

These files are self-melting-type compressed files. Copy the file storing the drivers you want to use to an appropriate directory, then melt the file.

Installation example

(1) On X:¥LABVIEW (X is the drive on which LabVIEW is installed), create a directory "MS9710A.LIB."

(2) Copy the file (MS9710A.EXE or MS9710AS.EXE) storing the drivers you want to use to this directory.

(3) Select "Specify File Name and Execute" from the icon menu of the program manager, and enter

X:¥LABVIEW¥MS9710A.LIB¥MS9710AG.EXE (GPIB driver)
or
X:¥LABVIEW¥MS9710A.LIB¥MS9710AS.EXE (RS-232C driver)

Then, press the OK button.
The following VI library files storing measuring instrument drivers will be created in the directory MS9710A.LIB:

MS9710AG.LLB	(GPIB)
MS9710AS.LLB	(RS-232C)

(4) After making sure the above library file has been created, delete MS9710AG.EXE (or MS9710AS.EXE).

11.2 Program Examples

This section gives examples of programs created using the measuring instrument drivers.
For example, let's create a program for obtaining a peak wavelength and peak level under GPIB control.
For how to connect a computer to the MS9710C, see Section 2, "How to Connect." For this program, set the GPIB address to 8.

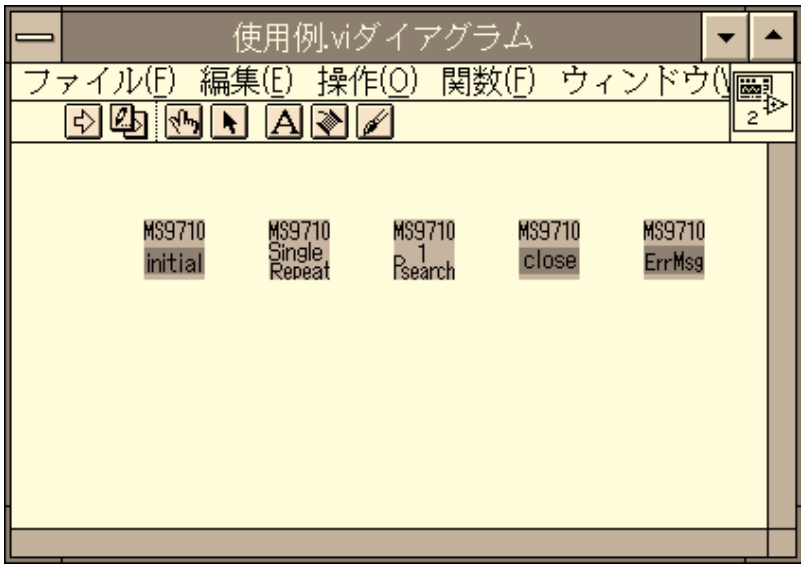
Here, we will five drivers listed below.
The GPIB driver is in the VI library "MS9710AG.LLB."
(The RS-232C driver is in the VI library "MS9710AS.LLB." To perform RS-232C control, create a similar program using this driver.)

MS9710A Initialize
MS9710A Close
MS9710A Sweep Start
MS9710A Peak/Dip;Peak/DipSearch(1)
MS9710A Error Message

(1) Arranging drivers in the block diagram

Arrange MS9710A Initialize and MS9710A Close VIs.
MS9710A Initialize and MS9710A Close VIs are used to start or terminate communication with the MS9710C.

Arrange driver VIs between them.
MS9710A Sweep Start — Start single sweep.
MS9710A Peak/Dip;Peak/Dip Search(1) — Carry out peak search to read the trace marker values.
MS9710A Error Message — If an error occurred in a sequence of operations, displays a message.

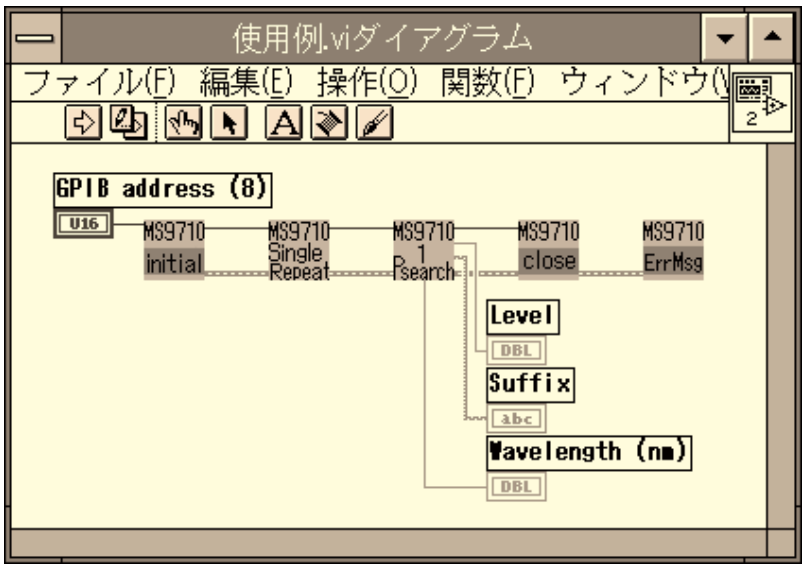


(2) Arranging controllers and displays

Double-clicking on the MS9710A Initialize.vi on the diagram window will open the measuring instrument driver window. Copy the controllers subject to GPIB address input from this window onto the front panel window. Double-click on the MS9710A Peak/Dip;Peak/Dip Search.vi icon and copy the displays for displaying a marker wavelength, level, and unit onto the front panel window.

(3) Connecting displays, controllers, and icons

Connect driver terminals with wires as shown below.



(4) Executing the program

After completion of sweep, carry out peak search to output the trace marker values to the display on the front panel.

11.3 List of Measuring Instrument Drivers

The measuring instrument driver file name is assigned as follows:
MS9710A □ (card name);(function key name).vi
or
MS9710A (name corresponding to panel key or function).vi
(For he GPIB driver □, is left blank. For the RS-232C driver, is entered with S.)
Icons resemble the keys on the main unit.

You can select drivers according to the main unit key operation. In most case, you can select the drivers to be used by imagining the main unit key operations.

Drivers for Front Card Functions

Wavelength Setting

Card name	;Function key name	Function
Wavelength	;Center/Start/Stop	Set a center wavelength, start wavelength, stop wavelength.
Wavelength	;Set Span	Set a sweep span.
Wavelength	;Air/Vacuum	Switch between wavelength display modes (in Air/in Vacuum).
Wavelength	;Mkr Value WL/Freq	Switch between marker wavelength/frequency display modes.

Level Setting

Card name	;Function key name	Function
LevelScale	;Log/Linear	Set a log/linear scale.
LevelScale	;REF Level	Set a reference level.
LevelScale	;Opt.Att Off/On	Turn on/off the optical attenuator.

Resolution and Averaging Setting

Card name	;Function key name	Function
Res/BW/Avg	;Res/BW/SmplPt	Set resolution, light video bandwidth, or number of sampling points.
Res/BW/Avg	;Actual Resolution	Switches between actual resolution display modes.
Res/BW/Avg	;Average	Set Point/Sweep Average or Smooth.

Peak and Dip Detection

Card name	;Function key name	Function
Peak/Dip	;Peak/DipSearch(1)	Detect a peak/dip whose level is highest.
Peak/Dip	;Peak/DipSearch(2)	Detect a peak/dip whose level is the second lowest/highest or a peak/dip whose wavelength is the second shortest/longest.

Waveform Analysis

Card name	;Function key name	Function
Analysis	;Threshold	Threshold
Analysis	;ndB-Loss	ndB-Loss
Analysis	;SMSR	Side mode suppression ratio
Analysis	;Envelope	Envelop
Analysis	;RMS	RMS
Analysis	;Spectrum Power	Power integration
Analysis	;Analysis Off	Analysis mode end

Memory and Trace Setting

Card name	;Function key name	Function
Trace	;Set Memory/Trace	Switches between memories A and B or select a trace mode among A, B, A & B, A-B, and B-A.

Measurement data save/recall

Card name	;Function key name	Function
Save/Recall	;FD	Save, recall, delete data or format a FD.

Drivers for Back Card Function

Graph Display Method Switching

Card name	;Function key name	Function
Graph	;Graph Clear	Clear a graph
Graph	;Set Display Mode	Switch between normal, overlap, max. hold, normalize, and 3-dimensional display modes.

Applicable measurement functions

Card name	;Function key name	Function
Appli	;DFB-LD Test	Evaluate DFB-LD.
Appli	;FP-LD Test	Evaluate FP-LD.
Appli	;LED Test	Evaluate LED.
Appli	;PMD Test	Evaluate PMD.
Appli	;Multi Peak	Detect multiple peaks. (WDM Analysis)
Appli	;Opt.Amp Mode	Evaluate the optical amplifier.
Appli	;Opt.Amp Result	Evaluate the optical amplifier (output the measurement result).
Appli	;Opt.Amp-Parameter	Evaluate the optical amplifier (set parameters).
Appli	;Opt.Amp-Pin/Pout	Evaluate the optical amplifier (switches between signal light types (Pin/Pout).
Appli	;Opt.Amp-Pout to ase	Evaluate the optical amplifier (copy Pout data to the Pase memory)
Appli	;Opt.Amp-Res.Cal	Evaluate the optical amplifier (calibrate resolution).

Continued

Card name	;Function key name	Function
Appli	;WDM	Analize WDM(Parameter setting)
Appli	;WDM Result	Analize WDM(Measurement result output)
Appli	;Application Off	Terminate the applicable measurement mode.

Special Measurement Modes

Card name	;Function key name	Function
MeasurMode	;D.Range Norm/High	Wide dynamic range measurement
MeasurMode	;Ext.Trigger	External synchronization measurement
MeasurMode	;TLS Tracking	Tracking measurement
MeasurMode	;Adjust to TLS	Calibration with tunable laser source
MeasurMode	;Power Monitor	Power monitor measurement
MeasurMode	;Measure Mode Off	Terminate the measurement mode.

Title Setting

Card name	Function
Title	Title Setting

Calibration Functions

Card name	;Function key name	Function
Cal	;Offset	Set a wavelength or level offset.
Cal	;Wl Calibration	Calibrate a wavelength or initialize calibration data.
Cal	;Auto Alignment	Aligns the optical axis automatically.

