BERTWave Series
Remote Control
Operation Manual

Eighth Edition

- **For safety and warning information, please read this manual before attempting to use the equipment.**
- **Additional safety and warning information is provided within the MP2100A/MP2101A/MP2102A BERTWave Operation Manual, MP2100B BERTWave Operation Manual or MP2110A BERTWave Operation Manual. Please also refer to them before using the equipment.**
- **Keep this manual with the equipment.**

ANRITSU CORPORATION

Document No.: M-W3773AE-8.0

Safety Symbols

To prevent the risk of personal injury or loss related to equipment malfunction, Anritsu Corporation uses the following safety symbols to indicate safety-related information. Ensure that you clearly understand the meanings of the symbols BEFORE using the equipment. Some or all of the following symbols may be used on all Anritsu equipment. In addition, there may be other labels attached to products that are not shown in the diagrams in this manual.

Symbols used in manual

⚠ DANGER This indicates a very dangerous procedure that could result in serious injury or death if not performed properly.

⚠ WARNING This indicates a hazardous procedure that could result in serious injury or death if not performed properly.

⚠ CAUTION This indicates a hazardous procedure or danger that could result in light-to-severe injury, or loss related to equipment malfunction, if proper precautions are not taken.

Safety Symbols Used on Equipment and in Manual

The following safety symbols are used inside or on the equipment near operation locations to provide information about safety items and operation precautions. Ensure that you clearly understand the meanings of the symbols and take the necessary precautions BEFORE using the equipment.

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

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

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

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

These indicate that the marked part should be recycled.

BERTWave Series
Remote Control Operation Manual

4 September 2015 (First Edition)
30 March 2018 (Eighth Edition)

Notes On Export Management

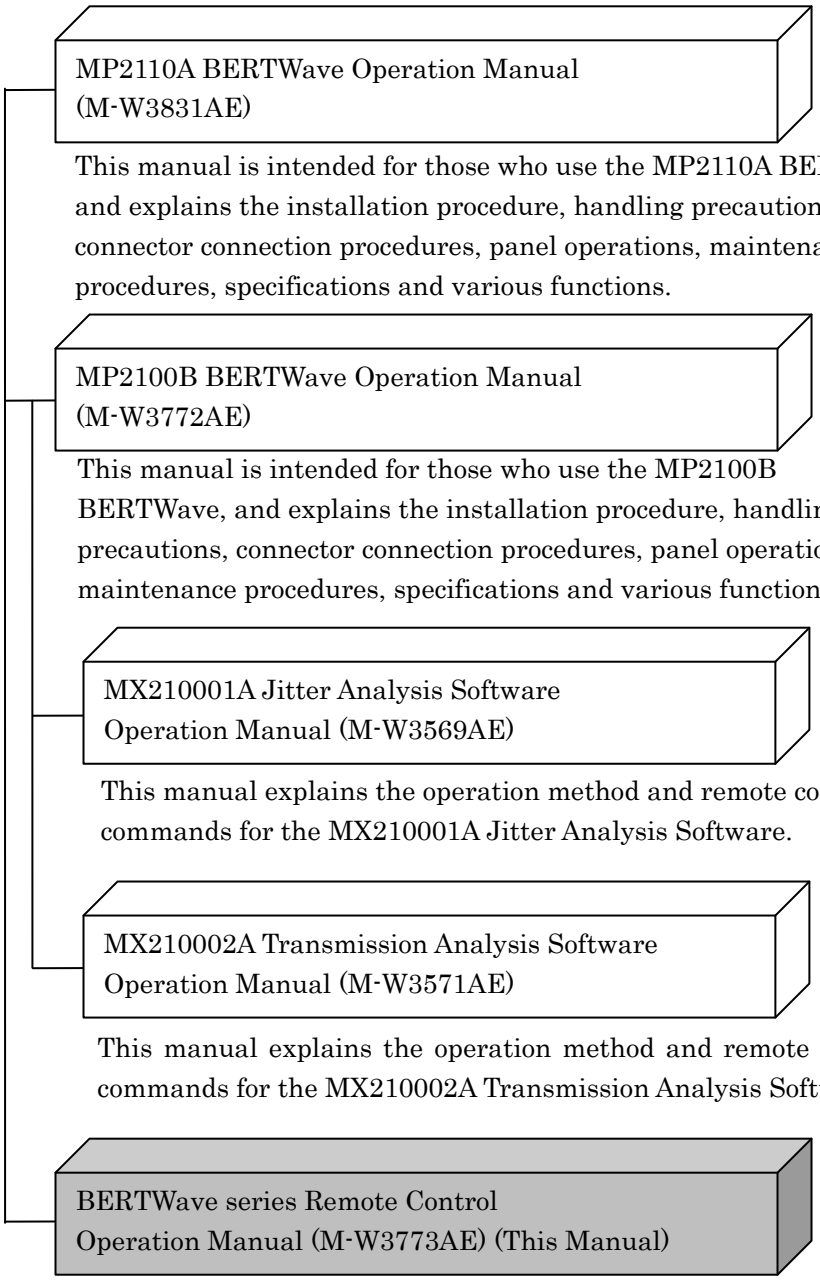
This product and its manuals may require an Export License/Approval by the Government of the product's country of origin for re-export from your country.

Before re-exporting the product or manuals, please contact us to confirm whether they are export-controlled items or not.

When you dispose of export-controlled items, the products/manuals need to be broken/shredded so as not to be unlawfully used for military purpose.

About This Manual

The manual set for the BERTWave Series consists of the following five operation manuals:



This manual is intended for those who use the MP2110A BERTWave, and explains the installation procedure, handling precautions, connector connection procedures, panel operations, maintenance procedures, specifications and various functions.

This manual is intended for those who use the MP2100B BERTWave, and explains the installation procedure, handling precautions, connector connection procedures, panel operations, maintenance procedures, specifications and various functions.

This manual explains the operation method and remote control commands for the MX210001A Jitter Analysis Software.

This manual explains the operation method and remote control commands for the MX210002A Transmission Analysis Software.

This manual explains the commands to control the BERTWave, status register structure, and sample programs.

Also, there is the manual “MP2100A BERTWave MP2101A BERTWave PE MP2102A BERTWave SS Operation Manual (Operation) (M-W3349AE)”. MP2100A, MP2101A, and MP2102A are the discontinued products.

This manual explains the remote control commands.

This manual assumes the reader has the following information:

- The reader has read through the *MP2100B BERTWave Operation Manual* or *MP2110A BERTWave Operation Manual*.
- The reader can create the C or Basic program.

For the connection of the power source and peripheral devices, panel operation, and maintenance, refer to the following manual:
MP2100B BERTWave Operation Manual
MP2110A BERTWave Operation Manual

In this manual, the models of BERTWave are described as follows.

MP210xA:	MP2100A, MP2101A, and MP2102A
this instrument:	MP2100A, MP2101A, MP2102A, MP2100B, and MP2110A
BERTWave:	MP2100A, MP2101A, MP2102A, MP2100B, and MP2110A

Table of Contents

About This Manual..... I

Chapter 1 Overview 1-1

1.1 What is Remote Control?..... 1-2
1.2 Main Uses for Remote Control 1-3
1.3 Abbreviations 1-5
1.4 Restrictions on Software Versions..... 1-6

Chapter 2 Before Use 2-1

2.1 Connection types of Remote Controlling 2-2
2.2 Multiple Sessions (Version 5 or later)..... 2-3
2.3 Required Equipment..... 2-4
2.4 Connecting Equipment 2-5
2.5 Setting Interface..... 2-8
2.6 Checking Connection..... 2-10
2.7 Message Format..... 2-12
2.8 Checking Instrument Status..... 2-17
2.9 Checking If Message Execution Is Completed 2-29

Chapter 3 Message List..... 3-1

3.1 Rules for Describing Messages..... 3-3
3.2 Correspondence Between Panel and Messages 3-4
3.3 Messages Corresponding to Common Operations 3-41
3.4 Status Register 3-59
3.5 Specifying Module and Channel..... 3-82
3.6 BERT(PPG/ED) messages..... 3-84
3.7 XFP/SFP+ specific messages
(MP210xA, MP2100B)..... 3-132
3.8 O/E messages 3-135
3.9 Scope-specific messages..... 3-136

Appendix A Command Compatibility With Existing Products A-1

Appendix B Error Codes B-1

Appendix C Sample Program C-1

Appendix D Bibliography D-1

Index Index-1

Table of Messages

TRM 3-41
GTL (Go to local) 3-41
:SYSTem:BEEPer:SET 3-42
:SYSTem:VERSion? 3-43
*IDN? (Identification) 3-43
*OPT? (Option Identification Query) 3-43
:SYSTem:INFormation? 3-46
:SYSTem:{DATE|TIME}? 3-47
:SYSTem:ERRor? 3-47
:SYSTem:ERRor:HCLear 3-48
:SYSTem:ERRor:HISTory? 3-48
:SYSTem:DISPlay:ALARm 3-49
:SYSTem:INFormation:ERRor? 3-49
*RST (Reset) 3-50
:SYSTem:MMEMory:RECall 3-51
:SYSTem:MMEMory:STORe 3-52
:SYSTem:DISPlay:RESult 3-54
:SYSTem:PRINt:COPY 3-55
:SYSTem:DISPlay:DATA? 3-56
:DISPlay:ACTive 3-58
:DISPlay:ACTive:ACResult 3-58
*CLS (Clear Status) 3-59
:STATus:PRESet 3-60
*STB? (Status Byte) 3-61
*SRE (Service Request Enable) 3-62
*ESR? (Standard Event Status Register) 3-63
*ESE (Event Status Enable) 3-64
:STATus:OPERation:CONDition? 3-65
:STATus:OPERation[:EVENt]? 3-65
:STATus:OPERation:ENABle 3-66
:STATus:OPERation:NTRansition 3-67
:STATus:OPERation:PTRansition 3-68
:INSTrument:PE<ch>:RESet 3-69
:INSTrument:PE<ch>:CONDition? 3-69
:INSTrument:PE<ch>[:EVENt]? 3-70
:INSTrument:PE<ch>:NTRansition 3-71
:INSTrument:PE<ch>:PTRansition 3-72
:INSTrument:WAV:RESet 3-73
:INSTrument:WAV:CONDition? 3-73
:INSTrument:WAV[:EVENt]? 3-74
:INSTrument:WAV:NTRansition 3-74

:INSTrument:WAV:PTRansition 3-75
:INSTrument:XSFP:RESet 3-76
:INSTrument:XSFP:CONDition? 3-76
:INSTrument:XSFP[:EVENt]? 3-76
:INSTrument:XSFP:NTRansition 3-77
:INSTrument:XSFP:PTRansition 3-78
*OPC (Operation Complete) 3-79
*WAI (Wait to Continue) 3-79
:SOURce:OUTPut:ASET 3-80
*TRG (Trigger) 3-80
:SENSe:MEASure:ASTP 3-81
:SENSe:MEASure:ASTate? 3-81
:MODule:ID 3-82
:DISPlay:RESult 3-84
:OUTPut:RCLock 3-84
:OUTPut:RCLock:SELect 3-85
:OUTPut:CMU:EXTClock 3-87
:OUTPut:RCLock:STATus? 3-87
:OUTPut:RCLock:APPLy 3-88
:OUTPut:CLOCk:SOURce:CHANnel 3-89
:OUTPut:CLOCk:DIVRate? 3-89
:OUTPut:SYNC:SOURce 3-90
:BERT:ALL:PARam:TRACking 3-91
:SENSe:PARam:TRACking 3-92
:OUTPut:BITRate:STANdard 3-93
:INPut:BITRate:STANdard 3-96
:OUTPut:BITRate 3-97
:INPut:BITRate 3-98
:OUTPut:BITRate:OFFSet 3-99
:OUTPut:BITRate:DIVRate 3-100
:INPut:BITRate:DIVRate? 3-101
:SOURce:PATTern:TYPE 3-102
:SENSe:PATTern:TYPE 3-103
:SOURce:PATTern:LOGic 3-104
:SENSe:PATTern:LOGic 3-104
:SOURce:MMEMory:PATTern:RECall 3-105
:SENSe:MMEMory:PATTern:RECall 3-106
:SOURce:PATTern:DATA:LENGth? 3-106
:SENSe:PATTern:DATA:LENGth? 3-107
:OUTPut:DATA:OUTPut 3-108
:OUTPut:DATA:AMPLitude 3-109

:OUTPut:DATA:ATTFactor 3-109
:OUTPut:DATA:RELative? 3-110
:SOURce:PATTern:EADDition:SET 3-110
:SOURce:PATTern:EADDition:VARiation 3-111
:SOURce:PATTern:EADDition:SINGle 3-111
:SOURce:PATTern:EADDition:RATE 3-112
:INPut:DATA:INTerface 3-113
:INPut:DATA:ATTFactor 3-114
:INPut:DATA:THReshold 3-115
:SENSe:PATTern:SYNC:ASYNc 3-116
:SENSe:PATTern:SYNC:THReshold 3-117
:SENSe:PATTern:SYNC:PSMode 3-118
:SENSe:PATTern:SYNC:FPOSition 3-119
[:BERT:ALL]:DISPlay:RESult:EALarm:HRESet 3-120
[:BERT:ALL]:CALCulate:DATA:MONitor? 3-120
[:BERT:ALL]:CALCulate:DATA:MONitor:HISTory? 3-121
[:BERT:ALL]:SENSe:MEASure:IMMediate? 3-122
[:BERT:ALL]:CALCulate:DATA:EALarm? 3-124
:SENSe:MEASure:EALarm:MODE 3-126
:SENSe:MEASure:EALarm:PERiod 3-127
:DISPlay:RESult:EALarm:MODE 3-128
[:BERT:ALL]:SENSe:MEASure:STARt 3-129
[:BERT:ALL]:SENSe:MEASure:STOP 3-129
[:BERT:ALL]:SENSe:MEASure:EALarm:STATe? 3-129
:SENSe:MEASure:EALarm:STARt? 3-130
:SENSe:MEASure:EALarm:STOP? 3-130
:SENSe:MEASure:EALarm:ELAPsed? 3-131
:SENSe:MEASure:EALarm:TIMed? 3-131
:CALCulate:OPTical:STATus? 3-132
:SOURce:OPTical:SIGNal:WLENgth? 3-132
:SOURce:OPTical:SIGNal:OUTPut 3-133
:SOURce:OPTical:XFP:REFClock 3-134
:INPut:CHA|CHB 3-136
:CONFigure:MEASure:CHANnel 3-136
:CONFigure:MEASure:TYPe 3-137
:CONFigure:MEASure:AMPTime:DISPlay 3-138
:DISPlay:WINDow:GRAPhics:CLEar 3-139
:SAMPling:STATus 3-139
:DISPlay:WINDow:AUTOscale 3-139
:DISPlay:WINDow:ZOOM 3-140
:DISPlay:SIGNal:CHA|CHB 3-141

:DISPlay:MODE 3-142
:DISPlay:MODE:EYE:FAST 3-142
:OPTion:MAX:SAMPles:NUMber 3-143
:ACCUmulation:TYPe 3-144
:ACCUmulation:LIMit 3-144
:ACCUmulation:PERSistency 3-146
:ACCUmulation:AVERaging 3-146
:TIME:CRU 3-147
:TIME:CRU:RATE:STANdard 3-147
:TIME:CRU:RATE 3-148
:TIME:CRU:LBWidth 3-148
:TIME:CRU:STATus? 3-149
:TIME:CRU:FREQuency? 3-149
:INPut:CLKRecovery 3-150
:CONFigure:CLKRecovery 3-150
:EYEPulse:PRINt:COPY 3-151
:PRINt:GRATicule 3-152
:PRINt:INVerse 3-152
:DISPlay:WAVeform:COLor 3-153
:DISPlay:MASK:COLor 3-153
:DISPlay:INFormation 3-154
:DISPlay:LABel 3-154
:DISPlay:LABel:DALL 3-155
:TMEMory:REFerence:SET 3-155
:TMEMory:REFerence:CLEar 3-156
:TMEMory:CHANnel 3-156
:CALibrate:TEMPerature? 3-157
:CALibrate:AMPLitude? 3-157
:CALibrate:APPLication 3-158
:TIME:TRACking 3-159
:TIME:TRACking:STATus? 3-160
:CONFigure:TRACking:DRATe 3-160
:CONFigure:TRACking:DRATe:MASTer 3-161
:TIME:ACQClock? 3-162
:TIME:CLKRate 3-163
:TIME:DATRate 3-164
:TIME:DIVRatio 3-165
:TIME:AUTodetect 3-166
:TIME:PTRigger 3-166
:TIME:PTRigger:RESet 3-167
:DISPlay:WINDow:X:UNIT 3-168

:DISPlay:WINDow:X:BITs 3-168
:DISPlay:WINDow:X:OFFSets 3-169
:TIME:PATTern:TYPE 3-170
:TIME:PATLength 3-170
:CONFigure:TRACking:PATLength 3-171
:CONFigure:TRACking:PATLength:MASTer 3-171
:CONFigure:SKEW:CHA|CHB 3-172
:CONFigure:SKEW:ALIGn 3-173
:DISPlay:WINDow:CHANnel:BOTH 3-174
:DISPlay:WINDow:Y:DIVision:CHA|CHB 3-175
:DISPlay:WINDow:Y:OFFSets:CHA|CHB 3-175
:INPut:ATTenuation:CHA|CHB 3-176
:CALCulate:CHANnel:MATH 3-177
:CALCulate:CHANnel:MATH:DEFine 3-178
:DISPlay:WINDow:Y:DIVision:CHMath 3-178
:DISPlay:WINDow:Y:OFFSets:CHMath 3-179
:CALibrate:OEPower[:JUDGe] 3-180
:INPut:WAVLength[:CHA|CHB] 3-181
:CALibrate:CGain[:CHA|CHB] 3-181
:CALibrate:SYSTem:CGain 3-182
:CALibrate:RESPonsivity[:CHA|CHB] 3-183
:CALibrate:AUTocorrect[:CHA|CHB] 3-183
:CALibrate:CALPower[:CHA|CHB] 3-184
:FILTer[:CHA|CHB] 3-185
:INPut:FILTer:ENABle 3-187
:INPut:FILTer 3-188
:CONFigure:EXRCorrection[:CHA|CHB] 3-189
:CONFigure:EXRCorrection:FACTor[:CHA|CHB] 3-190
:CONFigure:MEASure:DISPlay:ADD 3-191
:CONFigure:MEASure:DISPlay:ADELete 3-191
:CONFigure:MEASure:AMPTIME{1|2|3|4} 3-192
:CONFigure:MEASure:AREa:DISPlay 3-193
:CONFigure:MEASure:AREa:ITEM 3-194
:CONFigure:MEASure:PAM:TIMing 3-195
:CONFigure:MEASure:PAM:CENTer 3-195
:CONFigure:MEASure:PAM:EOPening 3-196
:CONFigure:MEASure:PAM:TEQualizer:CHA|CHB 3-196
:CONFigure:MEASure:PAM:TEQualizer:DISPlay:CHA|CHB 3-197
:CONFigure:MEASure:PAM:TEQualizer:CALCulate:CHA|CHB
..... 3-197

:CONFigure:MEASure:PAM:TEQualizer:OPTimization:CHA|CHB 3-197
:CONFigure:MEASure:PAM:TEQualizer:TAPS:COUNt:CHA|CHB 3-198
:CONFigure:MEASure:PAM:TEQualizer:TAPS:CHA|CHB . 3-198
:CONFigure:MEASure:DEFine 3-200
:CONFigure:MEASure:TRANsition:CORRection 3-200
:CONFigure:MEASure:TRANsition:CORRect:FACTor 3-201
:CONFigure:MEASure:EYEBoundary:OFFSet 3-201
:CONFigure:MEASure:EYEBoundary:WIDTh 3-202
:CONFigure:MASK:RECall 3-203
:CONFigure:MASK:TYPe 3-204
:CONFigure:MASK:ALGorithm 3-205
:CONFigure:MASK:UPDate 3-205
:CONFigure:MASK:USER:MARKer 3-206
:CONFigure:MASK:USER:LOCation:X1|XDELta 3-206
:CONFigure:MASK:USER:LOCation:Y1|YDELta 3-207
:CONFigure:MASK:MARGin:CONTupdate 3-208
:CONFigure:MASK:MARGin 3-208
:MEASure:MASK:MARGin? 3-209
:SAMPles:JUDGe:TYPE 3-209
:SAMPles:JUDGe 3-210
:SAMPles:JUDGe:RATE 3-211
:CONFigure:MASK:AREa:RESTriction 3-212
:CONFigure:MASK:AREa:RESTriction:ANGLe 3-212
:CONFigure:MASK:AREa:RESTriction:WIDTh 3-213
:CONFigure:HISTogram:AXIS 3-214
:HISTogram:CENTer 3-214
:HISTogram:X1|X2 3-215
:HISTogram:Y1|Y2 3-216
:FETCh:AMPTime:QUEStionableeye? 3-217
:FETCh:AMPLitude:<meas_item>[:CHA|CHB]? 3-217
:FETCh:TIME:<meas_item>[:CHA|CHB]? 3-220
:FETCh:MASK:<meas_item>? 3-222
:FETCh:HISTogram:AMPLitude:<meas_item>? 3-224
:FETCh:HISTogram:TIME:<meas_item>? 3-225
:TRACe:CHANnelA|CHANnelB|CHANnels? 3-226
:TRACe:PREPare 3-228
:TRACe:END 3-229
:CALCulate:MARKer:AOFF 3-230
:CALCulate:MARKer:CENTer 3-230

