2394A/2395A Spectrum Analyzers

Programming Manual

Note
2394A and 2395A use the identical command set to 2394 and 2395 instruments. Hence references in this manual to ‘2394 ‘ and ‘2395’ apply equally to ‘A’ versions.

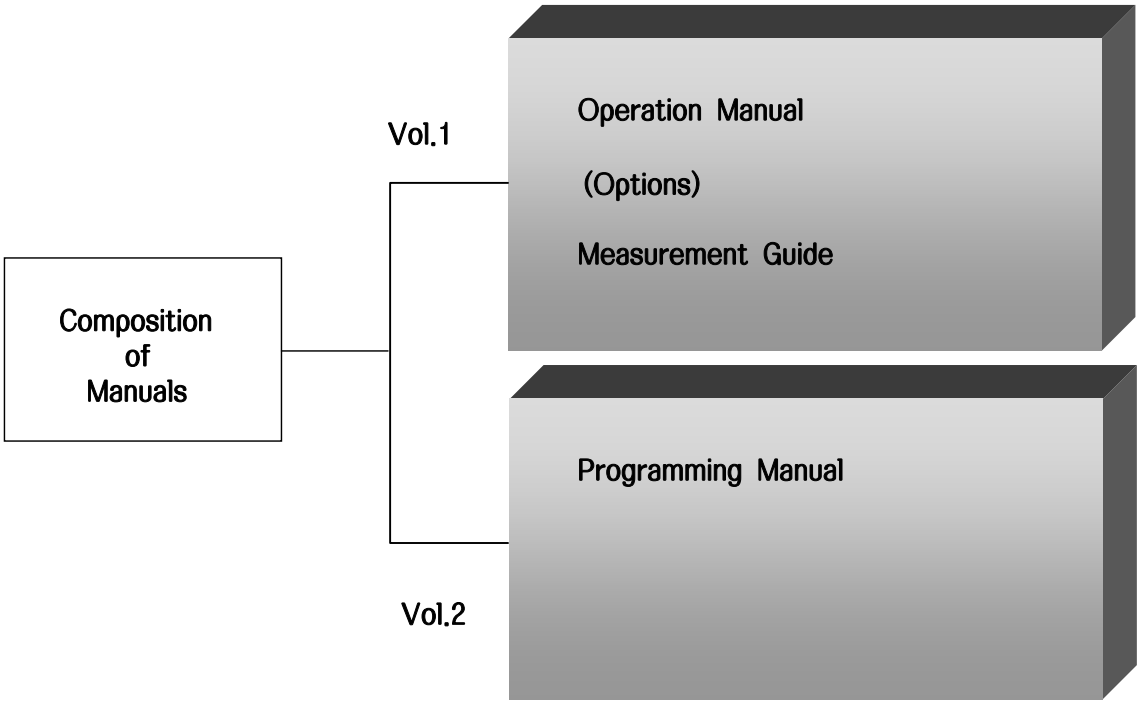
Document part no. 46892/592 (PDF version)

Issue 3
11 June 2007

ABOUT THIS MANUAL

Composition of 2394/2395 Manuals

The 2394/2395 Spectrum Analyzer manuals of the standard type are composed of the following three parts. About Safety and Warranty was referred on Operation **Manual.**



Operation Manual : Provides information on the 2394/2395 outline. Preparation before use, panel description, operation procedure, soft-key menu and performance tests.

Measurement Guide : Provides basic measurements with examples of typical measurements.

Programming Manual : Provides information on RS-232C remote control, GPIB remote control and sample programs.

TABLE OF CONTENTS

<BLANK>

SECTION 1
GENERAL

This section outlines the remote control and gives examples.

TABLE OF CONTENTS

<BLANK>

SECTION 1
GENERAL

General Description

This Spectrum Analyzer, when combined with an external controller (host computer, personal computer, etc.), can automate your measurement system. For this purpose, the spectrum analyzer is equipped with an RS-232C interface port, GPIB interface.

Remote Control Functions

The remote control functions of the equipment are used to do the following:

(1) Control most of functions except the power switch and SYSTEM key.

(2) Read setting value.

(3) Configure the automatic measurement system when the equipment is combined with a personal computer and other measuring equipment.

* Set the RS-232C interface settings from the front panel.
* Set the GPIB address from the front panel.

Interface Port Selection Functions

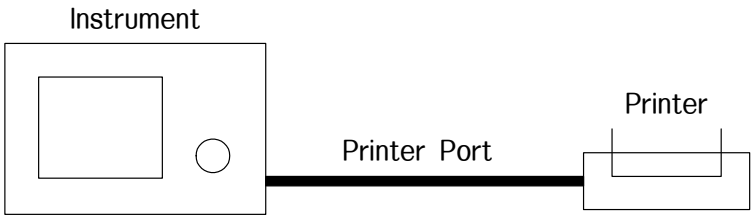
The Spectrum Analyzer has a standard RS-232C interface and a GPIB interface and parallel (printer) interface. Use the panel to select the interface port to be used to connect external devices as shown below.

Port for the external controller : Select RS-232C or GPIB.
Port for the printer : Select parallel port.
Each interface can connect only one device.

Examples of Configurations Using RS-232C and GPIB, Printer

(1) Stand-alone type

Waveforms measured with the equipment is output to the printer.

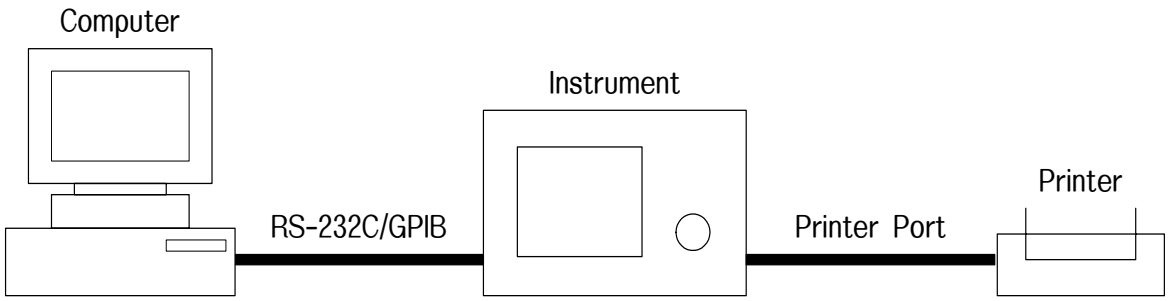


(2) Control by the host computer ①

The equipment is controlled automatically or remotely from the computer.

(3) Control by the host computer ②

The waveforms measured by controlling equipment automatically or remotely are output to the printer. The printer must be connected using printer port.



Specifications of RS-232C

The table below lists the standard specifications of RS-232C in the 2394/2395.

ITEM	SPECIFICATION
Function	Control from the external controller (except for power-ON/OFF, [System] key)
Communication system	Asynchronous (start-stop synchronous System), half-duplex
Communication control system	NONE, XON_XOFF, RTS_CTS, DTR_DSR
Baud rate	110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Data bits	7 or 8 bits
Parity	NONE, ODD, EVEN, MARK, SPACE
Start bit	1 bit
Stop bit (bits)	1 or 2bits
Connector	D-sub 9-pin, female

Specifications of GPIB

The table below lists the specifications with the GPIB provided for the spectrum analyzer.

ITEM	SPECIFICATION AND SUPPLEMENTARY EXPLANATION
Interface function	SH1 : All source handshake functions are provided. Synchronizes the timing of data transmission. AH1 : All acceptor handshake functions are provided. Synchronizes the timing of data reception. T6 : The basic talk functions and serial poll functions are provided. The talk only functions are not provided. The talker can be canceled by MLA. L4 : The basic listener functions are provided. The listen only function is not provided. The listener can be canceled by MTA. SR1 : All service request and status byte functions are provided. RL1 : All remote/local functions are provided. The local lockout function is provided. PP0 : The parallel poll functions are not provided. DC1 : All device clear functions are provided. E2 : Output is tri-state. LE0 : No extended listener capabilities. TE0 : No extended talker capabilities.

SECTION 2
CONNECTING DEVICE

This section describes how to connect external devices such as the host computer, Personal computer, with RS-232C, GPIB cables, This section also describes how to setup the interface of the equipment.

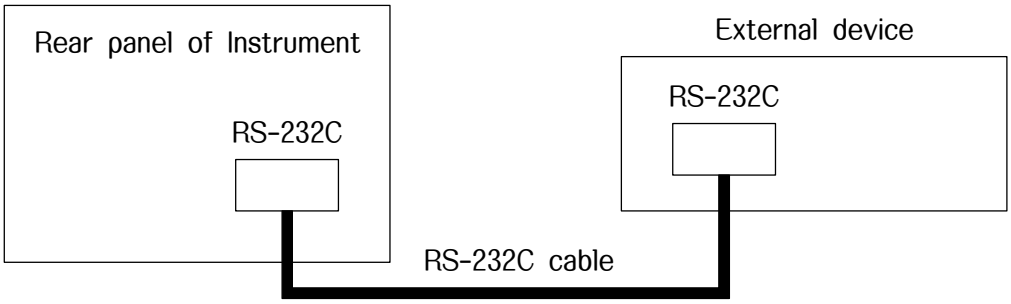
TABLE OF CONTENTS

<BLANK>

SECTION 2
CONNECTING DEVICE

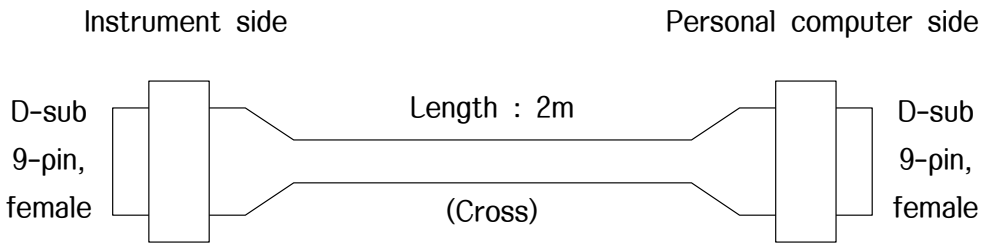
Connection of an External Device with an RS-232C Cable

Connect the RS-232C connector (D-sub 9-pin, male) on the rear panel of the equipment to the RS-232C connector of the external device(Host Controller, Computer) with an RS-232C cable.



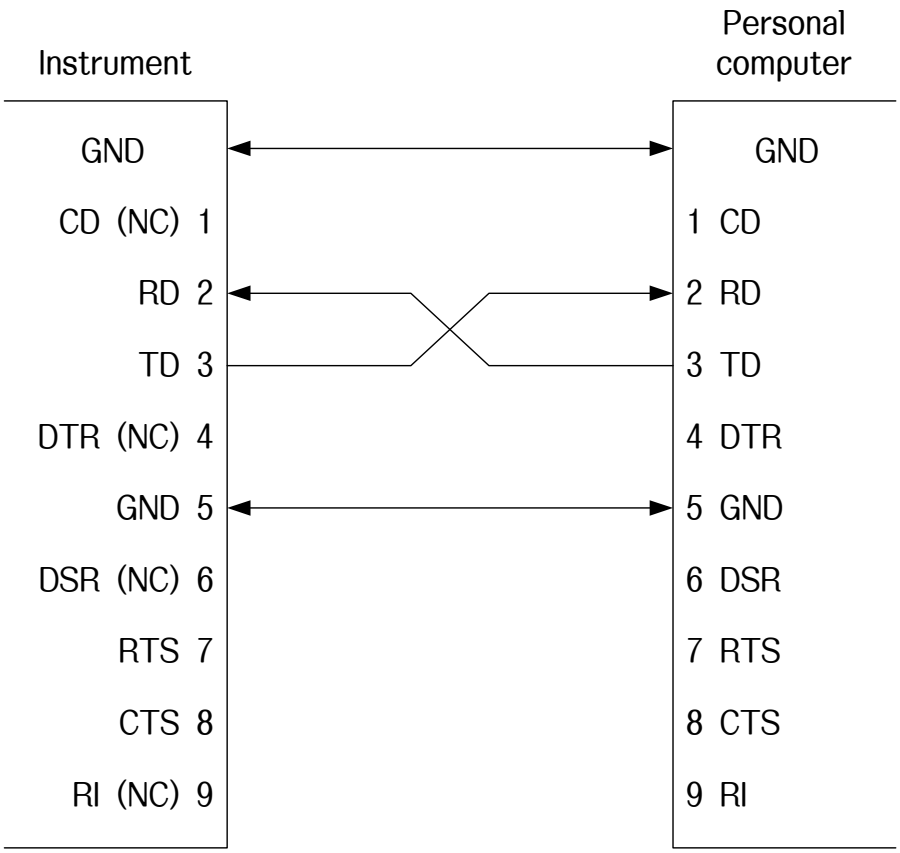
Notes :	RS-232C connectors with 9 pins are available, When purchasing the RS-232C cable, check the pins on the RS-232C connector of the external device. Also, the following RS-232C cables are provided as peripheral parts of the equipment.

● RS-232C cable (for personal computer)



Connection Diagram of RS-232C Interface Signals

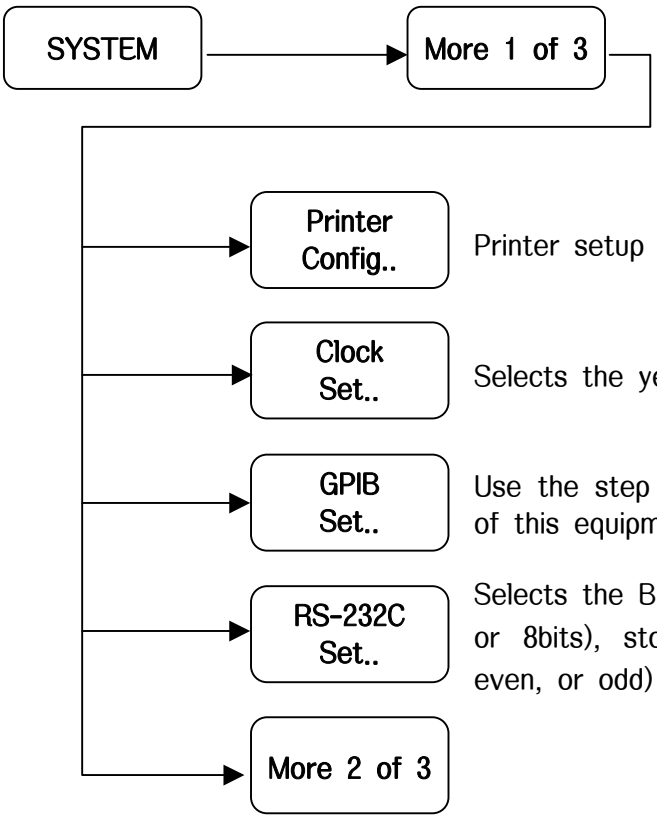
The diagram below shows the RS-232C interface signal connections the between equipment and host system such as a personal computer.



< Connection with personal computer >

Setting the Connection Port interfaces

Set the interfaces between connection ports of the equipment and host system such as a personal computer.



Printer setup menu.

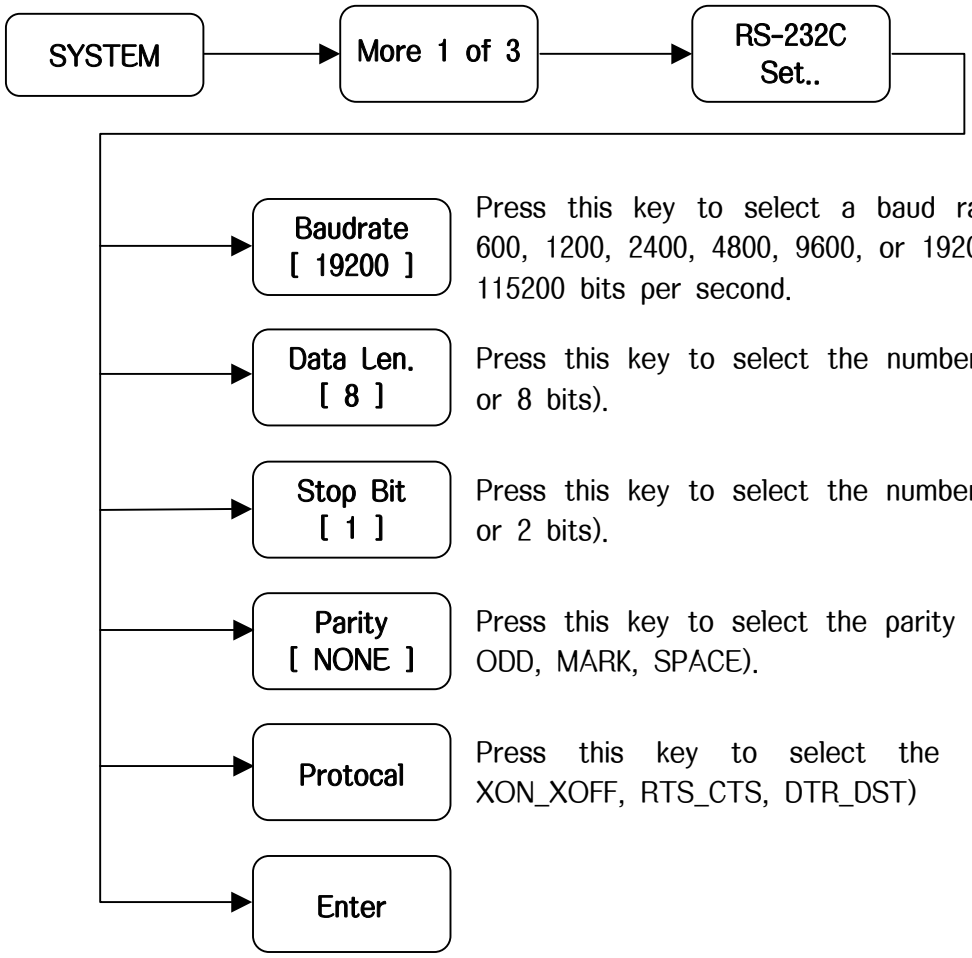
Selects the year, month, day hour, minute, second.

Use the step or knob key to enter the GPIB address of this equipment.

Selects the Baud rate (110 to 115200), data length (7 or 8bits), stop bit (bit 1 or 2), and parity (none, even, or odd) of the RS-232C.

Setting the RS-232C Interface Conditions

Set the RS-232C interfaces conditions of this equipment to those of the external device to be connected.



Press this key to select a baud rate of 110, 300, 600, 1200, 2400, 4800, 9600, or 19200, 38400, 57600, 115200 bits per second.

Press this key to select the number of data bits (7 or 8 bits).

Press this key to select the number of stop bits (1 or 2 bits).

Press this key to select the parity (NONE, EVEN or ODD, MARK, SPACE).

Press this key to select the protocol (NONE, XON_XOFF, RTS_CTS, DTR_DST)

Connection of a Device with a GPIB Cable & Requirements

Connect the GPIB connector on the rear panel of this equipment to the GPIB connector of an external device with a GPIB cable.

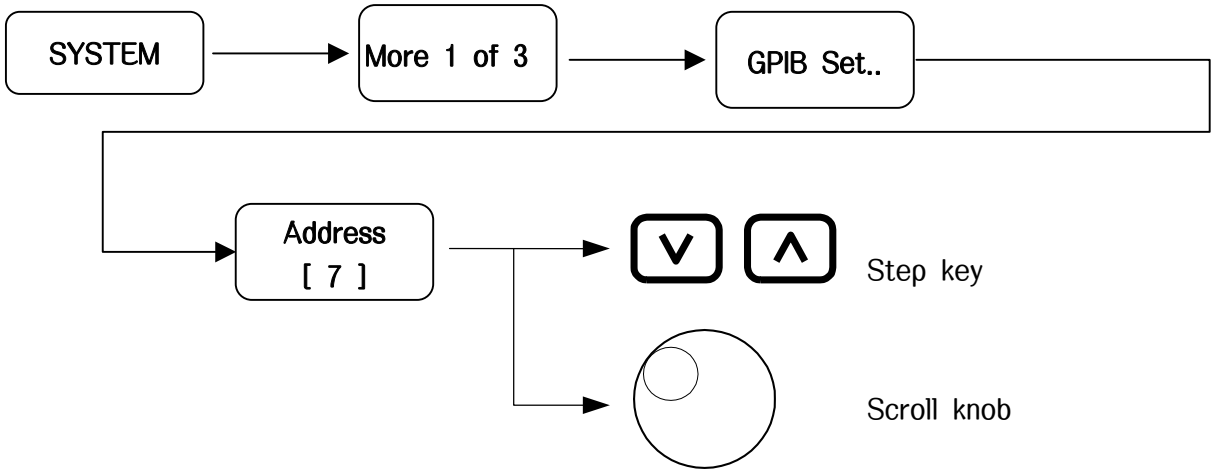
CAUTION	Be sure to connect the GPIB cable before turning the equipment power on.

GPIB Constraints.

1. Number of Interconnected Devices :	15 maximum
2. Interconnection Path Maximum Cable Length :	20 meters maximum or 2 meters per device (whichever is less).
3. Message Transfer Scheme :	Byte serial, bit parallel a synchronous data transfer using a 3-line handshake system.
4. Data Rate :	Maximum of 1 megabyte-per-second over the specified distances with tri-state drivers. Actual data rate depends on the transfer rate of the slowest device connected to the bus.
5. Address Capability :	Primary address : 31 talk, 31 listen. A Maximum of 1 talk and 14 listeners can be connected to the interface at given time.
6. Multiple-controller capability :	In system with more than one controller, only one controller can be active at any given time. The active controller can pass control to another controller, but only the system controller can assume unconditional control. Only one system controller is allowed.

Setting the GPIB Address

Set the GPIB address of this equipment as follows.



Use the step or knob key to enter the GPIB address of this equipment. The initial value is 7.

SECTION 3
DEVICE MESSAGE FORMAT

This section describes the format of the device messages transmitted on the bus between a controller (host computer) and equipment via the RS-232C or GPIB system.

TABLE OF CONTENTS

<BLANK>

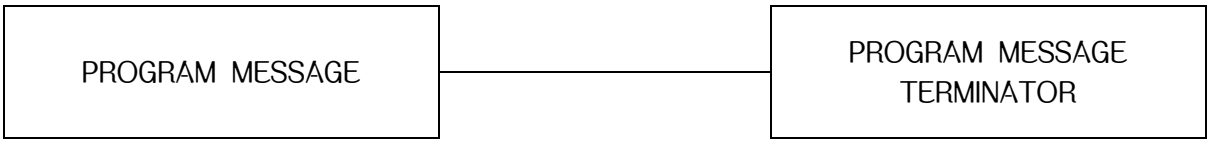
SECTION 3
DEVICE MESSAGE FORMAT

General Description

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

Program Message Format

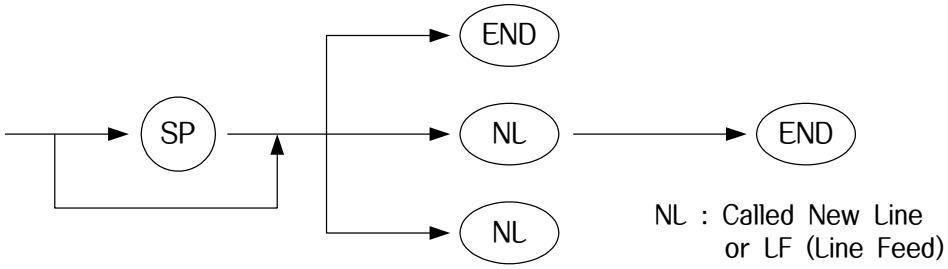
To transfer a program message from the controller program to this equipment using the "Send" statement, the program message formats are defined as follows.



<Example> Send ("CF 1 GHz;") :

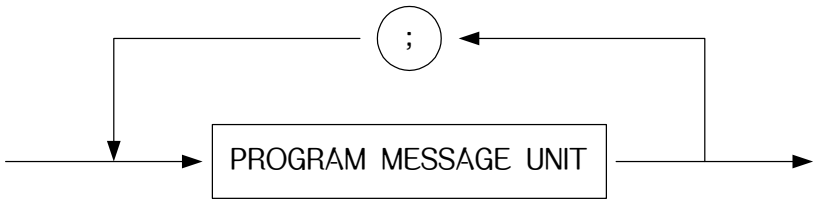
PROGRAM MESSAGE : When the program message is transmitted from the controller to this equipment, the specified terminator is attached to the end of the program message to terminate its transmission.

(1) PROGRAM MESSAGE TERMINATOR



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

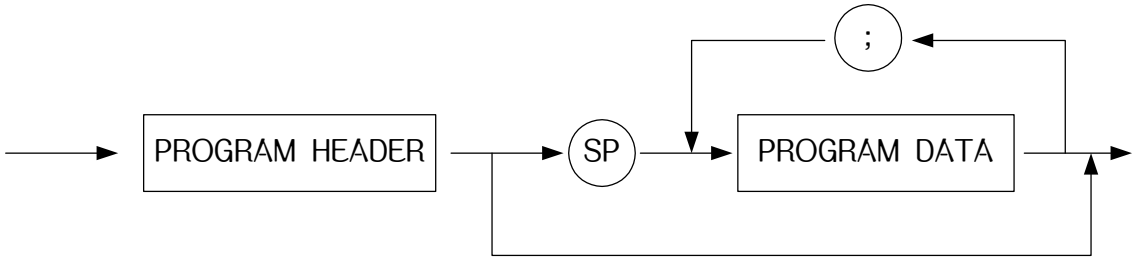
(2) PROGRAM MESSAGE



Multiple program message units can be output sequentially by separation them with a semicolon.