Measurement Condition Saving, Recalling, and Initialization

Card name	;Function key name	Function
Condition	;Save / Recall	Save, recall, or initialize measurement conditions.

Other Functions

Card name	;Function key name	Function
Others	;Back Light	Set the backlight-off time.
Others	;Buzzer On/Off	Set the buzzer to ON a OFF

Drivers for Front Panel Keys

Name corresponding to panel key or function	Function
Light Output Off/On	Turn on/off the optional light source.
Peak to Center	Set the peak wavelength to the center wavelength.
Peak to REF Level	Sets the peak level to the reference level.
Trace Marker to Center	Set the trace marker to the center wavelength.
Sweep Start	Start single/repeat sweep.
Sweep Stop	Stop sweep.
Auto Measurement Start	Start automatic measurement.
Printer Copy / Feed	Copy/feed to the internal printer.

Markers

Card name	;Function key name	Function
Marker	;Wl Marker	Set/read a wavelength marker.
Marker	;Lvl Marker	Set/read a level marker.
Marker	;Trace Marker	Set/read a trace marker.
Marker	;Delta Marker	Set/read a delta marker.
Marker	;Marker Erase	Erase a marker.

Zone Markers

Card name	;Function key name	Function
ZoneMarker	;Set / Erase	Set/erase zone markers.
ZoneMarker	;Zone to Span	Set the center and span of zone markers.
ZoneMarker	;Zoom Out/In	Zoom in/out the zone markers.

Memory Data Read

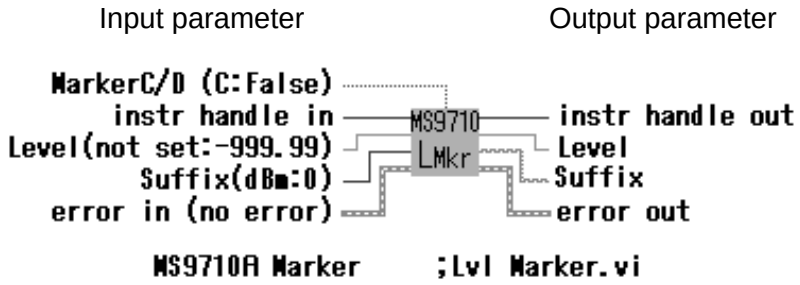
Name corresponding to panel key or function	Function
Read Memory Data	Read memory data.

Initialization, Error Messages, etc.

Name corresponding to panel key or function	Function
Initialize	Start communication.
Close	Terminate communication (for GPIB only)
Error Message	Display error messages.
Error Message Japanese	Display error messages (Japanese).
Reset	Rest the MS9710C.

11.4 Description of Measuring Instrument Driver Functions

This section explains functions and input/output parameters of measuring instrument drivers.
A measuring instrument driver VI receives data and setting values through the terminals on the left of the icon, performs the specified processing according to the input parameters, and outputs the processing results through the terminals on the right side of the icon.



11.4.1 Common parameters

This section explains most of the input/output parameters used with measuring instrument drivers.

instr handle in
instr handle out

"instr handle" is generated by "initialize.vi." It becomes the index for referencing information such as an RS-232C port number.
Arrange drivers in the order of execution and connect "instr handle out" terminals and "instr handle in" terminals, one after another, with wires.

error in
error out

The error information before execution is input to the "error in" terminal. When information indicating an error has occurred is input to the "error in" terminal, the VI performs nor processing and outputs the "error in" value through the "error out" terminal. When information indicating that no error has occurred is input to the "error in" terminal, VI performs the specified processing and outputs the post-processing error state through the "error out" terminal. Thus, errors can be checked. Connecting "error in" and "error out" terminals of VIs, one after another, with wires and using an MS9710A Error Message VI at the end of the diagram allow an error location, error code, and error message to be displayed.

status True if an error has occurred.

code Error code

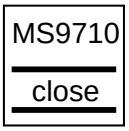
source Error location

timeout (ms) If processing is not completed within the timeout time, it is terminated and an error message is output. For processing requiring long time, set a sufficiently long time.

11.4.2 Description of functions

For input parameter setting ranges, output data formats, and so forth, see the help window.
( ) following an input parameter indicates an initial value (default).
The file name of the RS-232C is “MS9710AS...”.

MS9710A Close



This driver terminates communication with the device.
Execute it at the end of the program.
This driver is not included in the RS-232C library.

MS9710A Error Message
MS9710A Error Message Japanese

Executing this VI after executing a measuring instrument driver will display an error location, error code, and error message if any error has occurred.
If the error is an MS9710C-specific error, the value “MS9710C error code + 5000” is output.

For details on MS9710C error codes, see Appendix A.
(“101 Can’t Find Peak” and “102 Can’t Find Dip” are not handled as errors.)

“MS9710A Error Message Japanese” displays error messages in Japanese.

Input parameter:

type of dialog (OK msg:1) ..... Select an error message display dialog type.

Output parameters:

status ..... Indic puts an error message.
code ..... Error code
error message ..... Error location

MS9710A Initialize

This driver initializes the device and generates “instr handle.”
It sets the response data header to OFF.
To use a measuring instrument, this VI must be executed first.
Terminate the VI after completion of initialization.
RS-232C interface conditions are as follows:
Parity = Even; Stop Bit = 1; Character length = Fixed at 8 bits
Set MS9710C interface conditions to the above conditions. (For the setting method, see 2.2.3, “Setting RS-232C interface conditions.”)

Input parameters

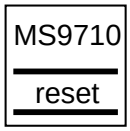
< GPIB >

GPIB address (8) ..... Input a GPIB address.
ID query (Yes:T) ..... Request the ID of the measuring instrument to check the device against it.
reset (No:F)..... Reset the measuring instrument.

< RS-232C >

Port No. (COM1:0)..... Input an RS-232C port number.
Speed (bps) (9600:4) ..... Set a transmission speed (600/1200/2400/4800/9600 bps).
ID query (Yes:T) ..... Request the ID of the measuring instrument to check the device against it.
reset (No:F)..... Reset the measuring instrument.

MS9710A Reset



This driver resets the measuring instrument.
After completion of resetting, the VI is terminated.

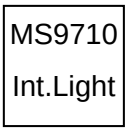
Input parameter:
timeout (ms) (600000) ..... Set a reset timeout time.

MS9710A Auto Measurement Star

This driver starts automatic measurement.
After completion of measurement, the VI is terminated.

Input parameter:
imeout (ms) (600000) ..... Set an automatic measurement timeout time.

MS9710A Light Output Off/On

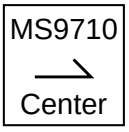


This driver turns on/off the internal light source (white light source, reference light source: option).

Input parameter:

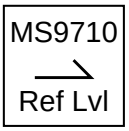
Off/On (Off:F) ..... Turn on/off optical output.

MS9710A Peak to Center



This driver sets the peak wavelength to the center wavelength.

MS9710A Peak to REF Level



This driver sets the peak level to the reference level.

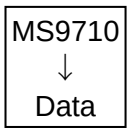
MS9710A Printer Copy/Feed

This driver produce a hard copy of or feeds the internal printer screen.
When a hard copy is produced, the VI is terminated after completion of data transfer to the printer.

Input parameter:

Number of Line Feed (Copy: 0)... When no value is input, a hard copy is produced.
When 1 or greater value is input, the internal printer feeds lines equivalent to the input value.

MS9710A Read Memory Data



This driver outputs the measurement data (Data/Suffix) and data measurement conditions (Condition Data) from the memory.

Input parameter:

Memory A/B (Memory A: F) ..... Select a memory from which data is to be read.

Output parameters:

Condition Data [cluster] ..... The following cluster elements are output:

1. Start Wavelength (nm) ..... Start wavelength
2. Stop Wavelength (nm) ..... Stop wavelength
3. Sampling Points ..... Number of sampling points

Data ..... Measurement data is output. The number of arrays is equal to the number of sampling points.

Suffix (dBm, mW) ..... The unit of measurement data is output. When the log scale is selected, dBm is output. When the linear scale is selected, mW is output.

MS9710A Sweep Start

This driver starts single/repeat sweep.

In the single sweep mode, the VI is terminated after completion of sweep.

In the repeat sweep mode, the VI is terminated after completion of sweep averaging if Sweep Average is ON. If Sweep Average is OFF, the VI is terminated after start of sweep.

Input parameters:

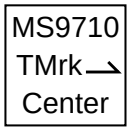
Single/Repeat (Single: F) ..... Select single or repeat sweep.

timeout (ms) (600000) ..... Set a single sweep timeout time.

MS9710A Sweep Stop

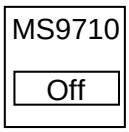
This driver stops sweep.

MS9710A Trace Marker to Center



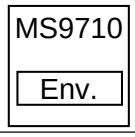
This driver sets the trace marker wavelength to the center wavelength.

MS9710A Analysis ;Analysis Off



This driver cancels the analysis mode.

MS9710A Analysis ;Envelope



This driver carries out spectrum analysis using the Envelope method and outputs the analysis result.

Input parameter:

Cut Level (dB) (no change) ..... Cut Level (dB) (no change): Set a cut level (1 to 20 dB).
If no value is input, the previous setting is used.

Output parameters:

λ c (nm) or (THz) ..... Outputs the center wavelength. If analysis is impossible, –1 is output.
Δ λ (nm) or (THz)..... Outputs the spectrum width. If analysis is impossible, –1 is output.

MS9710A Analysis ;ndB-Loss

This driver carries out spectrum analysis using an ndB-Loss method and outputs the analysis result.

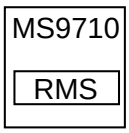
Input parameter:

Cut Level (dB) (no change) ..... Set a cut level (1 to 50 dB).
If no value is input, the previous setting is used.

Output parameters:

λ c (nm) or (THz) ..... Outputs the center wavelength. If analysis is impossible, –1 is output.
Δ λ (nm) or (THz)..... Outputs the spectrum width. If analysis is impossible, –1 is output.
N ..... Outputs the number of axis modes. If analysis is impossible, –1 is output.

MS9710A Analysis ;RMS



This driver carries out spectrum analysis using an RMS method and outputs the analysis result.