:CALCulate:MARKer:X1|X2|Y1|Y2 3-230
:CALCulate:MARKer:LOCation:X1|X2 3-231
:CALCulate:MARKer:LOCation:XDELta? 3-231
:CALCulate:MARKer:LOCation:CHA|CHB:Y1|Y2 3-232
:CALCulate:MARKer:LOCation:CHA|CHB:YDELta? 3-232
:JITTer:RESult:ERRor? 3-233
:JITTer:MEASure:PDJ 3-233
:JITTer:MEASure:PDJ:STANdard 3-234
:JITTer:MEASure:PDJ:FILTer 3-234
:JITTer:MEASure:EDGE:TYPE 3-235
:JITTer:MEASure:TJ:BER[:CHA|CHB] 3-236
:JITTer:MEASure:RJ:FIXed[:CHA|CHB] 3-236
:JITTer:MEASure:RJ:FIXed:VALue[:CHA|CHB] 3-237
:JITTer:MEASure:CORRection[:CHA|CHB] 3-237
:JITTer:MEASure:CORRection:DJ:SCALe[:CHA|CHB] 3-238
:JITTer:MEASure:CORRection:RJ:SCALe[:CHA|CHB] 3-238
:JITTer:MEASure:CORRection:RJ:RMS[:CHA|CHB] 3-239
:JITTer:MEASure:CROSsing[:CHA|CHB] 3-240
:JITTer:MEASure:CROSsing:MANual[:CHA|CHB] 3-240
:JITTer:GRAPh:ESTimate:RJDJ:CHA|CHB 3-241
:JITTer:GRAPh:ESTimate:RJPJ 3-241
:JITTer:GRAPh:DDJ:ALL|FALL|RISE 3-242
:JITTer:MEASure:PJ:FREQuency:CALCulate 3-242

RTM? A-8
INF? A-8
:SYSTem:MEMory:INITialize A-8
INI A-9
HCP A-9
OON A-9
:SENSe:MEASure:ASTRt A-10
SAT A-10
SOT A-10
STT? A-10
:OUTPut:CMU:REFClock A-11
RFC A-11
CRE A-12
CEC A-12
SOP A-13
:SENSe:PARam:AEXecute A-15
:OUTPut:CLOCk:OPERation A-16

OPE A-17
:OUTPut:CLOCk:FREQuency A-19
:OUTPut:CMU:FREQuency A-19
CRF A-20
:OUTPut:CMU:RESolution A-20
CRS A-21
:OUTPut:CLOCk:OFFSet:PPM A-22
COP A-22
PTS A-23
LGC A-24
DLN? A-25
DON A-25
DAP A-26
DAT A-26
PRO? A-27
EAD A-27
EAV A-28
ESI A-28
ERT A-29
DSD A-30
DTH A-30
SYN A-31
SYE A-32
SYM A-33
FPS A-33
HRE A-34
MTR? A-34
ERS? A-35
:SENSe:PARam:AEXecute? A-36
END? A-38
ER? A-39
EC? A-39
CC? A-40
FRQ? A-40
MOD A-41
PRD A-42
CUR A-42
STA A-43
STO A-43
MSR? A-43
MSA? A-43

MSO? A-44
MLP? A-44
ETI? A-44
:FETCh:AMPLitude:MEASurement? A-45
:FETCh:TIME:MEASurement? A-46
:MEASure:AMPLitude[:CHA|CHB]? A-47
:MEASure:TIME? A-48
:MEASure:MASK? A-49
:MEASure:HISTogram:AMPLitude? A-50
:MEASure:HISTogram:TIME? A-51

Chapter 1 Overview

This chapter explains the outline of the remote control, main uses, and glossary.

1.1 What is Remote Control? 1-2
1.2 Main Uses for Remote Control 1-3
1.3 Abbreviations 1-5
1.4 Restrictions on Software Versions 1-6

1.1 What is Remote Control?

The remote control function sends commands via the communications interface from the remote control PC to set the measuring instrument and read the measurement results and measuring instrument conditions.

The BERTWave supports the Ethernet or GPIB as a control interface. For MX210xA and MP2100B, the Option 030 is required to use GPIB interface.

When using either interface, set the number to distinguish the BERTWave from other equipment. This number must be an IP address and TCP port number when using the Ethernet interface, or the GPIB interface when using GPIB connection.
The character strings for controlling the BERTWave are called "command". The command is composed of the ASCII character strings. For example, the following command sets when the signal of the pulse pattern generator (hereafter, PPG) is output to the connector.

```
:OUTput:DATA:OUTput ON
```

A command for reading data from the BERTWave is called "query message". A query command has the question symbol (?) appended to the string. For example, sending the following command queries the PPG bit rate set at the instrument.

```
:OUTput:BITRate?
```

The controller PC receives the following response against the query message from the instrument.

```
1250000
```

The bit rate is 1250000 kbit/s.

When the BERTWave is measured via remote control, the Remote lamp on the screen is lit. Only the power switch and the key **Local/Panel Unlock** on the system menu are valid in this situation. This situation is called panel lock. To unlock the panel, touch **Local/Panel Unlock** on the system menu.

1.2 Main Uses for Remote Control

The main uses for remote control are:

Automation of measurement

To control measuring instruments by executing a program, instead of touch-panel operations. Measurement can be automated by describing the control procedures for controlling the measuring instruments, in the program.

Remote control of instruments

To collect measurement data by controlling measuring instruments installed at remote locations, over communications lines.

Control of multiple measuring instruments

To measure the characteristics of DUTs via the remote control of multiple measuring instruments.

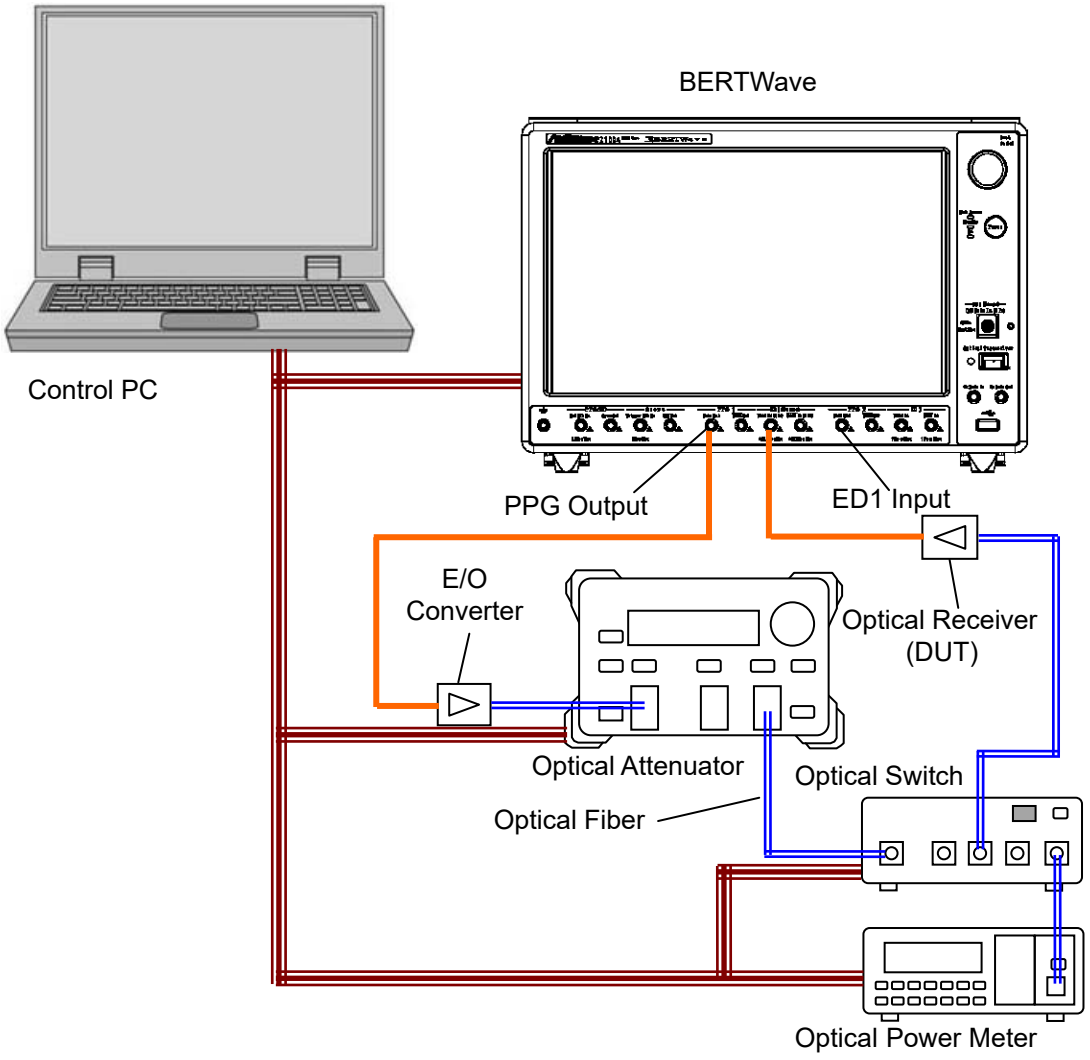


Figure 1.2-1 Example of Controlling Multiple Instruments

Figure 1.2-1 shows an example of controlling multiple instruments. In this example, the bit error rates are measured with changes in the optical input level of the optical receiver. On the control PC, set the attenuation of the optical attenuator to read the optical power level measured by the optical power meter, and the bit error rate measured by the BERTWave. Table 1.2-1 shows the measurement result.

Table 1.2-1 Bit Error Rate of Optical Receiver

Optical Power (dBm)	Bit Error Rate
–25.034	0.011442
–24.523	0.0048758
–24.031	0.001631
–23.536	0.00044241
–23.030	0.000078419
–22.523	0.0000088616
–22.031	0.000000616
–21.524	0.000000016
–21.037	0.00000000028235

1.3 Abbreviations

Table 1.3-1 shows the abbreviations used in this operation manual.

Table 1.3-1 Abbreviations

Abbreviation	Formal name
ASCII	American Standard Code for Information Interchange
CR	Carriage Return
ED	Error Detector
EOI	End or Identify
ESER	Event Status Enable Register
ESR	Event Status Register
GPIB	General Purpose Interface Bus
HiSLIP	High Speed LAN Instrument Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
LF	Line Feed
MAV	Message Available
MSS	Master Summary Status
OSER	Operation Status Enable Register
OSR	Operation Status Register
PC	Personal Computer
PPG	Pulse Pattern Generator
SCPI	Standard Commands for Programmable Interfaces
SRER	Service Request Enable Register
SRQ	Service Request
STB	Status Byte Register
TR	Transition Filter
VISA	Virtual Instrument Software Architecture
VXI-11	VMEbus Extensions for Instrumentation-11

1.4 Restrictions on Software Versions

Some of the commands described in this manual are only available in a specific version or later of the MX210000A BERTWave Control Software.

In MP2100B (version 4) and MP2110A (version 6), the MX210000A version is displayed at the upper right of the application window.

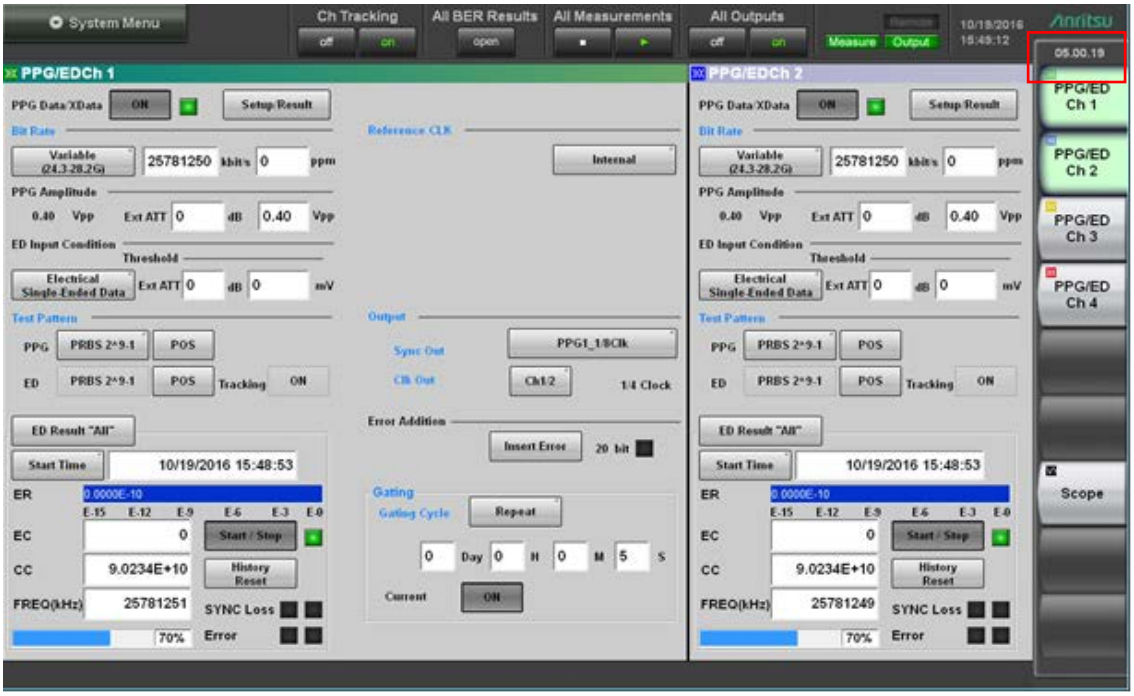


Figure 1.4-1 Display of Software Version Number (MP2100B, MP2110A)

For MP210xA (version 3 or earlier), confirm the version of MX210000A in Information dialog box of **Setup Utility**.

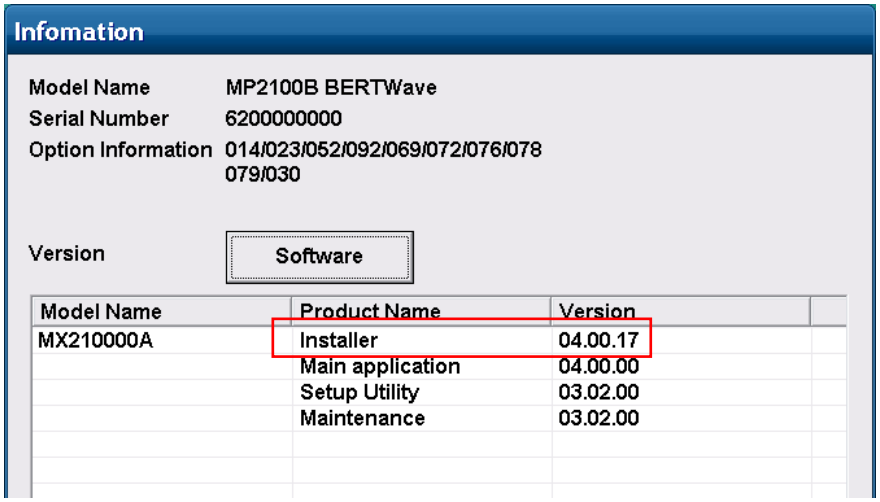


Figure 1.4-2 Display of Software Version Number (MP210xA, MP2100B)

Chapter 2 Before Use

This chapter explains the preparations for using remote control.

2.1 Connection types of Remote Controlling 2-2
2.2 Multiple Sessions (Version 5 or later) 2-3
2.3 Required Equipment 2-4
2.4 Connecting Equipment 2-5
2.4.1 Connecting Ethernet 2-5
2.4.2 Connecting GPIB 2-6
2.5 Setting Interface 2-8
2.5.1 MP210xA and MP2100B 2-8
2.5.2 MP2110A 2-9
2.6 Checking Connection 2-10
2.6.1 When using Ethernet (Windows 7) 2-10
2.6.2 When using GPIB 2-11
2.7 Message Format 2-12
2.7.1 Message Types 2-12
2.7.2 Message Configuration 2-13
2.7.3 Common Commands 2-16
2.7.4 Device Dependent Commands 2-16
2.8 Checking Instrument Status 2-17
2.8.1 Register Structure 2-17
2.8.2 Status Byte Register 2-19
2.8.3 Standard Event Status Register 2-21
2.8.4 Operation Status Register 2-23
2.8.5 Device Dependent Registers 2-26
2.9 Checking If Message Execution Is Completed 2-29
2.9.1 When using Ethernet 2-29
2.9.2 When using GPIB 2-30

2.1 Connection types of Remote Controlling

The following connections can be performed when controlling BERTWave remotely.

- Raw Socket connection
- VXI-11 connection (version 5 or later)
- HiSLIP connection (version 5 or later)
- GPIB connection

Table 2.1-1 lists the resource names when performing the connection using VISA.

Table 2.1-1 VISA Resource Name List

Protocol	VISA Resource Name	Example
Raw Socket	TCPIP[0]::<IP address|host name>::<Port number>::SOCKET	TCPIP::192.168.20.3::5001::SOCKET
VXI-11	TCPIP[0]:: <IP address|host name >::inst0::INSTR	TCPIP::192.168.20.3::inst0::INSTR
HiSlip	TCPIP[0]:: <IP address|host name >::hislip[0] [,<Port number >]::INSTR	TCPIP::192.168.20.3::hislip::INSTR
GPIB	GPIB[0]::<primary address>[::<secondary address>]::INSTR	GPIB::1::INSTR

2.2 Multiple Sessions (Version 5 or later)

When Raw Socket, VXI-11, or HiSLIP is used as the protocol, multiple PCs can connect with MP2110A simultaneously. If the different types of protocols are used, these PCs are not connected with MP2110A simultaneously. When controlling the multiple sessions using VXI-11 or HiSLIP, the session can be locked to perform the exclusion control of the sessions.

Table 2.2-1 lists the maximum number of sessions for each protocol.

Table 2.2-1 Maximum Number of the Sessions List

Protocol	Maximum Number of the Sessions
Raw Socket	30 sessions
VXI-11	30 sessions
HiSLIP	8 sessions
GPIB	1 session

2.3 Required Equipment

The equipment required for remote control is as follows

- Control PC
- Ethernet interface
- GPIB interface*
- Program development tool

Control PC

Prepare the PC that meets the operating environment for the GPIB interface and program development tools.

Ethernet Interface

Prepare Ethernet-compliant interface and cable.

GPIB Interface*

Prepare IEEE 488.2-compliant GPIB interface and cable.

Program Development Tool

Prepare a tool for developing and running programs for performing remote control. For the requirements specification of the program development tool, refer to the manuals that come with the prepared tool.

VISA is required for using the sample program explained in Appendix C.

*: The option 030 is required for using GPIB in MP210xA and MP2100B.

2.4 Connecting Equipment

2.4.1 Connecting Ethernet

Connect the Ethernet connector on the side-panel of the BERTWave and external devices using LAN cables.
The Ethernet connector is located on the left side panel for MP210xA and MP2100B, and on the rear panel for MP2110A
Use a LAN cable to connect the BERTWave and a control PC directly.
Use a LAN cable via a network hub when connecting to multiple external devices.

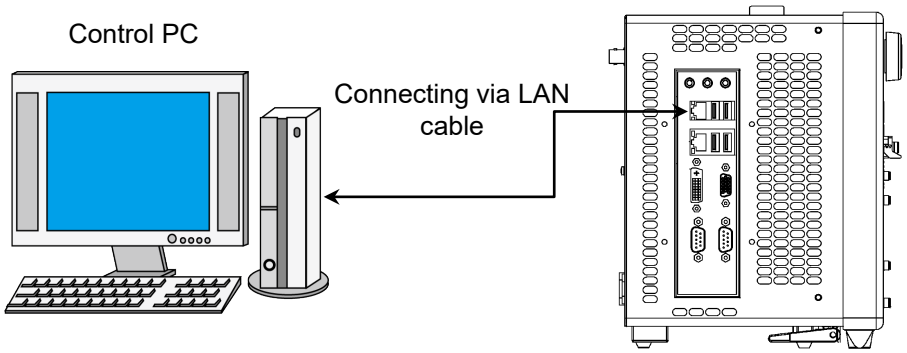


Figure 2.4.1-1 Direct Connection between BERTWave and Control PC

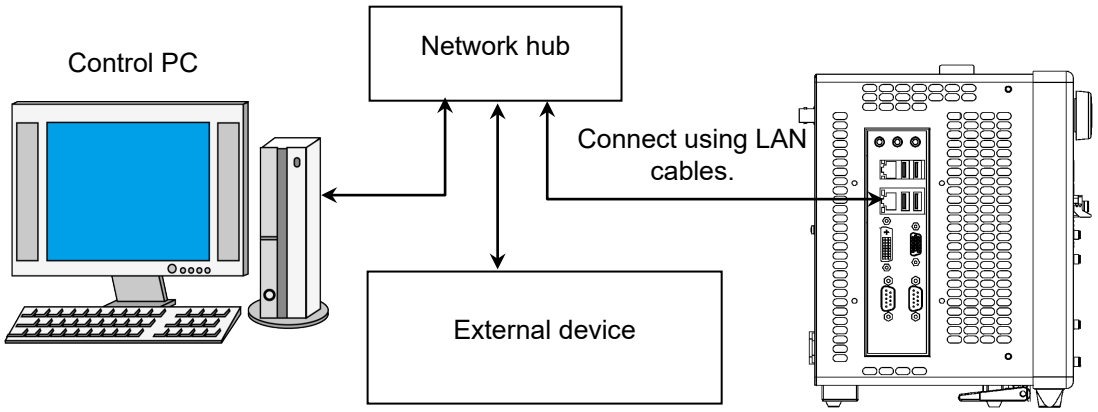


Figure 2.4.1-2 Sample Connection with Multiple External Devices

Notes:

- MP210xA and MP2100B cannot connect to the network including the address range from 192.168.1.0 to 192.168.1.255.
- The control PC may have difficulty in communicating with the BERTWave, depending on the status of communications between them. The direct connection is recommended to ensure communication stability.