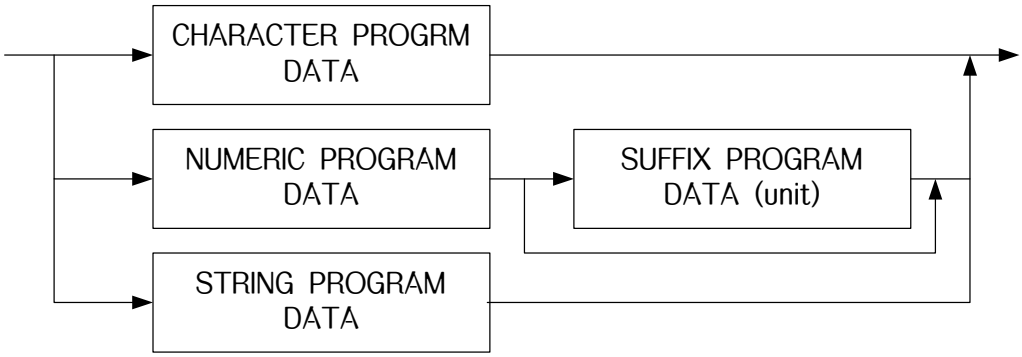
< Example > Send (“CF 1 ㎓; SP 500 ㎒;”)

(3) PROGRAM MESSAGE



Program message consists with program header and program data. The program header of an IEEE488.2 common command always begins with an asterisk. The program header of a program query always ends with a question mark.

(4) PROGRAM MESSAGE



(5) PROGRAM MESSAGE

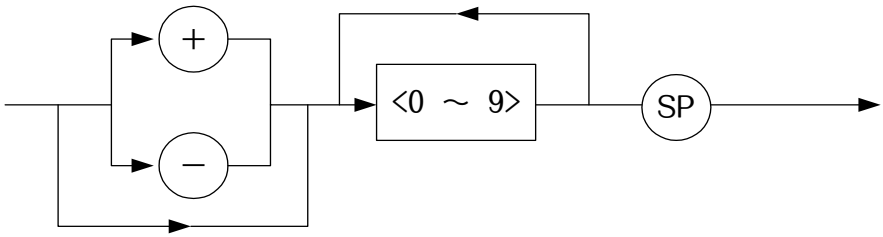
Character of program data is specific character string data consisting of the upper-case alphabetic characters from A to Z, number 0 TO 9, #, *, ?.

< Example > Send (“ST AUTO;”); ..... Sets Sweep Time to AUTO.

(6) NUMERIC PROGRAM DATA

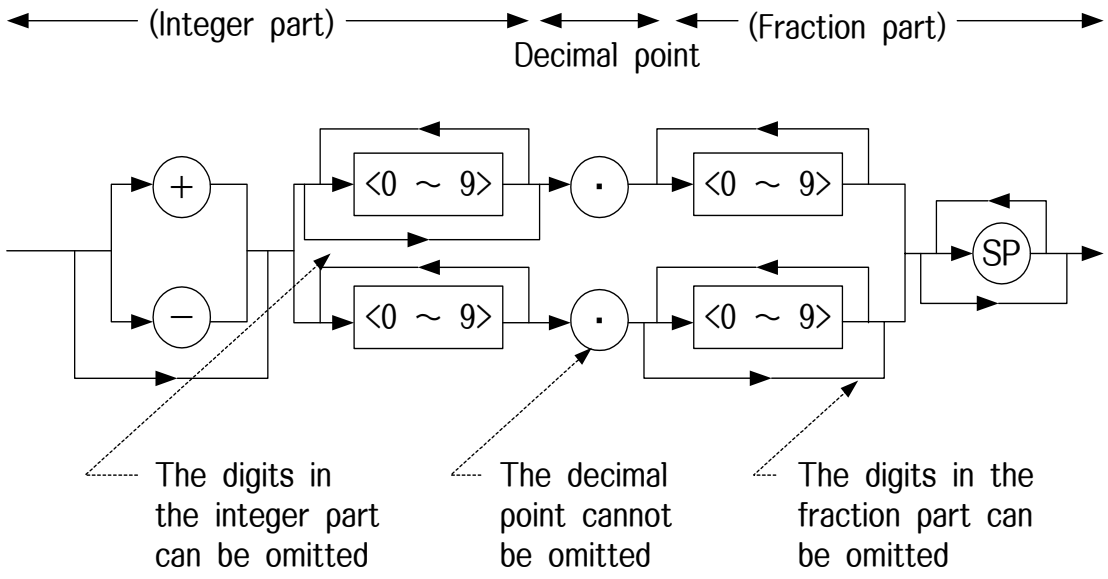
Numeric of program data has two types of formats : integer format (NR1) and fixed-point format (NR2).

< Integer format (NR1) >



- Zeros can be inserted at the beginning → 005, +000045
- There must be no spaces between a + or - sign and a number → +5, +△5 (×)
- Spaces be inserted after a number → +5△△△
- The + sign is optional → +5.5
- Commas cannot be used to separate digits → 1,234,567 (×)

< Fixed-point format (NR2) >



- The numeric expression of the integer format applies to the integer part.
- There must be no spaces between number and the decimal point
  → +753△.123 (×)
- Spaces can be inserted afterthe digits in the fraction part → +753.123△△△
- A number need not be placed before the decimal point → .05
- A + or - sign can be placed before the decimal point → +.05, -.05
- A number can end with a decimal point → 12.

(7) SUFFIX OF PROGRAM DATA (unit)

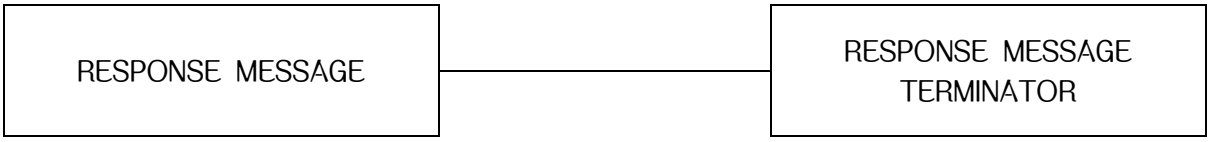
The table below lists the suffixes used for the spectrum analyzer.

Table of Suffix Codes of System

Classification	Unit	Suffix Code
Frequency	GHz	GHz
	MHz	MHz
	kHz	kHz
	Hz	Hz
	Default	Hz
Time	Second	SEC
	m second	MS
	μ second	US
	Default	MS
Level (dB system)	dB	DB
	dBm	DBM
	dBuV	DBUV
	dBmV	DBMV
	Default	Determined in conformance with the set scale unit.
Level (V system)	V	V
	mV	MV
	μV	UV
	Default	Determined in conformance with the set scale unit.
Level (W system)	W	W
	mW	MW
	μW	UW
	nW	NW
	pW	PW
	Default	Determined in conformance with the set scale unit.

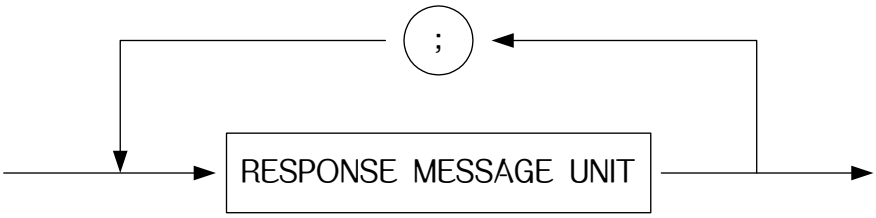
Response Message Format

To transfer the response messages from this equipment to the controller using the "Receive" statement, the response message formats are defined as follows.



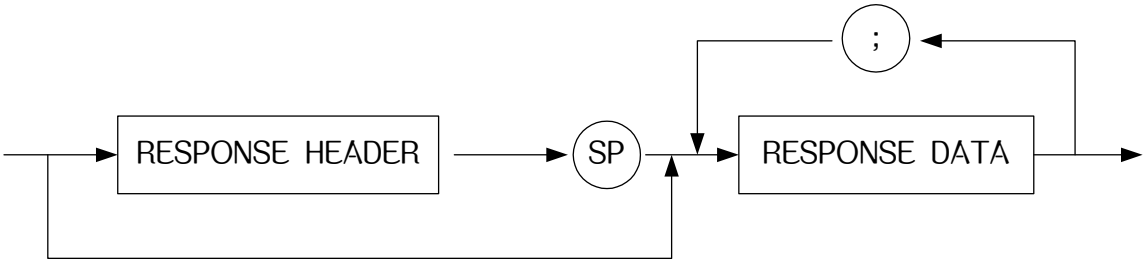
(1) RESPONSE MESSAGE TERMINATOR

(2) RESPONSE MESSAGE

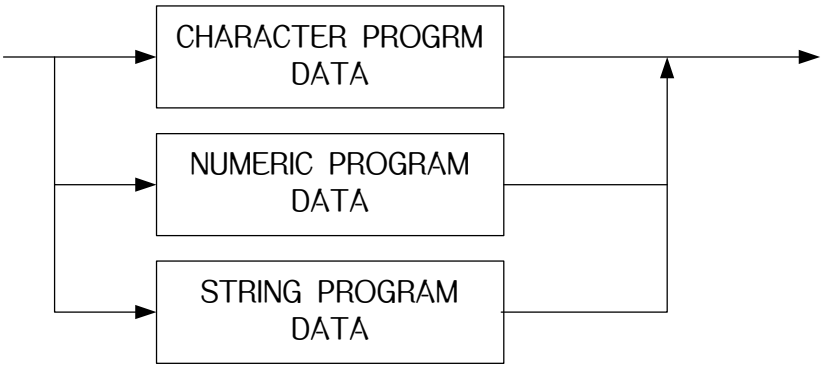


When a query is sent by the "Send" statement with one or more program queries, the response message also consists of one or more response message units.

(3) Usual RESPONSE MESSAGE UNIT



(4) RESPONSE DATA

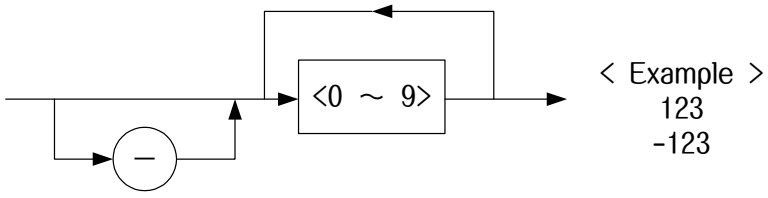


(5) CHARACTER RESPONSE DATA

Character response data is specific character string data consisting of the upper-case alphabetic characters from A to Z, lower-case alphabetic characters from a to z, 0 to 9, and [,], dot[.], minus (−), comma (,).

(6) NUMERIC RESPONSE DATA

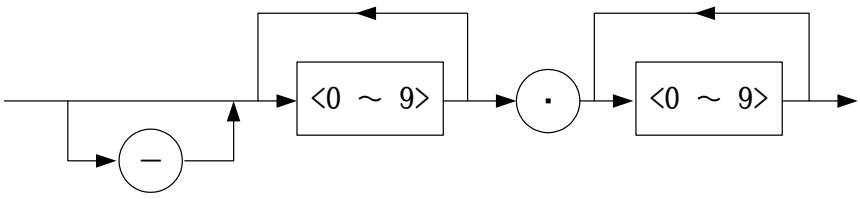
< Integer format (NR1) >



< Example >
123
-123

● The number at the most-significant position is other than 0.
● A +sign need not be placed before a positive number.

< Integer format (NR2) >



● The number at the most-significant position is other than 0.
● A +sign need not be placed before a positive number.

<BLANK>

SECTION 4
DETAILED DESCRIPTION OF COMMANDS

This section describes the usable device and response messages in functional order.

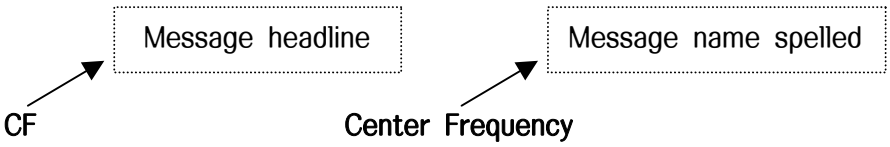
TABLE OF CONTENTS

SECTION 4
DETAILED DESCRIPTION OF COMMANDS

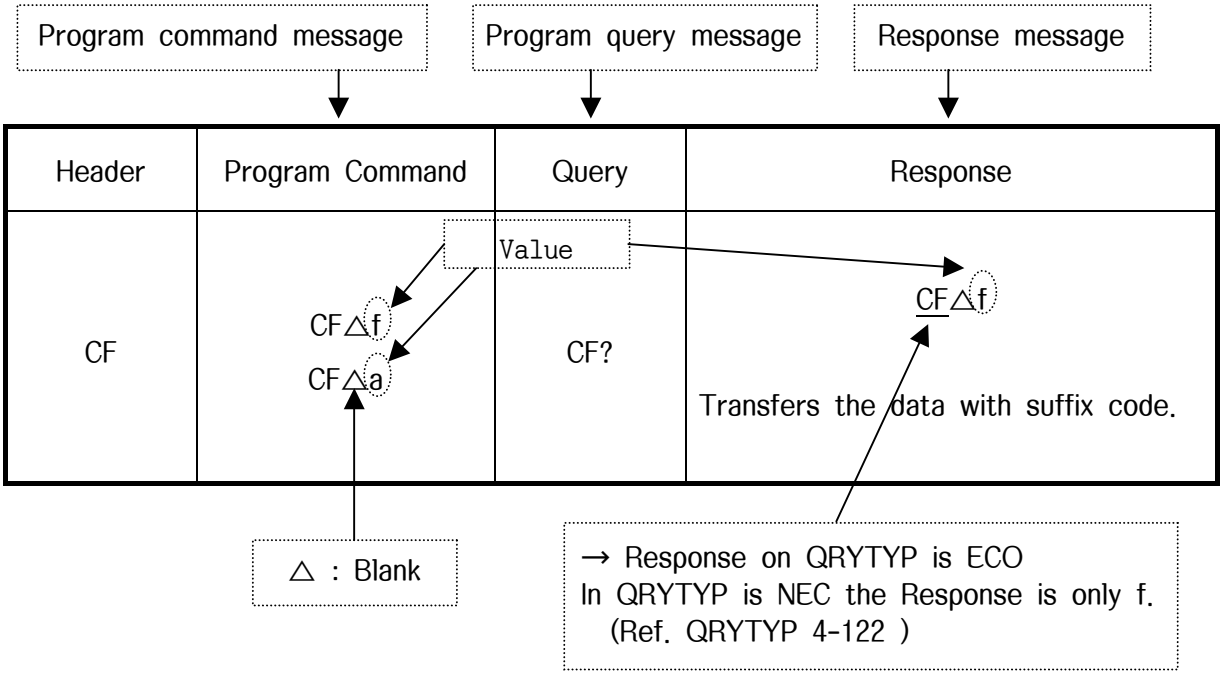
General Description

This section gives detailed descriptions of the device messages for the spectrum analyzer in functional order.

CF



■ Function — Sets the center frequency and sets the spectrum analyzer to center frequency/span mode.



- Value of f — 9.5 ㎑ to 13.2 ㎓ (2394) / 26.5 ㎓ (2395)
- Value of a — OA, UP, DN
- Suffix code — None : ㎐ (10^0)
  HZ : ㎐ (10^0)
  KHZ : ㎑ (10^3)
  MHZ : ㎒ (10^6)
  GHZ : ㎓ (10^9)
- Initial setting — Value of f = 1.5 ㎓
- Example — CF△123456;
  CF△50 MHZ;
  CF?;

; is used to consecutive command
cf) CF 10 ㎒; RL 10dBm; SP 1 ㎒

Frequency

CF

CF **Center Frequency**

■ Function Sets the center frequency and sets the spectrum analyzer to center frequency/span mode.

Header	Program Command	Query	Response
CF	CF△f CF△a	CF?	CF△f

■ Value of f 9 ㎑ to 13.2 ㎓ (2394) / 26.5 ㎓ (2395)

■ Value of a OA : Function Query (same as ?)
UP : Increment size is 1/10 of the current span.
DN : Decrement size is 1/10 of the current span.

■ Suffix code None : ㎐ (10^0)
HZ : ㎐ (10^0)
KHZ : ㎑ (10^3)
MHZ : ㎒ (10^6)
GHZ : ㎓ (10^9)

■ Initial setting Value of f = 6605.0 ㎒

■ Example
```
CF 123456;
CF 50MHZ;
CF?;
```

If the center frequency is set to near the frequency of boundary the span value would not be satisfied. In this time span will be adjusted automatically.

FA

FA **Start Frequency**

■ Function — Sets the start frequency and sets the spectrum analyzer to start frequency/stop frequency mode.

Header	Program Command	Query	Response
FA	FA△f FA△a	FA?	FA△f

■ Value of f — 9 ㎑ to 13.2 ㎓ (2394) / 26.5 ㎓ (2395)
(span value is adjusted automatically)

■ Value of a
- OA : Function Query (same as ?)
- UP : Increment size is 1/10 of the current span.
- DN : Decrement size is 1/10 of the current span.

■ Suffix code
- None : ㎐ (10^0)
- HZ : ㎐ (10^0)
- KHZ : ㎑ (10^3)
- MHZ : ㎒ (10^6)
- GHZ : ㎓ (10^9)

■ Initial setting — Value of f = 10 ㎒

■ Example
```
FA 123456;
FA 50 MHZ;
FA?;
```

FB

FB **Stop Frequency**

■ Function — Sets the stop frequency and sets the spectrum analyzer to start frequency/stop frequency mode.

Header	Program Command	Query	Response
FB	FB△f FB△a	FB?	FB△f

■ Value of f — 9 ㎑ to 13.2 ㎓ (2394) / 26.5 ㎓ (2395)
(Span value is adjusted automatically)

■ Value of a
OA : Function Query (same as ?)
UP : Increment size is 1/10 of the current span.
DN : Decrement size is 1/10 of the current span.

■ Suffix code
None : ㎐ (10^0)
HZ : ㎐ (10^0)
KHZ : ㎑ (10^3)
MHZ : ㎒ (10^6)
GHZ : ㎓ (10^9)

■ Initial setting — Value of f = 13.2 ㎓ (2394) / 26.5 ㎓ (2395)

■ Example
FB 123456;
FB 50 MHZ;
FB?;

SS

SS **Center Frequency Step Size**

■ Function Sets the center frequency step size. Changing the Step Size value is able in Manual Mode.

Header	Program Command	Query	Response
SS	SS△f SS△a	SS?	SS△f (AUTO/MAN)

■ Value of f 9 ㎑ to 13.2 ㎓ (2394) / 26.5 ㎓ (2395)

■ Value of a OA : Function Query (same as ?)
UP : Increment size is 1/10 of the current span.
DN : Decrement size is 1/10 of the current span.
AUTO : 10% of span
MAN : Manual

■ Suffix code None : ㎐ (10^0)
HZ : ㎐ (10^0)
KHZ : ㎑ (10^3)
MHZ : ㎒ (10^6)
GHZ : ㎓ (10^9)

■ Initial setting Value of f = 10% of span

■ Example SS 123456;
SS 50 MHZ;
SS?;

FOFFS

FOFFS **Frequency Offset**

■ Function Set frequency offset value. Changing the offset value is able in FOFFS mode ON.

Header	Program Command	Query	Response
FOFFS	FOFFS△f FOFFS△sw	FOFFS?	FOFFS△f(ON/OFF)

■ Value of f -999 ㎓ to +999 ㎓

■ Value of sw ON : On
OFF : Off

■ Suffix code NONE : ㎐
HZ : ㎐ (10^0)
KHZ : ㎑ (10^3)
MHZ : ㎒ (10^6)
MHZ : ㎓ (10^9)

■ Example FOFFS ON;
FOFFS 3410.7MHZ;

Reference Clock

REFLO

REFLO **Reference Clock**

■ Function Selects the reference clock.

Header	Program Command	Query	Response
REFLO	REFLO△sw	REFLO?	REFLO△sw

■ Value of sw INT : Internal 10 ㎒ clock
EXT : External 10 ㎒ clock

■ Initial setting INT

■ Example REFLO INT;

Auto Tune

AUTOTUNE

AUTOTUNE Auto Tune

■ Function Detects the maximum peak point in full span, and displays its spectrum in the center of the screen, and then changes to a small span width, and the last span width to 1 ㎒.

Header	Program Command	Query	Response
AUTOTUNE	AUTOTUNE	-----	----

■ Example AUTOTUNE;

Span

SP

SP **Frequency Span**

■ Function Sets the frequency span.

Header	Program Command	Query	Response
SP	SP△f SP△a	SP?	SP△f

■ Value of f 0, 100 Hz to 13.2 GHz (2394) / 26.5 GHz (2395)

■ Value of a
- OA : Function Query (same as ?)
- UP : Increment the Parameter. 1, 2, 5, 10 sequence
- DN : Decrement the Parameter. 1, 2, 5, 10 sequence
- FULL : 13.2 GHz (2394) / 26.5 GHz (2395)
- ZERO : 0 Hz
- LAST : Last Span
- ZIN : Previous span / 2
- ZOUT : Previous span × 2

■ Suffix code
- None : Hz (10^0)
- HZ : Hz (10^0)
- KHZ : kHz (10^3)
- MHZ : MHz (10^6)
- GHZ : GHz (10^9)

■ Initial setting Value of f = 13210 MHz

■ Example
```
SP 123456;
SP 50MHZ;
SP?;
```

FS

FS **Full Span**

■ Function Selects the full frequency span.

Header	Program Command	Query	Response
FS	FS	----	----

■ Example FS;

ZS

ZS **Zero Span**

■ Function Sets zero frequency span.

Header	Program Command	Query	Response
ZS	ZS	----	----

■ Example ZS;

ZI

ZI **Zoom-In**

■ Function Changes to 1/2 the previous span.

Header	Program Command	Query	Response
ZI	ZI	----	----

Example ZI;

ZO

ZO **Zoom-Out**

■ Function Changes to two times the previous span. Span varies in the range that allows holding the center frequency.

Header	Program Command	Query	Response
ZO	ZO	----	----

Example ZO;

Amplitude

RL

RL **Reference Level**

■ Function Sets the reference level.

Header	Program Command	Query	Response
RL	RL△ ℓ RL△a	RL?	RL△ ℓ

■ Value of ℓ Value from -110dBm to +30dBm (0.1dBm step)

■ Value of a OA : Function Query (same as ?)
UP : Increment the Parameter. 1 division level
DN : Decrement the Parameter. 1 division level

■ Suffix code None : dBm
DBM : dBm
DBMV : dB mV
DBUV : dB μV
V : V
MV : mV (10^-3 V)
UV : μV (10^-6 V)
W : W
MW : mW (10^-3 W)
UW : μW (10^-6 W)
NW : nW (10^-9 W)
PW : pW (10^-12 W)
FW : fW (10^-15 W)

■ Initial setting ℓ = -10 dBm

■ Example
```
RL 30 DBM;
RL UP;
```

AT

AT **Input Attenuation**

■ Function — Sets the amount of attenuation for the input attenuator. It possible in manual mode.

Header	Program Command	Query	Response
AT	AT△a AT△n	AT?	AT△n dB (AUTO/MAN)

■ Value of n — 0 to 55 (12 step) : 0 to 55 dB (5dB step)

■ Value of a
- AUTO : Auto
- MAN : Manual
- UP : Increment the Parameter. 10dB step.
- DN : Decrement the Parameter. 10dB step.

■ Suffix code
- None : dB
- DB : dB

■ Initial setting — n = Calculated value when AUTO is selected for AT

■ Example
```
AT 10DB;
AT?;
```

LG

LG **Logarithm Amplitude Mode & Scale**

■ Function — Selects 1, 2, 5 or 10 dB logarithmic amplitude mode & scale. When not in LOG mode, querying 'LG?' return a zero.

Header	Program Command	Query	Response
LG	LG△ ℓ	LG?	LG△ ℓ

■ Value of ℓ — 1, 2, 5, 10 : dB/div
NONE : 10dB/div

■ Suffix code — NONE : dB/div
DB : dB/div

■ Initial setting — ℓ = 10 dB/div

■ Example — LG 5; LG?;

LN

LN **Linear Mode**

■ Function — Selects linear amplitude mode. When not in linear mode, querying 'LN?' returns a zero.

Header	Program Command	Query	Response
LN	LN	LN?	LN△sw

■ Value of sw — 0 : Not Linear Mode
1 : Linear Mode

■ Suffix code — None

■ Example — LN;LN?;

AUNITS

AUNITS **Absolute Amplitude Units**

■ Function Sets the absolute amplitude units for the input signal display.

Header	Program Command	Query	Response
AUNITS	AUNITS△a	AUNITS?	AUNITS△u

■ Value of a
- DBM : dBm
- DBMV : dB mV
- DBUV : dB μV
- V : Volt
- W : Watt

■ Value of u
- DBM : dBm
- DBMV : dB mV
- DBUV : dB μV
- VOLT : Volt
- WATT : Watt

■ Suffix code None

■ Initial setting DBM : dBm

■ Example AUNITS DBM;

INPUTZ

INPUTZ **Input Impedance**

■ Function Change the input impedance. This just using the calculation value do not considering of signal reflection.