Input parameters:

Sigma Factor:Δ λ (no change) ... Δ λ (σ, 2σ, 2.35σ 3σ)
If no value is input, the previous setting is used.
S.Level (dB) (no change) ..... Set a slice level (1 to 30 dB).
If no value is input, the previous setting is used.

Output parameters:

λ c (nm) or (THz) ..... Outputs the center wavelength. If analysis is impossible, –1 is output.
k σ (nm) or (THz)..... Outputs the spectrum width. If analysis is impossible, –1 is output.

MS9710A Analysis ;SMSR

This driver performs SMSR analysis and outputs the analysis result.

Input parameter:

Side Mode (no change)..... Outputs the center wavelength.
If analysis is impossible, –1 is output.

Output parameters:

Δ λ (nm) or (THz)..... Outputs the difference in wavelength between the main peak and the side mode.
If analysis is impossible, –1 is output.

Δ l (dB) ..... Outputs the difference in level between the main peak and the side mode. If analysis is impossible, –999.99 is output.

MS9710A Analysis ;Spectrum Power

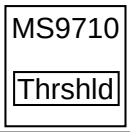
This driver carries out power integration and outputs the analysis result.

Output parameters:

Power (dBm) ..... Outputs the power integration value.

λ c (nm) or (THz) ..... Outputs the center wavelength.

MS9710A Analysis ;Threshold



This driver carries out spectrum analysis using a threshold method and outputs the analysis result.

Input parameter:

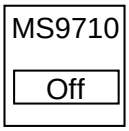
Cut Level (dB) (no change) ..... CutLevel(dB) (no change): Set a cut level (1 to 50 dB).
If no value is input, the previous setting is used.

Output parameters:

λ c (nm) or (THz) ..... Outputs the center wavelength. If analysis is impossible, –1 is output.

Δ λ (nm) or (THz)..... Outputs the spectrum width. If analysis is impossible, –1 is output.

MS9710A Appli ;Application Off



This driver terminates application measurement.

MS9710A Appli ;DFB-LD Test

MS9710
DFB-LD

This driver measures DFB-LD and outputs the measurement result.

Input parameters:

ndB Width (dB) (no change) ..... Set a ndB width (1 to 50 dB).
When no value is input, the previous setting is used.

Side Mode (no change) ..... Set a side mode (2nd Peak, Left, Right).
When no value is input, the previous setting is used.

Output parameters:

Result [cluster] ..... The following cluster elements are output.

1. SMSR (dB) ..... Side mode suppression ratio (If analysis is impossible, –999.99 is output.)
2. ndB Width (nm) ..... Width of spectrum at the level ndB lower than the peak (If analysis is impossible, –1 is output.)
3. Peak Wavelength (nm) ..... Main peak wavelength (If analysis is impossible, –1 is output.)
4. Peak Level (dBm) ..... Main peak level (If analysis is impossible, –999.99 is output.)
5. Side Mode Peak Wavelength (nm) ..... Side mode wavelength (If analysis is impossible, –1 is output.)
6. Side Mode Peak Level (dBm) ..... Side move level (If analysis is impossible, –999.99 is output.)
7. Mode Offset (nm) ..... Mode offset (difference in wavelength between main peak and sidemode) (If analysis is impossible, –1 is output.)
8. Stop Band (nm) ..... Stop band (interval between wavelengths of both side modes of main peak) (If analysis is impossible, –1 is output.)
9. Center Offset (nm) ..... Center offset (difference between main peak wavelength and center wavelength of both side modes) (If analysis is impossible, –999.99 is output.)

MS9710A Appli ;FP-LD Test

MS9710
FP-LD

This driver measures FP-LD and outputs the measurement result.

Input parameter:

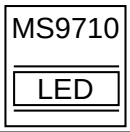
Mode Cut Level (dB) (no change) Mode Cut Level (dB) (no change)
..... Set a mode cut level (1 to 50 dB). When no value is input, the previous setting is used.

Output parameters:

Result [cluster] ..... Outputs the following cluster elements:

1. FWHM (2.35s) (nm) ..... Half-magnitude full width obtained through the RMS analysis (If analysis is impossible, –1 is output.)
2. Mean Wavelength (nm)..... Center wavelength obtained through the RMS analysis (If analysis is impossible, –1 is output.)
3. Peak Wavelength (nm) ..... Main peak wavelength (If analysis is impossible, –1 is output.)
4. Peak Level (dBm) ..... Main peak level (If analysis is impossible, –999.99 is output.)
5. Mode ..... Number of axis modes used in ndD-Loss analysis (If analysis is impossible, – 1 is output.)
6. Mode Spacing (nm) ..... Mode spacing (axis mode interval) (If analysis is impossible, –1 is output.)
7. Total Power (dBm) ..... Power integration value (If analysis is impossible, –999.99 is output.)

MS9710A Appli ;LED Test



This driver measured LED and outputs the measurement result.

Input parameters:

ndB Width (dB) (no change) ..... Set ndB Width (1 to 50 dB).
When no value is input, the previous setting is used.

Power Cal (dB) (no chage) ..... Set Power Cal (–10.00 to 10.00 dB).
When no value is input, the previous setting is used.

Output parameters:

Result [cluster] ..... The following cluster elements are output:

1. Mean Wavelength (FWHM) (nm)
   ..... Center wavelength obtained through the RMS measurement (If analysis is impossible, –1 is output.)
2. Mean Wavelength (ndB) (nm)
   ..... Center wavelength obtained through the threshold analysis (If analysis is impossible, –1 is output.)
3. FWHM (2.35σ) (nm) ..... Half-magnitude full width obtained through the RMS analysis (If analysis is impossible, –1 is output.)
4. ndB Width (nm) ..... Wavelength width obtained through the threshold analysis (If analysis is impossible, –1 is output.)
5. Peak Wavelength (nm) ..... Peak wavelength (If analysis is impossible, –1 is output.)
6. Peak Level (dBm) ..... Peak level (If analysis is impossible, –999.99 is output.)
7. Pk Dens (/1nm) (dBm) ..... Peak power per 1 nm (If analysis is impossible, –999.99 is output.)
8. Total Power (dBm) ..... Power integration value (If analysis is impossible, –999.99 is output.)

MS9710A Appli ;Multi Peak

This driver detects multiple peaks and outputs the detected peak wavelengths and levels.
This icon is to keep compatibility with the older version. Normally use the WDM and WDM-R.

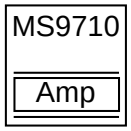
Input parameter:

S.Level (dB) (no change) ..... Set a slice level (1 to 50 dB).
When no value is input, the previous setting is used.

Output parameters:

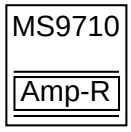
Peak Count ..... Outputs the number of detected peaks.
Wavelength (nm) ..... Outputs detected peak wavelengths as arrays in ascending order of wavelength.
Level (dBm) ..... Outputs detected peak levels as arrays in ascending order of wavelength.

MS9710A Appli ;Opt.Amp Mode



This driver selects the optical AMP measurement mode.

MS9710A Appli ;Opt.Amp Result



This driver outputs the optical AMP measurement result.

Output parameters:

Result [cluster] ..... Outputs the following cluster elements:
1. Gain (dB) ..... Optical amplifier gain (amplification ratio)
   (If analysis is impossible, –999.99 is output.)
2. NF (dB) ..... Optical amplifier NF (If analysis is impossible, –999.99 is output.)
3. Signal Wevelength (nm) ..... Signal light peak wavelength (If analysis is impossible, –1 is output.)
4. ASE Level (/Res) (dBm) ..... ASE level per resolution (If analysis is impossible, –999.99 is output.)
5. Res (nm) ..... Resolution data used for NF measurement

MS9710
Amp-P

MS9710A Appli ;Opt.Amp-Parameter

This driver sets parameters for optical AMP measurement.

Input parameters:

Parameter [cluster]..... Set the following cluster elements.
(When no value is input, the previous setting is used.)

1. NF Select (no change) ..... Select an NF calculation mode between NF (S-ASE) and NF (Total).
2. Method (no change) ..... Select an NF measurement method.
   1: NF measurement made without using a spectrum division method
   2: NF measurement made using a spectrum division method
   3: NF measurement using a polarization nullification method
   4: NF measurement made using a pulse method
   5: WDM measurement
3. ASE Fitting (no change)..... Select a fitting method used for obtaining an ASE level.
   1: Gauss fitting
   2: Mean
4. Fitting Span (nm) (no change).. Set a span subject to the fitting carried out to obtain an ASE level.
5. Masked Span (nm) (no change)
   ..... Set a span excluded from the fitting carried out to obtain an ASE level.
6. Pin Loss (dB) (no change).... Set the difference between the level of the signal light input to the optical amplifier and the level of the signal light input to the optical spectrum analyzer.
7. ..... Set the difference between the actual level of the optical amplifier output signal and the level of the optical amplifier output signal input to the optical spectrum analyzer.
8. NF Cal (no change) ..... Set a calibration value used for NF calculation.
9. O.BPF Lvl Cal (dB) (no change)
   ..... Set the difference in level between the pass and block ranges of the optical band pass filter inserted between the optical amplifier and the optical spectrum analyzer.
10. O.BPF BW (nm) (no change) .. Set an effective optical filter width used for NF (total) calculation.
11. Pol Loss (dB) (no change).... Set a level loss at the polarization control nullifying stage used for polarization nullification.

MS9710
Amp•in/out

MS9710A Appli ;Opt.Amp-Pin/Pout

This driver determines whether the signal light before amplification (Pin) or the signal light after amplification is to be measured.

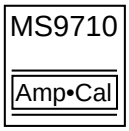
Input parameter:

Memory Pin/Pout (Pin: F) ..... Select Pin or Pout.

MS9710A Appli ;Opt.Amp-Pout to ase

During the measurement made using a polarization nullification method (PlznNull), this driver copies the spectrum temporarily written into the Pout memory to the internal memory Pase as an ASE.

MS9710A Appli ;Opt.Amp-Res.Cal



During the optical AMP measurement, this driver calibrates resolution of the optical spectrum analyzer. After completion of calibration, the VI is terminated.

Input parameters:

Mode Select (Initial: 0) ..... Initial sets the resolution calibration data to the default value.
Execute starts calibration.