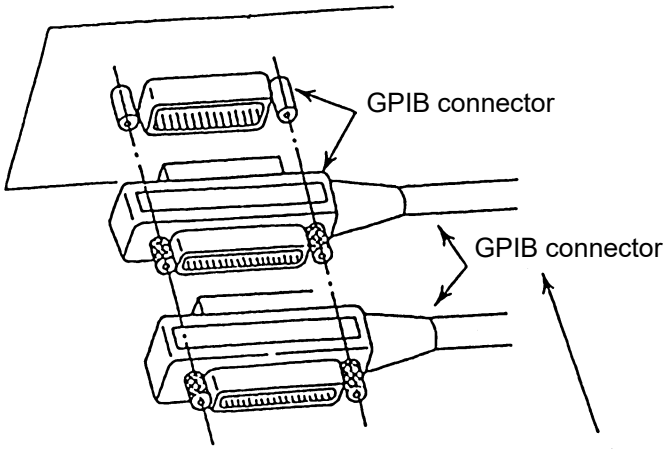
2.4.2 Connecting GPIB

Connect the GPIB connector on the rear panel of the BERTWave and an external device using a GPIB cable.

⚠ CAUTION

Always connect the GPIB cable BEFORE turning on the power to the BERTWave. Connecting it while the power is on may damage internal circuits.

Up to 15 devices, including the external PC controller can be connected to one BERTWave unit. Always follow the conditions shown below when connecting devices.



Total cable length:	Up to 20 m
Cable length between devices:	Up to 4 m
Number of devices that can be connected:	Up to 15

Figure 2.4.2-1 GPIB Cable Connection 1

Connect cables without forming loops.

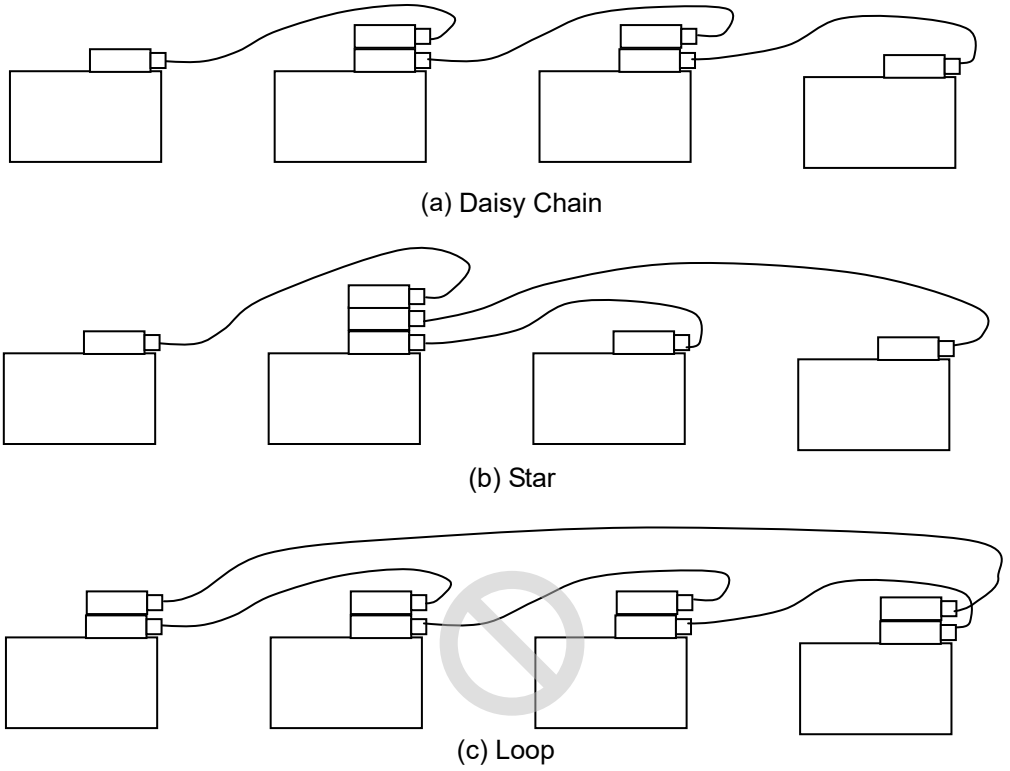


Figure 2.4.2-2 GPIB Cable Connection 2

2.5 Setting Interface

For details on how to set the interface, refer to 2.14 “Setting Interface for Remote Control” in the *MP2100B BERTWave Operation Manual* and 4.3.10 “Remote Control” in the *MP2110A BERTWave Operation Manual*.

2.5.1 MP210xA and MP2100B

Set the remote control interface to the Ethernet using the following method, and enter the IP address.

1. Switch on the power to the MP210xA or MP2100B.
2. Touch **Setup Utility** at the Selector screen.
3. Touch **Remote Control**.
4. In order to use Ethernet, touch the Active Interface button to set the button display to **Ethernet**. To use GPIB, set the button display to **GPIB**.
   When the Option 030 is not installed, the Active Interface button is disabled.
5. When using Ethernet, set the IP address, subnet mask, gateway and port number.
   The gateway address can be omitted.
   The port number can be set from 1024 to 5001.
   When using GPIB, set the GPIB address.
6. Touch **Apply**, and then the settings are completed.
   Touch **Exit**, and then the set value is deleted.

Note:

Do not set the following IP address.
192.168.1.xxx

2.5.2 MP2110A

Set the remote control interface to the Ethernet using the following method, and enter the IP address.

1. Switch on the power to the MP2110A.
2. Click **System Menu**.
3. Click **Remote Control**.
4. Set the IP address, subnet mask, gateway and port number.
   The gateway address can be omitted.
   The port number can be set from 1024 to 5001.
5. Click **OK**, and then the settings are completed.
   Click **Cancel**, and then the set value is deleted.

Note:

Do not set the following IP address.
169.254.1.xxx

2.6 Checking Connection

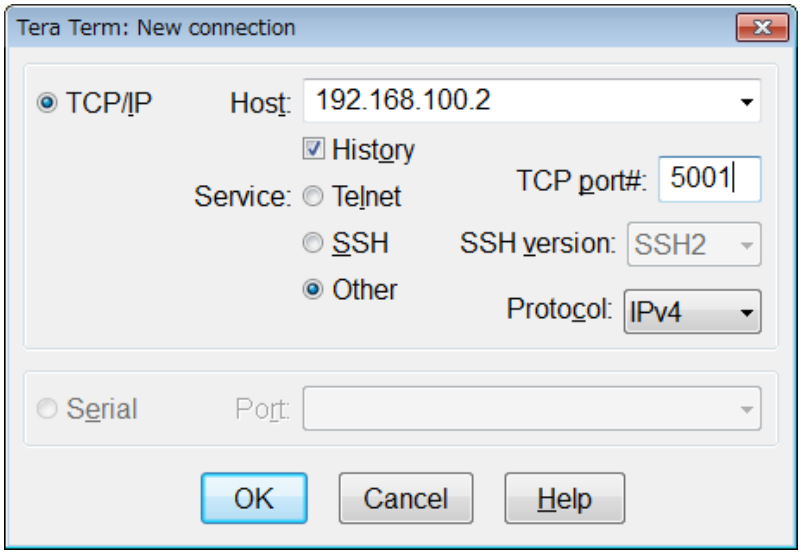
This section describes how to check if the Control PC can recognize the BERTWave

2.6.1 When using Ethernet (Windows 7)

This section explains how to use the free software, Tera Term Version 4.69.

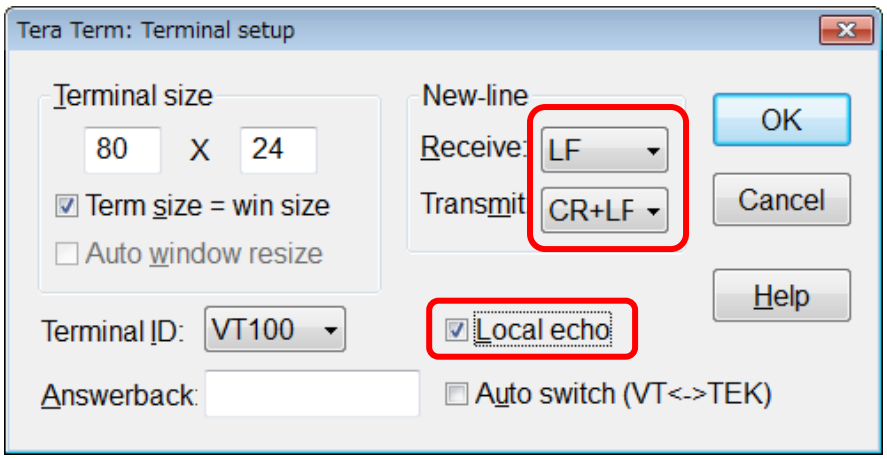
1. When starting Tera Term, the **New connection** window is opened.
   Enter the IP address and TCP port number in the **Host**.
   Set the service to **Others** and protocol to **IPv4**.
   Click **OK**.

   If the BERTWave IP address is set to 192.168.100.2, and the port number is set to 5001, set as follows.



2. When Tera Term recognizes the BERTWave, the communication window is displayed.
3. Click **Settings (S)** - **Terminal (T)..** on the menu.

4. Set the return cord reception to **LF** and those of counterpart to **CR+LF**. Check the local echo and click **OK**.



5. Send *IDN?.
   Confirm that the response is displayed from the BERTWave.

2.6.2 When using GPIB

1. Install the software drivers for the GPIB interface.
2. Run the software.
   For the operation method, refer to the GPIB interface operation manual.
3. Check the displayed instrument address.

2.7 Message Format

2.7.1 Message Types

Messages are composed of the character strings indicating message and message end. The character string indicating the message end is LF (Line Feed) or CR (Carriage Return) +LF.

Note:

If LF or CR+LF is not attached to the message end, a timeout error occurs because the communication does not end.

Messages are composed of the following types depending on the transmission direction:

Program Messages

Messages sent from PC to instrument

There are two types of the program messages:

- Command
  This can be used for measurement condition settings and measurement start.
- Query
  This queries the status and settings of the measuring instrument. When transmitting the query, the instrument creates a response message to the query.

Response Messages

Messages sent from instrument to PC controller

2.7.2 Message Configuration

The messages are composed of header and data parts separated by more than a half width space. Program messages always have a header but sometimes have no data. Response messages always have data but sometimes have no header.

Header

The command header has the following types:

- Simple header
  The header is composed of alphanumeric characters and underbars, and the initial character is an alphabetic character.
  Example: `STA`
- Common command header
  The header is composed of alphanumeric characters and underbars, and the initial character is an asterisk (*).
  Example: `*CLS`
- Multiple headers
  Single headers are linked by colons. Colons can be used at the header. Multiple headers can be used to configure layered processing.
  Example: `:SENSE:MEASURE:START`

Queries have a question mark (?) appended to the header.

Example:
```
*ESE?
:CONFIGURE?
```

Data

The data format is character string data, numeric data, and binary data.

String data is ASCII code enclosed in quotation marks.
An example of the program message when inputting `Model ANR-005` at the title is shown below.

Example:
```
:SYSYEM:MEMORY:STORE 'Model ANR-005',0,ALL
:SYESEM:MEMORY:STORE "Model ANR-005",0,ALL
```

When quotation marks are included in the character string, paired marks are used.

Example:
```
He said "Good product". → "He said ""Good Product""."
He said 'Good product'. → 'He said ''Good Product''.
'
```

In addition, paired quotation marks can be used inside other paired quotation marks.

Example:

```
He said "Good product". → 'He said "Good Product".'
He said 'Good product'. → "He said 'Good Product'."
```

The numeric values can be described by using numeric data, input numeric values either as decimal, binary, octal, or hexadecimal numbers. When using the binary, octal, or hexadecimal numbers, put #B, #O, or #H before the data.

Example:

```
10   #B1010            #O12        #HA
1550 #B11000001110     #O3016      #H60E
```

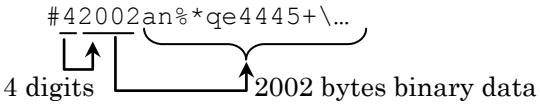
When using decimal numbers, use integer number, fixed point, and floating point. The following examples indicate the same values.

Example:

```
-10        -10.00      -1E1
1250       1250.000    1.25E3
0.0023     2.3E-4
```

For the binary data, the head string starts with a sign (#) and continues with data after a numeric value indicating the data length.

Example:



When there are multiple data in a message, separate each of them with commas (,).

Example: `:INPUT:DATA:ATTFACTOR 1,6`
`:SENSE:MEASURE:EALARM:PERIOD 0,0,1,0`

When concatenating multiple program messages, separate the message with semicolons (;).

Example: `:MOD:ID 5;:DISP:MODE EYE;:SAMP:STAT RUN`

Note:

When sending multiple messages separated by semicolons, the maximum length of the concatenated string is 1024 bytes.

When sending a concatenated string of query messages, response messages are separated by semicolons.

Example: `:MOD:ID 1;:OUTP:BITR:STAN?;:OUTP:BITR?`
`>"10G_LAN";10312500`

2.7.3 Common Commands

The GPIB specifications (IEEE 488.2) define equipment commands. In this manual, these defined commands are called common commands.

The common commands are divided into mandatory and option commands. The BERTWave supports the common commands listed in Table 2.7.3-1.

Table 2.7.3-1 Common Commands

Command	Explanation
*CLS	Clears stand event register and output queue
*ESE	Sets and queries standard event enable register
*ESR	Queries standard event register
*IDN	Queries product information
*OPC	Sets/queries bit setting and bit 0 for status byte indicating message processing completion
*OPT	Queries option information
*RST	Initializes BERTWave setting conditions
*SRE	Sets and queries SRER
*STB	Queries status byte register
*TRG	Starts measurement
*WAI	Waits previous sent message completion

2.7.4 Device Dependent Commands

In this manual, commands that differ according to the functions of the measuring instrument are called Device Dependent Commands.

This instrument has two types of Device Dependent Commands.

- SCPI
  Commands meeting SCPI standard
- Native
  Commands consisting of at least three ASCII characters

2.8 Checking Instrument Status

The BERTWave has registers indicating status, such as errors and command execution status. This section explains these registers.

2.8.1 Register Structure

Figure 2.8.1-1 shows the structure of the registers indicating the instrument status.

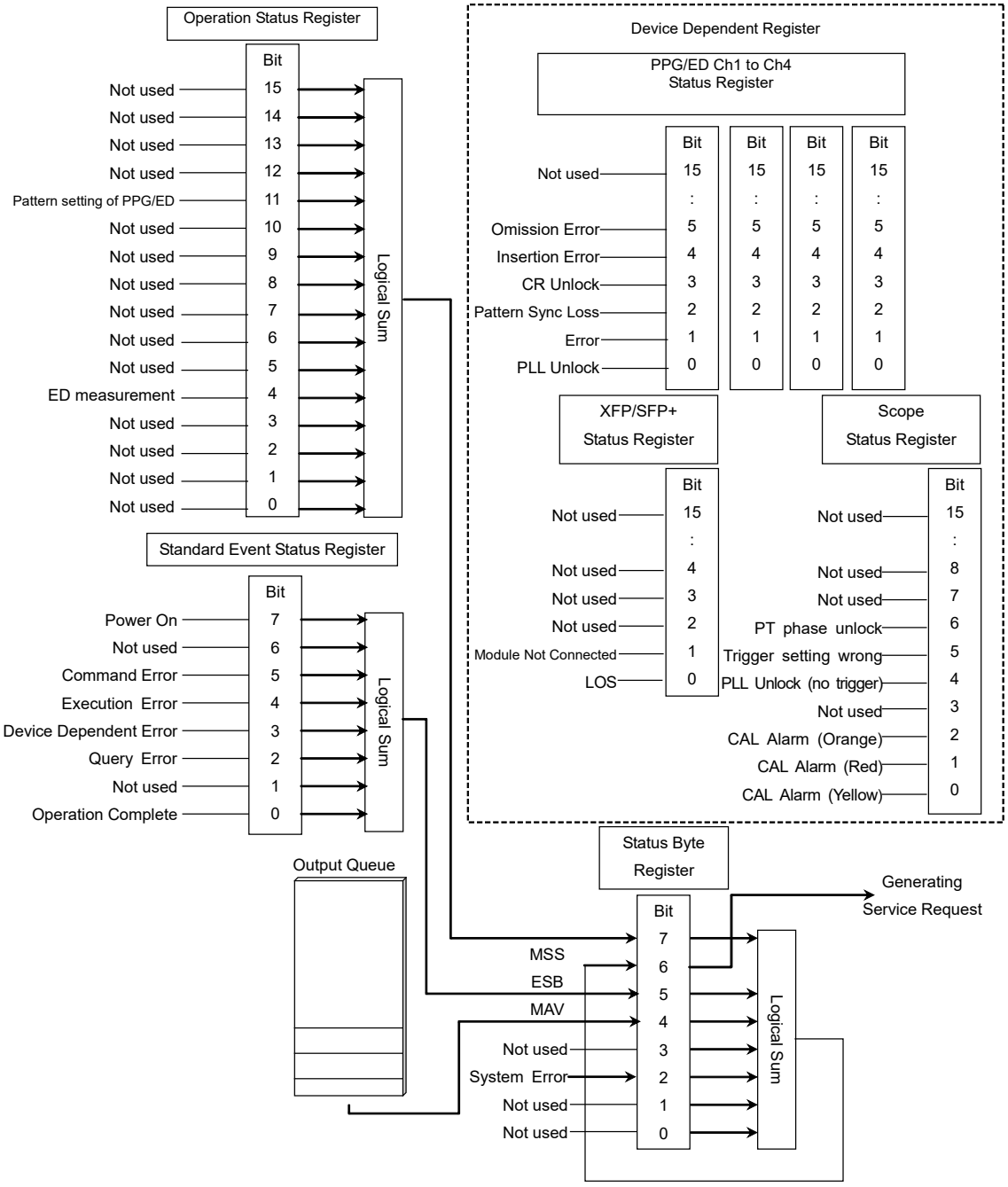


Figure 2.8.1-1 Register Structure

Each register uses 8-bit or 16-bit data. The register output values are the decimal totals for each bit shown in Table 2.8.1-1.

Table 2.8.1-1 Register Bit Decimal Conversion Values

Bit	Decimal value	Bit	Decimal value
0	1	8	256
1	2	9	512
2	4	10	1024
3	8	11	2048
4	16	12	4096
5	32	13	8192
6	64	14	16382
7	128	15	32764

The service request enable register (SRER) has a corresponding status byte register.

2.8.2 Status Byte Register

The status byte register (STB) displays the status of equipment defined by the GPIB standards. When the equipment status changes, the value in the STB changes too. It can be used to generate interrupts to the PC controller. These interrupts are called service requests.

There is a service request enable register (SRER) for the STB. The SRER can select the status byte bit generating the service request.

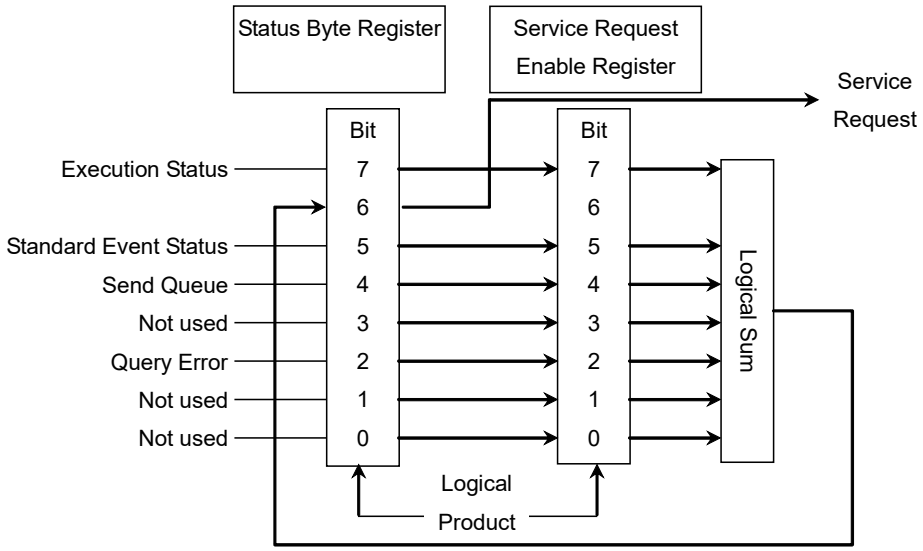


Figure 2.8.2-1 Configuration of Status Byte Register and Service Request Enable Register

Note:

When using the GPIB interface, the service request is enabled.

The following methods are used to read the status byte register.

- Using common *STB? command
- Using GPIB serial poll (when the Option 030 is installed for MP210xA or MP2100B)
  Read the GPIB interface manual for the serial poll method.
  When using serial polling, even if bit 6 is 1, it becomes 0 after reading once.

The `*SRE` and `*SRE?` common commands can be used for setting and reading the SRER for setting reading of the status byte register. To output the STB data, set the bit corresponding to the SRER to 1.

The meaning of each bit of the STB is shown in the following table.

Table 2.8.2-1 Meaning of Status Byte Register

Bit	Explanation
7	This is the logical sum of each bit of the logical product of the OSR and its event enable register.
6	MSS (Master Summary Register) It is the logical sum of the bit 5 to 0, bit 7 logical product of the STB and the SRER.
5	This is the logical sum of each bit of the logical product of the standard event status register and standard event enable register.
4	MAV (Message Available summary) This is always 1 when there is a response message in the output queue of this instrument
3	Not used; always 0
2	Becomes 1 at System Error
1	Not used; always 0
0	Not used; always 0

Bit 7 of the STB indicates information about the OSR.
For details about the information, refer to section 2.8.4 “Operation Status Register”.