Header	Program Command	Query	Response
INPUTZ	INPUTZ△sw	INPUTZ?	INPUTZ△sw

■ Value of sw 50 : 50 Ω
75 : 75 Ω

■ Suffix code NONE

■ Example INPUTZ 75;

RLO

RLO **Level Offset**

■ Function Set amplitude level offset value.

Header	Program Command	Query	Response
RLO	RLO△a RLO△sw	RLO?	RLO△a(ON/OFF)

■ Value of a -217.6 dB to +297.6 dB

■ Value of sw ON : On
OFF : Off

■ Suffix code NONE : dB
DB : dB

■ Example RLO ON;
RLO –200.5dB;

INTAMP

INTAMP **Internal Amplifier**

■ Function Activate the internal amplifier if installed.

Header	Program Command	Query	Response
INTAMP	INTAMP△sw	INTAMP?	INTAMP△sw

■ Value of sw ON : Activate
OFF : Deactivate

■ Example INTAMP ON;

CALSIG

CALSIG **Calibration Signal**

■ Function Turns the calibration signal(20 MHz, -20 dBm) on or off.
Connect the calibration signal output connector(BNC) and RF INPUT connector(N-type).

Header	Program Command	Query	Response
CALSIG	CALSIG△sw	CALSIG?	CALSIG△rsw

■ Value of sw 1, ON : On
0, OFF : Off

■ Value of rsw 0, 1

■ Initial setting 0

■ Example CALSIG ON;

Marker

SELMK

SELMK **Select Marker Number**

■ Function Select specific marker number for activation.

Header	Program Command	Query	Response
SELMK	SELMK△n	SELMK?	SELMK△n

■ Value of n 1 to 9
■ Suffix code NONE
■ Example SELMK 3;

NMKR

NMKR **New Marker Number**

■ Function Make new specific marker.

Header	Program Command	Query	Response
NMKR	NMKR△n	----	----

■ Value of n 1 to 9
■ Suffix code NONE
■ Example NMKR 2;

DMKR

DMKR **Delete Marker Number**

■ Function Delete specific marker.

Header	Program Command	Query	Response
DMKR	DMKR△n	----	----

■ Value of n 1 to 9
■ Suffix code NONE
■ Example DMKR 4;

SELMRO

SELMRO **Select Marker Read Out**

■ Function Define marker read out type in marker mode.

Header	Program Command	Query	Response
SELMRO	SELMRO△sw	SELMRO?	SELMRO△sw

■ Value of sw FREQ : Frequency
PRID : Period = 1/Frequqncy
TIME : Time
ITIME : Inverse Time = 1/Time
■ Example SELMRO PRID;

MKN

MKN **Normal Marker**

■ Function — Places an active marker on the specified frequency. If no frequency is specified, MKN places the marker at the center of trace. When zero span mode, the marker is set at the specified time.

Header	Program Command	Query	Response
MKN	MKN△f MKN△a MKN△t	MKN?	MKN△f MKN△t (time resolution : sweep time/500)

■ Value of f — 0 Hz to 13.2 GHz (2394) / 26.5 GHz (2395) : within the span

■ Value of t — 0 to 15 sec : within the sweep time

■ Value of a
- OA : Function Query (same as ?)
- UP : Increment the Parameter. 10% of span
- DN : Decrement the Parameter. 10% of span
- None : When normal marker is not specified, put the normal Marker on the center on grid.

■ Suffix code f
- None : Hz (10^0), In sweep mode
- HZ : Hz (10^0)
- KHZ : kHz (10^3)
- MHZ : MHz (10^6)
- GHZ : GHz (10^9)

■ Suffix code t
- None : ms (10^-3), In sweep mode
- US : μs (10^-6)
- MS : ms (10^-3)
- SEC : sec (10^0)

■ Initial setting — OFF

■ Example
```
MKN?;
MKN 100MHZ;
```

MKA

MKA **Marker Amplitude**

■ Function Returns on the amplitude data in marker mode.

Header	Program Command	Query	Response
MKA	----	MKA?	MKA△ ℓ MKA△v MKA△w MKA△f MKA△ρ

■ Value of ℓ When display unit system for marker level is dB.
■ Value of v When display unit system for marker level is V.
■ Value of w When display unit system for marker level is W.
■ Value of f For FM Demodulation, ㎑
■ Value of ρ For AM Demodulation, %
■ Example MKA?;

MKD

MKD **Marker Delta**

■ Function Places delta marker on the normal marker position. If it has some value marker places on the relative position of reference.

Header	Program Command	Query	Response
MKD	MKD	----	----

■ Value of f 0 Hz ~ 13.2 GHz (2394) / 26.5 GHz (2395) : inner span range

■ Value of r ON, OFF

■ Example MKD 154KHZ;
MKD?;

MKDTF

MKDTF **Marker 1/Delta**

■ Function Calculates 1/delta in the zero span mode or sweep mode.
The normal & delta marker must be on to work.
The only way to turn MKDTF off is to turn the marker off (MKOFF).

Header	Program Command	Query	Response
MKDTF	MKDTF	----	----

■ Example MKDTF;

MKTF

MKTF **Read the marker frequency or time**

■ Function Returns time or frequency of a marker.

Header	Program Command	Query	Response
MKTF	----	MKTF?	MKTF△f (frequency) MKTF△t (time)

■ Example MKTF?;

MKOFF

MKOFF **Marker Off**

■ Function Turns off the marker mode.

Header	Program Command	Query	Response
MKOFF	MKOFF	----	----

■ Example MKOFF;

Marker Noise

MKNOISE

MKNOISE **Marker Noise**

■ Function Sets the detector mode to sample and computes level at the current marker frequency position.

Header	Program Command	Query	Response
MKNOISE	MKNOISE△sw	MKNOISE?	MKNOISE△rsw

■ Value of sw ON : On
OFF : Off

■ Value of rsw OFF : Off
Result Value and Suffix code is dBc/Hz, MKA? Also response.

■ Suffix code None

■ Initial setting OFF

■ Example MKNOISE ON;

Phase Noise

MKPN

MKPN **Phase Noise**

■ Function Calculates carrier to noise value in the position of marker. For the result use MKA?.

Header	Program Command	Query	Response
MKPN	MKPN△sw	MKPN?	MKPN△rsw

■ Value of sw ON, 1 : On
OFF, 0 : Off

■ Value of rsw 0, 1

■ Suffix code None

■ Initial setting 0

■ Example MKPN ON;

MKPNO

MKPNO **Phase Noise Offset**

■ Function Set the offset of phase noise.

Header	Program Command	Query	Response
MKPNO	MKPNO△f	MKPNO?	MKPNO△f

■ Value of f 10 ㎐ to 100 ㎑ : phase noise offset
■ Suffix code None : ㎐ (10^0)
HZ : ㎐ (10^0)
KHZ : ㎑ (10^3)
■ Initial setting 10 ㎑
■ Example MKPNO 5KHZ;

Frequency Counter

MKFC

MKFC **Frequency Counter**

■ Function Activates a frequency counter that counts the frequency of the normal marker.

Header	Program Command	Query	Response
MKFC	MKFC△sw	MKFC?	MKFC△rsw

■ Value of sw ON : On
OFF : Off

■ Value of rsw OFF
0.01 ~ 13.2 ㎓ (2394) / 26.5 ㎓ (2395)

■ Suffix code None

■ Initial setting OFF

■ Example MKFC ON;

MKFCR

MKFCR **Frequency Counter Resolution**

■ Function Specifies the resolution of the frequency counter.

Header	Program Command	Query	Response
MKFCR	MKFCR△f	MKFCR?	MKFCR△f

■ Value of f 1, 10, 100, 1000 : (Hz)
■ Suffix code None
■ Initial setting 1000
■ Example MKFCR 1000;

Quasi Peak (Option)

MKFQP

MKFQP **Maker Quasi Peak**

■ Function Measure Quasi Peak in the place of active marker.
Use the MKTF for frequency and MKA for amplitude value.

Header	Program Command	Query	Response
MKFQP	MKFQP△sb	MKFQP?	MKFQP△sb

■ Value of sb ON
OFF : Release Quasi Peak

■ Initial setting OFF

■ Example MKFQP ON;

SELMKQP

SELMKQP **Select band Maker Quasi Peak**

■ Function Select Quasi Peak measuring band.

Header	Program Command	Query	Response
SELMKQP	SELMKQP △sb	SELMKQP?	SELMKQP △sb

■ Value of sb BNDB : RBW 9 ㎑
BNDC : RBW 120 ㎑

■ Initial setting BNDB

■ Example SELMKQP BNDB;

Marker shift

MKCF

MKCF **Marker to Center Frequency**

■ Function Sets the center frequency to the frequency value of the normal marker. The normal marker must be active to work.

Header	Program Command	Query	Response
MKCF	MKCF	----	-----

■ Example MKCF;

MKSS

MKSS **Marker to Center Frequency Step Size**

■ Function Sets the center frequency step-size equal to the frequency value of the active marker.
The normal marker must be active to work.

Header	Program Command	Query	Response
MKSS	MKSS	----	-----

■ Example MKSS;

MKFA

MKFA **Marker to Start Frequency**

■ Function Set the current active marker frequency to start frequency.

Header	Program Command	Query	Response
MKFA	MKFA	----	-----

■ Example MKFA;

MKFB

MKFB **Marker to Stop Frequency**

■ Function Set the current active marker frequency to stop frequency.

Header	Program Command	Query	Response
MKFB	MKFB	----	-----

■ Example MKFB;

MKRL

MKRL **Marker to Reference Level**

■ Function Sets the reference level to the amplitude of the normal marker. The normal marker must be active to work.

Header	Program Command	Query	Response
MKRL	MKRL	----	-----

■ Example MKRL;

MKSP

MKSP **Marker Delta to Span**

■ Function Sets the frequency span equal to the frequency difference between two markers on a trace.
If normal & delta marker is not active, MKSP cannot work.

Header	Program Command	Query	Response
MKSP	MKSP	----	-----

■ Example MKSP;

MKDSS

MKDSS **Marker Delta to Center Frequency Step Size**

■ Function Set the current marker Delta frequency to Center Frequency Step Size. The delta marker must be active to work.

Header	Program Command	Query	Response
MKDSS	MKDSS	----	-----

■ Example MKDSS;

MKZI

MKZI **Marker Zoom-In**

■ Function Sets the center frequency to the frequency value of an active marker and the frequency span changes to 1/2 the previous span. The normal marker must be active to work.

Header	Program Command	Query	Response
MKZI	MKZI	----	-----

■ Example MKZI;

MKZO

MKZO **Marker Zoom-Out**

■ Function Sets the center frequency to the frequency value of an active marker and the frequency span changes to the two times the previous span. The normal marker must be active to work.

Header	Program Command	Query	Response
MKZO	MKZO	----	-----

■ Example MKZO;

Peak

MKPK

MKPK **Peak Search**

■ Function Places a marker on the highest point on the trace, the next highest Point, the next-left peak, the next-right peak.

Header	Program Command	Query	Response
MKPK	MKPK△a	----	----

■ Value of a
HI : Finds the highest point on the trace
NH : Finds the next-highest point on the trace.
NR : Finds the next-right peak.
NL : Finds the next-left peak
None : Finds the highest point on the trace.

■ Example MKPK HI;

MKMIN

MKMIN **Marker Minimum Search**

■ Function Place the marker in the minimum level point of signal.

Header	Program Command	Query	Response
MKMIN	MKMIN	----	-----

■ Example MKMIN;

MKPP

MKPP **Marker Peak to Peak Search**

■ Function Ref marker positioned minimum level and active marker positioned maximum level.

Header	Program Command	Query	Response
MKPP	MKPP	----	-----

■ Example MKPP;

MKTRACK

MKTRACK **Signal Track**

■ Function Locates the active marker and sets the center frequency to the marker value. This is done after sweep, thus maintaining the marker value at the center frequency.

Header	Program Command	Query	Response
MKTRACK	MKTRACK△sw	MKTRACK?	MKTRACK△rsw rsw = 0, 1

■ Value of sw 1, ON : On
0, OFF : Off

■ Suffix code None

■ Initial setting 0, OFF

■ Example MKTRACK ON;

MKPX

MKPX **Marker Peak Search Excursion**

■ Function For peak search, set the peak least amplitude.
It is valid when MKPS is set to MANL. (ref. 4-43 MKPS)

Header	Program Command	Query	Response
MKPX	MKPX△d	MKPX?	MKPX△rd

■ Value of d 0 ~ n : dB (range : 0.03 ~ 100 dB)
■ Value of rd 0 ~ n dB
■ Suffix code none
■ Initial setting 3
■ Example MKPX 6; MKPX?;

MKPT

MKPT **Marker Peak Search Threshold**

■ Function Set the low limit line for peak search.
It is valid when MKPS is set to MANL. (ref. 4-43 MKPS)

Header	Program Command	Query	Response
MKPT	MKPT△d	MKPT?	MKPT△rd

■ Value of d n : dB (range : Ref level ~ -150 dB)
■ Value of rd n dB
■ Unit code None
■ Initial setting -100
■ Example MKPT -80; MKPT?;

MKPS

MKPS **Marker Peak Search Parameter**

■ Function Enable to use user define peak search parameter.

Header	Program Command	Query	Response
MKPS	MKPS△sw	MKPS?	MKPS△sw

■ Value of sw DFLT, MANL
■ Suffix code None
■ Initial setting DFLT
■ Example MKPS MANL;

MMPN

MMPN **Marker Multi Peak Number**

■ Function set the multi peak number.

Header	Program Command	Query	Response
MMPN	MMPN△d	MMPN?	MMPN△d

■ Value of d 1 ~ 9
■ Suffix code none
■ Initial setting 9
■ Example MMPN 5; MMPN?;

MMP

MMP **Marker Multi Peak**

■ Function Search Multi Peak and place each marker.

Header	Program Command	Query	Response
MMP	MMP		

■ Example MMP;

Trigger

TRGSWP

TRGSWP **Trigger Sweep**

■ Function Selects the continuous-sweep mode or the single-sweep mode.

Header	Program Command	Query	Response
TRGSWP	TRGSWP△sw	TRGSWP?	TRGSWP△rsw

■ Value of sw 0, CNT : Continuous-sweep Mode
1, SNG : Single-sweep Mode

■ Value of rsw 0 : Continuous-sweep Mode
1 : Single-sweep Mode

■ Suffix code None

■ Initial setting 0 : Continuous-sweep Mode

■ Example TRGSWP 0;

TM

TM **Trigger Source**

■ Function Sets the trigger switch and trigger source.

Header	Program Command	Query	Response
TM	TM△sw	TM?	TMP△sw

■ Value of sw FREE : Selects the free-run mode.
VID : Selects the video mode.
LINE : Selects the line mode.
EXT : Selects the external mode

■ Suffix code None

■ Initial setting FREE

■ Example TM FREE;
TM VID;

TLV

TLV **Trigger Level**

■ Function — Sets the threshold level of sweep the start trig when the trigger source is video. Sweep trigger level x is vertical position on graticule and ranges form 0 to 360 (0 is Bottom).

Header	Program Command	Query	Response
TLV	TLV△x	TLV?	TLV△x

■ Value of x — 0 to 360
■ Suffix code — None
■ Initial setting — 0
■ Example — TLV 100;
TLV?;

TDLY

TDLY **Delay Time**

■ Function Sets the delay time from point where trace time triggering occurs. Available only zero span mode.

Header	Program Command	Query	Response
TDLY	TDLY△t	TDLY?	TDLY△t

■ Value of t -Sweep Time < t < Sweep Time
Resolution : Sweep Time / 500

■ Suffix code None : ㎳
US : ㎲
MS : ㎳
SEC : sec

■ Initial setting 0

■ Example TDLY 50 ㎳;
TDLY?;

TE

TE **Trigger Edge**

■ Function Select Trigger edge type.

Header	Program Command	Query	Response
TE	TE△e	TE?	TE△e

■ Value of e FALL
RISE

■ Suffix code None

■ Initial setting FALL

■ Example TE RISE;
TE ?;

Time Gate

GATE

GATE Time Gate

■ Function Activate time gate function.

Header	Program Command	Query	Response
GATE	GATE△sw	GATE?	GATE△rsw

■ Value of sw OFF, 0
ON, 1
■ Value of rsw 0, 1
■ Suffix code none
■ Initial setting 0
■ Example GATE ON;
GATE ?;

GATECTL

GATECTL **Time Gate Control**

■ Function Set the control method of time gate.

Header	Program Command	Query	Response
GATECTL	GATECTL△sw	GATECTL?	GATECTL△sw

■ Value of sw EDGE : controlled by edge
LEVEL : controlled by level

■ Suffix code none

■ Initial setting LEVEL

■ Example GATECTL EDGE;
GATECTL ?;

GD

GD **Time Gate Delay**

■ Function set the delay time before open time gate.

Header	Program Command	Query	Response
GD	GD△t	GD?	GD△t

■ Value of t 2 us ~ 65.5 ms

■ Suffix code none : ms
US : μs
MS : ms

■ Initial setting 15 ms

■ Example GD 30MS;
GD ?;

GL

GL **Time Gate Length**

■ Function Set the opened time of time gate.

Header	Program Command	Query	Response
GL	GL△t	GL?	GL△t

■ Value of t 2 us ~ 65.5 ms
■ Suffix code none : ms
US : μs
MS : ms
■ Initial setting 20 ms
■ Example GL 5MS;
GL ?;

GP

GP **Time Gate Polarity**

■ Function Set the time gate edge type.

Header	Program Command	Query	Response
GP	GP△ρ	GP?	GP△ρ

■ Value of ρ NEG
POS

■ Initial setting NEG

■ Example GP POS;
GP ?;

Coupling

AUTOCPL

AUTOCPL **Auto Coupled**

■ Function Sets the resolution bandwidth, the video bandwidth, the input attenuator, and the sweep time in AUTO mode.

Header	Program Command	Query	Response
AUTOCPL	AUTOCPL	----	----

■ Example AUTOCPL;

RB

RB **Resolution Bandwidth**

■ Function Sets the resolution bandwidth.

Header	Program Command	Query	Response
RB	RB△f RB△a	RB?	RB△f (AUTO/MAN)

■ Value of f 300 Hz to 3 MHz (1, 3, 10 sequence)

■ Value of a
- UP : Increments in a 1, 3, 10 sequence.
- DN : Decrements in a 1, 3, 10 sequence.
- OA : Function Query (same as ?)
- AUTO : RBW Auto coupling
- MAN : RBW Manual coupling

■ Suffix code f
- None : Hz (10^0)
- HZ : Hz (10^0)
- KHZ : kHz (10^3)
- MHZ : MHz (10^6)

■ Initial setting f = calculated value when AUTO is selected for RBW.

■ Example RB 3KHZ;

VB

VB **Video Bandwidth**

■ Function Sets the video bandwidth.

Header	Program Command	Query	Response
VB	VB△f VB△a	VB?	VB△f (AUTO/MAN)

■ Value of f 10 Hz to 1 MHz (1, 3, 10 sequence)
(when input 3 MHz set to NONE)

■ Value of a UP : Increments in a 1, 3, 10 sequence.
DN : Decrements in a 1, 3, 10 sequence.
OA : Function Query (same as ?)
AUTO : VBW Auto coupling
MAN : VBW Manual coupling
NONE : Not Filtering

■ Suffix code f None : Hz (10^0)
HZ : Hz (10^0)
KHZ : kHz (10^3)
MHZ : MHz (10^6)

■ Initial setting f = calculated value when AUTO is selected for VBW.

■ Example VB 3 KHZ;

ST

ST **Sweep Time**

■ Function Sets the sweep time.

Header	Program Command	Query	Response
ST	ST△t ST△a	ST?	ST△t (AUTO/MAN)

■ Value of t 20 ㎳ to 1000 s : Sweep mode
25 ㎲ to 15 s : Zero Span mode

■ Value of a UP : Increments in a 1, 2, 5, 10 sequence.
DN : Decrements in a 1, 2, 5, 10 sequence.
OA : Function Query (same as ?)
AUTO : Sweep time Auto coupling
MAN : Sweep time Manual coupling

■ Suffix code t None : ㎳ (10^-3)
US : ㎲ (10^-6)
MS : ㎳ (10^-3)
SEC : sec (10^0)

■ Initial setting t = calculated value when AUTO is selected for Sweep time.

■ Example ST AUTO;
ST 20 ㎳;

Display Control

DL

DL **Display Line**

■ Function — Activates a horizontal line for use as visual aid or for computation purposes.

Header	Program Command	Query	Response
DL	DL△v	DL?	DL△rv

■ Value of v
Number : real. Dependent upon the selected amplitude units.
OFF : Off
ON : On
OA : Function Query (same as ?)

■ Value of rv
Number : real. Dependent upon the selected amplitude units.
OFF : Off

■ Suffix code
DBM : dBm
DBMV : dB mV
DBUV : dB μV
V : (V:V, MV:mV, UV:μV)
W : (W:W, MW:mW, UW:μW, NW:nW, PW:pW, FW:fW)
KHZ : kHz (FM Mode)
None : % (AM Mode)

■ Initial setting — OFF

■ Example — DL –50DBM;

Note : When set the DL displays top of screen. The value is not defined.

TH

TH **Threshold**

■ Function Set the minimum amplitude level and ignores data below this value.

Header	Program Command	Query	Response
TH	TH△ ℓ TH△sw	TH?	TH△r ℓ

■ Value of ℓ Number : real. Dependent upon the selected amplitude units.

■ Value of sw OFF : Off
ON : On
OA : Function Query (same as ?)

■ Value of r ℓ OFF : Off
Number : real. Dependent upon the selected amplitude units.

■ Suffix code DBM : dBm
DBMV : dB mV
DBUV : dB μV
V : (V:V, MV:mV, UV:μV)
W : (W:W, MW:mW, UW:μW, NW:nW, PW:pW, FW:fW)
KHZ : kHz (FM Mode)
None : % (AM Mode)

■ Initial setting OFF : Off

■ Example TH -50DBM;

Note : When set the TH displays bottom of screen. The value is not defined.

TITLE

TITLE **Screen Title Entry**

■ Function Places character data in the title area of the display. Available characters are Alpha-numeric.

Header	Program Command	Query	Response
TITLE	TITLE△text	TITLE?	TITLE△text

■ Value of text String within 8 characters. (following DOS Filename rule)

■ Example TITLE SA_970;
TITLE ?;

GRAT

GRAT **Graticule**

■ Function Turns the display graticule on or off

Header	Program Command	Query	Response
GRAT	GRAT△sw	GRAT?	GRAT△rsw

■ Value of sw 0, OFF : Off
1, ON : On
■ Value of rsw 0, 1
■ Initial setting ON
■ Example GRAT OFF;

Trace Function

TRS

TRS **Trace Status**

■ Function Sets the trace status.

Header	Program Command	Query	Response
TRS	TRS△sw	TRS?	TRS△sw

■ Value of sw TRA : Trace A
TRB : Trace B

■ Suffix code None

■ Initial setting None

■ Example TRS TRA;
TRS?:

TRF

TRF **Trace Function**

■ Function Sets the chosen trace's function.

Header	Program Command	Query	Response
TRF	TRF△sw	TRF?	TRF A=rsw B=rsw

■ Value of sw
CLEW : Clear and Write
MXMH : Max. Hold
MINH : Min. Hold
VIEW : View
BLANK : Blank

■ Value of rsw
0 : CLEW
1 : MXMH
2 : MINH
3 : VIEW
4 : BLANK

■ Suffix code None

■ Initial setting A=0 B=4

■ Example TRF CLEW;

TRA/TRB

TRA/TRB **Trace Data Input / Output**

■ Function — Provides a method for transferring trace data to or from a computer. The available data formats are decimal number(d) format, binary (b) format (only GPIB). [ref. 4-67 TDF]

Header	Program Command	Query	Response
TRA TRB	TRA△#ns△#ne,<d0,…dn> TRB△#ns△#ne,<d0,…dn> TRA△#ns△#ne,<b0…bn> TRB△#ns△#ne,<b0…bn>	TRA△#ns△#ne,? TRB△#ns△#ne,?	TRA ver : [d] TRB ver : [d] TRA ver : [b] TRB ver : [b]

■ Value of ns, ne — 1 to 500 : ns = start point, ne = stop point
(X axis position. Equal to trace data count)

■ Value of ver — Version information

■ Value of d — d1,d2,…d500 : decimal format(ASCII CODE)
LOG, AM, FM mode : 0 ~ 3600
Linear mode : 0 ~ 1800
QP mode : 0 ~ 4000 (ref. 4-128 Quasi peak)

■ Value of b — b1b2…b1000 : binary format(binary 2 bytes)
LOG, AM, FM mode : 0000h ~ 0E10h
Linear mode : 0000h ~ 0708h
QP mode : 0000h ~ 0FA0h

■ Suffix code — None

■ Example — TRA #1 #3, <2048, 1248, 200>
TRA #1 #500,?;

* Caution : Binary format Trace Data is read only from equipment!

TRAALL/TRBALL

TRAALL/TRBALL Trace All Data Output

■ Function Provides a method for transferring all trace data to the computer. The available data formats are decimal number (d) format, binary (b) format (only GPIB).