Timeout (ms) (600000) ..... Set a resolution calibration timeout time.

MS9710A Appli ;PMD Test

This driver measures PMD (Polarization Mode Dispersion) and outputs the result. The measurement mode is set to Auto.

Input parameter:

Mode Cpl Factor (no change) ..... Mode Cpl Factor (no change): Set Mode Cpl Factor (0.01 to 1.00).
When no value is input, the previous setting is used.

Output parameters:

Result [cluster] ..... The following cluster elements are output:
If analysis is impossible, –1 is output.

1. Diff.Group Delay (fs) ..... Differential group delay time
2. 1st Peak Wavelength (nm) ... First peak wavelength
3. Last Peak Wavelength (nm) .. Last peak wavelength
4. Peak Count ..... Number of peaks included in analysis range

MS9710A Application ;WDM

MS9710
WDM

This driver sets parameters used in the WDM application.

Input parameters:

Display Mode (0. No Change) ... Sets display mode
0. No Change
1. Multi Peak ..... Sets Multi Peak display
2. SNR ..... Sets SNR display
3. Relative ..... Sets Relative display
4. Table ..... Sets Table display
Level (dB) (No Change) ..... Sets slice level
Input Parameter (No Change) ... Sets the following parameters. This is cluster.
SNR Dip Direction (No Change) .... Sets search direction in SNR Display
0. No Change
1. Higher
2. Left
3. Right
Delta Lambda (nm) (No Change) ... Sets D l in SNR Display
Reference Peak No. (No Change) ... Sets Ref Peak No. in Relative display

MS9710A Application ;WDM-R

MS9710
WDM-R

This reads the results of the WDM application.

Input parameter:

Sort Wl/Lvl (Wl) ..... Whether the data are aligned in ascending order of wavelength or in descending order of level.

Output parameters:

Peak Count ..... Outputs number of peaks.
Gain Tilt. (dB) ..... Outputs Gain Tilt in SNR Display. When analysis is impossible, –999.99 is output.
Result ..... Outputs analysis results. This is cluster in one order array.
Wavelength (mm) ..... Outputs wavelength at peak level or –1 when analysis is impossible
Level (dB) ..... Outputs peak level or –999.99 when analysis is impossible
SNR (dB) ..... Outputs SNR value in SNR Display or –999.99 when analysis is impossible
SNR Direction ..... Outputs SNR detection direction in SNR Display or –999.99 when analysis is impossible.
Spacing (mm) ..... Outputs spacing in Relative Display or –999.99 when analysis is impossible
Relative Wl (nm) ..... Outputs relative wavelength in Relative Display or –999.99 when analysis is impossible.
Relative Lvl (dB) ..... Outputs relative level in Relative Display or –999.99 when analysis is impossible.
Frequency (THz) ..... Output peak frequency.
Spacing F (GHz) ..... Output spacing frequency.

MS9710A Cal ;Auto Alignment

MS9710 Auto Align

This driver aligns the optical axis automatically.
After completion of alignment, the VI is terminated.

Input parameters:

Mode Select (Initial: 0) ..... "Initial" sets the alignment position data to the default value.
Execute starts calibration.
timeout (ms) (600000) ..... Set an auto alignment timeout time.

MS9710A Cal ;Offset

MS9710 Offset

This driver sets the offset of the level axis.

Input parameters:

Wavelength Offset (nm) (no change)
..... Set the offset of the wavelength axis.
When no value is input, the previous setting is used.
Level Offset (dB) (no change) .... Set the offset of the level axis.
When no value is input, the previous setting is used.

MS9710A Cal ;Wl Calibration

MS9710 λ Cal

This driver creates wavelength calibration data by performing calibration using an external or reference light source.
After completion of calibration, the VI is terminated.

Input parameters:

Mode Select (Initial: 0) ..... "use int.Light" calibrates the wavelength using the internal reference light source (option), creating wavelength calibration data.
"us Ext.Light" calibrates the wavelength using the external reference light source, creating wavelength calibration data.
"Initial" sets the wavelength calibration data to the default value.
timeout (ms) (600000) ..... Set a wavelength calibration timeout time.

MS9710A Condition ;Save / Recall

MS9710 Condition

Recalls or saves measurement conditions from/in the internal memory.

Input parameters:

Save/Recall (Recall: F) ..... Select saving or recalling.
Memory No.(1) ..... Set a memory number (between 0 and 5 when recalling (0 is a factory-set memory number); between 1 and 5 when saving).

MS9710A Graph ;Graph Clear

MS9710 Graph Clear

This driver clears the displayed graph.

MS9710A Graph ;Set Display Mode

MS9710 Disp Mode

This driver sets a display mode.

Input parameters:

Display Mode (Normal: 0) ..... Select a display mode.
0. Normal ..... Set a normal display mode.
1. Over Lap ..... Set an overlap display mode.
2. Max Hold ..... Set a max. hold display mode.
3. Normalize ..... Set a normalize display mode.
4. 3D ..... Set a 3-dimensional display mode.

3D Type (Type1: 1) ..... Set a 3-dimensional display mode type (1,2, or 3).
(Effective only when the display mode is 3D.)

3D Angle (degree) (45 degree: 1) ... Set a 3-dimensional display angle (30, 45, 60, or 90 degrees).
(Effective only when the display mode is 3D. When the 3D Type is set to 3, only 45 may be set.)

MS9710A LevelScale ;Log/Linear

MS9710 Log/Lin

This driver switches between log and linear scales and sets a scale value.

Input parameters:

Log/Linear (Log: F) ..... Select a log or linear scale.
Scale Value (no change) ..... Set a scale value.
When no value is input, the previous setting is used.

Output parameter:

Scale Value · Suffix ..... Outputs a scale value and unit.

MS9710A LevelScale ;Opt.Att Off/On

MS9710 Opt.Att

This driver turns on/off the internal attenuator.

Input parameter:

Off/On (Off: F) ..... Turn on/off the attenuator.

MS9710A LevelScale ;Ref Level

MS9710
Ref Lvl

This driver sets a reference level when a log scale is selected.

Input parameter:

Ref Level (dBm,dB) (no change) ... Set a reference level.

Output parameter:

Reference Level ..... Outputs a reference level.

MS9710A Marker ;Delta Marker

MS9710
ΔMkr

This driver sets a delta marker and outputs the differences in wavelength and level between the delta marker and the trace marker.

Input parameter:

Wavelength (nm) or (THz) ..... Set a wavelength of the delta marker.
When no value is input, the previous setting is used. (This parameter must not be omitted when no delta marker is displayed.)

Output parameters:

Delta Wavelength (nm) or (THz) ... Outputs the difference in wavelength between the delta marker and trace marker.
Delta Level ..... Outputs the difference in level between the delta marker and the trace marker.
If analysis is impossible, –1 is output. (Effective only when the linear scale is selected.)

MS9710A Marker ;Lvl Marker

MS9710
LMkr

This driver sets a level marker and marker values (level and unit).

Input parameters:

MarkerC/D (C: False) ..... Select marker C or D.
Level (no change) · Suffix(dBm: 0)
..... When no value is input, the previous setting is used. (This parameter must not be omitted when no level marker is displayed.)

Output parameter:

Level · Suffix ..... Output a marker level and unit.

MS9710A Marker ;Marker Erase

MS9710
OffMkr

This driver erases all markers other than zone markers.

MS9710A Marker ;Trace Marke

MS9710
TMkr

This driver sets a trace marker and outputs marker values (wavelength, level, and unit).

Input parameter:

Wavelength (nm) or (THz) ..... Set a trace marker wavelength.
When no value is input, the previous setting is used. (This parameter must not be omitted when no trace marker is displayed.)

Output parameters:

Wavelength (nm) or (THz) ..... Outputs a trace marker wavelength.
Level · Suffix ..... Outputs a trace marker level and unit.

MS9710A Marker ;Wl Marker

MS9710
λMkr

This driver sets a wavelength marker and outputs marker wavelength.

Input parameters:

MarkerA/B (A: False) ..... Select marker A or B.
Wavelength (nm) or (THz) ..... Set a marker wavelength.
When no value is input, the previous setting is used. (This parameter must not be omitted when no wavelength marker is displayed.)

Output parameter:

Wavelength (nm) or (THz) ..... Outputs a marker wavelength.

MS9710A MeasurMode ;D.Range Norm/High

MS9710
Norm/Hi

This driver selects a normal or high dynamic range.

Input parameter:

Normal/High (Normal: F) ..... Set a normal or high dynamic range.

MS9710A Measure Mode ;TLS Tracking

MS9710
TLS
Tracking

This sets the TLS tracking measurement mode to ON or OFF.

Input parameter:

YLS Tracking On/Off (0: Off) ... Sets the TLS tracking measurement mode to ON or OFF.

0. OFF
1. ON

MS9710A Measure Mode: Adjust to TLS

MS9710
Adjust
to TLS

This performs Adjust to TLS of the TLS tracking mode.

Input parameter:

Adjust to TLS (0: Cancel) ..... Performs Adjust to TLS of the TLS tracking mode

0. Cancel
1. Execute

MS9710A MeasurMode ;Ext.Trigger

MS9710
Ext.Trig

This driver carries out EXT trigger measurement.
When carrying out EXT trigger measurement, execute “Measure Mode Off.vi”.

Input parameter:

Delay Time (ms) (no change)..... Delay Time (ms) (no change): Set a delay time (0 to 5000000 us).
When no value is input, the previous setting is used.

MS9710A MeasurMode ;Measure Mode Off

MS9710
Mes.Mode
Off

This driver passes through EXT trigger and power monitor measurement modes and return to the normal measurement mode.

MS9710A MeasurMode ;Peak Hold

MS9710
PkHold

(Not used by MS9710C.)

MS9710A MeasurMode ;Power Monitor

MS9710
Pow.M

This driver sets power monitor measurement and outputs a power monitor measurement value.
The VI terminates when a power monitor value is read.
To terminate power monitor measurement, execute “Measure Mode Off.vi”.

Input parameter:

Wl Range (nm) (no change) ..... Set a wavelength range (632.8/850.0/1300.0/1550.0 nm).
When no value is input, the previous setting is used.

Output parameter:

Power (dBm) ..... Outputs a power monitor measurement value.