Bit 6 of the STB is called the master summary status (MSS) bit. When it is 1, there is a notification from BERTWave to the PC controller. When it changes to 1 from 0, a service request is generated.

Bit 5 of the STB indicates information about the standard status register.
For details about the information, refer to section 2.8.3 “Standard Event Status Register”.

The device dependent register data is not indicated in the STB.
Bits 7 and 5 of the STB can be set to 0 using the `*CLS` common command.
When `*CLS` is sent after a command or when a query is sent after `*CLS`, the send queue is cleared and bit 4 is set to 0.

The SRER cannot be set to 0 by `*CLS`, so use `*SRE`.

2.8.3 Standard Event Status Register

There is a standard event status enable register (ESE) for the standard event status register (ESR). The logical product of these two registers and the logical sum of each bit of this result is output to bit 5 of the STB.

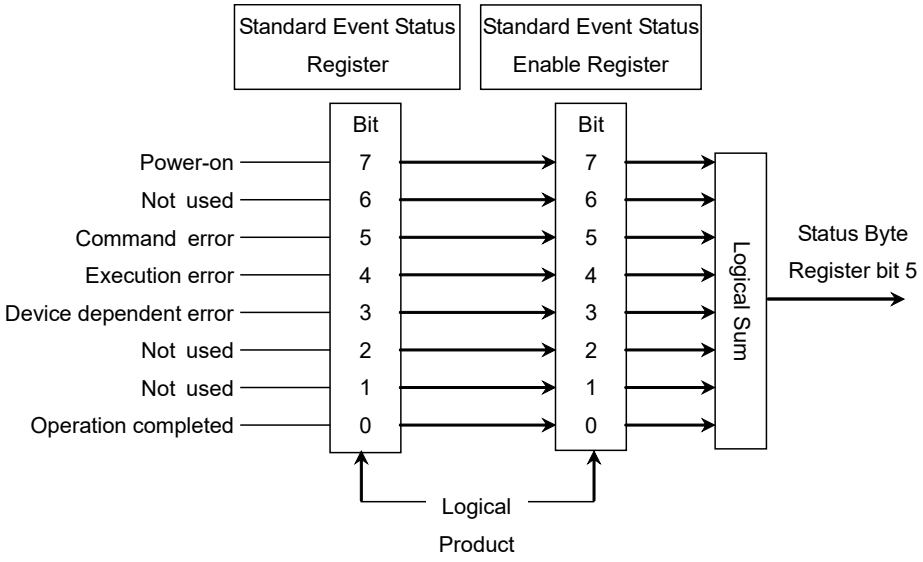


Figure 2.8.3-1 Configuration of Standard Event Status Register and Standard Event Status Enable Register

The meaning of each bit of the ESR is listed in the table below.

Table 2.8.3-1 Meaning of Standard Event Status Register

Bit	Explanation
7	1 if the BERTWave is powered on.
6	Not used; always 0
5	1 if a command error occurs. Refer to Appendix B, “Error Codes” for details.
4	1if an execution error occurs. Refer to Appendix B, “Error Codes” for details.
3	1if a device-dependent error occurs. Refer to Appendix B, “Error Codes” for details.
2	Not used; always 0
1	Not used; always 0
0	Operation Complete Becomes 1 when entire command operation completed after *OPC command operation

Bit 7 to bit 0 of the ESR can be read by the `*ESR?` command.
The standard event register returns to 0 when read.

The ESE can be set and read using the `*ESE` and `*ESE?` commands. To output standard event register data, set the bit corresponding to the enable register to 1.

The bit 0 can be read using the `*OPC` command.

The standard register can be set to 0 using the `*CLS` command.

2.8.4 Operation Status Register

The operation status register (OSR) is composed of the following registers:

- Operation status condition register
- Transition filter
- Operation status event register
- Operation status enable register (OSER)

The operation status condition register indicates changes in the status. When the status changes, the value of this register also changes.

The OSER records changes in the value of the execution status condition register. There is a transition filter that defines the write condition before the OSER. The transition filter sets the OSER to 1 under any of the following conditions:

- When bit changes from 0 to 1
- When bit changes from 1 to 0
- When bit changes from 0 to 1 or bit changes from 1 to 0

The OSER sets the OSER output at each bit. The logical product these two registers is obtained and the logical sum of each bit of the result is output at bit 7 of the STB.

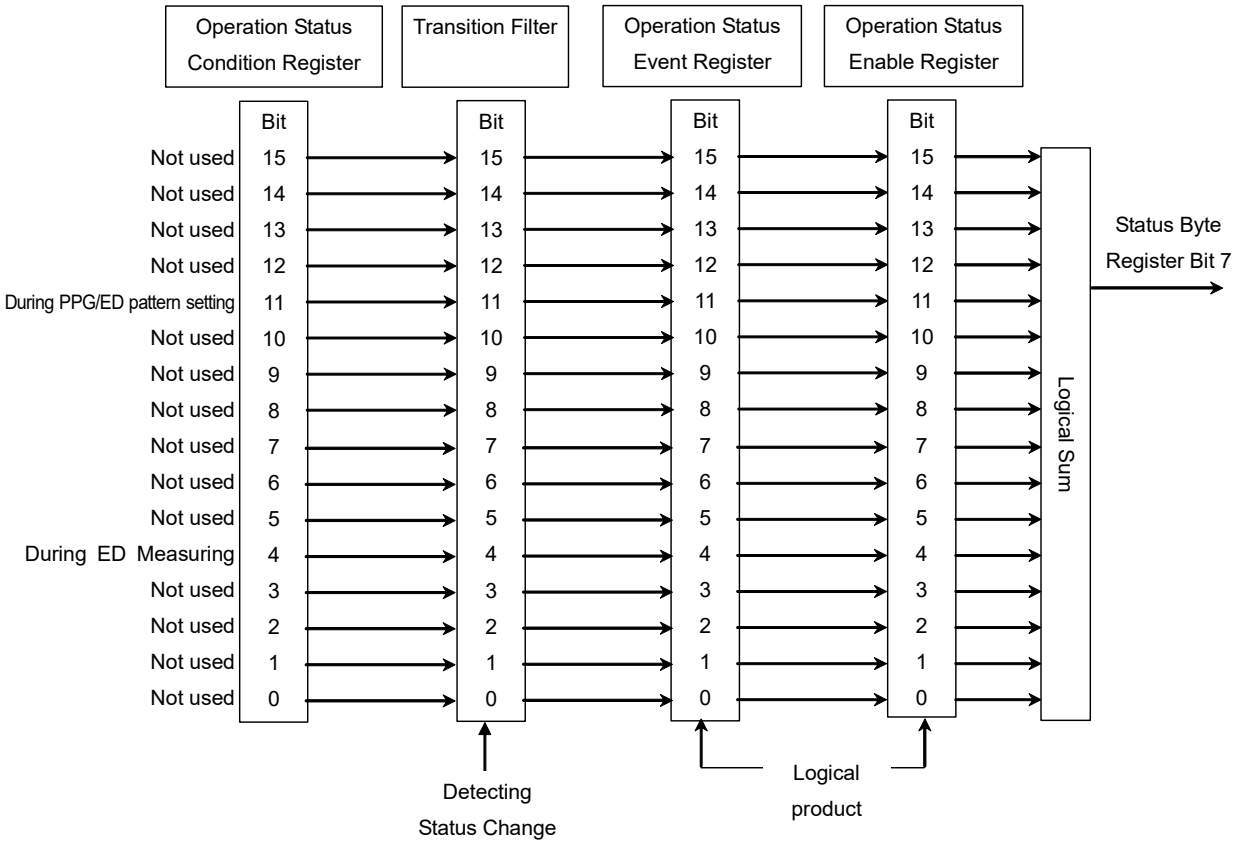


Figure 2.8.4-1 Configuration of Operation Status Condition Register, Operation Status Event Register, Operation Status Enable Register, and Transition Filter

Each bit definition of the execution status is as follows.

Table 2.8.4-1 Bit Definition of Operation Status Register

Bit	Explanation
15 to 12	Not used; always 0
11	1 during execution of PPG/ED pattern setting.
10 to 5	Not used; always 0
4	1 during execution of ED measurement.
3 to 0	Not used; always 0

The commands for confirming the execution start or end time at the OSR are shown in the following table.

Table 2.8.4-2 Commands for Confirming Execution of operation at Operation Status Register

Operation Status Register Bit	Command
11	:SENSe:MMEMory:PATTern:RECall :SENSe:PATTern:TYPE :SOURce:MMEMory:PATTern:RECall :SOURce:PATTern:TYPE
4	[:BERT:ALL]:SENSe:MEASure:STARt [:BERT:ALL]:SENSe:MEASure:STOP

To detect the execution start, the transition filter bit response to STATus:OPERation:PTRansition is set to 1.
To detect the execution end, the transition filter bit response to :STATus:OPERation:NTRansition.
The OSER can be read using :STATus:OPERation[:EVENt]?. When the register is read, the OSR returns to 0.

The operation status condition register can be read using:STATus:OPERation:CONDition?.

To set the OSER, use:STATus:OPERation:ENBle. To read the OSER, use STATus:OPERation:ENBle?. To output the OSR data, set the bit for the status setting enable register to 1.

When sending :STATus:OPERation:RESet, the operation status event register is set to 0.
Even when sending :STATus:OPERation:RESet, the OSER is not changed.

2.8.5 Device Dependent Registers

The following registers are called the device dependent registers.

- PPG/ED Ch1 to 4 Status Register
- XFP/SFP+ Status Register
- Scope Status Register

The device dependent status register has the same type of condition register, transition filter, and event register. However there is no enable register for switching the output at each bit on/off.

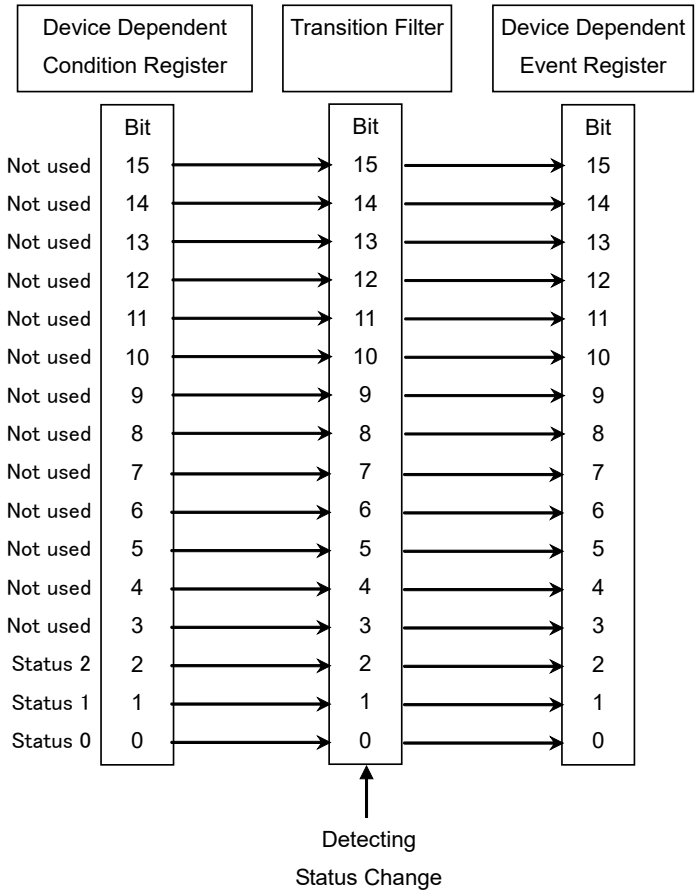


Figure 2.8.5-1 Configuration of Device Dependent Register

When the value of the device dependent status register changes, there is no effect on the STB. As a result, a service request is not generated to the PC controller.

Each bit definition of the device dependent register is as follows.

Table 2.8.5-1 Meaning of PPG/ED Ch1 to Ch4 Status Register

Bit	Explanation
15 to 6	Not used; always 0
5	Indicates Omission Error occurs
4	Indicates Insertion Error occurs
3	Indicates CR Unlock occurs
2	Indicates SYNC Loss occurs
1	Indicates Bit Error occurs
0	Indicates PLL Unlock occurs

Table 2.8.5-2 Meaning of XFP/SFP+ Status Bit

Bit	Explanation
15 to 2	Not used; always 0
1	Indicates LOS occurs
0	Indicates Ready status

Table 2.8.5-3 Bit Meaning of Scope Status Register

Bit	Explanation
15 to 9	Not used; always 0
8	Indicates CRU Unlock occurs
7	Not used; always 0
6	Indicates that Precision Trigger is out of synchronized status.
5	Indicates that the frequency of trigger input signal is abnormal.
4	Indicates PLL Unlock (No trigger input) occurs
3	Not used; always 0
2	Indicates CAL alarm (Orange) occurs
1	Indicates CAL alarm (Red) occurs
0	Indicates CAL alarm (Yellow) occurs

To detect the occurrence of these phenomena, set the transition filter bit to 1 using the following commands:

```
:INSTrument:PE{1|2|3|4}:PTRansition
:INSTrument:XSFP:PTRansition
:INSTrument:WAV:PTRansition
```

To detect the end of these phenomena, set the transition filter bit to 1using the following commands:

```
:INSTrument:PE{1|2|3|4}:NTRansition
:INSTrument:XSFP:NTRansition
```

```
:INSTrument:WAV:NTRansition
```

The device dependent event register can be read using the following queries:

```
:INSTrument:PE{1|2|3|4}[:EVENt]?
:INSTrument:XSFP[:EVENt]?
:INSTrument:WAV[:EVENt]?
```

The device dependent condition register can be read using the following queries:

```
:INSTrument:PE{1|2|3|4}:CONDition?
:INSTrument:XSFP:CONDition?
:INSTrument:WAV:CONDition?
```

The device dependent event register can be initialized using the following queries:

```
:INSTrument:PE{1|2|3|4}:RESet
:INSTrument:XSFP:RESet
:INSTrument:WAV:RESet
```

2.9 Checking If Message Execution Is Completed

Some of the BERTWave program messages may take several seconds to several ten seconds to execute. Depending on the interface (Ethernet or GPIB) you are using, the procedure to check if time-consuming message execution is completed is different.

2.9.1 When using Ethernet

Even during execution of a program message sent to the BERTWave, the subsequent message(s) can be sent. However, until execution of the previously sent message(s) is completed, the subsequent message(s) is not processed, and is stored in the buffer of the BERTWave.

Therefore, note that execution of the previously sent program message(s) may not have always been completed, even if the subsequent message(s) can be sent. To check if execution of a program message is completed, send a query to receive a response.

Example:

`:CALibrate:AMPLitude`	Starts Level calibration for Scope
`:SYSTem:ERRor?`	Query for error code and error message
`> 0,"No Error"`	No error

It may take around fifty seconds to complete execution of the `:CALibrate:AMPLitude` command in this example. `AMPLitude` is sent, and subsequently `:SYSTem:ERRor?`, and then execution completion of `AMPLitude` is checked by receiving a response message.

Note:

If it may take long time to process the command that precedes the query, set the sufficient response timeout, with respect to the command processing time. (Timeout needs to be longer by at least 10 seconds than the command processing time.)

2.9.2 When using GPIB

Sending the subsequent message is forced to wait, on the control PC, until execution of the sent message is completed. Therefore, prevent a communication timeout from occurring during message execution by the BERTWave, when sending a time-consuming command. Set the sufficient timeout for GPIB interface of the control PC, with respect to the command processing time. (Timeout needs to be longer by at least 10 seconds than the command processing time.)

Example: Sending a command that takes about 20 seconds to execute

1. Set the timeout for GPIB interface to 30 seconds.
2. Send a command which takes a time to complete.
3. Reset the timeout to the previous setting.

Chapter 3 Message List

This chapter describes the message details of remote control commands for BERTWave.

3.1 Rules for Describing Messages 3-3
3.2 Correspondence Between Panel and Messages 3-4
3.2.1 Messages corresponding to common operations 3-4
3.2.2 Messages corresponding to PPG/ED 3-6
3.2.3 Messages corresponding to XFP/SFP+ (MP210xA, MP2100B) 3-13
3.2.4 Messages corresponding to O/E (MP210xA, MP2100B) 3-14
3.2.5 Messages corresponding to Scope 3-15
3.2.6 Messages corresponding to Information 3-39
3.2.7 Messages with no corresponding panel operation 3-40
3.3 Messages Corresponding to Common Operations 3-41
3.3.1 Setting system configuration 3-41
3.3.2 Obtaining system information 3-43
3.3.3 System alarm (MP210xA, MP2100B) 3-48
3.3.4 Resetting/recalling settings 3-50
3.3.5 Storing settings and results 3-52
3.3.6 Turning on/off plotting processing 3-54
3.3.7 Screen Copy 3-55
3.3.8 Specifying screen display 3-58
3.4 Status Register 3-59
3.4.1 Clearing register 3-59
3.4.2 Status byte register 3-61
3.4.3 Standard event status register 3-63
3.4.4 Operation status register 3-65
3.4.5 PPG/ED status register 3-69
3.4.6 Scope status register 3-73
3.4.7 XFP/SFP+ status register 3-76
3.4.8 Register operation without dependence on module 3-79
3.5 Specifying Module and Channel 3-82
3.5.1 Specifying with command 3-82
3.5.2 Specifying in header (Added in Version 3.02) 3-83
3.6 BERT(PPG/ED) messages 3-84
3.6.1 Window Operation 3-84
3.6.2 Clock Input 3-84
3.6.3 Clock Output 3-89
3.6.4 Tracking Setting 3-91

3.6.5 Bit Rate Setting 3-93
3.6.6 Test Pattern Setting 3-102
3.6.7 PPG 3-108
3.6.8 ED 3-113
3.7 XFP/SFP+ specific messages (MP210xA, MP2100B) 3-132
3.8 O/E messages 3-135
3.9 Scope-specific messages 3-136
3.9.1 Setting active channel and On/Off display ... 3-136
3.9.2 Basic operation 3-139
3.9.3 Setup 3-141
3.9.4 Time 3-159
3.9.5 Amplitude, O/E 3-174
3.9.6 Measure 3-191
3.9.7 Acquiring results 3-217
3.9.8 Marker 3-230
3.9.9 Jitter 3-233

3.1 Rules for Describing Messages

The following table shows the rules for describing messages.

Table 3.1-1 Rules for Describing Messages

Symbols	Usage
<>	Parameters in angled bracket are input by the programmer.
[]	Messages or parameters in square brackets can be omitted.
|	Select one of choices separated by vertical bars. For example, if A | B | C | D are choices, select one of them.
{}	Group the choices. For example, A | B({C | D}) means that A, B(C), or B(D) is available.
<binary_data>	This string is in binary data format.
<character>	Alphabet or numeric characters
<file_name>	The string indicates file name and path. The double quotation marks or single quotation marks are needed at the beginning and end of the data. \,/,:,*,?,",<,>,| are not used in the file name. Example: "PATTERN005"
<integer>	Decimal integer Example: –100, 12500000
<numeric>	Decimal number Example: 0, 1.2E–6, 2.35
<string>	String data The double quotation marks or single quotation marks are needed at the beginning and end of the data.
<enable>	On/Off setting To turn off, specify 0 or OFF. (Response: 0) To turn on, specify 1 or ON. (Response: 1) Example: 0, 1, OFF, ON
…	Indicates that multiple parameters or responses are omitted.
>	Precedes a response, in Example of Use.

Some parts of the header strings can be omitted.
The lower-case characters can be omitted, but the upper-case characters cannot be omitted.
Example: `:STATus:OPERation:EVENt?`

The following are also acceptable:

```
:STAT:OPER:EVEN?
:STAT:OPERATION:EVEN?
:STATUS:OPERAT:EVENT?
:STATUS:OPERATION:EVEN?
:STATUS:OPERATION:EVENT?
```

BERTWave interprets them in the same way.

3.2 Correspondence Between Panel and Messages

This section explains correspondence between panel and messages.