Header	Program Command	Query	Response
TRAALL TRBALL	----	TRAALL? TRBALL?	TRAALL ver:[d] TRBALL ver:[d] TRAALL ver:[b] TRBALL ver:[b]

■ Value of ver Version information

■ Value of d d1,d2,…d500 : decimal format(ASCII CODE)
LOG, AM, FM mode : 0 ~ 3600
Linear mode : 0 ~ 1800
QP mode : 0 ~ 4000 (ref. 4-128 Quasi peak)

■ Value of b b1b2…b1000 : binary format(binary 2 bytes)
LOG, AM, FM mode : 0000h ~ 0E10h
Linear mode : 0000h ~ 0708h
QP mode : 0000h ~ 0FA0h

■ Suffix code None

■ Initial setting Current trace data

■ Example TRAALL?;
TRBALL?;

```
* Caution : Binary format Trace Data is read only from equipment!
```

TDF

TDF **Trace Data Format**

■ Function Selects the format for input and output trace data.
You must specify the desired format when transferring data form the spectrum analyzer to a computer.

Header	Program Command	Query	Response
TDF	TDF△sw	TDF?	TDF△sw

■ Value of sw BIN : Binary data format
DEC : Decimal data format (ASCII Code)

■ Suffix code None

■ Initial setting DEC

■ Example TDF BIN;

Mathematic

AMB

AMB **Trace A Minus Trace B**

■ Function Subtracts the contents of Trace B from Trace A and places the result in Trace A.

Header	Program Command	Query	Response
AMB	AMB	----	----

■ Example AMB;

BML

BML **Trace B Minus Display Line**

■ Function Subtracts the display line form Trace B and places the result in Trace B.

Header	Program Command	Query	Response
BML	BML	----	----

■ Example BML;

APB

APB **Trace A Plus Trace B**

■ Function Adds the contents of Trace B to Trace A and the result in Trace A.

Header	Program Command	Query	Response
APB	APB	----	----

■ Example APB;

AMBPL

AMBPL **Trace A Minus Trace B Plus Display Line**

■ Function Subtracts the contents of Trace B from Trace A, adds the display line to this value, and stores the result in Trace A.

Header	Program Command	Query	Response
AMBPL	AMBPL	----	----

■ Example AMBPL;

AXB

AXB **Trace A Exchange Trace B**

■ Function Exchanges the contents of Trace A with those of Trace B.

Header	Program Command	Query	Response
AXB	AXB	----	----

■ Example AXB;

Detect Mode

DET

DET **Detection Mode**

■ Function Selects the detection mode for the waveform data being displayed.

Header	Program Command	Query	Response
DET	DET△d	DET?	DET△d

■ Value of d
- POS : Positive Peak
- NEG : Negative Peak
- SAM : Sample
- NRM : Normal
- AVG : Average

■ Suffix code None

■ Initial setting POS

■ Example DET NRM;

Average

AVG

AVG **Trace Average**

■ Function Trace average on or off. Depends on the condition of average count and cycling setup.

Header	Program Command	Query	Response
AVG	AVG△sw	AVG?	AVG△rsw

■ Value of sw 1, ON : On
0, OFF : Off

■ Value of rsw 0, 1

■ Suffix mode None

■ Initial setting OFF

■ Example AVG ON;

AVGC

AVGC **Number of Trace Average**

■ Function Sets the averaging rate (number of sweep repetitions).

Header	Program Command	Query	Response
AVGC	AVGC△n	AVGC?	AVGC△n

■ Value of n 2 ~ 999 : times of average
■ Suffix mode None
■ Initial setting 20
■ Example AVGC 32;

AVGCYL

AVGCYL **Average Cycle**

■ Function Set averaging cycle on means that the trace stops after the number of times of averaging has completed.

Header	Program Command	Query	Response
AVGCYL	AVGCYL△sw	AVGCYL?	AVGCYL△rsw

■ Value of sw 1, ON : On
0, OFF : Off
■ Value of rsw 0, 1
■ Suffix mode Nonde
■ Initial setting OFF
■ Example AVGCYL ON;

AVGS

AVGS **Average Control**

■ Function Control the averaging process.

Header	Program Command	Query	Response
AVGS	AVGS△sw	AVGS?	AVGS△sw

■ Value of sw STOP : Stop
CONT : Continue
RESET : Reset

■ Value of rsw CONT, STOP

■ Suffix mode None

■ Initial setting CONT

■ Example AVGS CONT;

File Management

Note : Can not be used disk drive name or directory, only filename is used in filename field.
(A:DEMO.STS [X] C:\DEMO.BMP [X])

FDRV

FDRV **File Disk Drive Selection**

■ Function Select source or destination drive for file transfer.

Header	Program Command	Query	Response
FDRV	FDRV△d	FDRV?	FDRV△d Drive

■ Value of d A: : 3.5” Floppy diskette
C: : Internal Memory

■ Example FDRV A:

FSAVE

FSAVE **File Save**

■ Function Save file which type was defined by extension of its name.

Header	Program Command	Query	Response
FSAVE	FSAVE△d	----	----

■ Value of d File name

■ Example FSAVE demo1.sts
FSAVE demo1.lmt
FSAVE demo1.trc

File extension & Data type
*.sts : Status data
*.lmt : Limit data
*.trc : Trace data
*.bmp : Screen dump

Note : FSAVE and FLOAD do not use wild card in filename. [ex. *.sts (×)

FLOAD

FLOAD **File Load**

■ Function Load file which type was defined in extension of its name.

Header	Program Command	Query	Response
FLOAD	FLOAD△d	----	----

■ Value of d File name

■ Example FLOAD demo1.sts
FLOAD demo1.lmt
FLOAD demo1.trc

File extension & Data type
*.sts : Status data
*.lmt : Limit data
*.trc : Trace data
*.bmp : Screen dump

FDEL

FDEL **File Delete**

■ Function Delete file.

Header	Program Command	Query	Response
FDEL	FDEL△d	----	----

■ Value of d File name :

■ Example FDEL demo1.sts;

FCPT

FCPT **File Copy To**

■ Function Copy file to other driver.(A: ↔ C:)

Header	Program Command	Query	Response
FCPT	FCPT△d	----	----

■ Value of d File name :

■ Example FCPT demo1.sts;

MEM

MEM **Memory Available**

■ Function Display current driver's remain memory.

Header	Program Command	Query	Response
MEM	MEM	MEM?	MEM r

■ Value of r n Bytes

■ Example MEM ?;

Limit Line

LMTPC

LMTPC **Limit Line Function**

■ Function Sets the limit line function on or off.

Header	Program Command	Query	Response
LMTPC	LMTPC△sw	LMTPC?	LMTPC△rsw rsw = OFF, UFIL, LFIL, PASS, AFIL

■ Value of sw ON : On
OFF : Off

■ Value of rsw OFF : Off
UFIL : Upper Failure
LFIL : Lower Failure
PASS : Pass
AFIL : All Failure

■ Suffix code None

■ Initial setting OFF

■ Example LMTPC ON;

ALARM

ALARM **Pass/Fail Alarm**

■ Function Sets alarm when limit-pass/fail check.

Header	Program Command	Query	Response
ALARM	ALARM△sw	ALARM?	ALARM△rsw

■ Value of sw ON : On
OFF : Off
OA : Function Query (same as ?)
■ Value of rsw ON, OFF
■ Initial setting OFF
■ Example ALARM ON;

LMTUP

LMTUP **Limit Line Upper Area**

■ Function Sets the upper limit line area on or off.

Header	Program Command	Query	Response
LMTUP	LMTUP△sw	LMTUP?	LMTUP△rsw

■ Value of sw 1, ON : On
0, OFF : Off
■ Value of rsw 0, 1
■ Initial setting OFF
■ Example LMTUP ON;

LMTLW

LMTLW **Limit Line Lower Area**

■ Function Sets the lower limit line area on or off.

Header	Program Command	Query	Response
LMTLW	LMTLW△sw	LMTLW?	LMTLW△rsw

■ Value of sw 1, ON : On
0, OFF : Off

■ Value of rsw 0, 1

■ Initial setting OFF

■ Example LMTLW ON;

Note! : Limit Line data file is .LMT. This file contents the data which is same as trace data(*.TRC). So the same data convert calculation can be adopted.

Measurement

MSTART

MSTART **Measure Start**

■ Function Start measurement.

Header	Program Command	Query	Response
MSTART	MSTART△sw	----	----

■ Value of sw
OBW : Occupied Bandwidth
ACP : Adjacent Channel Power
CP : Channel Power
HD : Harmonic Distribution
XDB : X dB Down

■ Example MSTART CP;

MSTOP

MSTOP **Measure Stop**

■ Function Stop measurement.

Header	Program Command	Query	Response
MSTOP	MSTOP	----	----

■ Example MSTOP;

MCONT

MCONT **Measure Continuous**

■ Function Set the measure state continuous or single.
When set on, the state is continuous and set off for single.

Header	Program Command	Query	Response
MCONT	MCONT△sw	MCONT?	MCONT△rsw

■ Value of sw 1, ON : On
0, OFF : Off
OA : Query (same as ?)

■ Value of rsw 0, 1

■ Initial setting 1

■ Example MCONT ON;

MCLRA

MCLRA **Measure All Clear**

■ Function Stopping the measurement and clear the measurement window.

Header	Program Command	Query	Response
MCLRA	MCLRA	----	----

■ Example MCLRA;

X dB Down

SXDBPT

SXDBPT **Set X dB Point**

■ Function Set the X dB value.

Header	Program Command	Query	Response
SXDBPT	SXDBPT△d	SXDBPT?	SXDBPT△d

■ Value of d 0 to 100 dB
■ Initial setting 3.00
■ Example SXDBPT 10.5 dB;

XDL

XDL **X dB Left**

■ Function Return left frequency distance from make frequency to left frequency it corresponds to X dB below mark level.

Header	Program Command	Query	Response
XDL	XDL△sw	XDL?	XDL△f

■ Value of sw OA : Query (same to ?)
■ Value of f X㎒
■ Example XDL?;

XDR

XDR **X dB Right**

■ Function Return right frequency distance from mark frequency to right frequency it corresponds to X dB below mark level.

Header	Program Command	Query	Response
XDR	XDR△sw	XDR?	XDR△f

■ Value of sw OA : Query (same to ?)
■ Value of f X ㎒
■ Example XDR?;

XDRL

XDRL **X dB Relative**

■ Function Return relative frequency which is the sum of XDL and XDR.

Header	Program Command	Query	Response
XDRL	XDRL△sw	XDRL?	XDRL△f

■ Value of sw OA : Query (same to ?)
■ Value of f X ㎒
■ Example XDRL?;

ACP Channel BW

SACPMCH

SACPMCH **Set ACP Main Channel**

■ Function Set main channel bandwidth in Adjacent Channel power measurement.

Header	Program Command	Query	Response
SACPMCH	SACPMCH△f	SACPMCH?	SACPMCH△r

■ Value of f 10 ㎐ ~
■ Value of r X ㎒
■ Suffix code NONE : ㎐
HZ : ㎐
KHZ : ㎑
MHZ : ㎒
GHZ : ㎓
■ Example SACPMCH 300MHZ;

Note ! : Adjust this value before programming, until disappear warning or error message in the bottom of measurement window. In ACP measurement least span value is 1 ㎒.

SACPACH

SACPACH **Set ACP Adjacent Channel**

■ Function Set adjacent channel bandwidth in ACP measurement.

Header	Program Command	Query	Response
SACPACH	SACPACH△f	SACPACH?	SACPACH△r

■ Value of f 10 ㎐ ~

■ Value of r X ㎒

■ Suffix code NONE : ㎐
HZ : ㎐
KHZ : ㎑
MHZ : ㎒
GHZ : ㎓

■ Example SACPACH 170.0MHZ;

Note ! : Adjust this value before programming, until disappear warning or error message in the bottom of measurement window. In ACP measurement least span value is 1 ㎒.

SACPSCH

SACPSCH **Set ACP Channel Space**

■ Function Set channel space which is distance from main channel to adjacent channel.

Header	Program Command	Query	Response
SACPSCH	SACPSCH△f	SACPSCH?	SACPSCH△r

■ Value of f 10 Hz ~

■ Value of r X MHz

■ Suffix code NONE : Hz
HZ : Hz
KHZ : kHz
MHZ : MHz
GHZ : GHz

■ Example SACPSCH 700.0 MHZ;

Note ! : Adjust this value before programming, until disappear warning or error message in the bottom of measurement window. In ACP measurement least span value is 1 MHz.

SACPAVG

SACPAVG **Set ACP Average**

■ Function Set the measurement mode to averaging.
The measurement value is averaged continuous. But in some case the measurement value was reset, the cases is calling start measurement or measurement off and on or measurement stop and start.

Header	Program Command	Query	Response
SACPAVG	SACPAVG△sw	SACPAVG?	SACPAVG△rsw

■ Value of sw 1, ON : On
0, OFF : Off
OA : Query (same as ?)

■ Value of rsw 0, 1

■ Example SACPAVG ON;

AMC

AMC **ACP Main Channel**

■ Function Return the main channel power level value.

Header	Program Command	Query	Response
AMC	AMC△sw	AMC?	AMC△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dBm
■ Example AMC?;

ALC

ALC **ACP Lower Channel**

■ Function Return the lower adjacent channel power level value. This value is relative difference level of main channel power.

Header	Program Command	Query	Response
ALC	ALC△sw	ALC?	ALC△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dB
■ Example ALC?;

AUC

AUC **ACP Upper Channel**

■ Function Return the upper adjacent channel power level value. This value is relative difference level of main channel power.

Header	Program Command	Query	Response
AUC	AUC△sw	AUC?	AUC△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dB
■ Example AUC?;

Channel Power

SCPBW

SCPBW **Set Channel Power BandWidth**

■ Function Set channel power bandwidth for measuring power level in the limited frequency range.

Header	Program Command	Query	Response
SCPBW	SCPBW△f	SCPBW?	SCPBW△r

■ Value of f X Hz (10 Hz ~)
■ Value of r X MHz
■ Suffix code NONE : Hz
HZ : Hz
KHZ : kHz
MHZ : MHz
GHZ : GHz
■ Example SCPBW 300 MHZ;

Note ! : Adjust this value before programming, until disappear warning or error message in the bottom of measurement window. In CP measurement least span value is 1 MHz.

SCPSP

SCPSP **Set Channel Power Span**

■ Function Set channel power span. This is the same value in frequency span.

Header	Program Command	Query	Response
SCPSP	SCPSP△f	SCPSP?	SCPSP△r

■ Value of f 1㎒ ~
■ Value of r X㎒
■ Suffix code HZ:㎐, KHZ:㎑, MHZ:㎒, GHZ:㎓, None:㎐
■ Example SCPSP 700.0 MHZ;

SCPMH

SCPMH **Set Channel Power MaxHold**

■ Function Set channel power trace holding maximum power level value.

Header	Program Command	Query	Response
SCPMH	SCPMH△sw	SCPMH?	SCPMH△rsw

■ Value of sw 1, ON : On
0, OFF : Off
OA : Query (same as ?)
■ Value of rsw 0, 1
■ Example SCPMH ON;

SCPAVG

SCPAVG **Set Channel Power Average**

■ Function Set the measurement mode to averaging. The measurement value is averaged continuously. But the measurement value was reset, in case of measurement stop and start.

Header	Program Command	Query	Response
SCPAVG	SCPAVG△sw	SCPAVG?	SCPAVG△rsw

■ Value of sw 1, ON : On
0, OFF : Off
OA : Query (same as ?)

■ Value of rsw 0, 1

■ Example SCPAVG ON;

CPWR

CPWR **Channel Power**

■ Function Return the channel power level value

Header	Program Command	Query	Response
CPWR	CPWR△sw	CPWR?	CPWR△d

■ Value of sw OA : Query (same as ?)

■ Value of d X dBm

■ Example CPWR?;

Occupied BandWidth

SOBWSP

SOBWSP **Set Occupied BandWidth Span**

■ Function Set occupied bandwidth span.

Header	Program Command	Query	Response
SOBWSP	SOBWSP△f	SOBWSP?	SOBWSP△r

■ Value of f 100 Hz ~

■ Value of r X MHz

■ Suffix code NONE : Hz
HZ : Hz
KHZ : kHz
MHZ : MHz
GHZ : GHz

■ Example SOBWSP 300MHZ;

SOBWP

SOBWP **Set OBW Power Percentage**

■ Function Set power percentage for measurement channel power within specific percentage of total channel power.

Header	Program Command	Query	Response
SOBWP	SOBWP△d	SOBWP?	SOBWP△d

■ Value of d 5 to 100 %
■ Suffix code NONE : %
%
■ Example SOBWP 99.6%;

OCP

OCP **OBW Channel Power**

■ Function Return the measurement power level which is measured within Occupied channel bandwidth.

Header	Program Command	Query	Response
OCP	OCP△sw	OCP?	OCP△d

■ Value of sw OA : Query (same to ?)
■ Value of d X dBm
■ Example OCP?:

OCF

OCF **OBW Channel Frequency**

■ Function Return the bandwidth frequency that is limited by OBW.

Header	Program Command	Query	Response
OCF	OCF△sw	OCF?	OCF△f

■ Value of sw OA : Query (same to ?)

■ Value of f X ㎒

■ Example OCF?:

Harmonic Distortion

SHDN

SHDN **Set Harmonic Distortion Number**

■ Function Set the count of harmonic order to be measured.

Header	Program Command	Query	Response
SHDN	SHDN△n	SHDN?	SHDN△n

■ Value of n 2 to 5 : default = 2
■ Suffix code NONE
■ Example SHDN 3;

SHDAVG

SHDAVG **Set Harmonic Distortion Average**

■ Function Set the trace average on for more stable harmonic distortion measuring.

Header	Program Command	Query	Response
SHDAVG	SHDAVG△sw	SHDAVG?	SHDAVG△rsw

■ Value of sw 1, ON : On
0, OFF : Off
OA : Query (same to ?)
■ Value of rsw 0, 1
■ Example SHDAVG ON;

HF1

HF1 **Harmonic Distortion Frequency (1st)**

■ Function Return first distortion frequency of harmonic distortion measurement.

Header	Program Command	Query	Response
HF1	HF1△sw	HF1?	HF1△f

■ Value of sw OA : Query (same to ?)
■ Value of f X ㎒
■ Example HF1?;

HA1

HA1 **Harmonic Distortion Amplitude (1st)**

■ Function Return first distortion power level of harmonic distortion measurement.

Header	Program Command	Query	Response
HA1	HA1△sw	HA1?	HA1△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dBm
■ Example HA1?;

HF2

HF2 **Harmonic Distortion Frequency (2nd)**

■ Function Return second distortion frequency of harmonic distortion measurement.

Header	Program Command	Query	Response
HF2	HF2△sw	HF2?	HF2△f

■ Value of sw OA : Query (same to ?)
■ Value of f X ㎒
■ Example HF2?;

HA2

HA2 **Harmonic Distortion Amplitude (2nd)**

■ Function Return second distortion power level of harmonic distortion measurement.

Header	Program Command	Query	Response
HA2	HA2△sw	HA2?	HA2△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dBm
■ Example HA2?;

HF3

HF3 **Harmonic Distortion Frequency (3rd)**

■ Function Return third distortion frequency of harmonic distortion measurement.

Header	Program Command	Query	Response
HF3	HF3△sw	HF3?	HF3△f

■ Value of sw OA : Query (same to ?)
■ Value of f X ㎒
■ Example HF3?;

HA3

HA3 **Harmonic Distortion Amplitude (3rd)**

■ Function Return third distortion power level of harmonic distortion measurement.

Header	Program Command	Query	Response
HA3	HA3△sw	HA3?	HA3△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dBm
■ Example HA3?;

HF4

HF4 **Harmonic Distortion Frequency (4th)**

■ Function Return fourth distortion frequency of harmonic distortion measurement.

Header	Program Command	Query	Response
HF4	HF4△sw	HF4?	HF4△f

■ Value of sw OA : Query (same to ?)
■ Value of f X ㎒
■ Example HF4?;

HA4

HA4 **Harmonic Distortion Amplitude (4th)**

■ Function Return fourth distortion power level of harmonic distortion measurement.

Header	Program Command	Query	Response
HA4	HA4△sw	HA4?	HA4△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dBm
■ Example HA4?;

HF5

HF5 **Harmonic Distortion Frequency (5th)**

■ Function Return fifth distortion frequency of harmonic distortion measurement.

Header	Program Command	Query	Response
HF5	HF5△sw	HF5?	HF5△f

■ Value of sw OA : Query (same to ?)
■ Value of f X ㎒
■ Example HF5?;

HA5

HA5 **Harmonic Distortion Amplitude (5th)**

■ Function Return fifth distortion power level of harmonic distortion measurement.

Header	Program Command	Query	Response
HA5	HA5△sw	HA5?	HA5△d

■ Value of sw OA : Query (same as ?)
■ Value of d X dBm
■ Example HA5?;

THD

THD **Total Harmonic Distortion**

■ Function Measure Total Harmonic Distortion.

Header	Program Command	Query	Response
THD	THD△sw	THD?	THD△ρ

■ Value of sw OA : Query (same as ?)
■ Value of ρ X %
■ Example THD?;

Auxiliary

DEMOD

DEMOD **Demodulation**

■ Function Activates either AM or FM demodulation or turns the demodulation off.

Header	Program Command	Query	Response
DEMOD	DEMOD△sw	DEMOD?	DEMOD△sw

■ Value of sw AM : AM demodulation
FM : FM demodulation
OFF : Off
■ Initial setting OFF
■ Example DEMOD AM;

AUDIO

AUDIO **Speaker**

■ Function Selects Speaker on or off

Header	Program Command	Query	Response
AUDIO	AUDIO△sw	AUDIO?	AUDIO△rsw

■ Value of sw 1, ON : On
0, OFF : Off
■ Value of rsw 0, 1
■ Initial setting 0
■ Example AUDIO ON;

SQL

SQL Squelch level

■ Function Adjusts squelch level.

Header	Program Command	Query	Response
SQL	SQL△n	SQL?	SQL△n

■ Value of n 0 to 255 (1 step)
■ Initial setting 127
■ Example SQL 80;

AUDIOVR

AUDIOVR Speaker Volume

■ Function Adjusts the volume of the speaker.

Header	Program Command	Query	Response
AUDIOVR	AUDIOVR△n	AUDIOVR?	AUDIOVR△n

■ Value of n 0 to 7 (1 step)
■ Suffix code None
■ Initial setting 3
■ Example AUDIOVR 5;

Preset

PRST

PRST **Preset**

■ Function Executes preset. All equipment parameters are set to default values.

Header	Program Command	Query	Response
PRST	PRST	----	----

■ Example PRST;

CALALL

CALALL **All Calibrations**

■ Function Executes all calibration.

Header	Program Command	Query	Response
CALALL	CALALL	----	----

■ Example CALALL;

PCAL

PCAL **Periodic Temperature Calibrations**

■ Function Initiates periodic temperature calibration execution.

Header	Program Command	Query	Response
PCAL	PCAL△sw	PCAL?	PCAL△rsw

■ Value of sw 1, ON : ON
0, OFF : Off

■ Value of rsw 0, 1

■ Initial setting 1

■ Example PCAL ON;

YIGCAL

YIGCAL **First Local Oscillator Calibration**

■ Function Executes First local oscillator Calibration.

Header	Program Command	Query	Response
YIGCAL	YIGCAL	----	----

■ Example YIGCAL;

RCAL

RCAL **RBW Calibration**

■ Function Executes RBW calibration.

Header	Program Command	Query	Response
RCAL	RCAL	----	----

■ Example RCAL;

TMPCAL

TMPCAL **Temperature Calibration**

■ Function Executes temperature Calibration.

Header	Program Command	Query	Response
TMPCAL	TMPCAL	----	----

■ Example TMPCAL;

LVLC

LVLC **Level Calibration**

■ Function Executes level calibration.

Header	Program Command	Query	Response
LVLC	LVLC	----	----

■ Example LVLC;

SPCAL

SPCAL **Span Calibration**

■ Function Executes span calibration.

Header	Program Command	Query	Response
SPCAL	SPCAL	----	----

■ Example SPCAL;

Configuration

Printer

HCOPY

HCOPY **Hard Copy**

■ Function Prints entire screen image.

Header	Program Command	Query	Response
HCOPY	HCOPY	----	----

■ Example HCOPY;

Clock Set

DATE

DATE **Set Date**

■ Function Sets the built-in clock to the specified date.

Header	Program Command	Query	Response
DATE	DATE△yymmdd	DATE?	DATE△yymmdd

■ Value of yy 00 to 99 (year), 96 ~ 99, : 1996~1999, 00 ~ 95 : 2000~2095
■ Value of mm 01 to 12 (month)
■ Value of dd 01 to 31 (day)
■ Suffix code None
■ Example DATE 020228;

TIME

TIME **Set Time**

■ Function Sets the built-in clock to the specified time.

Header	Program Command	Query	Response
TIME	TIME△hhmmss	TIME?	TIME△hhmmss

■ Value of hh 00 to 23 (hour)
■ Value of mm 00 to 59 (minute)
■ Value of ss 00 to 59 (sec)
■ Suffix code None
■ Example TIME 091122;

GPIB Common Command

*CLS

***CLS** **Clear Status Command**

■ Function Clears the status byte register.