MS9710 Others ;Buzzer Off/On

MS9710
Buzzer

This turns the buzzer ON or OFF.

Input parameter:

Off/On (Off: F) ..... Selects On or Off

MS9710A Others ;Back Light

MS9710
Back
Light

This driver sets a backlight off time.

Input parameter:

BackLight Time (min) (10) ..... Set a backlight off time.
If 0 is input, the backlight will not turn off.

MS9710A Peak/Dip ;Peak/DipSearch(1)

MS9710
1
Psearch

This driver moves the trace marker to the peak or dip and output a trace marker wavelength and level.
When processing is complete, the VI is terminated.

Input parameter:

Peak/DipSearch (Peak: F) ..... Peak/Dip Search (Peak: F): Select peak or dip search.

Output parameters:

Wavelength (nm) ..... Outputs a trace marker wavelength.
When no peak or dip is detected, –1 is output.
Level Suffix ..... Outputs a trace marker level and unit.
When no peak or dip is detected, –999.99 is output.
When a trace marker level cannot be obtained, –1 is output (linear scale only).

MS9710A Peak/Dip ;Peak/DipSearch(2)

MS9710
2
Psearch

This driver detects a peak or dip set by Search Mode among the peaks and dips detected by Peak/DipSearch(1), and moves the trace marker there.
When Peak/DipSearch has not been executed, PeakSearch is executed.
When processing is complete, the VI is terminated.

Input parameter:

Search Mode (Next: 0) ..... Set a search mode (Next, Last, Left, Right).

Output parameters:

Wavelength(nm) ..... Outputs a trace marker wavelength.
When no peak or dip is detected, –1 is output.
Level Suffix ..... Outputs a trace marker level and unit.
When no peak or dip is detected, –999.99 is output.
If a trace marker level cannot be obtained, –1 is output (linear scale only).

MS9710A Res/BW/Avg ;Actual Resolution

MS9710
Act-Res

This driver turns off actual resolution display and outputs an actual resolution value.

Input parameter:

Off/On (Off: F) ..... Turn on/off actual resolution display.

Output parameter:

Actual Resolution value (nm)..... Outputs an actual resolution value.

MS9710A Res/BW/Avg ;Average

MS9710
Avg

Sets or cancels point average processing, sweep average processing, and smoothing processing.

Input parameters:

Average Times (Off: 0) ..... Sets a number of point average processing points or a sweep average processing count.
Point/Sweep (Point: F) ..... Select point or sweep average.
Smooth Points (Off: 0) ..... Set a number of smoothing points (3, 5, 7, 9, or 11).
When no value is input, smoothing processing is canceled.

MS9710A Res/BW/Avg ;Res/BW/SmplPt

MS9710
Res/BW
SmplPt

This driver sets measurement resolution, video bandwidth, and number of sampling points in span.
When no value is input, the previous setting is used.

Input parameters:

Sampling Points (no change) ..... Set a number of sampling points in span.
Resolution (nm) (no change) ..... Set measurement resolution.
Video Band Width (no change) .. Set a video bandwidth.

Output parameters:

Sampling Points ..... Outputs a number of sampling points in span.
Resolution (nm) ..... Outputs measurement resolution.
Video Band Width (Hz) ..... Outputs a video bandwidth.

MS9710A Save/Recall ;FD

MS9710
FD

This driver saves data, recalls data, deletes data, or formats the FD.
When saving, recalling, deletion, or formatting is complete, the VI is terminated.

Input parameters:

Mode (Save: 0) ..... Select data saving, data recalling, data deletion, or FD formatting.
Data Addition (no change) ..... Determine whether bit-map-format data and text-format data are to be saved in addition to the basic-format data.
When no value is input, the previous setting is used.
File Name (“ ”) ..... Input a file name.
timeout (ms) (600000) ..... Set a timeout time.

MS9710A Title

MS9710
Title

This driver inputs a title to be displayed on the screen.

Input parameter:

Title (“ ”) ..... Set a title to be displayed on the screen (max. 30 characters).

MS9710A Trace ;Set Memory/Trace

MS9710
Trace

This driver switches between data saving memories and selects a display (trace) mode.

Input parameters:

Memory A/B (no change) ..... Select a memory for saving data.
Trace Mode (no change) ..... Select a trace type (A, B, A & B, A-B, or B-A).

MS9710A Wavelength ;Air/Vacuum

MS9710
Air/Vac

This driver sets the wavelength display mode to “in Air” or “in Vacuum.”

Input parameter:

Air/Vacuum (Air: F) ..... “in Air” or “in Vacuum.”

MS9710A Wavelength ;Center/Start/Stop

MS9710
Center
strt stp

This driver sets and reads center, start, and stop wavelengths.
When no value is input, the previous setting is used.

Input parameters:

Center Wl (nm) (no change) ..... Set a center wavelength.
Start Wl (nm) (no change) ..... Set a start wavelength.
Stop Wl (nm) (no change) ..... Set a stop wavelength.

Output parameters:

Center Wl (nm) ..... Outputs a center wavelength.
Start Wl (nm) ..... Outputs a start wavelength.
Stop Wl (nm) ..... Outputs a stop wavelength.

MS9710A Wavelength ;Set Span

MS9710
Span

This driver sets and reads a span wavelength.
When no value is input, the previous setting is used.

Input parameter:

Span (nm) (no change) ..... Set a span wavelength.

Output parameter:

Span (nm) ..... Outputs a span wavelength.

MS9710A ZoneMarker ;Set / Erase

MS9710
ZoneMkr

This driver sets or erases zone markers.
When a Center or Span parameter is not input, zone markers are erased.

Input parameters:

Center (nm) (Erase) ..... Sets the center.
Span (nm) (Erase) ..... Sets a span.

MS9710A ZoneMarker ;Zone to Span

MS9710
ZoneMkr
Span

This driver sets the area between zone markers as a span.

MS9710A ZoneMarker ;Zoom Out/In

MS9710
ZoneMkr
Zoom

This driver zooms in or out zone markers.

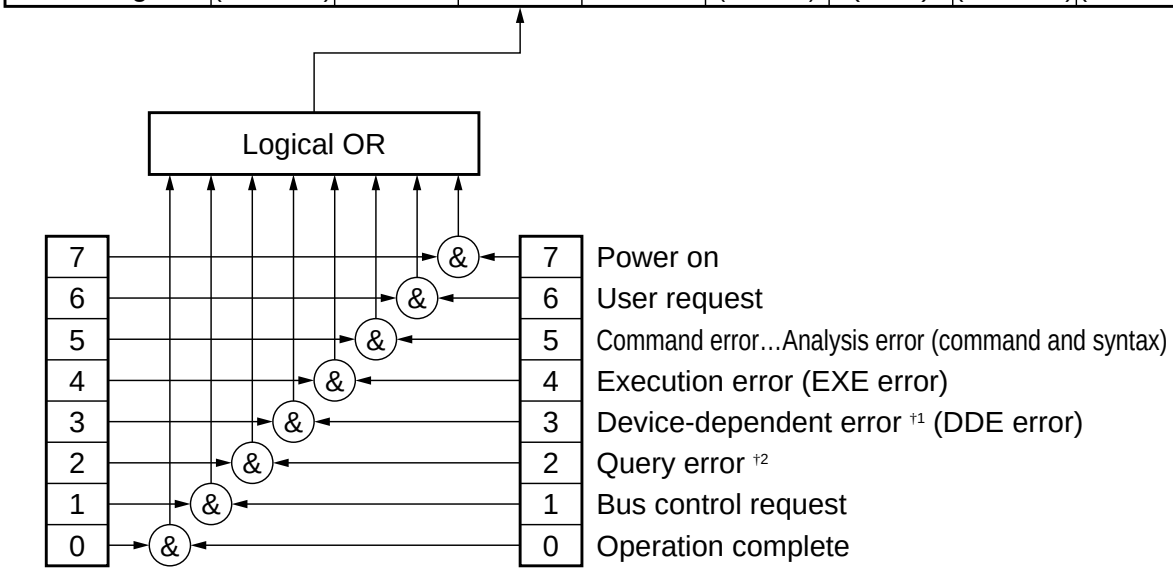
Input parameter:

Zoom Out/In (Out: F) ..... Sets “Zoom In” or “Zoom Out.”

Appendix A Error Messages

This appendix lists error messages summarized by bits 5 and 3 of the status byte register. Bit 5 indicates error messages reported by bits 2 to 5 of the standard event status register. Bit 3 indicates RES-Uncal and Peak/Dip errors.

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Line	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1
Summary message bit	ESB (reserved)	RQS	ESB	MAV	ESB (ERROR)	ESB (END)	ESB (reserved)	ESB (reserved)



†1:

Device-dependent error ..... Condition error (setting condition error, etc.)
Hardware error (CAL error)

†2:

Query error ..... The host has not read a received query message.

When an execution error (EXE error: bit 4) or device-specific error (DDE error: bit 3) of the standard event status register is set, the MS9710C reports its error number and message to the operator as the information about the error cause.
GPIB error numbers can be displayed using a ERR? command (see the explanation of device messages).

The lists on the following pages summarizes error numbers and messages.