3.2.1 Messages corresponding to common operations

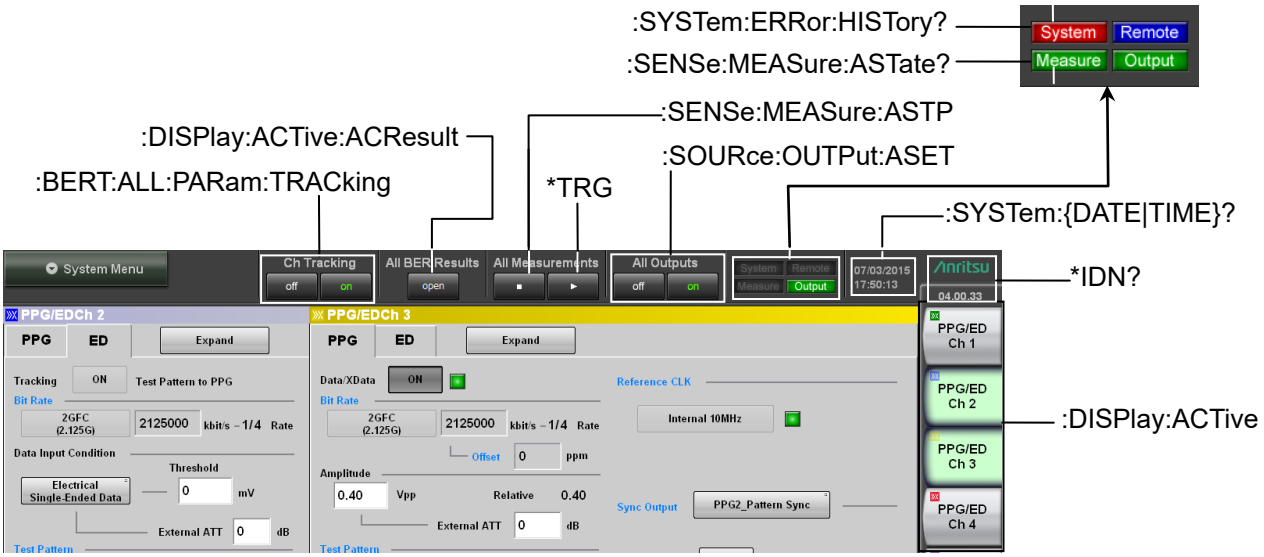


Figure 3.2.1-1 Messages Corresponding to Common Operations

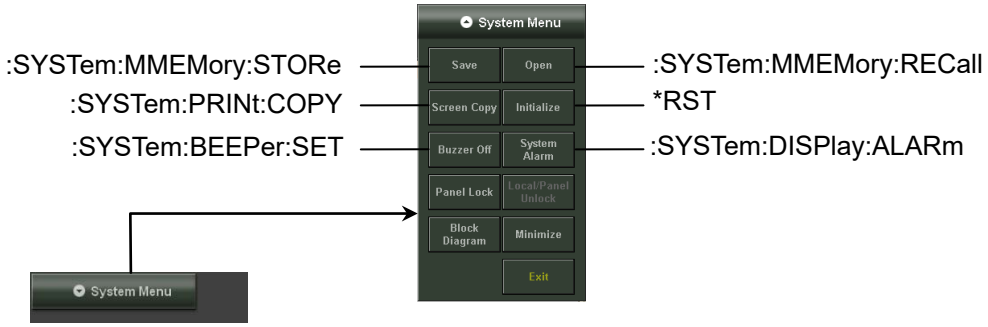


Figure 3.2.1-2 Messages Corresponding to System Menu (MP210xA, MP2100B)

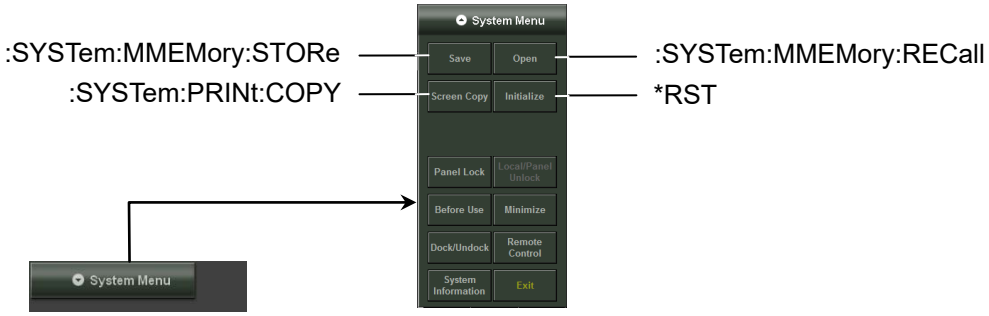


Figure 3.2.1-3 Messages Corresponding to System Menu (MP2110A)

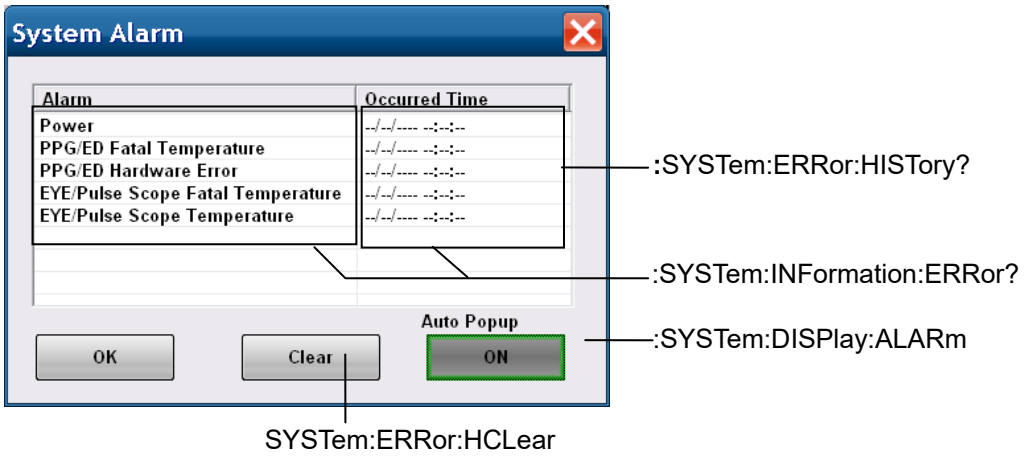


Figure 3.2.1-4 Messages Corresponding to System Alarm Dialog Box (MP210xA, MP2100B)

3.2.2 Messages corresponding to PPG/ED

When controlling PPG/ED, add :BERT[<ch>] to the beginning of the message header or specify a channel using the :MODule:ID command. Refer to Section 3.5, "Specifying Module and Channel" for details.

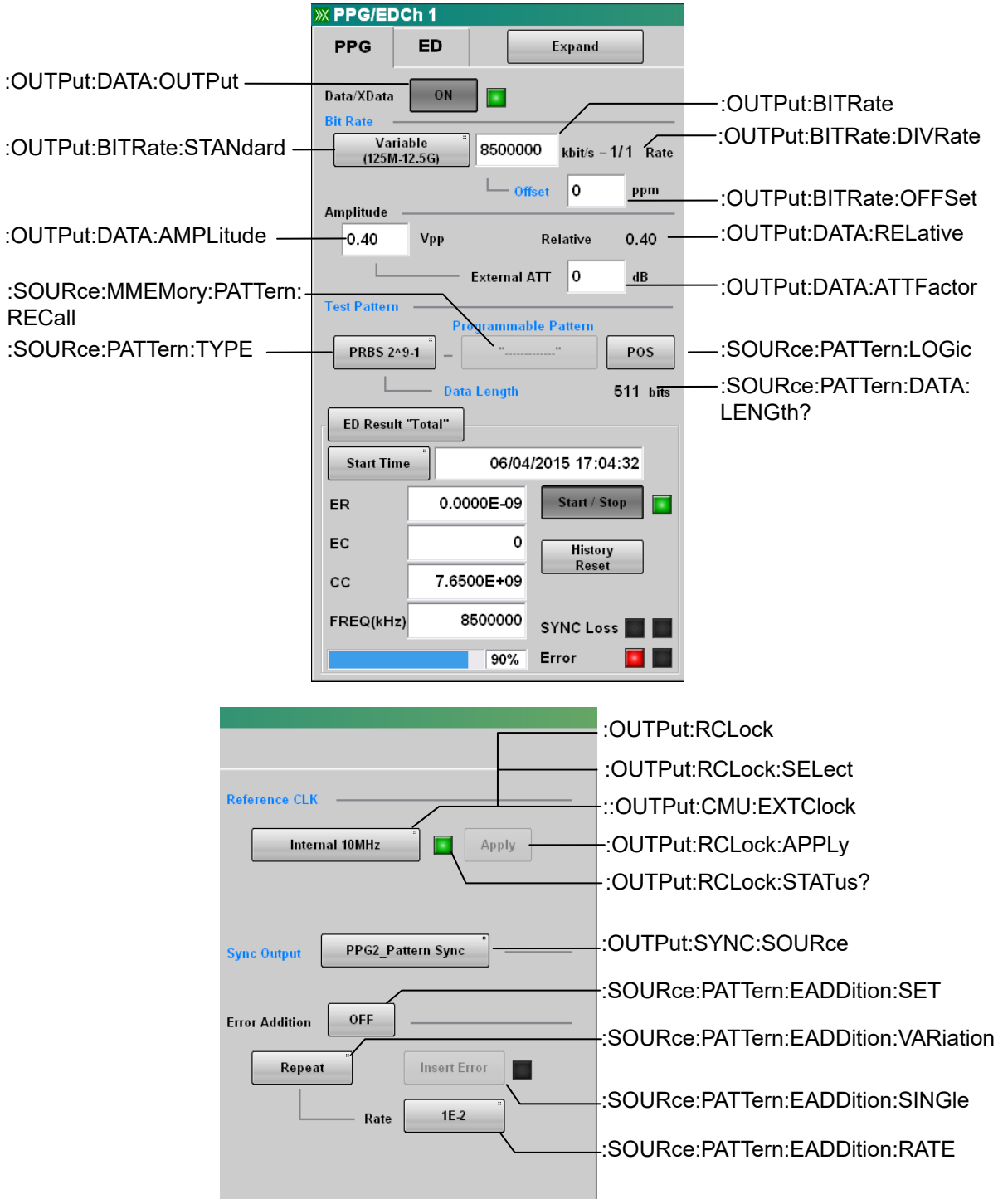


Figure 3.2.2-1 Messages Corresponding to PPG Panel (MP210xA, MP2100B)

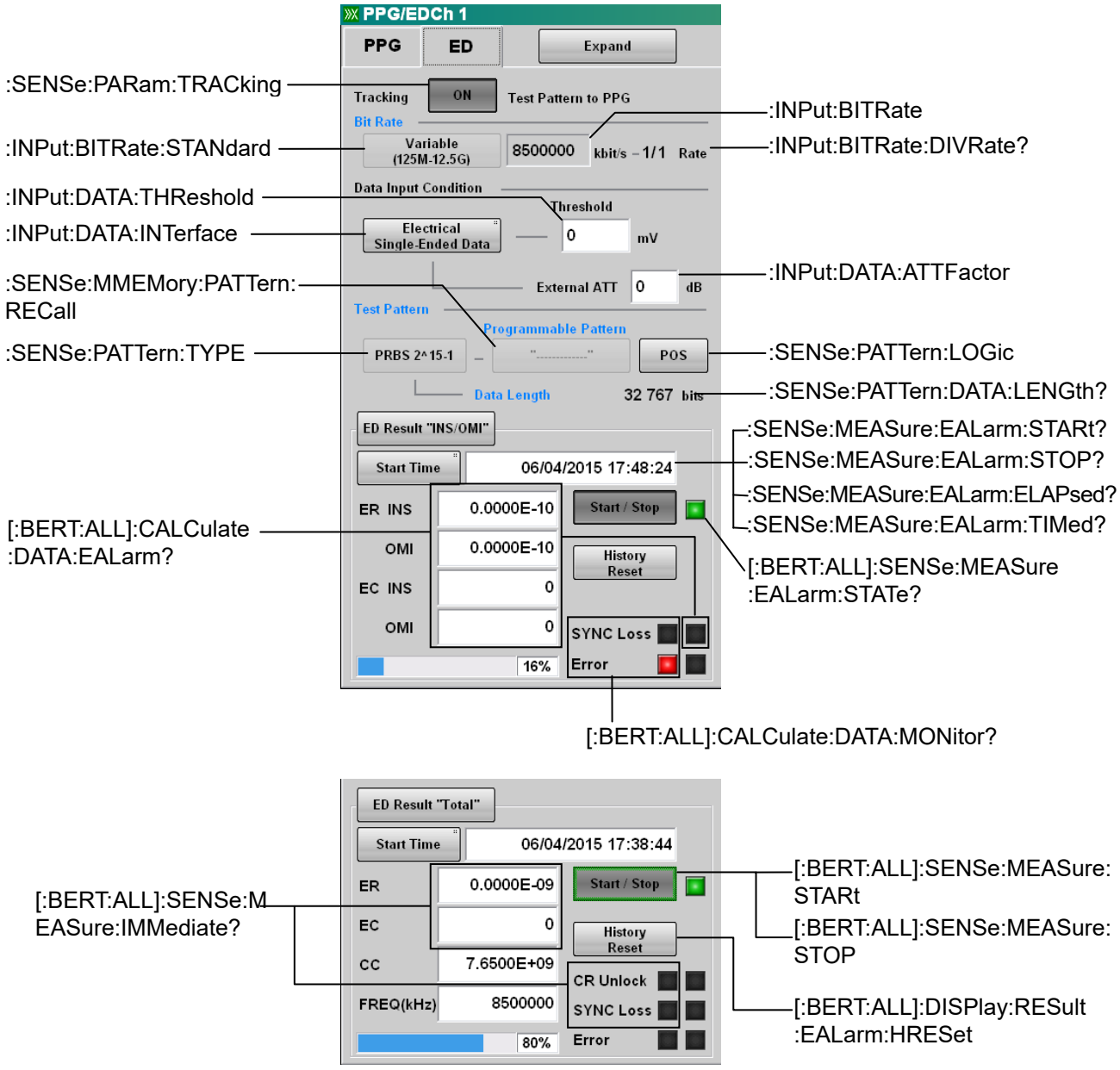


Figure 3.2.2-2 Messages Corresponding to ED Panel–1 (MP210xA, MP2100B)

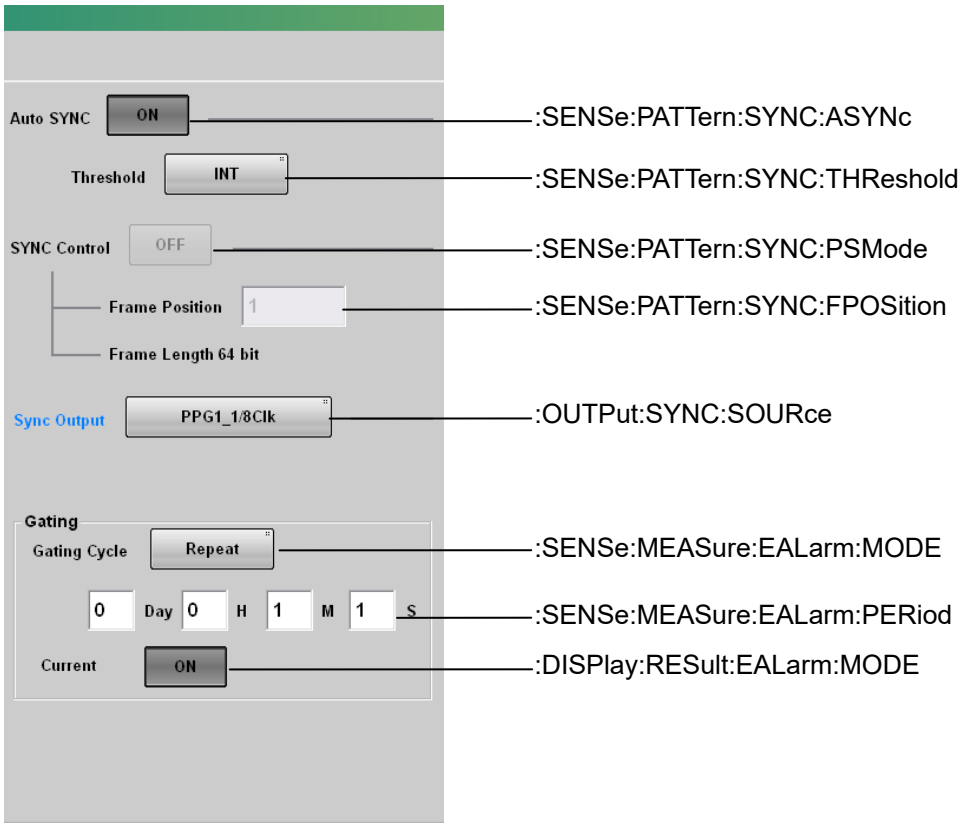


Figure 3.2.2-3 Messages Corresponding to ED Panel–2 (MP210xA, MP2100B)

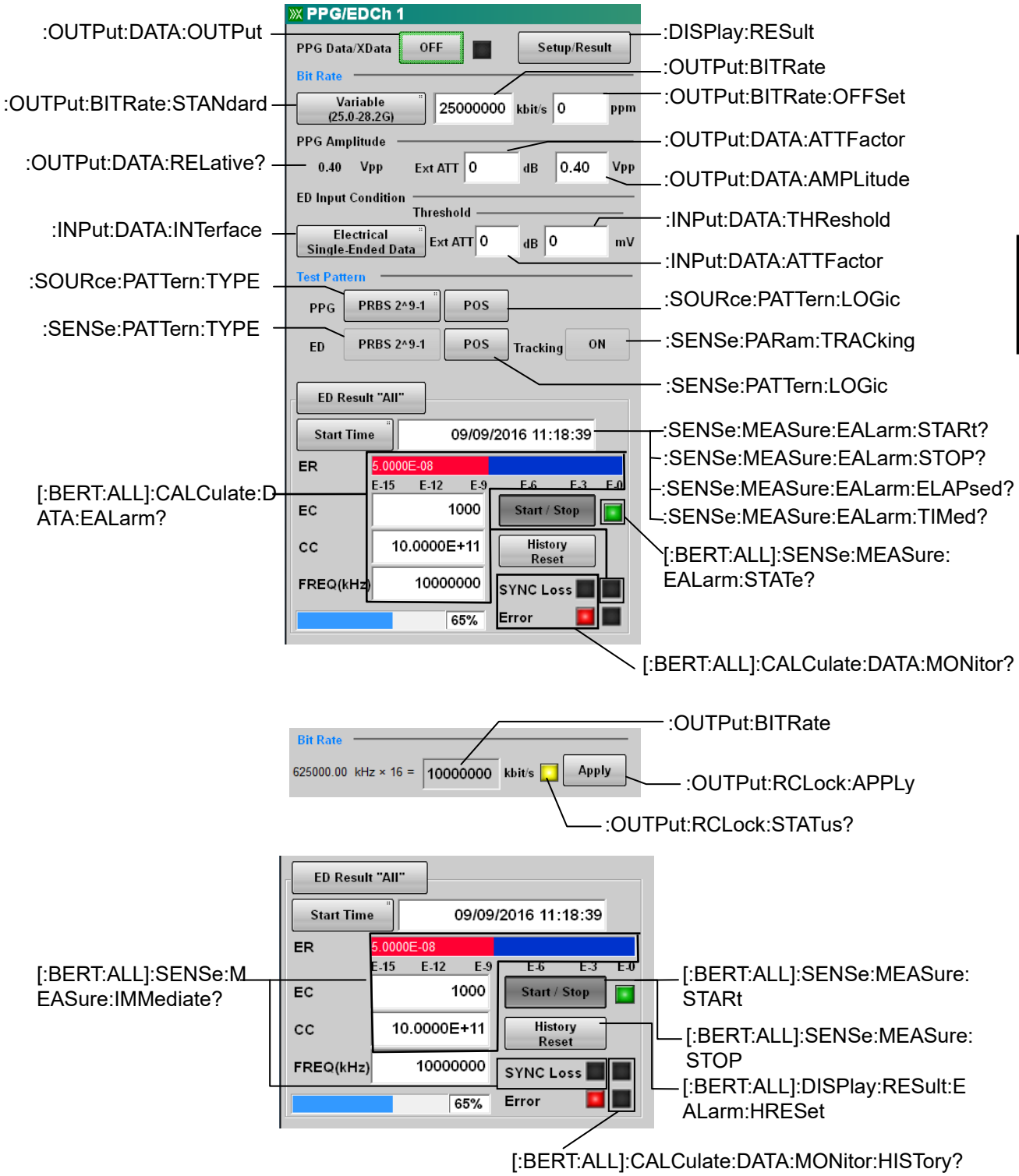


Figure 3.2.2-4 Messages Corresponding to PPG/ED Panel–1 (MP2110A)

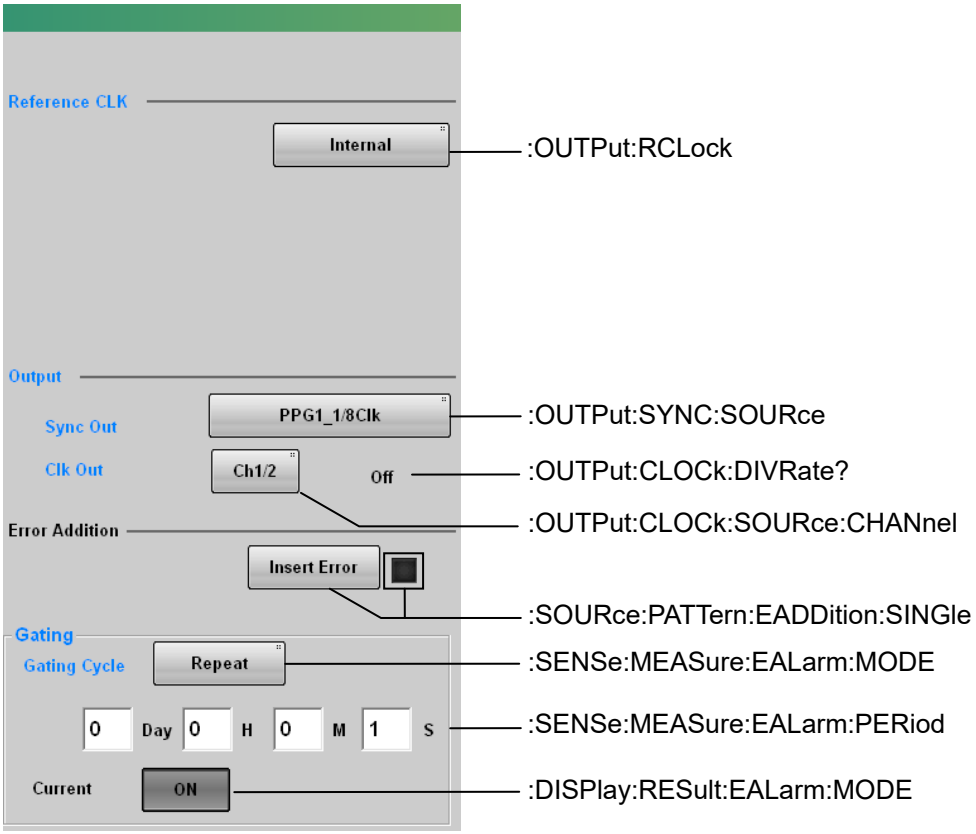


Figure 3.2.2-5 Messages Corresponding to PPG/ED Panel–2 (MP2110A)