Header	Program Command	Query	Response
*CLS	*CLS	----	----

■ Example *CLS;

*ESE

***ESE** **Standard Event Status Enable**

■ Function Sets or clears the standard status enable register.

Header	Program Command	Query	Response
*ESE	*ESE△n	*ESE?	*ESE△n

■ Value of n 0 to 255 : Represents the sum of the bit-weighted values.
■ Suffix code None
■ Example *ESE 20:
*ESE?;

*ESR?

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

■ Function Returns the current value in the standard event status register.

Header	Program Command	Query	Response
*ESR?	----	*ESR?	*ESR△n

■ Value of n 0 to 255 : Represents the sum of the bit-weighted values.
■ Suffix code None
■ Example *ESR?;

*IDN?

***IDN?** **Identification Query**

■ Function Return the model name, etc of the equipment

Header	Program Command	Query	Response
*IDN?	----	*IDN?	LGIT, mo, sn, v

■ Value of mo Model : 970
■ Value of sn Serial No.
■ Value of v Version
■ Suffix code None
■ Example *IDN?;

*OPC

***OPC** **Operation Complete Command**

■ Function Set the standard event register bit 0 to 1 when the requested action was competed.

Header	Program Command	Query	Response
*OPC	*OPC	----	-----

■ Example *OPC;

*OPC?

***OPC?** **Operation Complete Query**

■ Function Sets the output queue to 1 to generate a MAV summary message when all pending select device operations have been completed.

Header	Program Command	Query	Response
*OPC?	----	*OPC?	1

■ Example *OPC?;

*RST

***RST** **Rest Command**

■ Function Resets the device

Header	Program Command	Query	Response
*RST	*RST	----	----

■ Example *RST;

*SRE

***SRE** **Service Request Enable Command**

■ Function Sets the bits in the service request enable register.

Header	Program Command	Query	Response
*SRE	*SRE△n	*SRE?	*SRE△n

■ Value of n 0 to 255 : Represents the sum of the bit-weighted values.
■ Suffix code None
■ Example *SRE 32;
*SRE?;

*STB?

***STB?** **Returns Status Byte Command**

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

Header	Program Command	Query	Response
*STB?	----	*STB?	*STB△n

■ Value of n

Bit	Bit Weight	Bit Name	Condition of status byte register
7	128	----	0 = Not used
6	64	MSS	0 = Service not requested 1 = Service requested
5	32	ESB	0 = Event status not generated 1 = Event status generated
4	16	MAV	0 = No data in output queue 1 = Data in output queue
3	8	ESB2	0 = Event status not generated 1 = Event status generated
2	4	----	0 = Not used
1	2	----	0 = Not used
0	1	----	0 = Not used

■ Example *STB?;

Others

ESE2

ESE2 **Event Status Enable (End)**

■ Function — Allows the End Event Status Enable Register to select which bit in the corresponding Event Register cause a TRUE ESB summary message bit 3 when set.

Header	Program Command	Query	Response
ESE2	ESE2△n	ESE2?	ESE2△n

■ Value of n — 0 to 255 : Represents the sum of the bit-weighted values.
■ Suffix code — None
■ Example — ESE2 1;
ESE2?;

ESR2?

ESR2? **Event Status Register (End) Query**

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

Header	Program Command	Query	Response
ESR2?	----	ESR2?	ESR2△n

■ Value of n 0 to 255 : Represents the sum of the bit-weighted values.

Bit	Bit Weight	Event	Description
7	128	Not used	Not used
6	64	Not used	Not used
5	32	Not used	Not used
4	16	Measurement completed	Measurement has been completed (Peak search, OBW, X dB, Noise marker, Freq. Counter, Limit Pass/Fail..)
3	8	AUTO TUNE completed	AUTO TUNE has been completed.
2	4	Averaging completed	Sweeping according to the specified AVERAGE number has been completed.
1	2	Calibration completed	RBW Cal.. Power On Cal.. ALL ACL.. Temp Cal.. Span Cal.. Level Cal.. Log Cal.. has been completed.
0	1	Sweep completed	A single sweep has been completed or is in standby.

■ Suffix code None
■ Example ESR2?;

QRYTYP

QRYTYP **Query Response Type**

■ Function Sets query response type.

Header	Program Command	Query	Response
QRYTYP	QRYTYP△sw	QRYTYP?	QRYTYP△sw

■ Value of sw ECO : Query response type is echo.
NEC : Query response type is no echo.
■ Suffix code None
■ Initial setting NEC
■ Example QRYTYP NEC;

ERR

ERR **Error Number**

■ Function Returns the error number of the current function.
Error buffer size : 256

Header	Program Command	Query	Response
ERR	----	ERR?	ERR△n

■ Value of n 000 to 999 (Ref. APPENDIX-19 Error Code)
■ Suffix code None
■ Example ERR?;

Quasi Peak Mode (option)

QPM

QPM **Quasi Peak Mode**

■ Function Selects quasi peak mode bandwidth.

Header	Program Command	Query	Response
QPM	QPM△sw	QPM?	QPM△sw

■ Value of sw BNDB : RBW 9 ㎑
BNDC : RBW 120 ㎑
OFF : Off

■ Suffix code None

■ Initial setting OFF

■ Example QPM BNDB;

<BLANK>

SECTION 5
STATUS STRUCTURE

This section describes the device-status reporting and its data structure defined by the IEEE488.2 when GPIB interface bus is used. This section also describes the synchronization techniques between a controller and device.
These functions are used to control a device from an external controller using the GPIB interface bus. Most of these functions can also be used to control a device from an external controller using the RS-232C interface.

TABLE OF CONTENTS

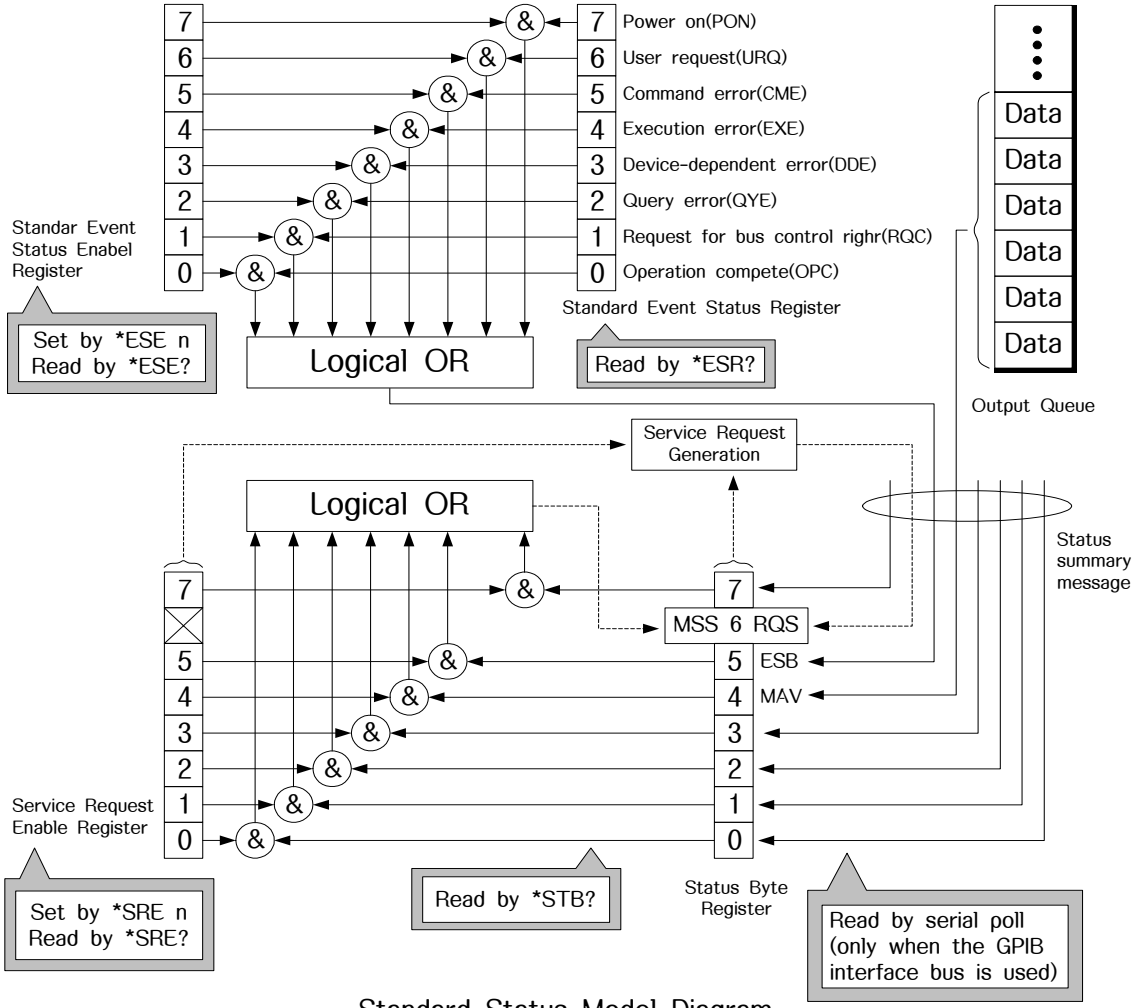
<BLANK>

SECTION 5
STATUS STRUCTURE

The Status Byte (STB) sent to the controller is based on the IEEE488.1 standard. The bits comprising the STB are called status summary messages because they represent a summary of the current data in registers and queues.

IEEE488.2 Standard Status Model

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



Standard Status Model Diagram

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

Register Model	Queue Model
The register model consists of two registers used for recording events and conditions encountered by a device. These two registers are the Event Status Register and Event Status Enable Register. When the results of the AND operation of both register contents are 1, the corresponding bit of the status bit becomes 1. In other cases, the corresponding bit becomes 0. When the result of their Logical OR is 1, the summary message bit also becomes 1. If the Logical OR result is 0, the summary message bit also becomes 0. The other register model which consists of status Byte Register and Service Request Enable Register has the same organism as above.	The queue in the queue model is used to sequentially record the waiting status values or information. If the queue is not empty, the queue structure summary message becomes 1. If the queue is empty, the message becomes 0.

In IEEE488.2, there are three standard models for the status data structure. Two are register models and one is a queue model based on the register model and queue model described above. The three standard models are:

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

Standard Event Status Register	Status Byte Register	Output Queue
The Standard Event Status Register has the same structure as the previously described register model. In this register, the bits for eight types of standard events encountered by a device are set at follows: ① Power on ② User request ③ Command error ④ Execution error ⑤ Device-dependent error ⑥ Query error ⑦ Request for bus control right ⑧ Operation complete The Logical OR output bit is represented by Status Byte Register bit 5 (DIO6) as a summary message for the Event Status Bit (ESB)	The Status Byte Register is a register in which the RQS bit and the seven summary message bits from the status data structure can be set. This register is used together with the Service Request Enable Register. When the results of the OR operation of both register contents are other than 0, SRQ becomes ON. To indicate this, bit 6 of the Status Byte Register (DIO7) is reserved by the system as the RQS bit. The RQS bit is used to indicated that there is a service request for the external controller. The mechanism of SRQ conforms to the IEEE488.1 standard.	The Output Queue has the structure of the queue model described above. Status Byte Register bit 4 (DIO5) is set as a summary message for Message Available (MAV) to indicate that there is data in the output buffer.

Status Byte (STB) Register

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

ESB and MAV Summary Messages

This paragraph describes the ESB and MAV summary message.

(1) ESB Summary Message

The ESB (Event Summary Bit) is a message defined by IEEE488.2 which uses bit 5 of the STB register. When the setting permits events to occur, the ESB summary message bit becomes 1 if any one of the events recorded in the Standard Status Register becomes 1. Conversely, the ESB summary message bit becomes 0 if one of recorded events occurs, even if events are set to occur.

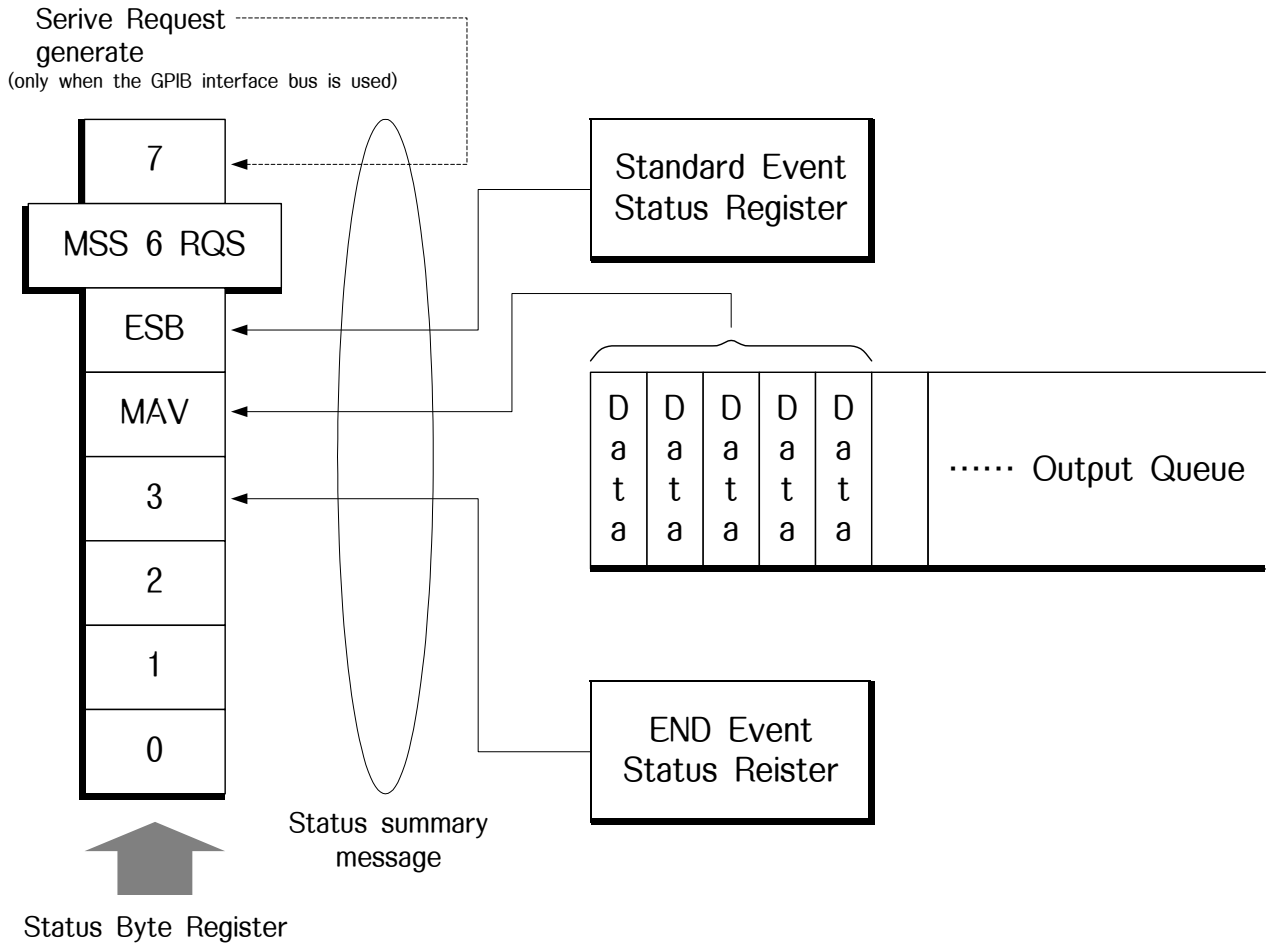
This bit becomes o when the ESR register is read by the *ESR? Query or when it is cleared by the *CLS command.

(2) MAV Summary Message

The MAV (Message Available) summary bit is a message defined by IEEE488.2 which uses bit 4 of the STB register. This bit indicates whether the output queue is empty. The MAV summary message bit is set to 1 when a device is ready to receive a request for a response message from the controller. When the output queue is empty, this bit is set to 0. This message is used to synchronize the information exchange with the controller. For example, this message is available when, after controller sends a query command to a device, the controller waits until MAV becomes 1. While the controller is waiting for a response from the device, other jobs can be processed. Reading the Output Queue without first checking MAV will cause all system bus operations to be delayed until the device responds.

Device-Dependent Summary Message

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



Reading and Clearing the STB Register

The STB register can be read using serial polling or the *STB? common query.
The IEEE488.1 STB message can be read by either method, but the value sent to bit 6 (position) is different for each method.
The STB register contents can be cleared using the *CLS command.

(1) Reading by serial polling (only when the GPIB interface bus is used)
The IEEE488.1 serial polling allows the device to return a 7-bit statue byte and an RQS message bit which conforms to IEEE488.1. The value of the status byte is not changed by serial polling. The device sets the RQS message to 0 immediately after being polled.

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

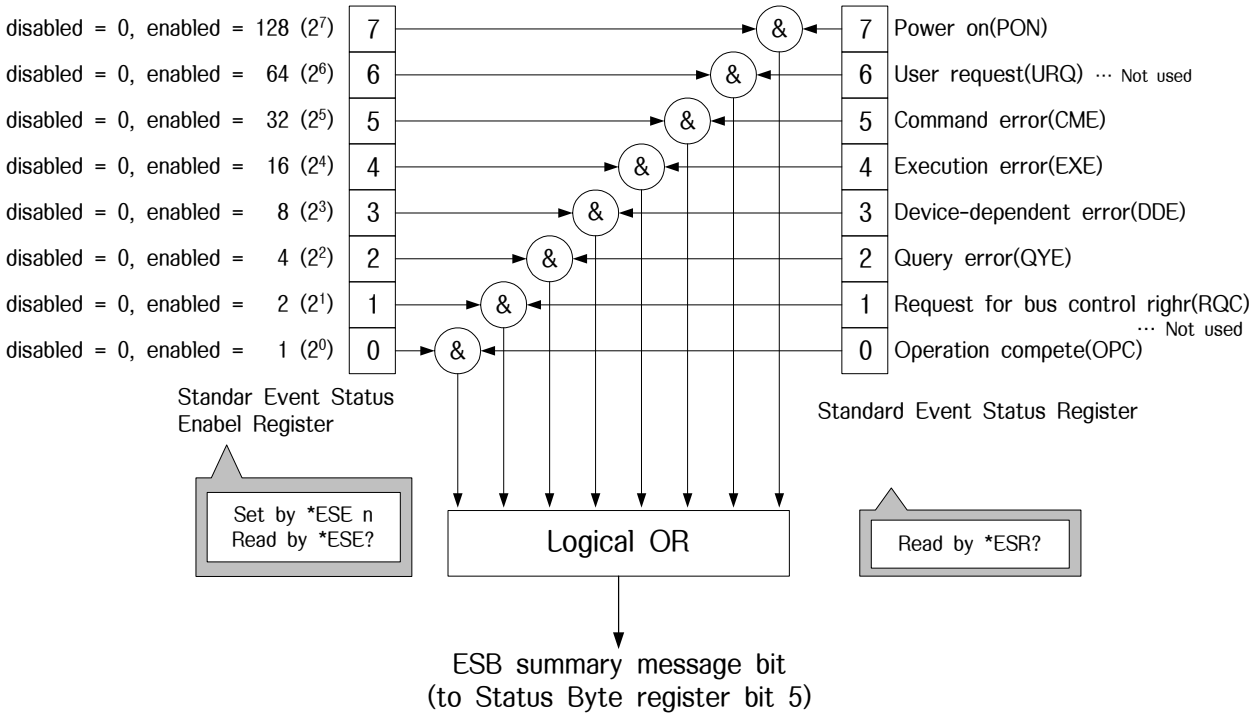
(3) Definition of MSS (Master Summary Message)
MSS indicates that there is at least one cause for a service request. The MSS message is represented at bit 6 response to an *STB? Query, but it is not status byte specified by IEEE488.1.
MSS is configured by the over all logical OR in which the STB register and SRQ enable (SRE) register are combined.

(4) Clearing the STB register using the *CLS common command
The *CLS common command clears all status data structures as well as the summary message corresponding to them.
The *CLS command does not affect the setting in the Enable Register.

Service Request (SRQ) Enabling Operation

Bits 0 to 7 of the Service Request Enable Register (SRE) determine which bit of the corresponding STB register can generate SRQ.

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



(1) Reading the SRE register

The contents of the SRE register are read using the *SRE? Common query.

The response message to this query is an integer from 0 to 255 which is the sum of the bit digit weighted values in the SRE register.

(2) Updating the SRE register

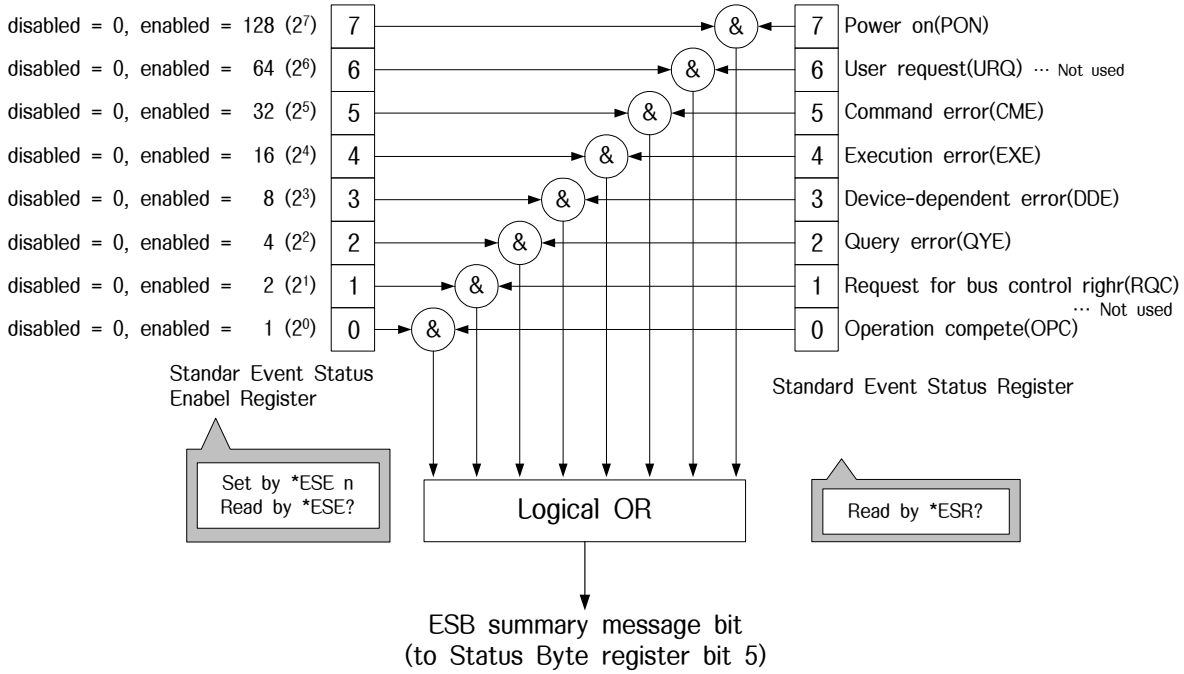
The SRE register is written using the *SRE common command. An integer from 0 to 255 is assigned as a parameter to set the SRE register bit to 0 or 1.

The value of bit 6 is ignored.

Standard Event Status Register

Bit Definition of Standard Event Status Register

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



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

Bit	Event name	Description
7	Power on (PON-Power on)	A transition from power~off to power~on occurred during the power~up procedure.
6	Not used	
5	Command error (CME-Command Error)	An illegal program message or misspelled command was received.
4	Execution error (EXE-Execution Error)	A legal but unexcitable program message was received.

Bit	Event name	Description
3	Device-dependent error (DDE-Device-dependent Error)	An error not caused by CME, EXE, or QYE occurred (parameter error, etc).
2	Query error (QYE-Query Error)	An attempt was made to read data in the output queue when it was empty. Or, the data in the output queue was lost before it was read.
1	Not used	
0	Operation complete (OPC-Operation Complete)	This bit becomes 1 when this equipment has processed the *OPC command.