A.1 System Errors (000 to 099)

No.	Error message	Status	Output condition
000	No error		
001	Optical Error (RAM)		RAM error
002	Optical Error (Slit-1)		Slit 1 error
003	Optical Error (Slit-2)		Slit 2 error
004	Optical Error (Wl Align)		Wavelength alignment error
005	Optical Error (Opt Att)		Optical ATT error
006	Not used		
007	Optical Error (Light Source)		Light source error
008	Optical Error (Grating)		Grating error
009	Optical Error (Offset)		Offset error
010	Optical Error (Over Power)		Excessive light input error

A.2 Measurement Errors (100 to 199)

No.	Error message	Status	Output condition
100	Auto Measure Incomplete	ESE-DDE	Auto measurement has not been completed normally.
101	Can’t Find Peak	ESE-DDE	No peak was found.
102	Can’t Find Dip	ESE-DDE	No dip was found.
103	Not used		
104	TMkr Not Display	ESE-DDE	No trace marker is displayed.
105 to 109	Not used		
110	Wl Cal Error (Optical Level)	ESE-DDE	Insufficient optical level during wavelength calibration
111	Wl Cal Error	ESE-DDE	Wavelength calibration error
112	Align Error (Optical Level)	ESE-DDE	Insufficient optical level during optical axis automatic alignment
113	Align Error	ESE-DDE	Optical axis automatic alignment error
114	Res Cal Error	ESE-DDE	Resolution calibration error
115	TLS Calibration Error	ESE-DDE	Error on calibration with TLS

A.3 Key Operation Errors (200 to 299)

No.	Error message	Status	Output condition
200	Reserved		
201	Input Value Error	ESE-EXE	The input value exceeds the range.
202 to 204	Not used		
205	Invalid In smooth	EXE-DDE	Invalid during smooth operation.
206	Invalid In Peak Hold	EXE-DDE	Invalid during peak hold measurement.
207	Invalid In Ext-Trig	EXE-DDE	Invalid during external trigger measurement.
208 209	Not used		
210	Valid Only In Spectrum Mode	ESE-DDE	Valid only during the spectrum measurement.
211	Invalid In Auto Measure	ESE-DDE	Invalid during automatic measurement.
212	Invalid In Condition	ESE-DDE	Invalid during condition.
213	Invalid In M emory Select	ESE-DDE	Invalid during memory list display.
214	Invalid In Title Input	ESE-DDE	Invalid during title input.
215	Invalid In Others Input	ESE-DDE	Invalid during Others input.
216	Invalid In Measurement	ESE-DDE	Invalid in Measurement.
217	Not used		
218	Valid Only In Power Monitor	ESE-DDE	Valid only during power monitor measurement.
219	Valid Only In Analysis	ESE-DDE	Valid only during waveform analysis.
220	Valid Only In Normal Disp	ESE-DDE	Valid only in the normal display mode.
221	Valid Only In 3D Disp	ESE-DDE	Valid only in the 3D display mode.
222	Not used		
223	Invalid In Normalize Disp	ESE-DDE	Invalid in the normalize display mode.
224	Invalid In 3D Disp	ESE-DDE	Invalid in the 3D display mode.
225	Invalid In 3D Disp Type3	ESE-DDE	Invalid in the 3D type-3 display mode.
226	Invalid In Peak/Dip Searching	ESE-DDE	Invalid during peak/dip search.
227	Invalid In Overlap Disp	ESE-DDE	Invalid in the overlap display mode.
228	Invalid In Max Hold Disp	ESE-DDE	Invalid in the max. hold display mode.
229	Invalid In Power Monitor	ESE-DDE	Invalid during power monitor measurement.
230	Valid Only In Log Scale	ESE-DDE	Valid only in the log scale mode.
231	Valid Only In Linear Scale	ESE-DDE	Valid only in the linea r scale mode.
232	Level Unit Error	ESE-EXE	Invalid level unit
233	Invalid In Application Mode	ESE-DDE	Invalid during application measurement.
234	Invalid In Opt. Amp	ESE-DDE	Invalid during optical amp. measurement.
235	Invalid In Zone Marker	ESE-DDE	Invalid when zone markers are displayed.

No.	Error message	Status	Output condition
236	Invalid In Zero Span	ESE-DDE	Invalid when zone markers are displayed.
237	Invalid In Auto PMD	ESE-DDE	Invalid in the auto mode (PMD).
238	Invalid In Analysis	ESE-DDE	Invalid in the analysis mode.
239	Invalid In Swp-Avg	ESE-DDE	Invalid in the sweep average mode.
240	Set A Trace or B Trace	ESE-DDE	Invalid when the trace mode is not A or B.
241	Invalid In A-B,B-A Trace	ESE-DDE	Invalid when the trace mode is A-B or B-A.
242	Invalid In A-B Trace	ESE-DDE	Invalid when the trace mode is A-B.
243	Invalid In B-A Trace	ESE-DDE	Invalid when the trace mode is B-A.
244	Invalid In A&B Trace	ESE-DDE	Invalid when the trace mode is A&B.
245	Invalid In Multi Peak	ESE-DDE	Invalid during multipeak detection.
246	Invalid In Opt. Amp-Test	ESE-DDE	Invalid during optical amp. test.
247	Invalid In Smplg-5001 pnt	ESE-DDE	Invalid when the number of sampling points is 5001.
248	Invalid In Plzn Null, Pulse, WDM	ESE-DDE	Invalid when in the PLzn-Null, Pulse, or WDM test mode.
249	Invalid In This State	ESE-DDE	Invalid under this measurement condition.
250	Set Same Wl For A,B	ESE-DDE	Difference in wavelength between memories A and B.
251	Set Same Resolution For A,B	ESE-DDE	Difference in resolution between memories A and B.
252	Set Same Points For A,B	ESE-DDE	Difference in number of points between memories A and B.
253	Set Same Air/Vac For A,B	ESE-DDE	Difference in Air/Vac mode between memories A and B.
254	Option Error	ESE-DDE	Option error
260	Invalid In TLS Tracking	ESE-DDE	Invalid in TLS Tracking.
261	Invalid In Frequency Marker	ESE-DDE	Invalid when the marker mode is frequency.

A.4 Device Errors (300 to 499)

No.	Error message	Status	Output condition
	– Errors related to FD –		
300	FD Does Not Exist	ESE-DDE	An FD has not been set.
301	FD Format Error	ESE-DDE	The FD format is invalid.
302	Can't Find File	ESE-DDE	The specified file does not exist on the FD.
303	FD Memory Full	ESE-DDE	Too many memory for the FD.
304	FD Write Protected	ESE-DDE	The FD is write-protected.
305	File Incomplete	ESE-DDE	Files on the FD are incomplete.
306	Data File Full	ESE-DDE	Too many files for the FD
	– Errors related to printer –		
320	No paper	ESE-DDE	End of paper
321	Lever Off	ESE-DDE	The paper hold level is released.
322	Printer Error	ESE-DDE	No printer is connected or the printer device type is invalid.
323	Printer Error	ESE-DDE	The thermistor connector is not connected.
324	Printer Error	ESE-DDE	Abnormal head temperature
	– Errors related to GPIB/RS-232C		
400	Reserved		
401	Command Error	ESE-CME	An undefined header has been received.
402	Command Error	ESE-CME	The integer part of the numeric data is invalid.
403	Command Error	ESE-CME	The real part of the numeric data is invalid, or invalid
404	Command Error	ESE-CME	The integer part of the numeric data is invalid, or invalid exponent-type data has been input.
405	Command Error	ESE-CME	The suffix part (unit) is invalid.
406	Command Error	ESE-CME	The number of arguments does not follow command syntax.
407	Command Error	ESE-CME	A ∗PCB command was received but there is no controller function for this command.
408 to 419	No used		
420	TLS Interface Error	–	Abnormality in connecting with TLS.
421	TLS Not Respond	–	No response from TLS.

Appendix B Binary Data Transfer Formats

This appendix explains formats of the binary data transferred in response to query commands DBA? and DBB?.

● Log scale

Data structure	16 bits/data
Transfer order	High-order byte → Low-order byte
Numeric representation	Signed 16 bit value (0.01 dBm = 1) Measured value = Input value × 0.01 dBm
Unit	dBm

Example: When 2 byte input values are 233 and 162
Hexadecimal notation: E9A2
Decimal notation: –5726
Measured value: –5726 × 0.01 dBm = –57.26 dBm

● Linear scale

Data structure	32 bits/data (exponent part, 16 bits; mantissa part, 16 bits)
Transfer order	High-order byte of exponent part → Low-order byte of exponent part → High-order byte of mantissa part → Low-order byte of mantissa part
Numeric representation	Mantissa part: Value (0.1000 to 1.0000) × 10000 Exponent part: Signed 16 bit value (+3 to –8) Measured value = (Mantissa part value × 0.0001) E + (Exponent part value) mW
Unit m	mW

Example: When 4-byte input values are 255, 247, 39, and 16
Exponent part value: –9
Mantissa part value: 10000
Measured value: (10000 × 0.0001)E + (–9) mW = 1E – 9 mW

Appendix C Comparison Table of GPIB Commands of Controller

Controller / Function	PACKET V	PC9801	IBM-PC	HP9000 series
Output data to device	WRITE @device-number:data	PRINT @listener-address;data	CALL IBWRT( )	OUTPUT device selector;data
Output binary data to device	BIN WRITE @device-number:data	WBYTE command;data		
Assign data input from device to variable	READ @device-number: variable	INPUT @talker-address,listener-address;variable LINE INPUT @talker-address,listener-address;variable	CALL IBRD( )	ENTER device selector;variab
Assign binary data input from device to variable	BIN READ @device-number:variable	RBYTE command;variable		
Initialize interface function	IFC @select-code	ISET IFC	CALL IBSIC( )	ABORT select-code
Turn on REN line	REN @select-code	ISET REN	CALL IBSRE( )	REMOTE Device selector (select-code)
Turn off REN line	LCL @LCL @select-code (sets all devices in local mode) LCL @ device-number (sets only the specified devices as listeners and issues 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)
Output interface message and data	COMMAND @select-code :message-character-string [;data]		CALL IBCMD( ) CALL IBCMDA( ) (asynchronous)	SEND select-code ;message-list

Controller / Function	PACKET V	PC9801	IBM-PC	HP9000 series
Trigger the specified device	TRG @ device-number	WBYTE &H3F, listener-address, secondary-address,&08;	CALL IBTRG( )	TRIGGER Device selector
Initialize device	DCL @select-code (all devices with corresponding to specified select code) DCL @ device-number (only specified devices)	WBYTE &H3F,&H14;WBYTE &H3F, listener-address, secondary-address, H04;		CLEAR Device selector (select-code) CLEAR Device selector (select-code + primary address)
Disable switching of device from remote to local	LLO @ select-code	WBYTE &H3F,&H11;		LOCAL LOCKOUT
Transfer control right to specified device	RCT @ device-number	WBYTE talker-address,&H09	CALL IBPCT( )	PASS CONTROL
Issue service request	SRQ @ select-code	ISET SRQ	CALL IBRSV( )	REQUEST select-code
Perform serial polling	STATUS @ device-number	POLL	CALL IBRSP( )	SPOLL (Device selector) (Function)
Set terminator code	TERM IS	CMD DELIM	CALL IBEOS( ) CALL IBEOT( )	
Set limits value for timeout check		CMD TIMEOUT	CALL IBTOM( )	

Appendix D Example of Program Used on PC9801

This appendix gives an example of a program that reads measurement data using a PC-9801 personal computer. The following program lines correspond to those of the PACKET V sample program on page 10-5.

```
 10  ' ****** GPIB initialize ******
 20    ISET IFC
 30    ISET REN
 40    CMD DELIM = 2
 50    CMD TIMEOUT = 5
 60  '
 70    DIM D $ (500)
 80  '
 90  ' ****** DATA READ TEST ******
100    PRINT @8 ; "DMA?"
110    FOR I = 0 TO 500
120    INPUT @8 ; D $ ( I )
130    PRINT I, D $ ( I )
140    NEXT I
150  '
160    END
```

Lines 20 to 50: GPIB initialization
Line 70: Array declaration
Lines 100 to 140: Equivalent to PACKET V sample program

Notes:

- When controlling the MS9710C from the PC-9801 via the GPIB, be sure to initialize the GPIB at the beginning of the program.