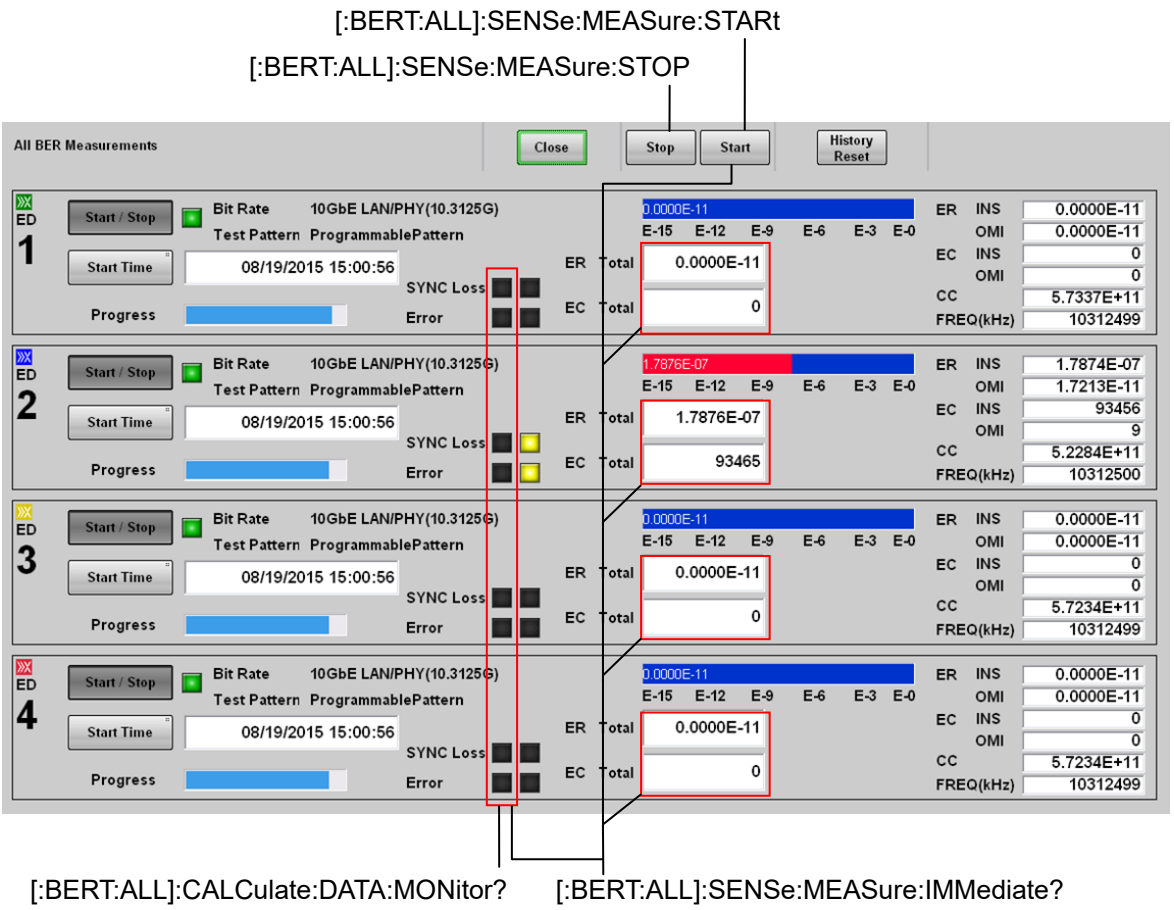


Figure 3.2.2-6 Messages Corresponding to All BER Measurement-1

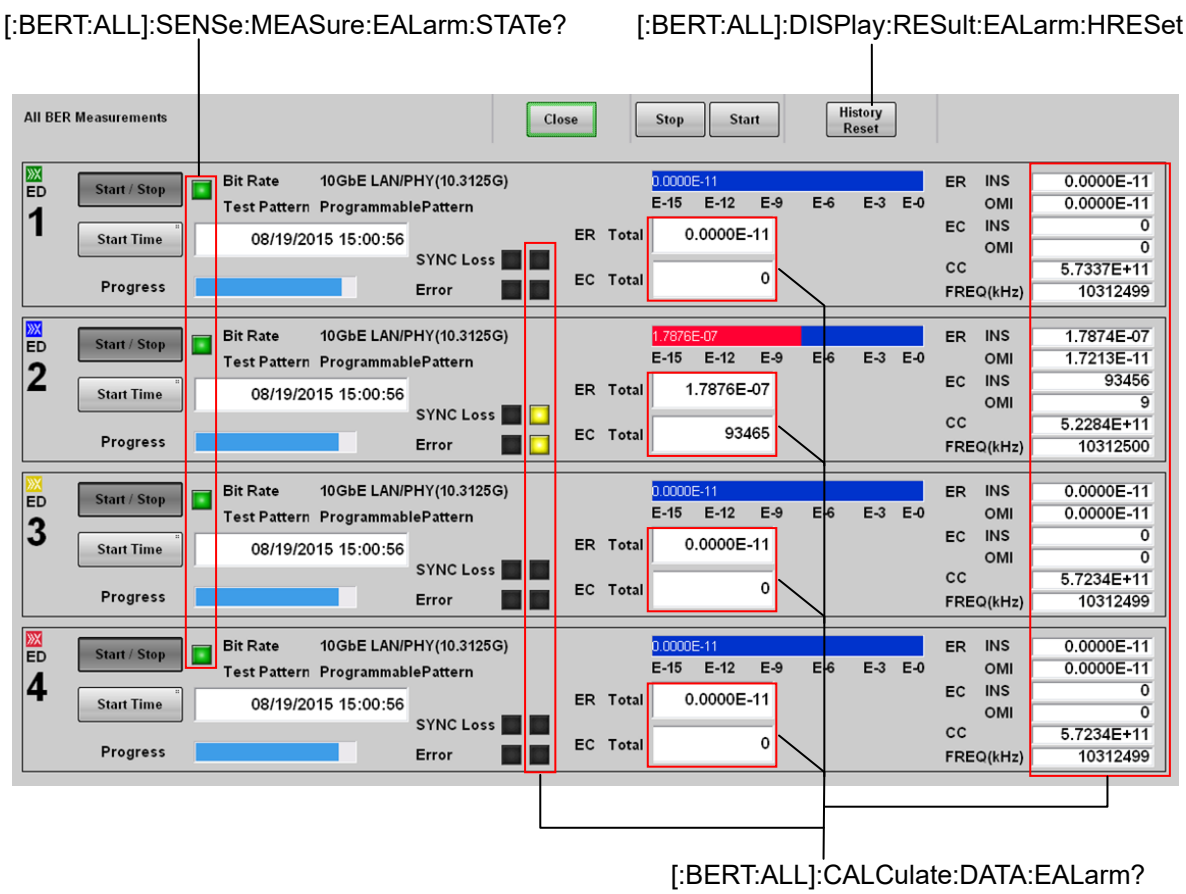


Figure 3.2.2-7 Messages Corresponding to All BER Measurement-2

3.2.3 Messages corresponding to XFP/SFP+ (MP210xA, MP2100B)

When controlling XFP/SFP+, add :PMODule to the beginning of the message header or send :MODule:ID 3 first. Refer to Section 3.5, "Specifying Module and Channel" for details.

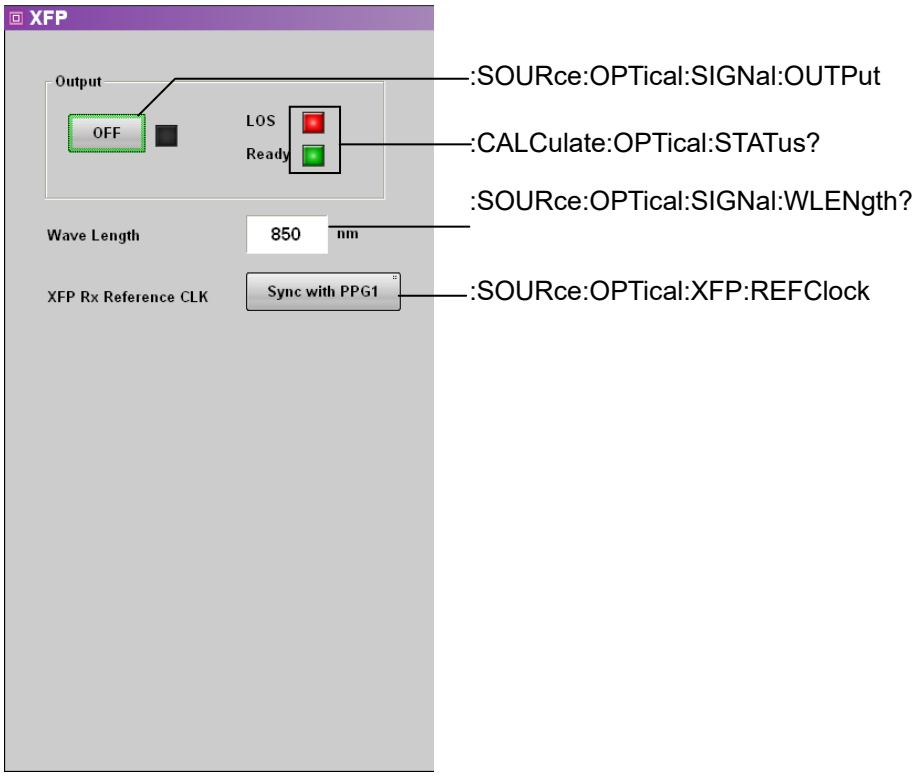


Figure 3.2.3-1 Messages Corresponding to XFP/SFP+ Panel

3.2.4 Messages corresponding to O/E (MP210xA, MP2100B)

When controlling O/E, add :OE to the beginning of the message header or send :MODule:ID 4 first. Refer to Section 3.5, "Specifying Module and Channel" for details.

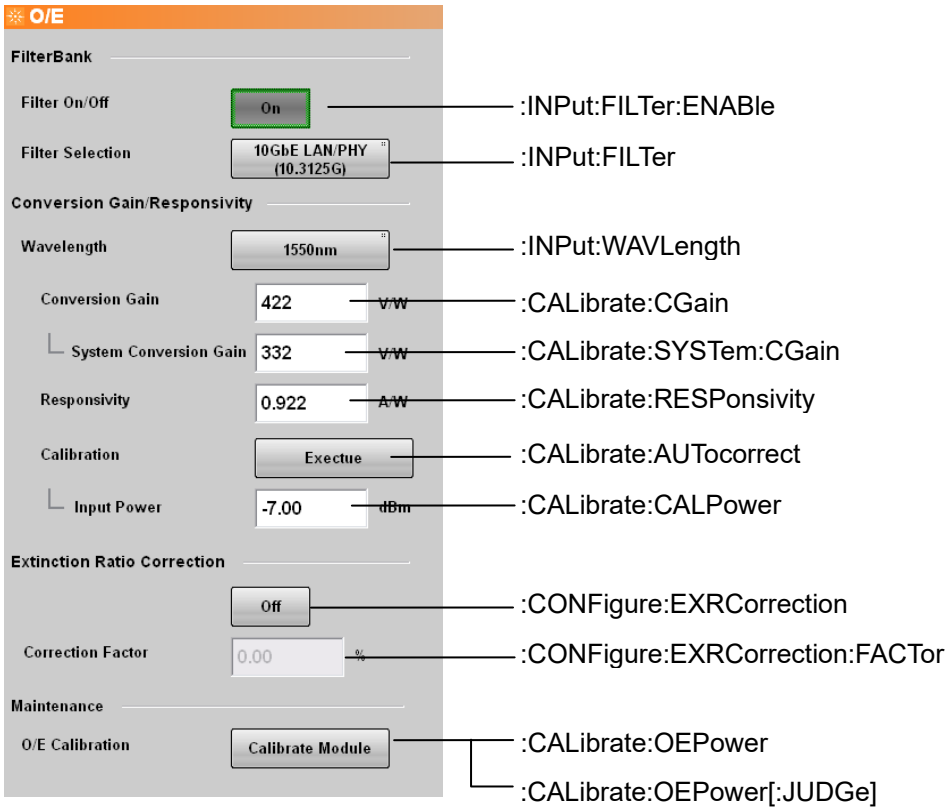


Figure 3.2.4-1 Messages Corresponding to O/E Panel

3.2.5 Messages corresponding to Scope

When controlling Scope, add :SCOPe to the beginning of the message header or send :MODule:ID 5 first. Refer to Section 3.5, "Specifying Module and Channel" for details.

3.2.5.1 Result Window

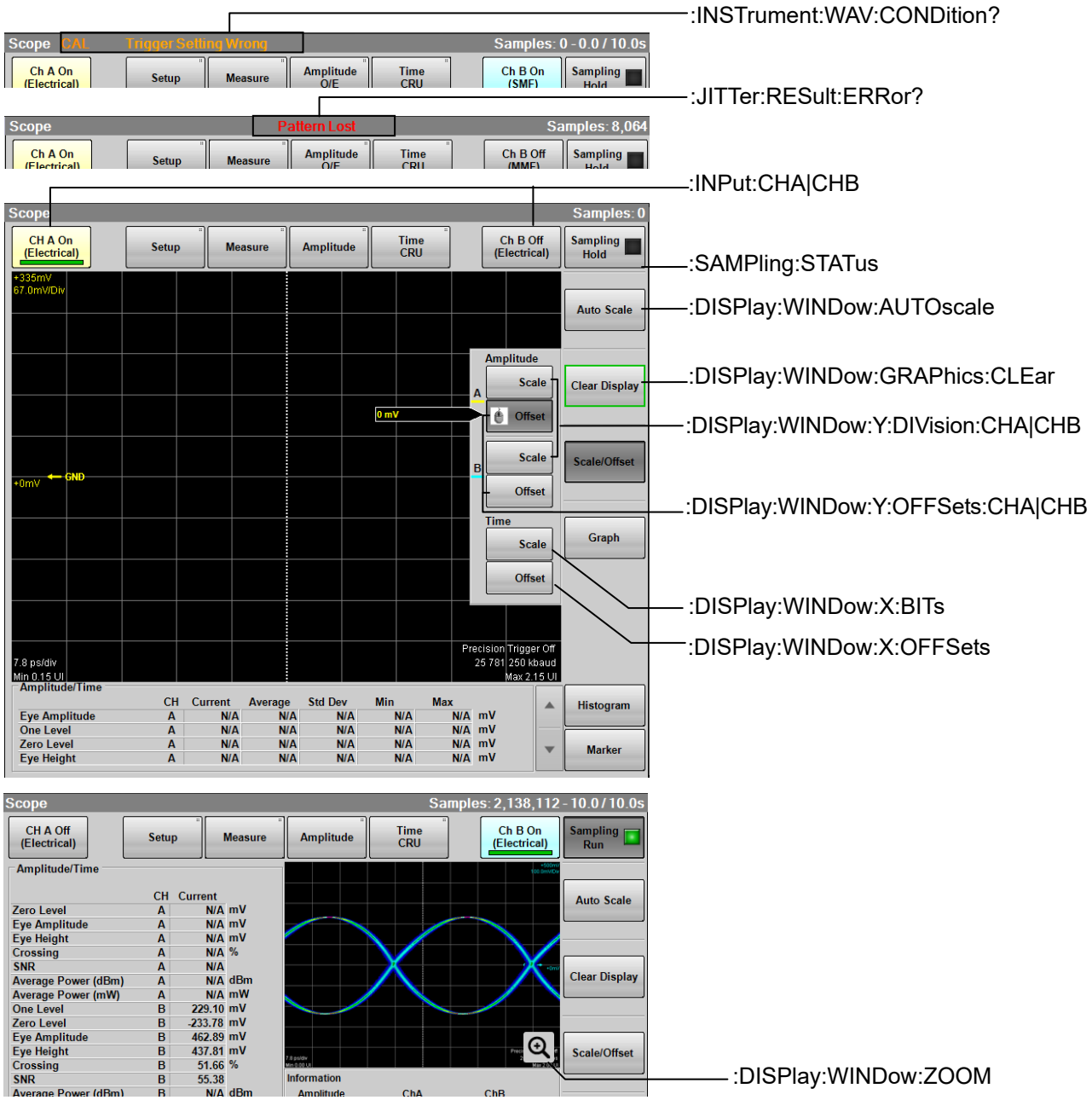


Figure 3.2.5.1-1 Messages Corresponding to Scope Panel (MP2110A)

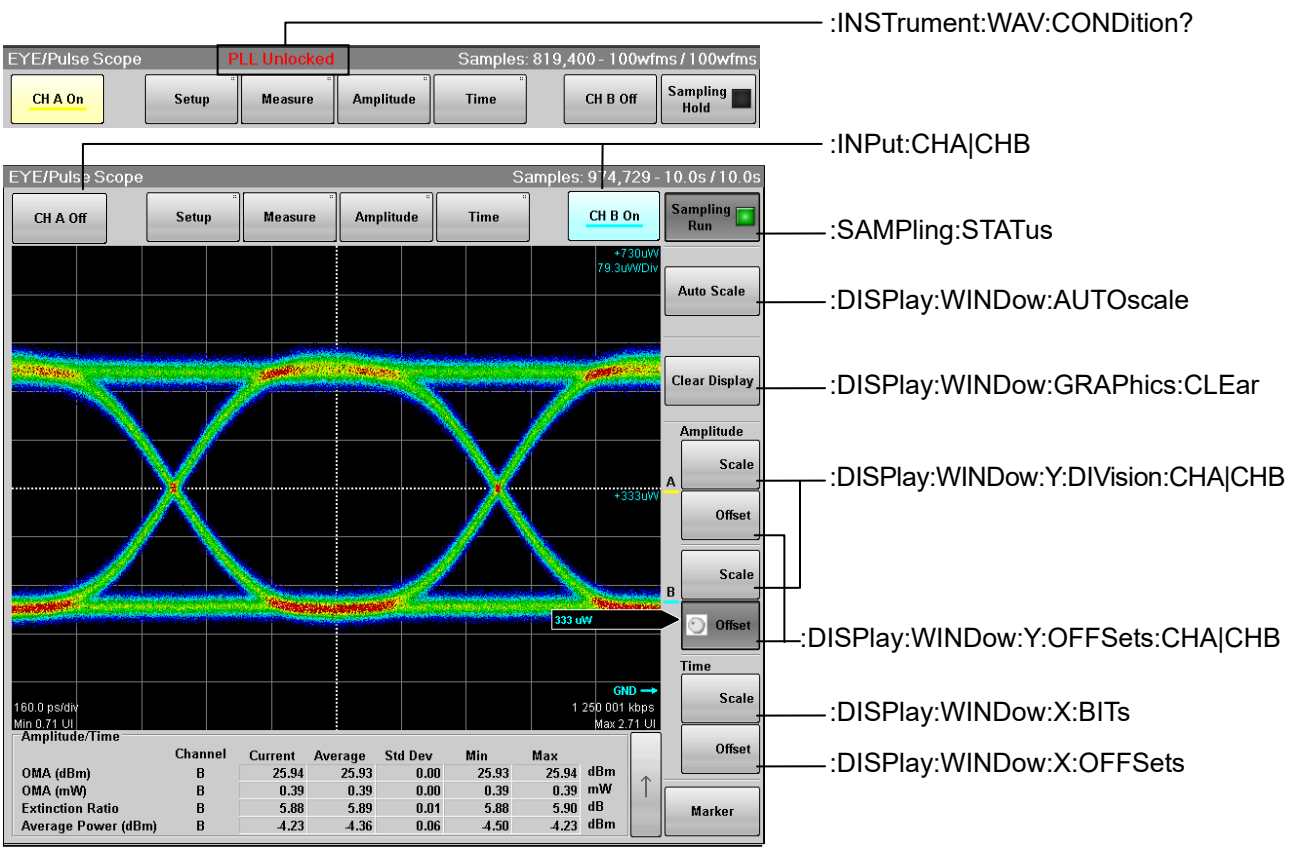


Figure 3.2.5.1-2 Messages Corresponding to Scope Panel (MP210xA, MP2100B)