Reading, Writing, and Clearing the Standard Event Status Register

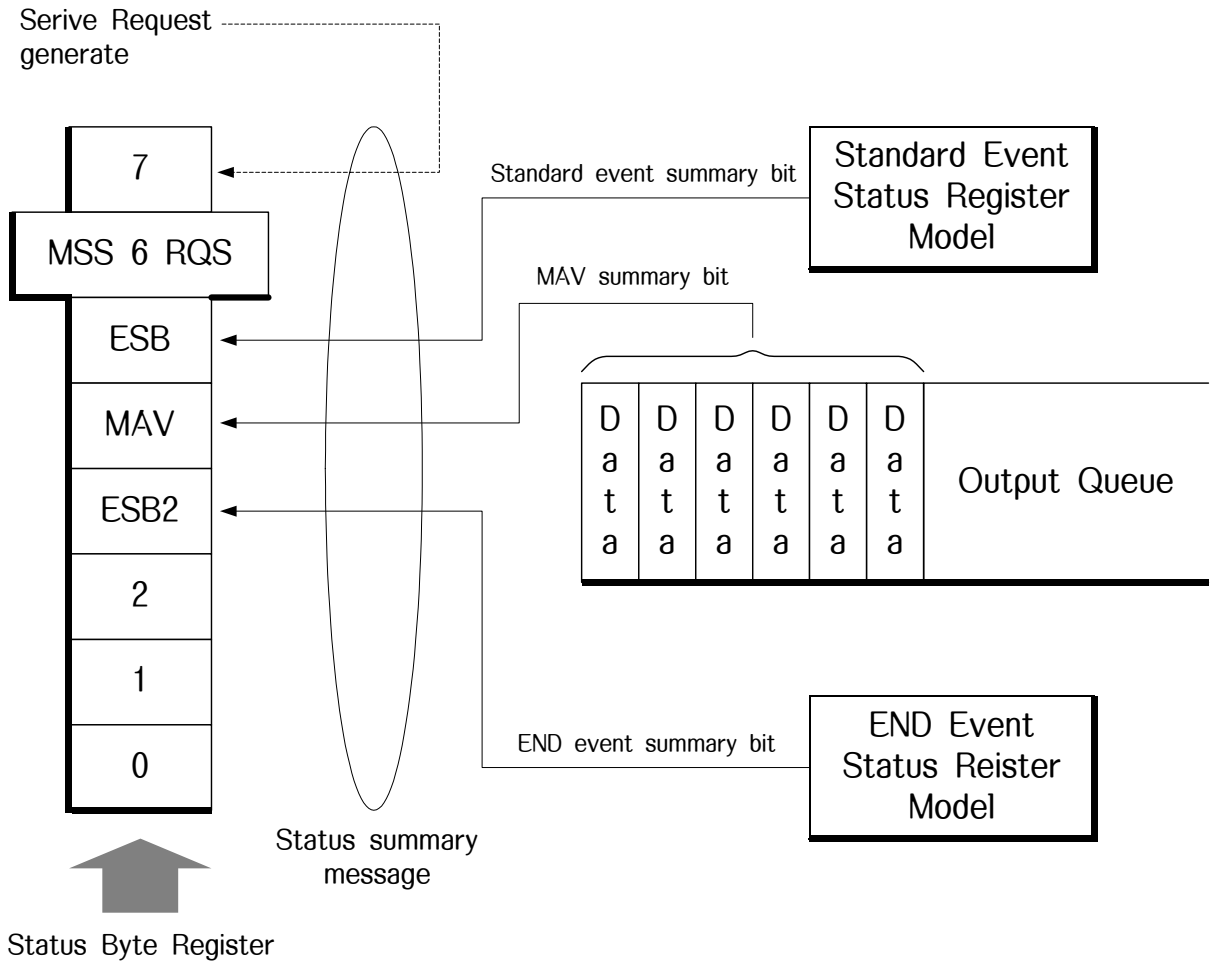
Reading	The register is read using the *ESR? Command query. The register is cleared after being read. The response message is integer -format data with the binary weight added to the event bit and the sum converted decimal
Writing	With the exception of clearing, data cannot be written to the register from outside.
Clearing	The register is cleared when : ① A *CLS command is received. ② The power is turned on Bit 7 is set to ON, and the other bits are cleared to 0. ③ An event is read for the *ESR? Query command.

Reading, Writing, and Clearing the Standard Event Status Enable Register

Reading	The registers is read using the *ESE? Command. The response message is integer-format data with the binary weight added to the event bit and the sum converted to decimal.
Writing	The register is written using the *ESE common command.
Clearing	The register is cleared when : ① An “ESE command with a data value of 0 is received. ② The power is turned on. The Standard Event Enable Register is not affected when : ① The device clear function status of IEEE488.1 is changed. ② A *RST common command is received. ③ A *CLS common command is received.

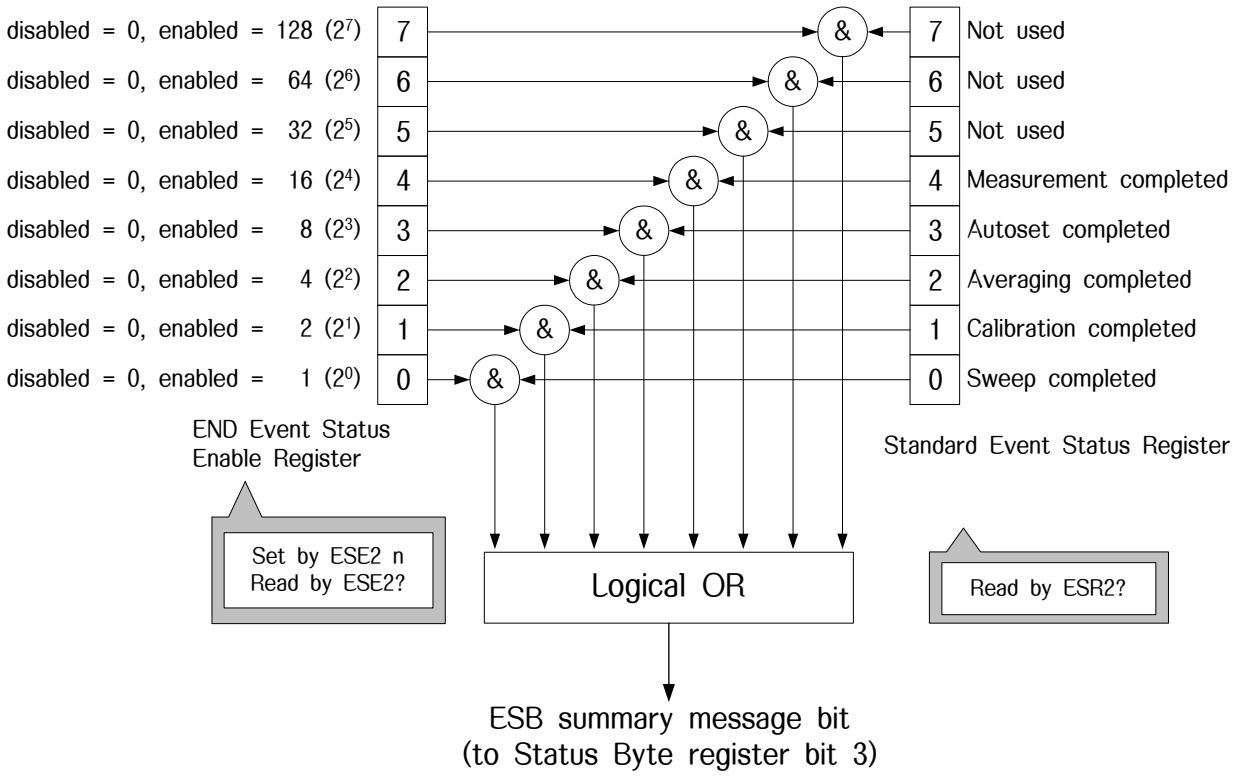
Extended Event Status Register

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



Bit Definition of END Event Status Register

The diagram below shows the operation and event-bit names of the END Event Status Register.



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

Bit	Event name	Description
7	Not used	Not used
6	Not used	Not used
5	Not used	Not used

Bit	Event name	Description
4	Measurement completed	Calculation processing for measurements (Peak search, OBX, X dB down, Noise marker, Frequency counter Limit pass/fail) has been completed.
3	AUTO SET completed	AUTO SET has been completed.
2	Averaging completed	Sweeping according to the specified AVERAGE number has been completed.
1	Calibration completed	RBW CAL, Power on CAL, All CAL, Temp CAL, Span CAL, Level CAL or LOG CAL has been completed.
0	Sweep completed	A single sweep has been completed or is standby.

Reading, Writing, and Clearing the Extended Event Status Register

Reading	The ESR? common query is used to read the register. The register is cleared after being read. The response message is integer-format data with the binary weight added to the event bit and the sum converted to decimal.
Writing	With the exception of clearing, data cannot be written to the register from outside.
Clearing	The register is cleared when : ① A *CLS command is received. ② The power is turned on. ③ An event is read for the ESR? query command.

Reading, Writing, and Clearing the Extended Status Enable Register

Reading	The ESE2? query is read the register. The response message is integer-format data with the binary weight added to the event bit and sum converted to decimals.
Writing	The ESE2 program command is used to write the register. Because bits 0 to 7 of the registers are weighted with values 1, 2, 4, 8, 16, 32, 64 and 128, respectively, the write data is transmitted as integer-format data that is the sum of the required bit digits selected form the weighted value.
Clearing	The register is cleared when : ① An ESE2 program command with a data value of 0 is received. ② The power is turned on. The Extended Event Status Enable register is not affected when : ① The device clear function status of IEEE488.1 is changed. ② A *RST common command is received. ③ A *CLS common command is received.

SECTION 6
EXAMPLE CODES

This section shows some example codes to transmit the message on the bus between a personal computer and spectrum analyzer via GPIB.

TABLE OF CONTENTS

<BLANK>

SECTION 6
EXAMPLE CODES

'C' language user with PC contained GPIB card program to control spectrum analyzer by use the library produced GPIB card manufacturer. For example here the example cords use the "DECL-32.h", "GPIB-32.obj" library from National Equipment.

Frequency and Level Measurement

Sets the normal marker on the peak point at the current waveform and measures the frequency and level on the normal marker.

1. Set
   a. Span : 1 ㎒
   b. Center Frequency : 1300 ㎒
   c. Reference Level : -10 dBm
   d. VBW, RBW, Input Attenuator : Auto
   e. Log 10 dB scale, Unit : dBm
   f. Sweep Time : 50 ㎳
2. Measuring
   a. Peak Search
   b. Read the frequency and the amplitude at the peak point

```
//-----------------------------------------------------------------------------------------------------
//                Frequency and Level measurement at the marker point
//-----------------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>

    int ud;
    char DataBuf[80]; // Set input message buffer size
    char SpollByte;

    void Send(char *buf)
    {
     ibwrt(ud, buf, (long)strlen(buf));
     if (ibsta & ERR) printf("ibwrt Error\n");
    }
```

```
void Receive(void)
{
 // Is data on 2394 buffer to read ?
 ibwait(ud, (TIMO | RQS));
 if (ibsta & (ERR | TIMO)) printf("ibwait Error\n");

 ibrsp(ud, &SpollByte);
 if (ibsta & ERR) printf("ibrsp Error\n");
 if (SpollByte != 0x50) printf("2394 Polling Error\n");

 // read data.
 ibrd(ud, DataBuf, 80L);
 DataBuf[ibcntl-1] = '\0';
 if (ibsta & ERR) printf("ibrd Error\n");
}

void InitGPIB(void)
{
 // Initialize GPIB bus and 2394
 ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
 if(ud<0) printf("2394 device open error\n");

 ibclr(ud);
 if(ibsta & ERR) { printf("ibclr error\n"); exit(1);}
 else  printf("Init Ok\n");
}

void main(void)
{
 prinf("<<<Frequency and Level measurement>>>\n");

 InitGPIB();
 Send("*CLS;*SRE 16;"); // Set 2394 to its initial state for programming

 // span: 1 MHz, Center Frequency: 1300 MHz, Reference Level: -10dBm
 Send("SP 1 MHz;CF 1300 MHz;RL -10 DBM;");
 Send("AUTOCPL;"); // VBW, RBW, Sweep Time, Input Attenuator : Auto
 Send("LG 10 DB;"); // Log 10 dB scale
 Send("AUNITS DBM;"); // dBm unit
 Send("ST MAN;ST 50MS;*OPC?;");
 Receive(); // Waiting the commands completed
 Send("MKPK;"); // Peak Search
 Send("MKA?;"); // Marker Amplitude Query ?
 Receive(); // Read the Amplitude value
 Printf("Amplitude = %s\n",DataBuf);

 Send("MKN?;"); // Marker Frequency Query ?
 Receive(); // Read the Frequency value
 Printf("Frequency = %s\n",DataBuf);

}// the end of main.
```

Delta Marker Measurement

Measuring the difference value of the frequency, amplitude between the normal and the delta marker, and 1/delta.

1. Set
   a. Center Frequency : 500 ㎒
   b. Span : 500 ㎑
   c. Reference Level : -20 dBm
   d. VBW : 30 ㎑
   e. RBW : 10 ㎑
   f. Sweep Time : 100 ㎳
   g. Input Attenuator : Auto
   h. Log 10 dB scale, Unit : dBm
2. Measuring
   a. Peak Search
   b. Marker Frequency to Center Frequency
   c. Marker Level to Reference Level
   d. Delta Marker : Peak Point, Normal Marker : 500.050 ㎒
   e. Read the Normal Marker frequency
   f. Read the difference between the Normal and the Delta Marker
   g. Read 1/Delta

```
//-----------------------------------------------------------------------------------------------
//                                Delta Marker measurement
//-----------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>

    int ud;
    char DataBuf[80]; // Set input message buffer size
    char SpollByte;

    void Send(char *buf)
    [
     ibwrt(ud, buf, (long)strlen(buf));
     if (ibsta & ERR) printf("ibwrt Error\n");
    ]
```

```
void Receive(void)
{
 // Is data on 2394 buffer to read ?
 ibwait(ud, (TIMO | RQS));
 if (ibsta & (ERR | TIMO)) printf("ibwait Error\n");

 ibrsp(ud, &SpollByte);
 if (ibsta & ERR) printf("ibrsp Error\n");
 if (SpollByte != 0x50) printf("2394 Polling Erro\n");

 // read data.
 ibrd(ud, DataBuf, 80L);
 DataBuf[ibcntl-1] = '\0';
 if (ibsta & ERR) printf("ibrd Error\n");
}

void InitGPIB(void)
{
 // Initialize GPIB bus and 2394
 ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
 if(ud<0) printf("2394 device open error\n");

 ibclr(ud);
 if(ibsta & ERR) { printf("ibclr error\n"); exit(1);}
 else  printf("Init Ok\n");
}

void main(void)
{
 prinf("<<<Delta Marker measurement>>>\n");

 InitGPIB();

 Send("*CLS;*SRE 16;"); // Set 2394 to its initial state for programming

 Send("CF 500 MHz;SP 500 kHz;"); // Center Frequency 500 MHz Span 500 kHz

 Send("VB MAN;VB 30 kHz;"); // VBW Manual, VBW 30 kHz
 Send("RB MAN;RB 10 kHz;"); // RBW Manual, RBW 10 kHz
 Send(ST MAN;ST 100MS;"); // Sweep Time Manual, Sweep Time 100 ms

 Send("AT AUTO;"); // Input Attenuator Auto
 Send("RL -20 DBM;") // Reference Level -20 dBm
 Send("LG 10 DB;"); // Log 10 dB scale
 Send("AUNITS DBM;*OPC?;"); // dBm unit
```

```
Receive(); // Waiting the commands completed

Send("MKPK;"); // Peak Search

Send("MKCF;*OPC?;"); // Marker Frequency → Center Frequency
Receive(); // Waiting the commands completed
Send("MKRL;*OPC?;"); // Marker Level → Reference Level
Receive(); // Waiting the commands completed

Send("MKD;"); // Delta Mark : Peak point, Normal Marker : Peak point
Send("MKN 500.050 MHz;"); // Delta Mark : Peak point, Normal Marker : 500.050 MHz

Send("MKN?;"); // Read the normal marker frequency
Receive();
printf("Normal Marker Frequency = %s\n", DataBuf);

Send("MKA?;"); // Read the amplitude difference between the Delta and the Normal marker
Receive();
printerf("Delta Amplitude = %s\n", DataBuf);

Send("MKTF?;"); // Read the difference frequency
Receive();
printerf("Delta Frequency = %s\n",DataBuf);

Send("MKDTF;"); // Set 1/Delta
Send("MKTF?;"); // Read 1/Delta
Receive();
printf("1/Delta = %s\n", DataBuf);

}// the end of main
```

Frequency Bandwidth

Searches the X dB point from the normal marker and measures X dB frequency bandwidth. (X is 6 dB on this example code.)

1. Set
   a. Center Frequency : 100 ㎒
   b. Span : 500 ㎑
   c. Reference Level : -10 dBm
   d. VBW : 10 ㎑
   e. RBW : 30 ㎑
   f. Sweep Time : Auto
   g. Input Attenuator : Auto
   h. Log 10 dB scale, Unit : dBm
2. Measuring
   a. Peak Search
   b. Marker Frequency to Center Frequency
   c. Marker Level to Reference Level
   d. Signal Sweep
   e. Set 6 dB down point from the normal marker
   f. Read 6 dB frequency bandwidth
   g. Stop X dB down
   h. Continuous Sweep

```
//-----------------------------------------------------------------------------------------------------
//                              Frequency Bandwidth measurement
//-----------------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>

    int ud;
    char DataBuf[80]; // Set input message buffer size

    void Send(char *buf)
    {
     ibwrt(ud, buf, (long)strlen(buf));
     if (ibsta & ERR) printf("ibwrt Error\n");
    }
```

```
void Receive(void)
{
 // read data
 ibwait(ud, DataBuf, 80L);
 DataBuf[ibcntl-1] = "\0";
 if (ibsta & ERR) printf("ibrd Error\n");
}

void InitGPIB(void)
{
 // Initialize GPIB bus and 2394
 ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
 if(ud<0) printf("2394 device open error\n");

 ibclr(ud);
 if(ibsta & ERR) {prinft("ibclr error\n"); exit(1);}
 else  printf("Init Ok\n");
}

void main(void)
{
 printf("<<<Frequency Bandwidth measurement>>>\n");

 InitGPIB();

 Send("*CLS;"); // Set 2394 to its initial state for programming

 Send("CF 100 MHz;"); // Center Frequency 100 MHz
 Send("SP 500 kHz;"); // Span 500 kHz

 Send("RL -10 DBM;"); // Reference Level -10dBm
 Send("AT AUTO;ST AUTO;"); // Input Attenuator Auto, Sweep Time auto
 Send("LG 10 DB; AUNITS DBM;"); // Log 10 dB scale, dBm unit

 Send("VB MAN;VB 10 kHz;"); // VBW Manual, VBW 10 kHz
 Send("RB MAM;RB 30 kHz;*OPC?;"); // RBW Manual, RBW 30 kHz
 Receive(); // Waiting the commands completed

 Send("MKPK;") // Peak Search

 Send("MKCF;*OPC?;"); // Marker Frequency → Center Frequency
 Receive(); // Waiting the commands completed
```

```
Send("MKRL:*OPC?;"); // Marker Amplitude → Reference Level
Receive(); // Waiting the commands completed

Send("TRGSWP SNG;"); // Single Sweep for measuring

Send("SXDBPT 6DB;*OPC?;"); // 6 dB down, Left and Right down from the Normal Marker
Receive(); // Waiting the command completed
Send("MSTART XDB;");

Send("XDRL?;"); // Read X dB down Frequency Bandwidth
Receive();
printf("Frequency Bandwidth = %s\n",DataBuf);

Send("MSTOP;"); // Stop X dB down measurement

Send("MCLRA;"); //  For screen Only

Send("TRGSWP CNT;"); // Continuous Sweep

}// the end of main
```

Occupied Bandwidth Measurement

Sets the normal marker on the carrier-frequency of the occupied band center frequency, and calculates OBW (Occupied Power Bandwidth).

1. Set
   a. Detection Mode : Sample
   b. Center Frequency : 100 ㎒
   c. Span : 2 ㎒
   d. Input Attenuator : Auto
   e. Unit : dBm, Log 10 dB scale
   f. Reference Level : -25 dBm
   g. RBW : 10 ㎑
   h. VBW : 1 ㎑
   i. Sweep Time : Auto
2. Measuring
   a. Peak Search
   b. Marker Frequency to Center Frequency
   c. Set OBW 50%
   d. Waiting for OBW calculation completed
   e. Read OBW

```
//-----------------------------------------------------------------------------------------------------------
//                                        OBW measurement
//-----------------------------------------------------------------------------------------------------------
   #include <windows.h>
   #include "Decl-32.h"  // NI Library
   #include <stdio.h>

   int ud;
   char DataBuf[80]; // Set input message buffer size
   char SpollByte;

   void Send(char *buf)
   {
    ibwrt(ud, buf, (long)strlen(buf));
    if (ibsta & ERR) printf("ibwrt Error\n");
   }

   void Receive(void)
   {
    ibrd(ud, DataBuf, 80L);
    DataBuf[ibcntl-1] = '\0';
    If (ibsta & ERR) printf("ibrd Error\n");
   }
```

```
viod InitGPIB(void)
[
 // Initialize GPIB bus and 2394
 ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
 if(ud<0) printf("2394 device open error\n");

 ibclr(ud);
 if(ibsta & ERR) [printf("ibclr error\n"); exit(1); ]
 else  printf("Init Ok\n");
]

void main(void)
[
 printf("<<<OBW measurement>>>\n");

 InitGPIB();

 Send("*CLS;"); // Set 2394 to its initial state for programming

 Send("DET SAM;"); // Sets the detection mode to SAMPLE

 Send("CF 100 MHz;") // Center Frequency 100 MHz
 Send("SP 2 MHz;") // Span 2 MHz

 Send("AT AUTO;ST AUTO;"); // Input Attenuator Auto, Sweep time auto

 Send("AUNITS DBM;"); // dBm unit
 Send("LG 10 DB;"); // Log 10 dB scale
 Send("RL -25 DBM;"); // Reference Level -25 dBm

 Send("RB MAN;RB 10 kHz;"); // RBW Manual, RBW 10 kHz

 Send("VB MAN;VB 1 kHz;*OPC?;") // VBW Manual, VBW 1 kHz
 Receive(); // Waiting the commands completed

 Send("MKPK;"); // Peak Search

 Send("MKCF;*OPC?;"); // Marker Frequency → Center Frequency
 Receive(); // Waiting the commands completed

 Send("SOBWP 50;*OPC?;"); // set OBW 50%
 Receive(); // Waiting the commands completed
 Send("SOBWSP 2 MHz;")' // Set OBW SPAN Value
 Send("MSTART OBW;"); //  OBW Measure Start
 Send("OCF?;"); // Query OBW ?
 Receive(); // Read OBW
 printf("OBW = %s\n",DataBuf);
 Send("MSTOP;"); // Measure Stop
]// the end of main
```

Marker Noise Measurement

Sets the reference marker on the signal, and the normal marker on the noise, then measures Marker Noise.