- DBA?, DBB?, DQA?, and DQB? are intended for personal computers that can input/output arrays. If your personal computer does not have an array input/output function or it does not use an equivalent program, use DMA? and DMB? commands.

Appendix E MS9710C and MV02 (MS9703A) Command Compatibility Table

This appendix summarizes compatibility between MS9710C spectrum analyzer device messages and MV02 (MS9703A) spectrum analyzer device messages.

MV02 device messages marked with ◎ can be used on the MS9710C.

MV02 device messages marked with × cannot be used on the MS9710C. However, since their functions are similar to those of the MS9710C, they are given just for reference purpose.

Device Message Compatibility Table

Function	MS9710C device message				MV02 device message			Compatibility
	Command	Data request	Response	Remarks	Command	Data request	Response	
Wave length								
Center	CNT λ λ=xxxx.xx	CNT?	λ λ=xxxx.xx	λ: Wavelength	CNT λ λ=x.xxxx (μm)	CNT?	λ	× Unit is different.
Span	SPN λ λ=xxxx.x	SPN?	λ λ=xxxx.x	λ: Wavelength (nm)	SPN Δλ Δλ=0,2 to 1000 (1, 2, or 5 steps) 1200	SPN?	Δλ	◎
Start	STA λ λ=xxxx.x	STA?	λ λ=xxxx.x	λ: Wavelength	STA λ λ=x.xxxx (μm)	STA?	λ	× Unit is different.
Stop	STO λ λ=xxxx.x	STO?	λ λ=xxxx.x	λ: Wavelength (nm)	STO Δλ Δλ=0,2 to 1000 (1, 2, or 5 steps) 1200	STO?	λ	× Wavelength to be added to start wavelength
Peak→Center	PKC				PKC			◎
Level Scale		LVS?	S S = LOG LIN	Log scale Linear scale		LVS?	s s = LOG LIN	◎
Log (/div)	LOG l l=xx.x	LOG?	l l=xx.x	Unit: db/div 0.1 to 10.0	LOG n n=0.2 to 10 (1, 2, or 5 steps) 10F	LOG?	n	◎ Except 10F
Ref Level	RLV L l=±xx.x	RLV?	l l=±xx.x	A, B, A&B; l=+30 to –90 A-B, B-A, normalize l=+100 to –100	RLV l A, B, A&B;l=+20 to -90 A-B, B-A, normalize ;l=+40 to -40	RLV?	l	◎
Linear	LLV l	LLV?	l	l: Level 1pW to 1W 1 to 200 %	LLV l	LLV?	l	◎
Peak→Level	PKL				PKL			◎
Resolution	RES n n: Shown on right	RES?	n n=0.07,0.1, 0.2,0.5,1	n: Resolution (nm)	RES n n=0.1,0.2,0.5,1,2,5	RES?	n	◎ Except n = 2, 5
Point Avg	AVT n n=2 to 1000 OFF	AVT?	n n=2 to 1000, OFF	n: Count	AVT n n=1 to 1000	AVT?	n	◎Corresponds to n = 2 to 1000
Smooth	SMT n n: Shown on right	SMT?	n n=3,5,7,9,11, OFF	n: Number of points	SMT n	SMT?	n	◎Corresponds to 3 to 11
Sweep								
Single	SSI				SSI			◎
Repeat	SRT				SRT			◎
Stop	SST				SST			◎
Auto Measure	AUT	AUT?	n n = 0: Measurement complete n = 1: During measurement		AUT	AUT?	n n = 0: Measurement complete n = 1: During measurement	◎
Measure Mode		MOD?	n n = 0: Spectrum not measured n = 1: Spectrum single sweep n = 2: Spectrum repeat sweep n = 3: Power monitor			MOD?	n	◎
Spectrum Go to spectrum measurement	SPC				SPC			◎

Function	MS9710C device message				MV02 device message			Compatibility
	Command	Data request	Response	Remarks	Command	Data request	Response	
Trace Marker	TMK λ λ=xxxx.xxxx	TMK?	λ,l λ=xxxx.xxxx l =xx.xx(dBm) = 4 significant digits (∗W,%)	λ= Wavelength l = Level dBm, ∗W, %	TMK λ λ=x.xxxxxx (μm)	TMK?	λ,l	× Unit of wavelength λis different
Wavelength Marker	MKA λ MKB λ λ=xxxx.xxxx	MKA? MKB?	λ λ λ=xxxx.xxxx	λ: Wavelength (nm)	MKA λ MKB λ λ=x.xxxxxx (μm)	MKA? MKB?	λ λ	× Unit is different.
Level Marker	MKC 1 MKD 1 l: Shown on right	MKC? MKD?	l l l=xx.xxx(dBm) = 7 significant digits (∗W, %)	l = Level dBm, ∗W, %	MKC l MKD l	MKC? MKD?	l l	◎
Marker Off	EMK				EMK			◎
Peak Search	PKS s s=PEAK =NEXT,LAST =LEFT,RIGHT	PKS?	m m=PAEK =NEXT,LAST =LEFT,RIGHT =ERROR	ERR: State other than peak search	PKS			×
Analysis								
Envelope	ANA ENV,r r=1 to 20	ANA?	ENV,r r=1 to 20	r = Cut level (dB)	ANA ENVH (1/2 level) ANA ENVQ (1/4 level)	ANA?	ENVH ENVQ	×
RMS	ANA RMS, r,k r=1 to 30 k=1,2, 2.35,3	ANA?	RMS,r,k r=1 to 30 k=1,2, 2.35,3	r = Slice level (dB) k = Constant of kσ	ANA RMS,r,k r=3,6,10,15,20,25,30 k=1,2,2.35,3	ANA?	RMS,r,k	◎
ndB-Loss	ANA NDB,r r=1 to 50	ANA?	NDB, r r=1 to 50	r = Attenuation (dB)	ANA NDB,r r=1 to 30	ANA?	NDB,r	◎
Threshold	ANA THR,r r=1 to 50	ANA?	THR,r r=1 to 50	r = Cut level (dB)	ANA THRH (1/2 level) ANA THRQ (1/4 level)	ANA?	THRH THRQ	×
SMSR	ANA SMSR,s s=2NDPEAK =LEFT =RIGHT	ANA?	SMSR,s s=2NDPEAK =LEFT =RIGHT		ANA SMSRL(Left) ANA SMSRR(Right)	ANA?	SMSRL SMSRR	×
Analysis Off	ANA OFF	ANA?	OFF		ANA OFF	ANA?	OFF	◎
Analysis Result								
Envelope Threshold RMS		ANAR?	λc, Δλ λc = xxxx.xxx Δλ=xxx.xxx	λc(nm) Δλ(nm)		ANAR?	λc, Δλ λc = x.xxxx (μm) Δλ=xx.xx (nm)	× Unit of λc is different.
ndB-Loss		ANAR?	λc,Δλ,n λc = xxxx.xxx Δλ=xxx.xxx n: Integer	λc(nm) Δλ(nm) n: Number of axis modes		ANAR?	λc, Δλ,n λc = x.xxxx (μm) Δλ=xx.xx(nm) n: Integer	× Unit of λc is different.
SMSR		ANAR?	Δλ,Δl Δλ=xxx.xxx Δl = xx.xx(dB)	Δλ (nm) Δl: Log (dB)		ANAR?	λc, Δ1 λc = x.xxxx (μm) Δ1 = xx.xx	× Unit of λc is different.
Display Mode								
Normal	DMD NRM	DMD?	NRM		DMD NRM	DMD?	NRM	◎
3D	DMD 3,m,n m, n = Shown on right	DMD?	3,m,n m=1,2,3 n=30,45,60,90	m = Type n = Angle	DMD 3,m,n	DMD?	3,m,n	◎
Normalize	DMD NRMZ	DMD?	NRMZ		DMD NRMZ	DMD?	NRMZ	◎
Memory Select	MSL s s=A,B	MSL?	s s=A,B		MSL s s=A,B	MSL?	s	◎
Trace Select	TSL s s=A,B, AB,A_B, B_A	TSL?	s s=A, B, AB,A_B, B_A		TSL s s=A,B,AB,A_B,B_A	TSL?	s	◎
Memory Data	d + Terminator d + Separator Binary	DMA? DMB? DQA? DQB? DBA? DBB?	Log scale ±xxx.xx Linear scale x.xxxxE±x Log: 2-byte data Linear: 4-byte data	Log: Unit(dBm) Linear: Unit (mW) Log: x 0.01 (dBm) Linear: x 0.0001 (mW)		DMA? DMB? DQA? DQB? DBA? DBB?		◎