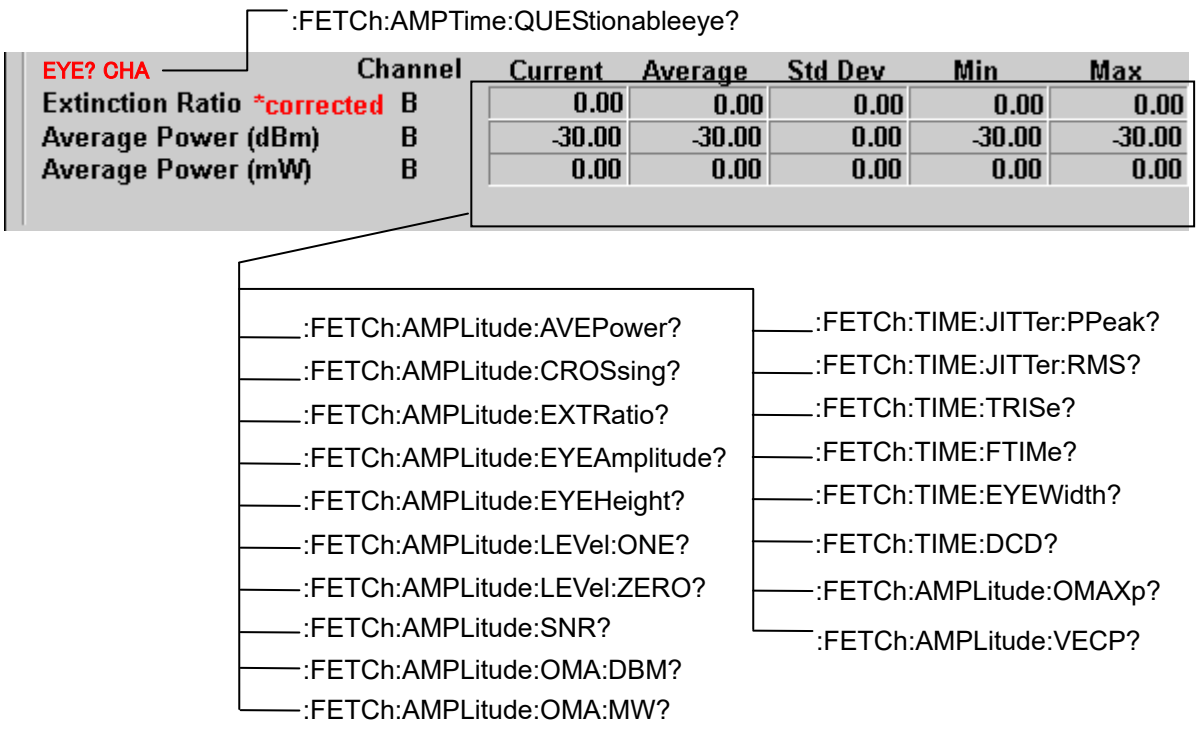


Figure 3.2.5.1-3 Messages Corresponding to Amplitude/Time Measurement Result (NRZ)

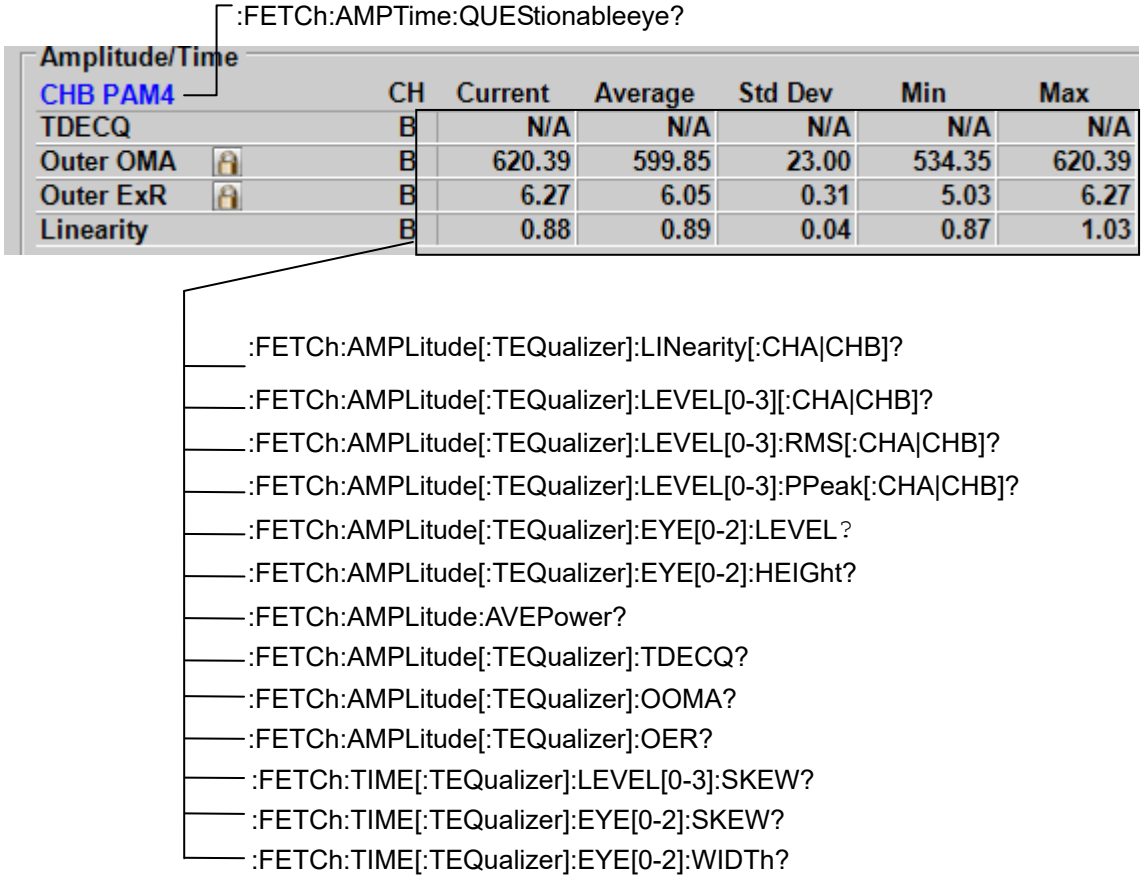


Figure 3.2.5.1-4 **Messages Corresponding to Amplitude/Time Measurement Result (PAM4)**

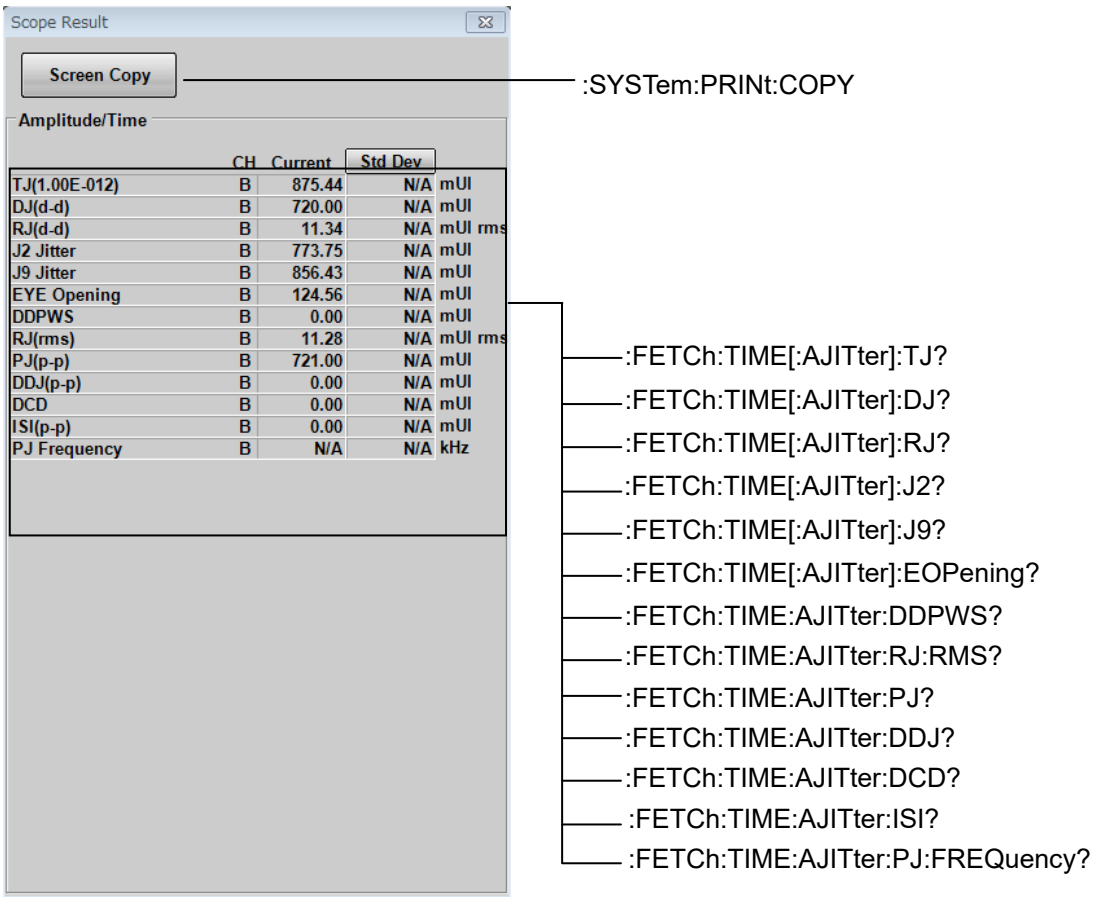


Figure 3.2.5.1-5 **Messages Corresponding to Jitter Measurement Result**

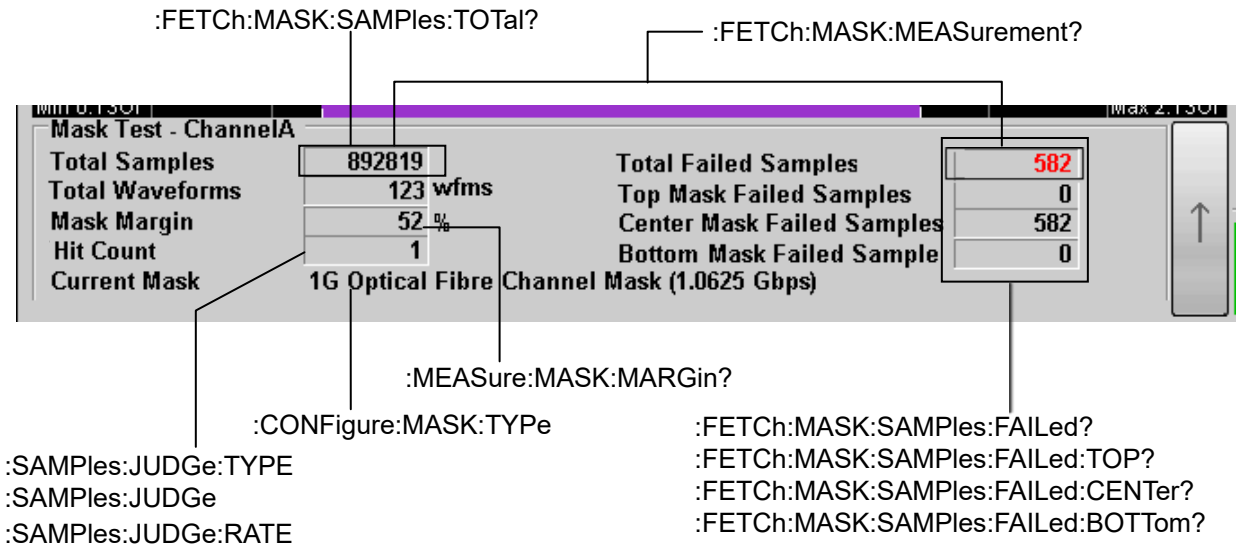


Figure 3.2.5.1-6 Messages Corresponding to Mask Test Measurement Result

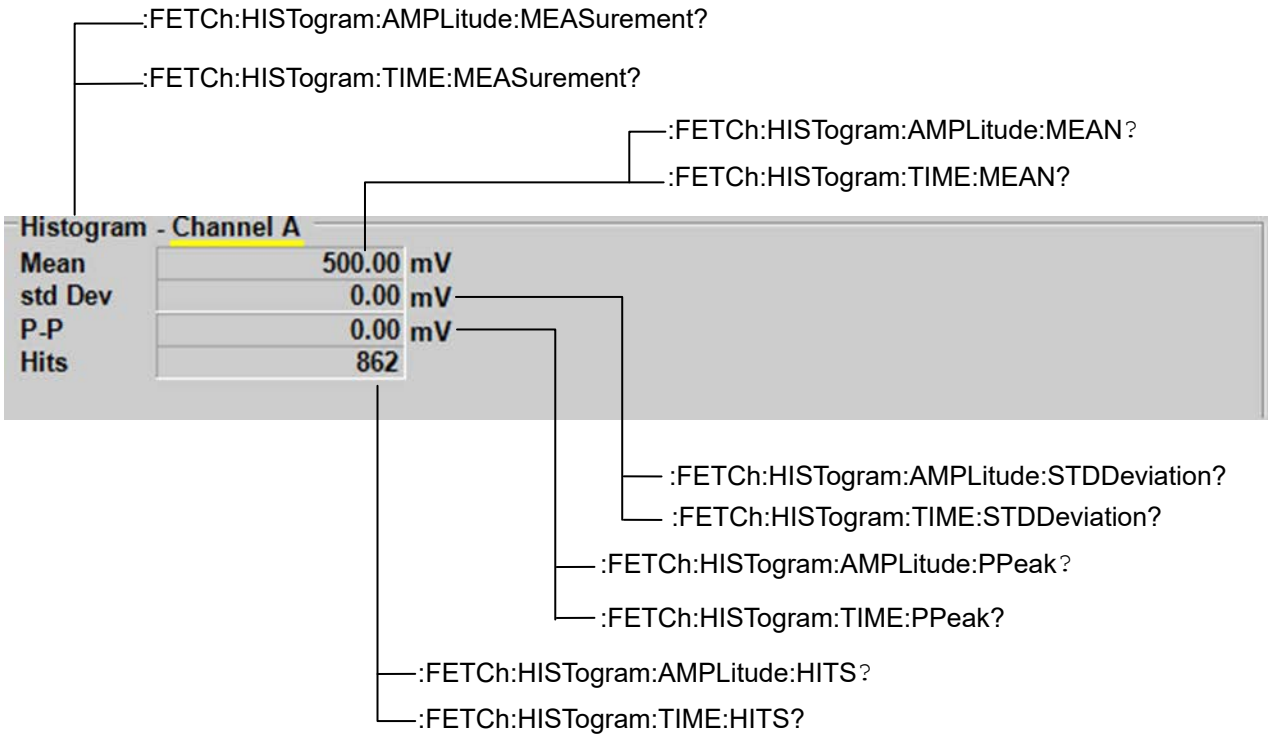


Figure 3.2.5.1-7 Messages Corresponding to Histogram Measurement Result

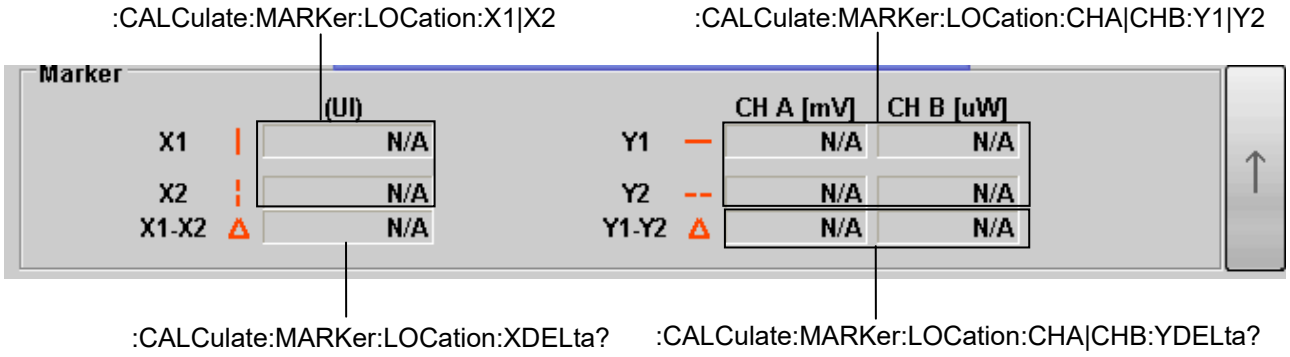


Figure 3.2.5.1-8 Messages Corresponding to Marker Display

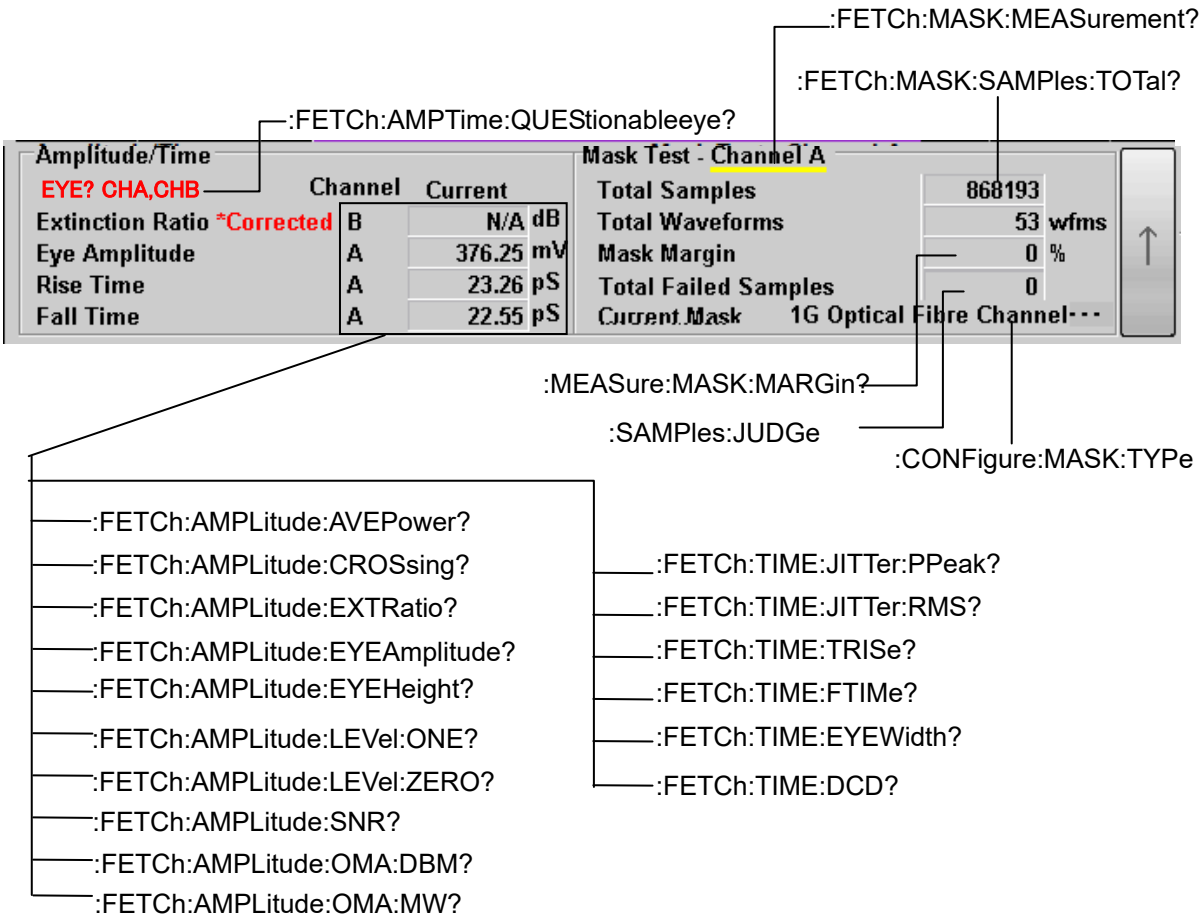


Figure 3.2.5.1-9 Messages Corresponding to Amplitude/Time&Mask Measurement Result

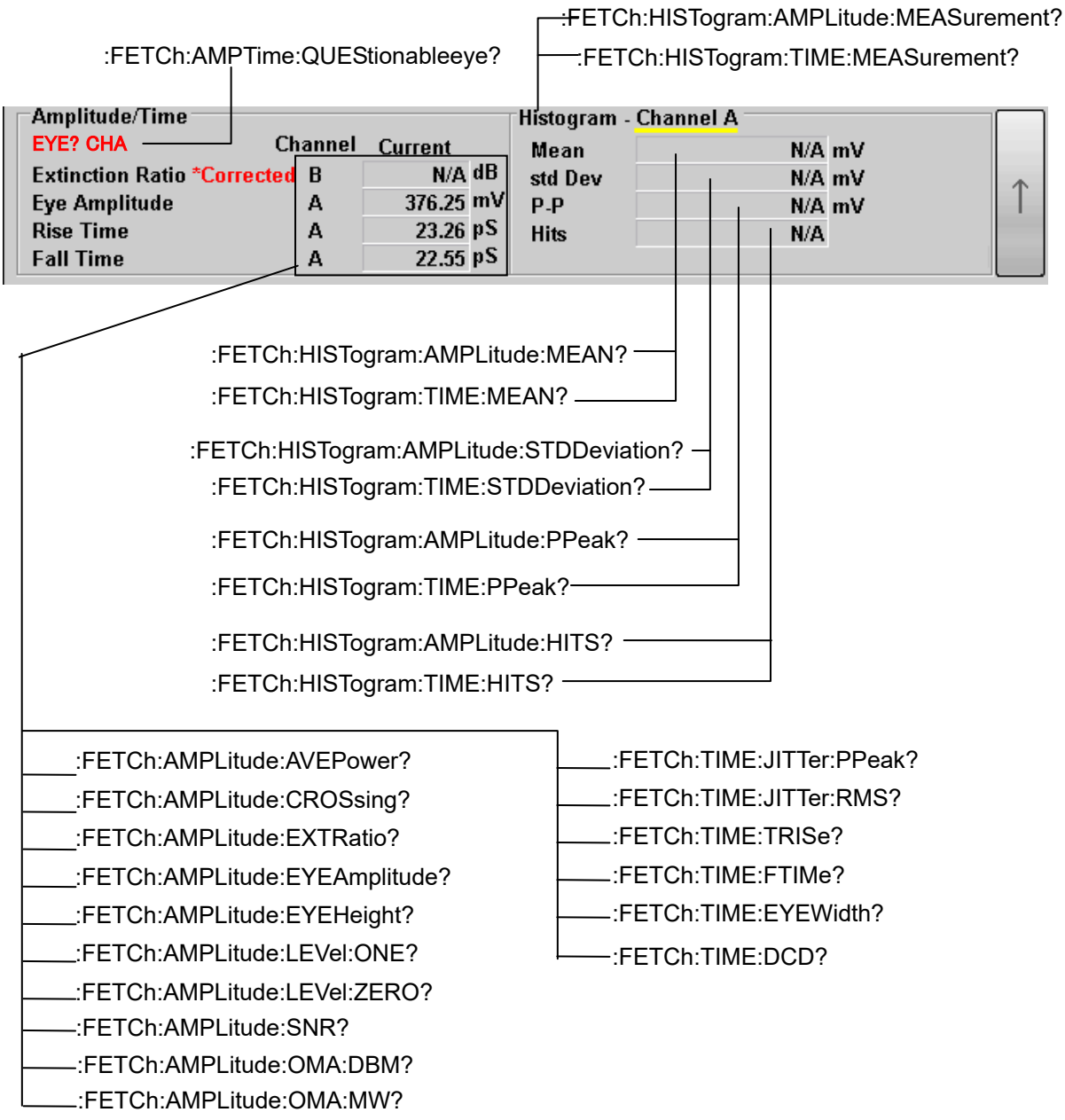
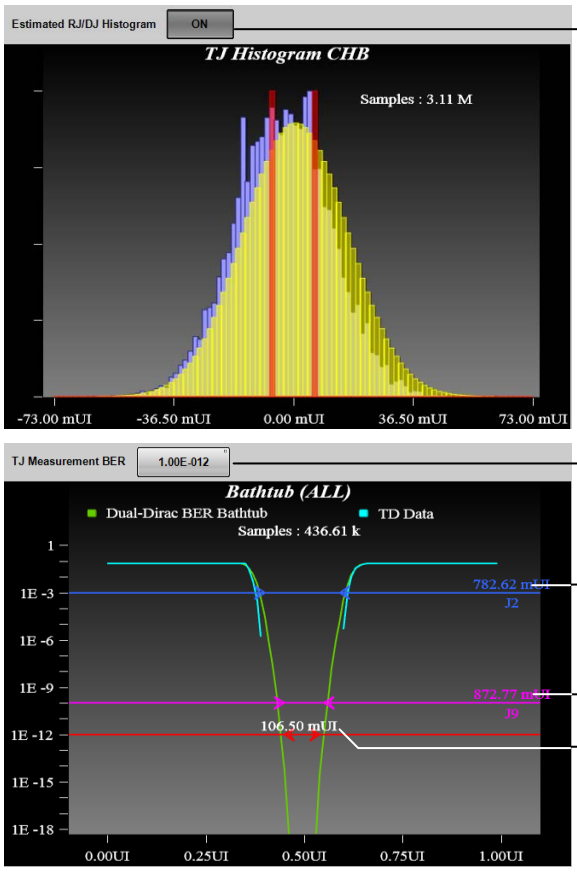


Figure 3.2.5.1-10 Messages Corresponding to Amplitude/Time&Histogram Measurement Result



:JITTer:GRAPh:ESTimate:RJDJ:CHA|CHB

:JITTer:MEASure:TJ:BER[:CHA|CHB]

:FETCh:TIME[:AJITter]:J2?