1. Set
   a. Center Frequency : 300 ㎒
   b. Span : 1 ㎒
   c. Reference Level : -20 dBm
   d. Input Attenuator : Auto
   e. Log 10 dB scale, Unit : dBm
   f. RBW : 10 ㎑
   g. VBW : Auto
   h. Sweep Time : 50 ㎳
2. Measuring
   a. Delta Marker : Peak point, Normal Marker : 300.100 ㎒
   b. Read Marker Noise

```
//-----------------------------------------------------------------------------------------------------------
//                                    Marker Noise measurement
//-----------------------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>

    int ud;
    char DataBuf[80]; // Set input message buffer size
    char SpollByte;

    void Send(char *buf)
    {
     ibwrt(ud, buf, (long)strlen(buf));
     if (ibsta & ERR) printf("ibwrt Error\n");
    }

    void Receive(void)
    {
     ibrd(ud, DataBuf, 80L);
     DataBuf[ibcntl-1] = '\0';
     If (ibsta & ERR) printf("ibrd Error\n");
    }

    viod InitGPIB(void)
    {
     // Initialize GPIB bus and 2394
```

```
 ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
 if(ud<0) printf("2394 device open error\n");

 ibclr(ud);
 if(ibsta & ERR) {printf("ibclr error\n"); exit(1); }
 else  printf("Init Ok\n");
}

void main(void)
{
 printf("<<<Marker Noise measurement>>>\n");

 InitGPIB();

 Send("*CLS;"); // Set 2394 to its initial state for programming

 Send("CF 300 MHz;") // Center Frequency 300 MHz
 Send("SP 1 MHz;") // Span 1 MHz

 Send("RL -20 DBM;") // Reference Level -20 dBm
 Send("AT AUTO;"); // Input Attenuator Auto
 Send("LG 10 DB;"); // Log 10 dB Scale
 Send("AUNITS DBM;"); // dBm Scale

 Send("RB MAN;RB 10 kHz;"); // RBW Manual, RBW 10 kHz

 Send("VB AUTO;"); // VBW Auto

 Send("ST MAN;ST 50MS;*OPC?;"); // Sweep Time Manual, Sweep Time 50 ms
 Receive(); // Waiting the commands completedRec

 Send("MKPK;"); // Peak Search

 Send("MKRL;*OPC?;"); // Marker Level → Reference Level
 Receive(); // Waiting the commands completed

 Send("MKD;"); // Delta Mark : Peak point, Normal Marker : Peak point

 Send("MKN 300.100 MHz;"); // Delta Mark : Peak point, Normal Marker : 300.100 MHz

 Send("MKNOISE ON;*OPC?;");
 Receive(); // Waiting the commands completed

 Send("MKNOISE?;");
 Receive(); // Read Marker Noise
 printf("Marker Noise = %s\n",DataBuf);

}// the end of main
```

Saving Data

Saves the current system status to current disk

```
//-------------------------------------------------------------------------------------------------
//                        Save the current status to current disk
//-------------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>
    #include <stdlib.h>

    int ud;
    char DataBuf[80]; // Set input message buffer size
    char SpollByte;

    void Send(char *buf)
    {
     ibwrt(ud, buf, (long)strlen(buf));
     if (ibsta & ERR) printf("ibwrt Error\n");
    }

    void Receive(void)
    {
     ibrd(ud, DataBuf, 80L);
     DataBuf[ibcntl-1] = '\0';
     If (ibsta & ERR) printf("ibrd Error\n");
    }

    viod InitGPIB(void)
    {
     // Initialize GPIB bus and 2394
     ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
     if(ud<0) printf("2394 device open error\n");

     ibclr(ud);
     if(ibsta & ERR) [printf("ibclr error\n"); exit(1); ]
     else  printf("Init Ok\n");
    }

    void main(void)
    {
     printf("<<<Save the current status to current disk>>>\n");
```

```
 InitGPIB();

 Send("*CLS;"); // Set 2394 to its initial state for programming

 Send("FSAVE TEST001.STS;");

}// the end of main
```

Recalling Data

Recalls the system status from current disk.

```
//-------------------------------------------------------------------------------------------------
//                    Recall the current status from current disk
//-------------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>
    #include <stdlib.h>

    int ud;
    char DataBuf[80]; // Set input message buffer size
    char SpollByte;

    void Send(char *buf)
    {
     ibwrt(ud, buf, (long)strlen(buf));
     if (ibsta & ERR) printf("ibwrt Error\n");
    }

    void Receive(void)
    {
     ibrd(ud, DataBuf, 80L);
     DataBuf[ibcntl-1] = '\0';
     If (ibsta & ERR) printf("ibrd Error\n");
    }

    viod InitGPIB(void)
    {
     // Initialize GPIB bus and 2394
     ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
     if(ud<0) printf("2394 device open error\n");

     ibclr(ud);
     if(ibsta & ERR) {printf("ibclr error\n"); exit(1); }
     else  printf("Init Ok\n");
    }

    void main(void)
    {
     printf("<<<Recall the current status from current disk>>>\n");
```

```
 InitGPIB();

 Send("*CLS;"); // Set 2394 to its initial state for programming

 Send("FLOAD TEST001.STS;");

}// the end of main
```

Get Trace Data

Get all trace data from the equipment.

```
//--------------------------------------------------------------------------------------------------
//                                  Get all the Trace Data
//--------------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>
    #include <stdlib.h>

    int ud;
    unsigned char DataBuf[4096]; // Set Input message buffer size
                        // [3600.255.....]→ "3600." : 5byte 5byte×500 + ... → about 4096
    char SpollByte;

    void Send(char *buf)
    {
     ibwrt(ud, buf, (long)strlen(buf));
     if (ibsta & ERR) printf("ibwrt Error\n");
    }

    void Receive(void)
    {
     // Is data on 2394 buffer to read ?
     ibwait(ud, (TIMO | RQS));
     if (ibsta & (ERR | TIMO)) printf("ibwait Error\n");

     ibrsp(ud, &SpollByte);
     if (ibsta & ERR) printf("ibrsp Error\n");
     if (SpollByte != 0x50) printf("2394 Polling Error\n");

     // read data.
     ibrd(ud, DataBuf, 4096L);
     DataBuf[ibcntl-1] =  '\0';
     if(ibsta & ERR) printf("ibrd Error\n");
    }

    void InitGPIB(void)
    {
```

```
 // Initialize GPIB bus and 2394
 ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
 if(ud<0) printf("2394 device open error\n");

 ibclr(ud);
 if(ibsta & ERR) { printf("ibclr error\n"); exit(1); }
 else  printf("Init Ok\n");
}

void main(void)
{
 int i,j,n;
 unsigned char c;
 char TempBuf[10];
 short int TraceData[510];
 unsigned char Title[30];
 short int Temp16BitInt;

 printf("<<<Get all the Trace Data>>>\n");

 InitGPIB();

 Send("*CLS;*SRE 16;"); // Set 2394 to its initial state for programming

 //---------------- For the binary type ----------------
    Send("TDF BIN;");
 //-----------------------------------------------------

 //---------------- For the decimal type ----------------
 // Send("TDF DEC;");
 // -----------------------------------------------------

 Sen("TRAALL?;");
 Receive();

 for(i=0; DataBuf[i] != '['; i++]
 {
  Title[i];
 }

 Title[i] = '\0';
 printf("--%s--\n",Title); // Display Title
```

```
 n = i + 1;

 //---------------- For the binary type ----------------
 for(i= 0;i <= 499; i++)
 {
  Temp16BitInt = DataBuf[n++]; // upper byte
  Temp16BitInt <<= 8;
  Temp16BitInt += DataBuf[n++]; // lower byte

  TraceData[i] = Temp16BitInt;
  printf(TRACE[%d] = %d\n", i, TraceData[i]);
 }
 //-----------------------------------------------------

 /*
 //---------------- For the decimal type ----------------
 for(i= 0;i <= 499; i++)
 {
  j=0;

  do{
    c = DataBuf[n++];
    TempBuf[j++]= c;
  }while ( isdigit(c) )

  TemBuf[j] = '\0';
  TraceData[i] = atoi(TempBuf);
  Printf("TRACE[%d] = %d\n", i, TraceData[i];
 }
 //-----------------------------------------------------
 */

}// the end of main
```

Pass/Fail Check

Check PASS or FAIL by comparing the current waveform with the upper limit line or the lower limit line.

```
//-----------------------------------------------------------------------------------------------------------
//                                        Pass / Fail Check
//-----------------------------------------------------------------------------------------------------------
    #include <windows.h>
    #include "Decl-32.h"  // NI Library
    #include <stdio.h>

     int ud;
     char DataBuf[80]; // Set Input message buffer size

     void Send(char *buf)
     {
      ibwrt(ud, buf, (long)strlen(buf));
      if (ibsta & ERR) printf("ibwrt Error\n");
     }

     void Receive(void)
     {
      ibrd(ud, DataBuf, 80L);
      DataBuf[ibcntl-1] = '\0';
      if (ibsta & ERR) printf("ibrd Error\n");
     }

     void InitGPIB(void)
     {
      // Initialize GPIB bus and 2394
      ud = ibdev(0,7,0,T10s,1,0); // GPIB initialization, set 2394 to address 7
      if(ud<0) printf("2394 device open error\n");

      ibclr(ud);
      if(ibsta & ERR) { printf("ibclr error\n"); exit(1); }
      else  printf("Init Ok\n");
     }
    void main(void)
    {
     printf("<<<Pass / Fail Check>>>\n");
```

```
InitGPIB();

Send("*CLS;"); // Set 2394 its initial state for programming

// Already, the Limit mask data had to be saved in current disk.
// Recall the limit mask data from current disk.
// When Loading is completed, The configuration is replaced by the data to have saved.
Send("FLOAD TEST001.STS;"); // load the limit mask data current disk.

Send("LMTUP ON;"); // upper limit on
Send("LMTLW ON;"); // lower limit on

Send("LMTPC ON;*OPC?;"); // pass/fail check RUN, check LMTPC completed.
Receive(); // Waiting the commands completed

Send("LMTPC?;"); // Query ? Pass/Fail
Receive(); // Read the result pass/fail check
printf("Pass/Fail Result = %s\n",DataBuf);

}// the end of main
```

<BLANK>

APPENDIX-PROGRAMMING COMMANDS

CATALOG ORDER

Function Group	Command	Description	Page
MKR/Counter	MKFC	Frequency Counter	4-31
MKR/Counter	MKFCR	Frequency Counter Resolution	4-32
MKR/Quasi Peak	MKFQP	Marker Quasi Peak	4-33
MKR/Quasi Peak	SELMKQP	Select band Maker Quasi Peak	4-33
MKR/MKR Shift	MKCF	Marker to Center Frequency	4-34
MKR/MKR Shift	MKSS	Marker to Center Frequency Step Size	4-34
MKR/MKR Shift	MKFA	Marker to Start Frequency	4-35
MKR/MKR Shift	MKFB	Marker to Stop Frequency	4-35
MKR/MKR Shift	MKRL	Marker to Reference Level	4-36
MKR/MKR Shift	MKSP	Marker Delta to Span	4-36
MKR/MKR Shift	MKDSS	Marker Delta to Center Frequency Step Size	4-37
MKR/MKR Shift	MKZI	Marker Zoom-In	4-38
MKR/MKR Shift	MKZO	Marker Zoom-Out	4-38
MKR/PEAK	MKPK	Peak Search	4-39
MKR/PEAK	MKMIN	Marker Minimum Search	4-39
MKR/PEAK	MKPP	Marker Peak to Peak Search	4-40
MKR/PEAK	MKTRACK	Signal Track	4-40
MKR/PEAK	MKPX	Marker Peak Search Excursion	4-41
MKR/PEAK	MKPT	Marker Peak Search Threshold	4-42
MKR/PEAK	MKPS	Marker Peak Search Parameter	4-43
MKR/PEAK	MMPN	Marker Multi Peak Number	4-44
MKR/PEAK	MMP	Marker Multi Peak	4-44
TRIG	TRGSWP	Trigger Sweep	4-45
TRIG	TM	Trigger Source	4-46
TRIG	TLV	Trigger Level	4-47
TRIG	TDLY	Delay Time	4-48
TRIG	TE	Trigger Edge	4-49
TRIG/Time Gate	GATE	Time Gate	4-50
TRIG/Time Gate	GATECTL	Time Gate Control	4-51
TRIG/Time Gate	GD	Time Gate Delay	4-52
TRIG/Time Gate	GL	Time Gate Length	4-53
TRIG/Time Gate	GP	Time Gate Polarity	4-54
CPL	AUTOCPL	Auto Coupled	4-55
CPL	RB	Resolution Bandwidth	4-56
CPL	VB	Video Bandwidth	4-57
CPL	ST	Sweep Time	4-58
DISP	DL	Display Line	4-59
DISP	TH	Threshold	4-60

Function Group	Command	Description	Page
DISP	TITLE	Screen Title Entry	4-61
DISP	GRAT	Graticule	4-62
TRACE	TRS	Trace Status	4-63
TRACE	TRF	Trace Function	4-64
TRACE	TRA/TRB	Trace Data Input/Output	4-65
TRACE	TRAALL/TRBALL	Trace All Data Output	4-66
TRACE	TDF	Trace Data Format	4-67
TRACE/Math..	AMB	Trace A Minus Trace B	4-68
TRACE/Math..	BML	Trace B Minus Display Line	4-68
TRACE/Math..	APB	Trace A Plus Trace B	4-69
TRACE/Math..	AMBPL	Trace A Minus Trace B Plus Display Line	4-69
TRACE/Math..	AXB	Trace A Exchange Trace B	4-70
TRACE/Detect..	DET	Detection Mode	4-71
TRACE/Average..	AVG	Trace Average	4-72
TRACE/Average..	AVGC	Number of Trace Average	4-73
TRACE/Average..	AVGCYL	Average Cycle	4-73
TRACE/Average..	AVGS	Average Control	4-74
FILE	FDRV	File Disk Driver Selection	4-75
FILE	FSAVE	File Save	4-76
FILE	FLOAD	File Load	4-76
FILE	FDEL	File Delete	4-77
FILE	FCPT	File Copy To	4-77
FILE	MEM	Memory Available	4-78
LIMIT	LMTPC	Limit Line Function	4-79
LIMIT	ALARM	Pass/Fail Alarm	4-80
LIMIT	LMTUP	Limit Line Upper Area	4-80
LIMIT	LMTLW	Limit Line Lower Area	4-81
MEAS	MSTART	Measure Start	4-82
MEAS	MSTOP	Measure Stop	4-82
MEAS	MCONT	Measure Continuous	4-83
MEAS	MCLRA	Measure All Clear	4-83
MEAS/XdB	SXDBPT	Set X dB Point	4-84
MEAS/XdB	XDL	X dB Left	4-84
MEAS/XdB	XDR	X dB Right	4-85
MEAS/XdB	XDRL	X dB Relative	4-85
MEAS/ACP	SACPMCH	Set ACP Main Channel	4-86
MEAS/ACP	SACPACH	Set ACP Adjacent Channel	4-87
MEAS/ACP	SACPSCH	Set ACP Channel Space	4-88

Function Group	Command	Description	Page
MEAS/ACP	SACPAVG	Set ACP Average	4-89
MEAS/ACP	AMC	ACP Main Channel	4-90
MEAS/ACP	ALC	ACP Lower Channel	4-90
MEAS/ACP	AUC	ACP Upper Channel	4-91
MEAS/CP	SCPBW	Set Channel Power BandWidth	4-92
MEAS/CP	SCPSP	Set Channel Power Span	4-93
MEAS/CP	SCPMH	Set Channel Power MaxHold	4-93
MEAS/CP	SCPAVG	Set Channel Power Average	4-94
MEAS/CP	CPWR	Channel Power	4-94
MEAS/OBW	SOBWSP	Set Occupied Bandwidth Span	4-95
MEAS/OBW	SOBWP	Set OBW Power Percentage	4-96
MEAS/OBW	OCP	OBW Channel Power	4-96
MEAS/OBW	OCF	OBW Channel Frequency	4-97
MEAS/HD	SHDN	Set Harmonic Distortion Number	4-98
MEAS/HD	SHDAVG	Set Harmonic Distortion Average	4-98
MEAS/HD	HF1	Harmonic Fundamental Frequency (1st)	4-99
MEAS/HD	HA1	Harmonic Fundamental Amplitude (1st)	4-99
MEAS/HD	HF2	Harmonic Distortion Frequency (2nd)	4-100
MEAS/HD	HA2	Harmonic Distortion Amplitude (2nd)	4-100
MEAS/HD	HF3	Harmonic Distortion Frequency (3rd)	4-101
MEAS/HD	HA3	Harmonic Distortion Amplitude (3rd)	4-101
MEAS/HD	HF4	Harmonic Distortion Frequency (4th)	4-102
MEAS/HD	HA4	Harmonic Distortion Amplitude (4th)	4-102
MEAS/HD	HF5	Harmonic Distortion Frequency (5th)	4-103
MEAS/HD	HA5	Harmonic Distortion Amplitude (5th)	4-103
MEAS/HD	THD	Total Harmonic Distortion	4-104
AUX	DEMOD	Demodulation	4-105
AUX	AUDIO	Speaker	4-105
AUX	SQL	Squelch Level	4-106
AUX	AUDIOVR	Speaker Volume	4-106
PRESET	PRST	Preset	4-107
PRESET	CALALL	All Calibrations	4-107
PRESET	PCAL	Periodic Temperature Calibration	4-108
PRESET	YIGCAL	First Local Oscillator Calibration	4-108
PRESET	RCAL	RBW Calibration	4-109
PRESET	TMPCAL	Temperature Calibration	4-109
PRESET	LVLC	Level Calibration	4-110
PRESET	SPCAL	Span Calibration	4-111

Function Group	Command	Description	Page
PRINT	HCOPY	Hard Copy	4-112
SYSTEM/Clock Set..	DATE	Set Date	4-113
SYSTEM/Clock Set..	TIME	Set Time	4-114
GPIB	*CLS	Clear Status Commend	4-115
GPIB	*ESE	Standard Event Status Enable	4-115
GPIB	*ESR?	Standard Event Status Register Query	4-116
GPIB	*IDN?	Identification Query	4-116
GPIB	*OPC	Operation Complete Commend	4-117
GPIB	*OPC?	Operation Complete Query	4-117
GPIB	*RST	Reset Command	4-118
GPIB	*SRE	Service Request Enable Commend	4-118
GPIB	*STB?	Returns Status Byte Commend	4-119
GPIB/Ext	ESE2	Event Status Enable (End)	4-120
GPIB/Ext	ESR2?	Event Status Register (End) Query	4-121
GPIB/Ext	QRYTYP	Query Response Type	4-122
GPIB/Ext	ERR	Error Number	4-122
Quasi Peak	QPM	Quasi Peak Mode	4-123

APPENDIX-PROGRAMMING COMMANDS

ALPHABET ORDER

Function Group	Command	Description	Page
GPIB	*CLS	Clear Status Commend	4-115
GPIB	*ESE	Standard Event Status Enable	4-115
GPIB	*ESR?	Standard Event Status Register Query	4-116
GPIB	*IDN?	Identification Query	4-116
GPIB	*OPC	Operation Complete Commend	4-117
GPIB	*OPC?	Operation Complete Query	4-117
GPIB	*RST	Reset Command	4-118
GPIB	*SRE	Service Request Enable Commend	4-118
GPIB	*STB?	Returns Status Byte Commend	4-119
LIMIT	ALARM	Pass/Fail Alarm	4-80
MEAS/ACP	ALC	ACP Lower Channel	4-90
TRACE/Math..	AMB	Trace A Minus Trace B	4-68
TRACE/Math..	AMBPL	Trace A Minus Trace B Plus Display Line	4-69
MEAS/ACP	AMC	ACP Main Channel	4-90
TRACE/Math..	APB	Trace A Plus Trace B	4-69
AMPL	AT	Input Attenuation	4-16
MEAS/ACP	AUC	ACP Upper Channel	4-91
AUX	AUDIO	Speaker	4-105
AUX	AUDIOVR	Speaker Volume	4-106
AMPL	AUINTS	Absolute Amplitude Units	4-18
CPL	AUTOCPL	Auto Coupled	4-55
Auto Tune	AUTOTUNE	Auto Tune	4-11
TRACE/Average..	AVG	Trace Average	4-72
TRACE/Average..	AVGC	Number of Trace Average	4-73
TRACE/Average..	AVGCYL	Average Cycle	4-73
TRACE/Average..	AVGS	Average Control	4-74
TRACE/Math..	AXB	Trace A Exchange Trace B	4-70
TRACE/Math..	BML	Trace B Minus Display Line	4-68
PRESET	CALALL	All Calibrations	4-107
AMPL	CALSIG	Calibration Signal	4-21
FREQ	CF	Center Frequency	4-5
MEAS/CP	CPWR	Channel Power	4-94
SYSTEM/Clock Set..	DATE	Set Date	4-113

Function Group	Command	Description	Page
AUX	DEMOD	Demodulation	4-105
TRACE/Detect..	DET	Detection Mode	4-71
DISP	DL	Display Line	4-59
MKR	DMKR	Delete Maker Number	4-23
GPIB/Ext	ERR	Error Number	4-122
GPIB/Ext	ESE2	Event Status Enable (End)	4-120
GPIB/Ext	ESR2?	Event Status Register (End) Query	4-121
FREQ	FA	Start Frequency	4-6
FREQ	FB	Stop Frequency	4-7
FILE	FCPT	File Copy To	4-77
FILE	FDEL	File Delete	4-77
FILE	FDRV	File Disk Driver Selection	4-75
FILE	FLOAD	File Load	4-76
FREQ	FOFFS	Frequency Offset	4-9
SPAN	FS	Full Span	4-13
FILE	FSAVE	File Save	4-76
TRIG/Time Gate	GATE	Time Gate	4-50
TRIG/Time Gate	GATECTL	Time Gate Control	4-51
TRIG/Time Gate	GD	Time Gate Delay	4-52
TRIG/Time Gate	GL	Time Gate Length	4-53
TRIG/Time Gate	GP	Time Gate Polarity	4-54
DISP	GRAT	Graticule	4-62
MEAS/HD	HA1	Harmonic Fundamental Amplitude (1st)	4-99
MEAS/HD	HA2	Harmonic Distortion Amplitude (2nd)	4-100
MEAS/HD	HA3	Harmonic Distortion Amplitude (3rd)	4-101
MEAS/HD	HA4	Harmonic Distortion Amplitude (4th)	4-102
MEAS/HD	HA5	Harmonic Distortion Amplitude (5th)	4-103
PRINT	HCOPY	Hard Copy	4-112
MEAS/HD	HF1	Harmonic Fundamental Frequency (1st)	4-99
MEAS/HD	HF2	Harmonic Distortion Frequency (2nd)	4-100
MEAS/HD	HF3	Harmonic Distortion Frequency (3rd)	4-101
MEAS/HD	HF4	Harmonic Distortion Frequency (4th)	4-102
MEAS/HD	HF5	Harmonic Distortion Frequency (5th)	4-103
AMPL	INPUTZ	Input Impedance	4-19
AMPL	INTAMP	Internal Amplifier	4-21
AMPL	LG	Logarithm Amplitude Mode & Scale	4-17
LIMIT	LMTLW	Limit Line Lower Area	4-81
LIMIT	LMTPC	Limit Line Function	4-79

Function Group	Command	Description	Page
LIMIT	LMTUP	Limit Line Upper Area	4-80
AMPL	LN	Linear Mode	4-17
PRESET	LVLC	Level Calibration	4-110
MEAS	MCLRA	Measure All Clear	4-83
MEAS	MCONT	Measure Continuous	4-83
FILE	MEM	Memory Available	4-78
MKR	MKA	Marker Amplitude	4-25
MKR/MKR Shift	MKCF	Marker to Center Frequency	4-34
MKR	MKD	Marker Delta	4-26
MKR/MKR Shift	MKDSS	Marker Delta to Center Frequency Step Size	4-37
MKR	MKDTF	Marker 1/Delta	4-26
MKR/MKR Shift	MKFA	Marker to Start Frequency	4-35
MKR/MKR Shift	MKFB	Marker to Stop Frequency	4-35
MKR/Counter	MKFC	Frequency Counter	4-31
MKR/Counter	MKFCR	Frequency Counter Resolution	4-32
MKR/Quasi Peak	MKFQP	Marker Quasi Peak	4-33
MKR/PEAK	MKMIN	Marker Minimum Search	4-39
MKR	MKN	Normal Marker	4-24
MKR/MKR Noise	MKNOISE	Marker Noise	4-28
MKR	MKOFF	Marker Off	4-27
MKR/PEAK	MKPK	Peak Search	4-39
MKR/Phase Noise	MKPN	Phase Noise	4-29
MKR/Phase Noise	MKPNO	Phase Noise Offset	4-30
MKR/PEAK	MKPP	Marker Peak to Peak Search	4-40
MKR/PEAK	MKPS	Marker Peak Search Parameter	4-43
MKR/PEAK	MKPT	Marker Peak Search Threshold	4-42
MKR/PEAK	MKPX	Marker Peak Search Excursion	4-41
MKR/MKR Shift	MKRL	Marker to Reference Level	4-36
MKR/MKR Shift	MKSP	Marker Delta to Span	4-36
MKR/MKR Shift	MKSS	Marker to Center Frequency Step Size	4-34
MKR	MKTF	Read the marker frequency or time	4-27
MKR/PEAK	MKTRACK	Signal Track	4-40
MKR/MKR Shift	MKZI	Marker Zoom-In	4-38
MKR/MKR Shift	MKZO	Marker Zoom-Out	4-38
MKR/PEAK	MMP	Marker Multi Peak	4-44
MKR/PEAK	MMPN	Marker Multi Peak Number	4-44
MEAS	MSTART	Measure Start	4-82
MEAS	MSTOP	Measure Stop	4-82

Function Group	Command	Description	Page
MKR	NMKR	New Maker Number	4-22
MEAS/OBW	OCF	OBW Channel Frequency	4-97
MEAS/OBW	OCP	OBW Channel Power	4-96
PRESET	PCAL	Periodic Temperature Calibration	4-108
PRESET	PRST	Preset	4-107
Quasi Peak	QPM	Quasi Peak Mode	4-123
GPIB/Ext	QRYTYP	Query Response Type	4-122
CPL	RB	Resolution Bandwidth	4-56
PRESET	RCAL	RBW Calibration	4-109
FREQ	REFLO	Reference Clock	4-10
AMPL	RL	Reference Level	4-15
AMPL	RLO	Level Offset	4-20
MEAS/ACP	SACPACH	Set ACP Adjacent Channel	4-87
MEAS/ACP	SACPAVG	Set ACP Average	4-89
MEAS/ACP	SACPMCH	Set ACP Main Channel	4-86
MEAS/ACP	SACPSCH	Set ACP Channel Space	4-88
MEAS/CP	SCPAVG	Set Channel Power Average	4-94
MEAS/CP	SCPBW	Set Channel Power BandWidth	4-92
MEAS/CP	SCPMH	Set Channel Power MaxHold	4-93
MEAS/CP	SCPSP	Set Channel Power Span	4-93
MKR	SELMK	Select Maker Number	4-22
MKR/Quasi Peak	SELMKQP	Select band Maker Quasi Peak	4-33
MKR	SELMRO	Select Maker Read Out	4-23
MEAS/HD	SHDAVG	Set Harmonic Distortion Average	4-98
MEAS/HD	SHDN	Set Harmonic Distortion Number	4-98
MEAS/OBW	SOBWP	Set OBW Power Percentage	4-96
MEAS/OBW	SOBWSP	Set Occupied Bandwidth Span	4-95
SPAN	SP	Frequency Span	4-12
PRESET	SPCAL	Span Calibration	4-111
AUX	SQL	Squelch Level	4-106
FREQ	SS	Center Frequency Step Size	4-8
CPL	ST	Sweep Time	4-58
MEAS/XdB	SXDBPT	Set X dB Point	4-84
TRACE	TDF	Trace Data Format	4-67
TRIG	TDLY	Delay Time	4-48
TRIG	TE	Trigger Edge	4-49
DISP	TH	Threshold	4-60
MEAS/HD	THD	Total Harmonic Distortion	4-104
SYSTEM/Clock Set..	TIME	Set Time	4-114