Function	MS9710C device message				MV02 device message			Compatibility
	Command	Data request	Response	Remarks	Command	Data request	Response	
Power Monitor	PWR λ	PWR?	λ λ=632.8, 850.0, 1300.0, 1550.0	λ= Wavelength (nm)	PWR λ,R,M λ=0.6328,0.85, 1.3,1.55 R = RON or ROFF M = MON or MOFF	PWR?	λ,R,M	×
Power monitor value		PWRR?	P1 P1=±xx.xx	P1 = Power (dBm)		PWRR?	P1, P2 P1 = Power (dBm) P2 = Max. value	×
Title								
Title	TTL character string	TTL?	Character string	30 characters	TTL row, column, character string	TTL?	Character string	×
Title Erase	TER				TER			◎
Time&Date Set								
Date	DATE yy,mm,dd	DATE?	yy,mm,dd		DATE yy,mm,dd	DATE?	yy,mm,dd	◎
Time	TIME hh,mi	TIME?	hh,mi	hh:00 to 23 mi:00 to 59	TIME hh,mi	TIME?	hh,mi	◎
Display Color	LCD P,R,G,B	LCD? P	R,G,B	P: Screen No.0-10 RGB: 0 to 7	CRT P,R,G,B P: 0 to 10, RGB: 0 to 7 each	CRT? P	R,G,B	×
Printor								
Copy	CPY				CPY			◎
Feed	FED n n=0 to 25			n = Number of character lines	FED n n=0 to 25			◎
Terminater	TRM 0 TRM 1	TRM?	0 1	0=LF,EOI 1=CR,LF, EOI	TRM 0 TRM 1	TRM?	0 1	◎
Header	HEAD ON HEAD OFF				HEAD ON HEAD OFF			◎
Error		ERR?	n n=xxx	n = Error No.		ERR?		◎
Extended Event Status Register		ESR1? ESR2? ESR3?	n n n	n = Register value 0 to 255		ESR1? ESR2? ESR3?		◎
Extended Event Status Enable Register	ESE1 n ESE2 n ESE3 n	ESE1? ESE2? ESE3?	n n n	n = Register value 0 to 255	ESE1 n ESE2 n ESE3 n	ESE1? ESE2? ESE3?	n n n	◎
Buzzer On/Off	BUZ s s=ON,OFF	BUZ?	s s=ON,OFF		BUZ s s=ON,OFF	BUZ?	s s=ON,OFF	◎

Appendix F MS9710C and HP Optical Spectrum Analyzer (HP71450A/71451A) Command Compatibility Table

This appendix summarizes compatibility between MS9710C spectrum analyzer device messages and HP71450A/ HP71451A spectrum analyzer device messages.

HP71450A/HP71451A device messages marked with o can be used on the MS9710C.

HP71450A/HP71451A device messages marked with x cannot be used on the MS9710C. However, since their functions are similar to those of the MS9710C, they are given just for reference purpose.

Device Message Compatibility Table

Function	MS9710C device message				HP optical spectrum analyzer device message			Compatibility
	Command	Data request	Response	Remarks	Command	Data request	Response	
Wave length								
Center	CNT λ λ=xxxx.xx	CNT?	λ λ=xxxx.xx	λ:Wavelength (nm)	CENTERWL λ λ=600 to 1750	CENTERWL?	λ	◎
Span	SPN λ λ=xxxx.x	SPN?	λ λ=xxxx.x	λ:Wavelength (nm)	SP Δλ Δλ=0,0.2 to 1200	SP?	Δλ	◎
Start	STA λ λ=xxxx.x	STA?	λ λ=xxxx.x	λ:Wavelength (nm)	STARTWL λ λ=600 to 1750	STARTWL?	λ	◎
Stop	STO λ λ=xxxx.x	STO?	λ λ=xxxx.x	λ:Wavelength (nm)	STOPWL λ λ=600 to 1750	STOPWL?	λ	◎
TMkr→Center	TMC				MKCWL	—		×
Level								
Log(/div)	LOG l l=xx.x	LOG?	l l=xx.x	Unit: dB/div 0.1 to 10.0	LG n n=0.1 to 10	LG?	n	◎
Ref Level	RLV L l=±xx.x	RLV?	l l=±xx.x	A,B,A & B; l=+30 to –90 A-B, B-A, normalize l=+100 to –100	RL l A,B,A & B;l=+30 to –90 A-B, B-A, normalize ;1=+100 to –100	RL?	l	Only in log scale mode
Liner	LLV l	LLV?	l	l = Level (1pW to 1W) 1 to 200 %	LN l	LN?	l	×
Peak→Level	PKL				MKRL	—		×
Resolution	RES n n=Shown in right	RES?	n n=0.07,0.1,0.2,0.5,1	n = Resolution (nm)	RB n n=0.08,0.1,0.2,0.5,1	RB?	n	◎
VBW	VBW s s= Shown in right	VBW?	s s=1 MHz, 100 KHz, 10 KHz, 1 KHz, 100 Hz,10 Hz	s = VBW value Default unit = Hz	VB s s=1 MHz, 300 KHz, 30KHz, 3 KHz, 300 Hz, 30 Hz	VB?	s	◎
Average								
Sweep Avg	AVS n n=2 to 1000 OFF	AVS?	n n=2 to 1000,OFF	n = Number of times	VAVG n n=2 to 1000,ON,OFF	VAVG?	n	◎
Smooth	SMT n n=Shown in right	SMT?	n n=3,5,7,9,11,OFF	n = Number of points	SMOOTH	—		×
Sweep								
Single	SSI				SINGLE,TS			×
Repeat	SRT				SWEEP s(s=ON,OFF),TS			×
Stop	SST				SWEEP s(s=ON,OFF),TS			×
Auto Measure	AUT	AUT?	n n = 0: Measurement complete 1: During measurement		AUTOMEAS	—		◎
Modulation Mode	MDM s s=NORMAL =TRIGGER	MDM?	s s=NORMAL =TRIGGER	Normal EXT trigger	TM s s=FREE,EXT FREE = NORMAL EXT = TRIGGER	—		◎
Spectrum Go to spectrum measurement	SPC				INSTMODE OSA	—		◎
Trace Marker	TMK λ λ=xxxx.xxxx	TMK?	λ,l λ=xxxx.xxxx l=xx.xx(dBm) = 4 significant digits (*W, %)	λ= Wavelength l = Level dBm,*W,%	MKWL λ λ=600 to 1700	MKWL?	λ	◎
ΔMarker	DMK λ λ=xxxx.xxxx	DMK?	Δλ,Δl Δλ=xxxx.xxxx Δl=xx.xx(dB) =xxx.xxx	λ: Wavelength(nm) Δλ: Difference in wavelength (nm) Δl: Difference in level Log: dB Linear: No unit	MKD λ	—		×

Function	MS9710C device message				HP optical spectrum analyzer device message			Compatibility
	Command	Data request	Response	Remarks	Command	Data request	Response	
Wave length Marker	MKA λ MKB λ λ=xxxx.xxxx	MKA? MKB?	λ λ λ=xxxx.xxxx	λ: Wavelength (nm)	MKDREFF λ	MKDREFF?	λ	×
Level Marker	MKC l MKD l l = Shown on right	MKC? MKD?	l l l=xx.xxx(dBm) = 7 significant digits (*W, %)	l = Level dbm,*W,%	MKDREFA l	MKDREFA?	l	×
Marker Off	EMK				MKOFF	—		◎
Peak Search	PKS s s=PEAK =NEXT,LAST =LEFT,RIGHT	PKS?	m m=PEAK =NEXT,LAST =LEFT,RIGHT =ERR	ERR = State other than peak search	MKPK HIP(;high peak) MKPK NL(;left peak) MKPK NR(;right peak)	—		◎
Dip(Pit) Search	DPS s s=DIP =NEXT,LAST =LEFT,RIGHT	DPS?	m m=DIP =NEXT,LAST =LEFT,RIGHT =ERR	ERR = State other than dip search	MKPK s	—		×
Graph Clear	GCL				CLRDSP	—		◎
Memory Select	MSL s s=A,B	MSL?	s s=A,B		CLRW TRA CLRW TRB	—		◎
Trace Select	TSL s s=A,B, AB,A_B, B_A	TSL?	s s=A,B, AB,A_B, B_A		VIEW TRA(;trace A) VIEW TRB(;trace B)	—		◎
Memory Data	d + Terminator	DMA? DMB?	Log scale ±xxx.xx	Log: Unit (dBm)	—	TRA? TRB?		×
	d + Separator	DQA? DQB?	Linear scale x.xxxE±x	Linear: Unit (mW)				
	Binary	DBA? DBB?	Log: 2-byte data Linear: 4-byte data	LOG: to 0.01(dBm) Linear: x 0.0001 (mW)				
Data Condition		DCA? DCB?	λ1,λ2, n λ1=xxxx.xx λ2=xxxx.xx n=251 to 5001	λ1,λ2:: Unit (nm) λ1,: Start wavelength λ2: Stop wavelength n: Number of sampling points				
Application								
DFB-LD Test	AP DFB,s,n s=2NDPEAK =LEFT =RIGHT n=1 to 50	AP?	DFB,s,n s=2NDPEAK =LEFT =RIGHT n=1 to 50	n = n of "ndB Width"	—	DFB_		×
FP-LD Test	AP FP,n n=1 to 50	AP?	FP,n n=1 to 50	n = Axis mode cut level	—	FP_		×
LED Test	AP LED,n,p n=1 to 50 p=–10.0 to +10.0	AP?	LED,n,p n=1 to 50 p=–10.0 to +10.0	n = n of "ndB Width" p=Total power CAL value(dB)	—	LED_		×
Application Result								
DFB-LD Test		APR?	SMSR,ndB-BW, λp,Lp,λSM,LSM, Mode-OFF,Stop-BW, Center-OFF	Wavelength = ****,***nm Level = **,**dBm(dB)	—	DFB_? FP_? LED_?		×
FP-LD Test		APR?	FWHM,λMean, λp,Lp,Mode, Mode-Space,Power					
LED Test		APR?	λFWHM,λ3dB, FWHM,3dBBW,λP,Lp PK-Dens,Power					

Function	MS9710C device message				HP optical spectrum analyzer device message			Compatibility
	Command	Data request	Response	Remarks	Command	Data request	Response	
Wave length Cal								
W-Offset	WOFS n	WOFS?	n n=±x. xx	Offset Wavelength (nm)	WLOFFSET λ	WLOFFSET?		◎
Auto Align	ALIN n n=0:ALIN INITIAL 1:ALIGN 2: Forced end	ALIN?	m m=0: End 1: In process 2: Insufficient optical level 3: Other fault		ALIGNPRST (;alignment resetting) AUTOALIGN (;auto alignment)			◎
Error		ERR?	n n=xxx	n = Error No.	—	ERR?		×
Extended Event Status Register		ESR1? ESR2? ESR3?	n n n	n = Register value 0 to 255	—	STB?		×