Function Group	Command	Description	Page
DISP	TITLE	Screen Title Entry	4-61
TRIG	TLV	Trigger Level	4-47
TRIG	TM	Trigger Source	4-46
PRESET	TMPCAL	Temperature Calibration	4-109
TRACE	TRA/TRB	Trace Data Input/Output	4-65
TRACE	TRAALL/TRBALL	Trace All Data Output	4-66
TRACE	TRF	Trace Function	4-64
TRIG	TRGSWP	Trigger Sweep	4-45
TRACE	TRS	Trace Status	4-63
CPL	VB	Video Bandwidth	4-57
MEAS/XdB	XDL	X dB Left	4-84
MEAS/XdB	XDR	X dB Right	4-85
MEAS/XdB	XDRL	X dB Relative	4-85
PRESET	YIGCAL	First Local Oscillator Calibration	4-108
SPAN	ZI	Zoom In	4-14
SPAN	ZO	Zoom Out	4-14
SPAN	ZS	Zero Span	4-13

APPENDIX – MUTUAL REFERENCE INDEX

OPERATION MANUAL ORDER

OPERATION MANUAL				PROGRAMMING MANUAL	
HARD KEY	UPPER MENU	SOFTKEY	PAGE	RCI COMMAND	PAGE
FREQ		Center	5-7	CF	4-5
FREQ		Start	5-8	FA	4-6
FREQ		Stop	5-9	FB	4-7
FREQ		CF Step	5-9	SS	4-8
FREQ		Freq. Offset	5-10	FOFFS	4-9
FREQ	More..	10 MHz Ref.	5-12	REFLO	4-10
SPAN		WidthSpan	5-8	SP	4-12
SPAN		Full Span	5-10	FS	4-13
SPAN		Zero Span	5-11	ZS	4-13
SPAN		Last Span	5-11	SP	4-12
SPAN		Zoom In	5-11	ZI	4-14
SPAN		Zoom Out	5-11	ZO	4-14
AMPL		Ref. Level	5-13	RL	4-15
AMPL		Atten.	5-15, 5-40	AT	4-16
AMPL		Log	5-13	LG	4-17
AMPL		Linear	5-13	LN	4-17
AMPL		Scale..	5-14	LG	4-17
AMPL		Unit..	5-14	AUNITS	4-18
AMPL	More..	Input Z	5-15	INPUTZ	4-19
AMPL	More..	Ref. Offset	5-16	RLO	4-20
AMPL	More..	Int Amp	5-16	INTAMP	4-21
AMPL	More..	Cal. Out[20M]	5-16	CALSIG	4-21
MEAS	X dB Down..	X[dB] Point	5-17	SXDBPT	4-84
MEAS	X dB Down..	Start	5-17	MSTART	4-82
MEAS	X dB Down..	Stop	5-17	MSTOP	4-82
MEAS	Adjacent CH Power..	MainChBW	5-18	SACPMCH	4-86

OPERATION MANUAL				PROGRAMMING MANUAL	
HARD KEY	UPPER MENU	SOFTKEY	PAGE	RCI COMMAND	PAGE
MEAS	Adjacent CH Power..	AdjChBW	5-18	SACPACH	4-87
MEAS	Adjacent CH Power..	ChSpacing	5-18	SACPSCH	4-88
MEAS	Adjacent CH Power..	Meas. Avg.	5-18	SACPAVG	4-89
MEAS	Adjacent CH Power..	Start	5-18	MSTART	4-82
MEAS	Adjacent CH Power..	Stop	5-18	MSTOP	4-82
MEAS	Channel Power..	Integ. BW	5-18	SCPBW	4-92
MEAS	Channel Power..	Ch PWR Span	5-18	SCPSP	4-93
MEAS	Channel Power..	Max Hold	5-18	SCPMH	4-93
MEAS	Channel Power..	Meas. Avg.	5-18	SCPAVG	4-94
MEAS	Channel Power..	Start	5-18	MSTART	4-82
MEAS	Channel Power..	Stop	5-18	MSTOP	4-82
MEAS	Occupied BandWidth..	OBW Span	5-19	SOBWSP	4-95
MEAS	Occupied BandWidth..	OBW %PWR	5-19	SOBWP	4-96
MEAS	Occupied BandWidth..	Start	5-19	MSTART	4-82
MEAS	Occupied BandWidth..	Stop	5-19	MSTOP	4-82
MEAS	Harmonic Distortion..	Harmonics	5-20	SHDN	4-98
MEAS	Harmonic Distortion..	Averaging	5-20	SHDAVG	4-98
MEAS	Harmonic Distortion..	Start	5-20	MSTART	4-82
MEAS	Harmonic Distortion..	Stop	5-20	MSTOP	4-82
MEAS		Clear Measurement	5-21	MCLRA	4-83
MEAS	More..	Continuous	5-21	MCONT	4-83
MEAS	More.. Quasi-Peak Measure..	QP_B	5-21	QPM	4-123

OPERATION MANUAL				PROGRAMMING MANUAL	
HARD KEY	UPPER MENU	SOFTKEY	PAGE	RCI COMMAND	PAGE
MEAS	More.. Quasi-Peak Measure..	QP_C	5-21	QPM	4-123
MKR		Sel. Marker	5-22	SELMK, NMKR,MKN	4-22, 4-24
MKR		Normal	5-23	NMKR, MKN	4-22, 4-24
MKR		Delta	5-23	MKD	4-26
MKR		OFF	5-24	DMKR	4-23
MKR		MKR Trace	5-24		
MKR	More..	ReadOut..	5-25	SELMRO	4-23
MKR	More.. Function..	MKR Noise	5-26	MKNOISE	4-28
MKR	More.. Function..	Phase Noise	5-26	MKPN, MKPNO	4-29, 4-30
MKR	More.. Function..	Quasi Peak	5-26	MKFQP	4-33
MKR	More.. Function..	Off	5-26		
MKR	More..	MKR Table	5-26		
MKR	More..	Mkr All OFF	5-27	MKOFF	4-27
FC		Frequency Counter	5-27	MKFC, MKFCR	4-31, 4-32
MKR>		Mkr>CF	5-28	MKCF	4-34
MKR>		Mkr>Cfstep	5-29	MKSS	4-34
MKR>		Mkr>Start	5-28	MKFA	4-35
MKR>		Mkr>Stop	5-28	MKFB	4-34
MKR>		Mkr>Ref	5-28	MKRL	4-36
MKR>		Undo			
MKR>	More..	dMkr>Span	5-29	MKSP	4-36
MKR>	More..	dMkr>CFstep	5-29	MKDSS	4-37
MKR>	More..	dMkr>ZoomIN	5-29	MKZI	4-38
MKR>	More..	dMkr>ZoomOUT	5-29	MKZO	4-38
MKR>	More..	Undo			
PEAK			5-30	MKPK	4-39
PEAK		Next Peak	5-30	MKPK	4-39
PEAK		NPeakLeft	5-31	MKPK	4-39
PEAK		NPeakRight	5-31	MKPK	4-39
PEAK		MinSearch	5-30	MKMIN	4-39
PEAK		Pk-Pk Search	5-31	MKPP	4-40

OPERATION MANUAL				PROGRAMMING MANUAL	
HARD KEY	UPPER MENU	SOFTKEY	PAGE	RCI COMMAND	PAGE
PEAK		Mkr Track	5-31	MKTRACK	4-40
PEAK	More.. Search Param..	Excur. [dB]	5-32	MKPX	4-41
PEAK	More.. Search Param..	Thresh. [dB]	5-32	MKPT	4-42
PEAK	More.. Search Param..	Search Par. [AUTO/MNL]	5-32	MKPS	4-43
PEAK	More..	Continuous	5-32		
PEAK	More..	Peak Number	5-32	MMPN	4-44
PEAK	More..	Multi Peak Search	5-32	MMP	4-44
TRIG		Continuous	5-33	TRGSWP	4-45
TRIG		Single	5-33	TRGSWP	4-45
TRIG	Source..	Free Run	5-34	TM	4-46
TRIG	Source..	Video	5-34	TM, TLV	4-46, 4-47
TRIG	Source..	Line	5-35	TM	4-46
TRIG	Source..	External	5-35	TM	4-46
TRIG		Trig Delay	5-35	TDLY	4-48
TRIG		Time Gate	5-36	GATE	4-50
TRIG	Time Gate Set..	Delay	5-37	GD	4-52
TRIG	Time Gate Set..	Length	5-37	GL	4-53
TRIG	Time Gate Set..	Control	5-37	GATECTL	4-51
TRIG	Time Gate Set..	Edge	5-37	GP	4-54
TRIG		Trig Edge	5-37	TE	4-49
CPL		All Auto	5-38	AUTOCPL	4-55
CPL		RBW	5-39	RB	4-56
CPL		RBW [AUTO/MNL]	5-40	RB	4-56
CPL		VBW	5-39	VB	4-57
CPL		VBW [AUTO/MNL]	5-39	VB	4-57
CPL		Swp Time	5-41	ST	4-58
CPL		Swp Time [AUTO/MNL]	5-41	ST	4-58
DISP		Disp. Line	5-43	DL	4-59
DISP		Disp. Line [OFF/ON]	5-43	DL	4-59
DISP		Thresh. Line	5-44	TH	4-60
DISP		Thresh. Line [OFF/ON]	5-44	TH	4-60

OPERATION MANUAL				PROGRAMMING MANUAL	
HARD KEY	UPPER MENU	SOFTKEY	PAGE	RCI COMMAND	PAGE
DISP		Screen Title..	5-44	TITLE	4-61
DISP	More..	Graticule	5-46	GRAT	4-62
DISP	More..	Annotation	5-46		
DISP	More..	White Mode	5-46		
TRACE		Cle & Wrt	5-47	TRF	4-64
TRACE		Max Hold	5-47	TRF	4-64
TRACE		Min Hold	5-48	TRF	4-64
TRACE		View	5-48	TRF	4-64
TRACE		Blank	5-48	TRF	4-64
TRACE		Select	5-47	TRS	4-63
TRACE	More.. Average..	Average	5-49	AVG	4-72
TRACE	More.. Average..	Count	5-49	AVGC	4-73
TRACE	More.. Average..	Cycle	5-49	AVGCYL	4-73
TRACE	More.. Average..	Stop	5-49	AVGS	4-74
TRACE	More.. Average..	Continuous	5-49	AVGS	4-74
TRACE	More.. Average..	Reset	5-49	AVGS	4-74
TRACE	More.. Detect..	Normal	5-50	DET	4-71
TRACE	More.. Detect..	Sample	5-50	DET	4-71
TRACE	More.. Detect..	Pos Peak	5-50	DET	4-71
TRACE	More.. Detect..	Neg Peak	5-50	DET	4-71
TRACE	More.. Detect..	Average	5-50	DET	4-71
TRACE	More.. Math..	A-B -> A	5-51	AMB	4-68
TRACE	More.. Math..	B-DL -> B	5-51	BML	4-68
TRACE	More.. Math..	A+B -> A	5-51	APB	4-69
TRACE	More.. Math..	A-B+DL -> A	5-51	AMBPL	4-69
TRACE	More.. Math..	A EXCH B	5-51	AXB	4-70
TRACE	More.. Math..	Math [OFF/ON]	5-51		
SAVE			5-52	FSAVE	4-76
FILE		Load	5-53	FLOAD	4-76
FILE		Delete	5-53	FDEL	4-77
FILE		Copy to	5-53	FCPT	4-77
FILE		Disk	5-53	FDRV	4-75

OPERATION MANUAL				PROGRAMMING MANUAL	
HARD KEY	UPPER MENU	SOFTKEY	PAGE	RCI COMMAND	PAGE
FILE		File Type	5-54		
FILE	More..	Sort Key	5-55		
FILE	More..	Sort Direct	5-55		
FILE	More..	Copy All	5-55		
FILE	More..	Delete All	5-55		
FILE	More..	Format A:	5-55		
FILE	More..	Filename [Auto/Title]	5-55		
LIMIT		UpPassChk	5-56	LMTPC, LMTUP	4-79, 4-80
LIMIT		LowPassChk	5-56	LMTPC, LMTLW	4-79, 4-81
LIMIT		Alarm	5-56	ALARM	4-80
LIMIT		Clear Limit	5-57		
SYSTEM		SA Mode..	None		
SYSTEM		CDMA Mode..	Option		
SYSTEM		EMC Mode..	Option		
SYSTEM		DTF Mode..	Option		
SYSTEM		Source Mode..	Option		
SYSTEM	More 1 of 3 Clock Set..	Time Set	5-59	TIME	4-114
SYSTEM	More 1 of 3 Clock Set..	Data Set	5-59	DATE	4-113
PRESET		Preset	5-62	PRST	4-107
PRESET		Last State	5-63		
PRESET	Alignment Mode..	All Align	5-63	CALALL	4-107
PRESET	Alignment Mode..	Yig. Align	5-63	YIGCAL	4-108
PRESET	Alignment Mode..	Span Align	5-63	SPCAL	4-111
PRESET	Alignment Mode..	Level Align	5-63	LVLC	4-110
PRESET	Alignment Mode..	RBW Align	5-63	RCAL	4-109
PRESET		Power On	5-64		
PRESET		Auto Align	5-64	PCAL	4-108
AUX		AM Demod.	5-65	DEMOD	4-105
AUX		FM Demod.	5-65	DEMOD	4-105
AUX		Audio Sound	5-66	AUDIO	4-105
AUX		Audio Level	5-66	AUDIOVR	4-106
AUX		Squelch Lev	5-66	SQL	4-106
TUNE			5-68	AUTOTUNE	4-11
PRINT			5-58	HCOPY	4-112

APPENDIX-ERROR CODE

ERROR CODE		ERROR DESCRIPTION
100	:	Data out of range
101	:	Center frequency out of range
102	:	Start frequency out of range
103	:	Stop frequency out of range
104	:	Center frequency step size out of range
105	:	Span frequency out of range
106	:	Reference level out of range
107	:	Attenuator level out of rang
108	:	Marker function out of rang
111	:	RBW out of range
112	:	VBW out of range
113	:	Sweep time out of range
118	:	Threshold level error
119	:	LG(Amplitude dB scale) data out of range
120	:	ESE data out of range
121	:	ESE2 data out of range
122	:	SRE data out of range
123	:	Trigger level data out of range
124	:	Trigger level, Trigger source is not video
125	:	Trig delay data out of range
126	:	Trig delay, Span is not zero
127	:	Average count out of range
128	:	Audio level out of range
129	:	Squelch level out of range
130	:	Date out of range
131	:	Time out of range
132	:	Save internal error
133	:	Save external error
134	:	Recall internal error
135	:	Recall external error

137 : Limit error
138 : DB down error
139 : OBW error
140 : TRA out of range
141 : TRB out of range
142 : Delta marker function error
143 : Normal marker function error
144 : Display line error
145 : Marker counter resolution out of range
146 : Noise marker error
147 : Printer not connected or not responded
148 : Printer off-line or paper empty
150 : Auto setup-signal is not found
151 : Peak search error
152 : RBW cal fail
153 : PWR on cal fail
154 : Temp. Cal fail
155 : Tracking generator is not on
156 : Tracking generator output level error
157 : Tracking generator manual frequency cal. Offset frequency range error
158 : Span cal fail
159 : Level cal fail
160 : Log cal fail
161 : AM scale range over
162 : Not AM mode
163 : FM scale range over
164 : Not FM mode
165 : Do not execute in Free Run
166 : Execute in Zero Span

200 : Disk Fail
201 : Disk Full
202 : Write Protection
203 : Disk Empty
210 : Not enough free space
211 : Over MAX Storage error

220 : File create error
221 : File write error
222 : File type error
223 : File name error
224 : No data error (No data for file write)
225 : No limit line data error
226 : Image file create error
227 : Can Not Save, go MEAS DataBase Menu
230 : File Open error
231 : File read error
232 : Can't load
233 : No trace data
234 : Load DCF instead in MEAS / Config..
235 : Can't Load in DTF Mode.
236 : Could not load File in EMC Mode
237 : Could not load File , go EMC Mode
238 : Could not load File , go DTF Mode
239 : Could not load File , go DTF Configuration
240 : Disk A: access Denied
241 : Disk A: bad Sector
242 Disk A: not Ready
243 : Disk A: protected
244 : Disk A: format fail
251 : Can't open source file
252 : Can't open target file
253 : Can't copy at same disk
260 : Read Only
261 : File already exists
262 File rename fail
270 : File delete fail
:
800 : Option Not Installed
810 Not Closed Other Option Mode
820 : Not in the Option Mode
:
995 : Processing the other function
996 Input data size over error

997 : Bad command
998 : Unnecessary suffix insertion
999 : Missing suffix and invalid suffix
:
:

AEROFLEX INTERNATIONAL LTD.
SOFTWARE LICENCE AND WARRANTY

This document is an Agreement between the user of this Licensed Software, the Licensee, and Aeroflex International Limited, the Licensor. By opening this Software package or commencing to use the software you accept the terms of this Agreement. If you do not agree to the terms of this Agreement please return the Software package unopened to Aeroflex International Limited or do not use the software.

1. DEFINITIONS

The following expressions will have the meanings set out below for the purposes of this Agreement:

Add-In Application Software	Licensed Software that may be loaded separately from time to time into the Equipment to improve or modify its functionality
Computer Application Software	Licensed Software supplied to run on a standard PC or workstation
Designated Equipment	the single piece of Equipment upon which the licensed software is installed
Downloaded Software	any software downloaded from an Aeroflex web site
Embedded Software	Licensed Software that forms part of the Equipment supplied by Aeroflex and without which the Equipment cannot function
Licence Fee	the consideration ruling at the date of this Agreement for the use of one copy of the Licensed Software on the Designated Equipment
Licensed Software	All and any programs, listings, flow charts and instructions in whole or in part including Add-in, Computer Application, Downloaded and Embedded Software supplied to work with Designated Equipment

2. LICENCE FEE

The Licensee shall pay the Licence Fee to Aeroflex in accordance with the terms of the contract between the Licensee and Aeroflex.

3. TERM

This Agreement shall be effective from the date hereof and shall continue in force until terminated under the provisions of Clause 9.

4. LICENCE

4.1 Unless and until terminated, this Licence confers upon the Licensee the non-transferable and non-exclusive right to use the Licensed Software on the Designated Equipment.

4.2 The Licensee may not use the Licensed Software on other than the Designated Equipment, unless written permission is first obtained from Aeroflex and until the appropriate additional Licence Fee has been paid to Aeroflex.

4.3 The Licensee may not amend or alter the Licensed Software and shall have no right or licence other than that stipulated herein.

4.4 The Licensee may make not more than two copies of the Licensed Software (but not the Authoring and Language Manuals) in machine-readable form for operational security and shall ensure that all such copies include Aeroflex's copyright notice, together with any features which disclose the name of the Licensed Software and the Licensee. Furthermore, the Licensee shall not permit the Licensed Software or any part to be disclosed in any form to any third party and shall maintain the Licensed Software in secure premises to prevent any unauthorised disclosure. The Licensee shall notify Aeroflex immediately if the Licensee has knowledge that any unlicensed party possesses the Licensed Software. The Licensee's obligation to maintain confidentiality shall cease when the Licensed Software and all copies have been destroyed or returned. The copyright in the Licensed Software shall remain with Aeroflex. The Licensee will permit Aeroflex at all reasonable times to audit the use of the Licensed Software.

4.5 The Licensee will not disassemble or reverse engineer the Licensed Software, nor sub-licence, lease, rent or part with possession or otherwise transfer the whole or any part of the Licensed Software.

5 WARRANTY

5.1 Aeroflex certifies that the Licensed Software supplied by Aeroflex will at the time of delivery function substantially in accordance with the applicable Software Product Descriptions, Data Sheets or Product Specifications published by Aeroflex.

5.2 The warranty period (unless an extended warranty for Embedded Software has been purchased) from date of delivery in respect of each type of Licensed Software is:

Embedded Software	12 months
Add-In Application Software	90 days
Computer Application Software	90 days
Downloaded Software	No warranty

5.3 If during the appropriate Warranty Period the Licensed Software does not conform substantially to the Software Product Descriptions, Data Sheets or Product Specifications Aeroflex will provide:

5.3.1 In the case of Embedded Software and at Aeroflex's discretion either a fix for the problem or an effective and efficient work-around.

5.3.2 In the case of Add-In Application Software and Computer Application Software and at Aeroflex's discretion replacement of the software or a fix for the problem or an effective and efficient work-around.

5.4 Aeroflex does not warrant that the operation of any software will be uninterrupted or error free.

6 The above Warranty does not apply to:

6.1 Defects resulting from software not supplied by Aeroflex, from unauthorised modification or misuse or from operation outside of the specification.

6.2 Third party produced Proprietary Software which Aeroflex may deliver with its products, in such case the third party Software Licence Agreement including its warranty terms shall apply.

7 The remedies offered above are sole and exclusive remedies and to the extent permitted by applicable law are in lieu of any implied conditions, guarantees or warranties whatsoever and whether statutory or otherwise as to the software all of which are hereby expressly excluded.

8. INDEMNITY

8.1 Aeroflex shall defend, at its expense, any action brought against the Licensee alleging that the Licensed Software infringes any patent, registered design, trademark or copyright, and shall pay all Licensor's costs and damages finally awarded up to an aggregate equivalent to the Licence fee provided the Licensee shall not have done or permitted to be done anything which may have been or become any such infringement and shall have exercised reasonable care in protecting the same failing which the Licensee shall indemnify Aeroflex against all claims costs and damages incurred and that Aeroflex is given prompt written notice of such claim and given information, reasonable assistance and sole authority to defend or settle such claim on behalf of the Licensee. In the defence or settlement of any such claim, Aeroflex may obtain for the Licensee the right to continue using the Licensed Software or replace it or modify it so that it becomes non-infringing.

8.2 Aeroflex shall not be liable if the alleged infringement:

8.2.1 is based upon the use of the Licensed Software in combination with other software not furnished by Aeroflex, or

8.2.2 is based upon the use of the Licensed Software alone or in combination with other software in equipment not functionally identical to the Designated Equipment, or

8.2.3 arises as a result of Aeroflex having followed a properly authorised design or instruction of the Licensee, or

8.2.4 arises out of the use of the Licensed Software in a country other than the one disclosed to Aeroflex as the intended country of use of the Licensed Software at the commencement of this Agreement.

8.3 Aeroflex shall not be liable to the Licensee for any loss of use or for loss of profits or of contracts arising directly or indirectly out of any such infringement of patent, registered design, trademark or copyright.

9. TERMINATION

9.1 Notwithstanding anything herein to the contrary, this Licence shall forthwith determine if the Licensee:

9.1.1 As an individual has a Receiving Order made against him or is adjudicated bankrupt or compounds with creditors or as a corporate body, compounds with creditors or has a winding-up order made against it or

9.1.2 Parts with possession of the Designated Equipment.

9.2 This Licence may be terminated by notice in writing to the Licensee if the Licensee shall be in breach of any of its obligations hereunder and continue in such breach for a period of 21 days after notice thereof has been served on the Licensee.

9.3 On termination of this Agreement for any reason, Aeroflex may require the Licensee to return to Aeroflex all copies of the Licensed Software in the custody of the Licensee and the Licensee shall, at its own cost and expense, comply with such requirement within 14 days and shall, at the same time, certify to Aeroflex in writing that all copies of the Licensed Software in whatever form have been obliterated from the Designated Equipment.

10. THIRD PARTY LICENCES

The software or part thereof may be the proprietary property of third party licensors. In such an event such third party licensors (as referenced on the package or the Order Acknowledgement) and/or Aeroflex may directly enforce the terms of this Agreement and may terminate the Agreement if the Licensee is in breach of the conditions contained herein.

11. EXPORT REGULATIONS

The Licensee undertakes that where necessary the Licensee will conform with all relevant export regulations imposed by the Governments of the United Kingdom and/or the United State of America.

12. NOTICES

Any notice to be given by the Licensee to Aeroflex shall be addressed to:

Aeroflex International Limited, Longacres House, Six Hills Way, Stevenage, SG1 2AN, UK.

13. LAW AND JURISDICTION

This Agreement shall be governed by the laws of England and shall be subject to the exclusive jurisdiction of the English courts. This agreement constitutes the whole Contract between the parties and may be changed only by memorandum signed by both parties.

CHINA Beijing
Tel: [+86] (10) 6539 1166
Fax: [+86] (10) 6539 1778

CHINA Shanghai
Tel: [+86] (21) 5109 5128
Fax: [+86] (21) 5150 6112

FINLAND
Tel: [+358] (9) 2709 5541
Fax: [+358] (9) 804 2441

FRANCE
Tel: [+33] 1 60 79 96 00
Fax: [+33] 1 60 77 69 22

GERMANY
Tel: [+49] 8131 2926-0
Fax: [+49] 8131 2926-130

HONG KONG
Tel: [+852] 2832 7988
Fax: [+852] 2834 5364

INDIA
Tel: [+91] 80 5115 4501
Fax: [+91] 80 5115 4502

KOREA
Tel: [+82] (2) 3424 2719
Fax: [+82] (2) 3424 8620

SCANDINAVIA
Tel: [+45] 9614 0045
Fax: [+45] 9614 0047

SPAIN
Tel: [+34] (91) 640 11 34
Fax: [+34] (91) 640 06 40

UK Burnham
Tel: [+44] (0) 1628 604455
Fax: [+44] (0) 1628 662017

UK Stevenage
Tel: [+44] (0) 1438 742200
Fax: [+44] (0) 1438 727601
Freephone: 0800 282388

USA
Tel: [+1] (316) 522 4981
Fax: [+1] (316) 522 1360
Toll Free: (800) 835 2352

web **www.aeroflex.com** Email **info-test@aeroflex.com**

November 2005