:FETCh:TIME[:AJITter]:J9?

:FETCh:TIME[:AJITter]:EOPening?

Figure 3.2.5.1-11 Messages Corresponding to Jitter Display 1 (MP2110A – Eye, Advanced Jitter)

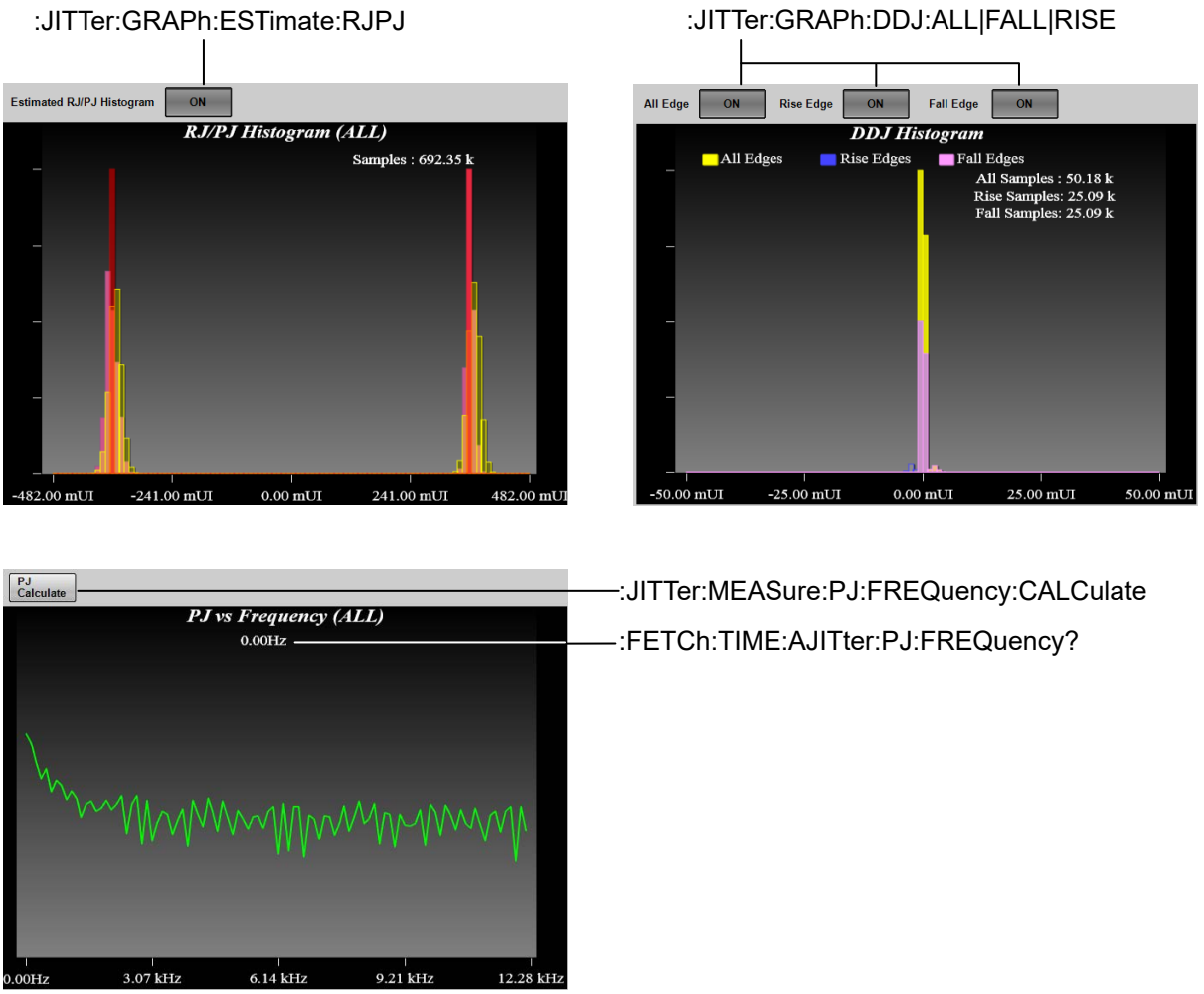


Figure 3.2.5.1-12 Messages Corresponding to Jitter Display 2 (MP2110A – Advanced Jitter)

3.2.5.2 Setup

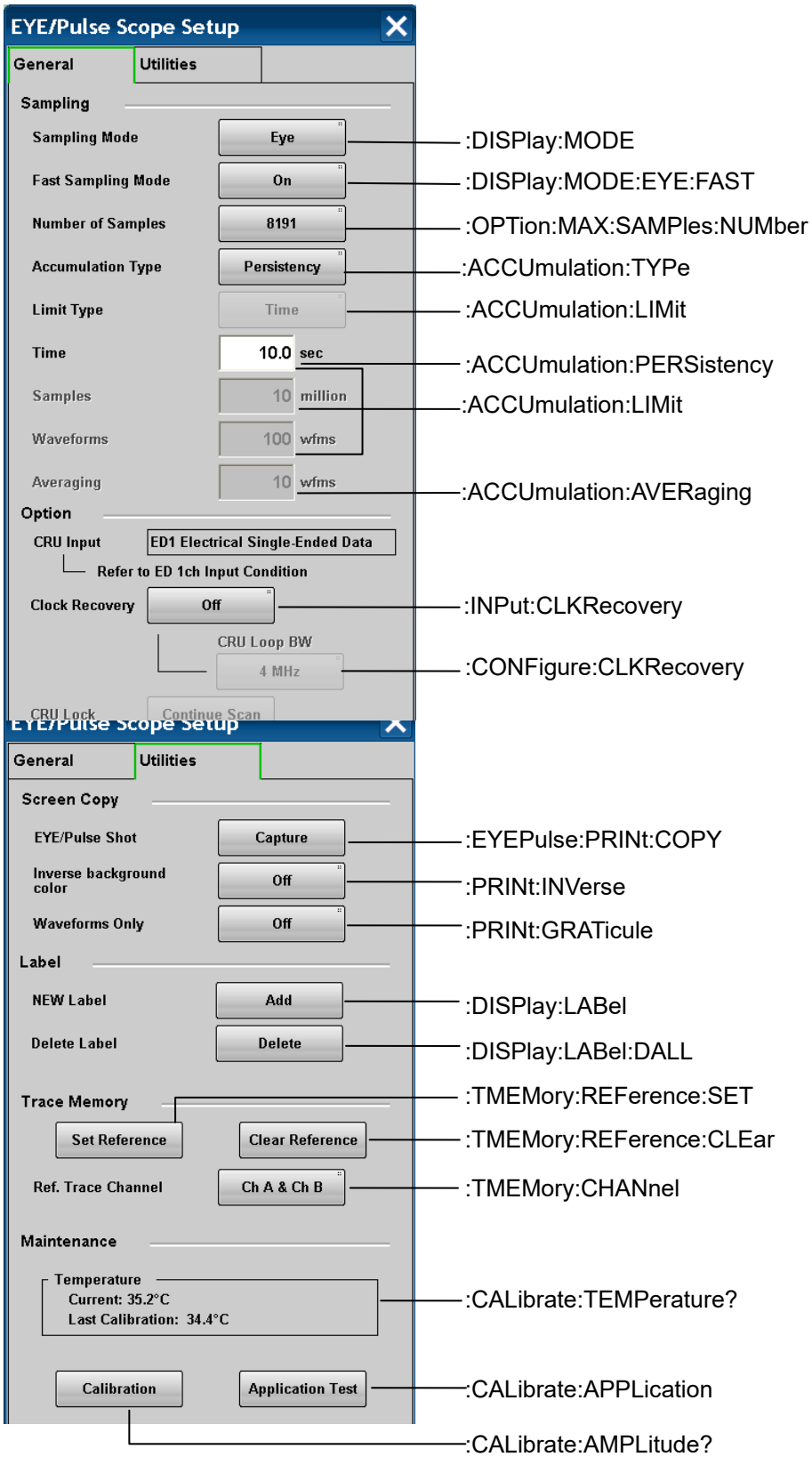


Figure 3.2.5.2-1 Messages Corresponding to Setup Dialog Box (MP210xA, MP2100B)

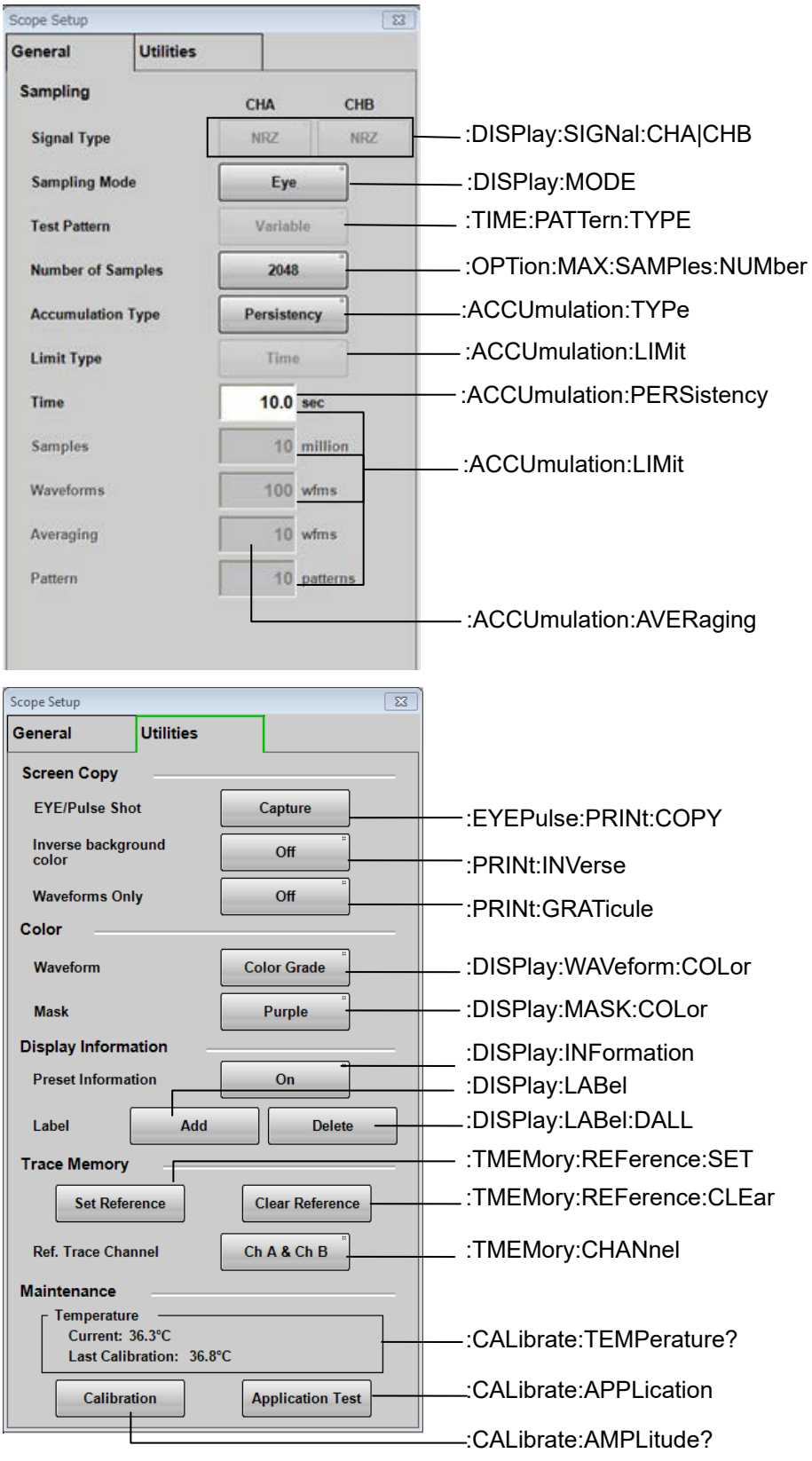


Figure 3.2.5.2-2 Messages Corresponding to Setup Dialog Box (MP2110A)

3.2.5.3 Measure

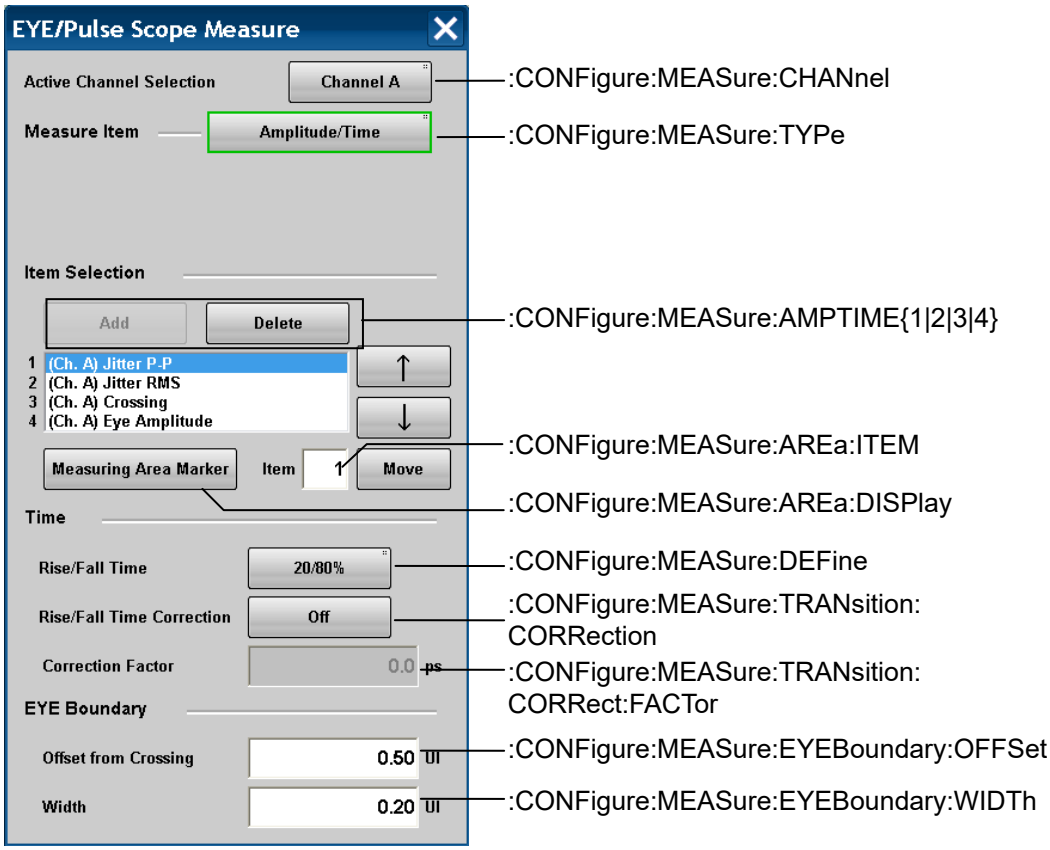


Figure 3.2.5.3-1 Messages Corresponding to Measure Dialog Box (Amplitude/Time,Amplitude/Time&Mask,Amplitude/Time&Histogram) (MP210xA, MP2100B)

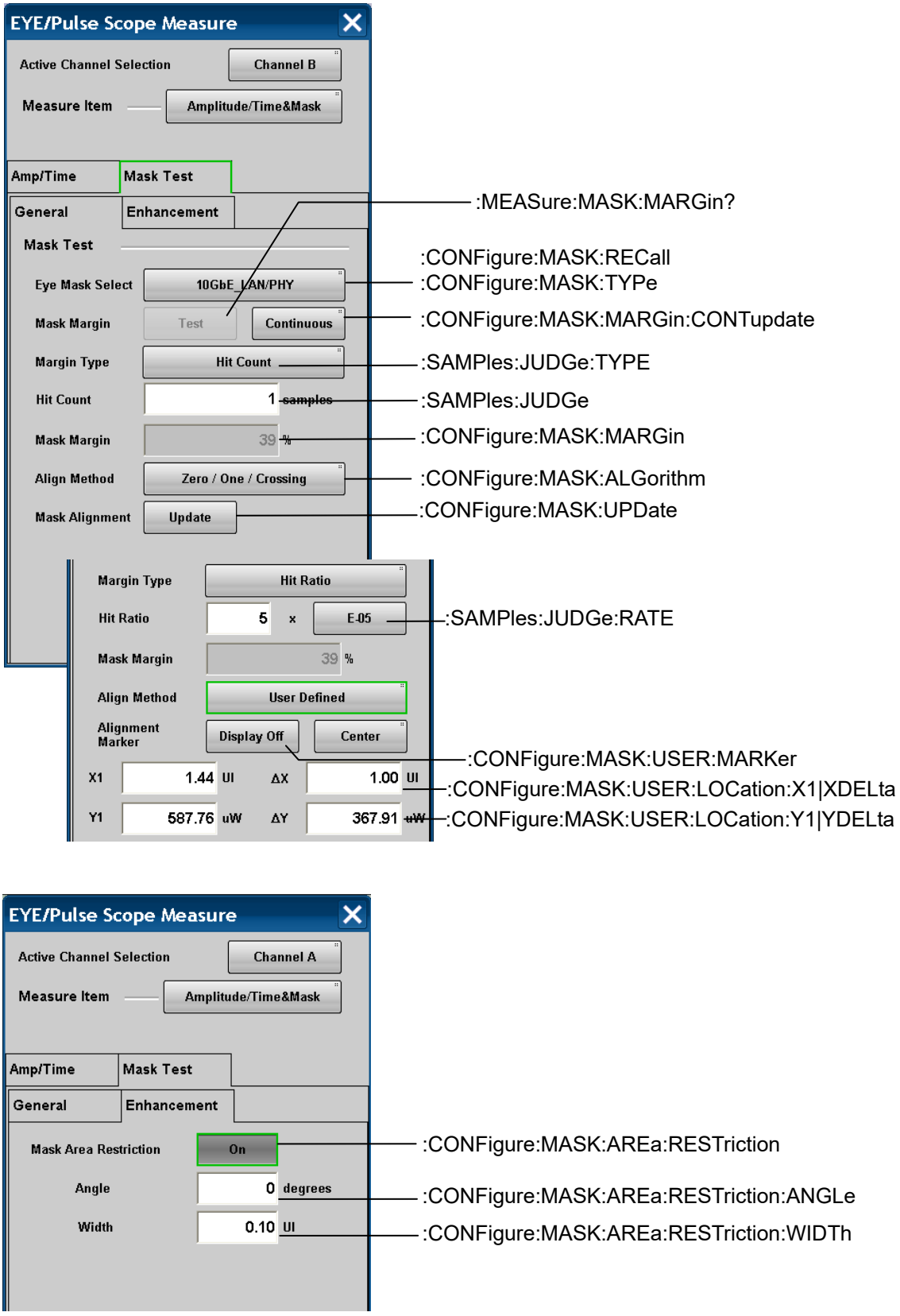


Figure 3.2.5.3-2 Messages Corresponding to Measure Dialog Box (Mask Test, Amplitude/Time&Mask) (MP210xA, MP2100B)

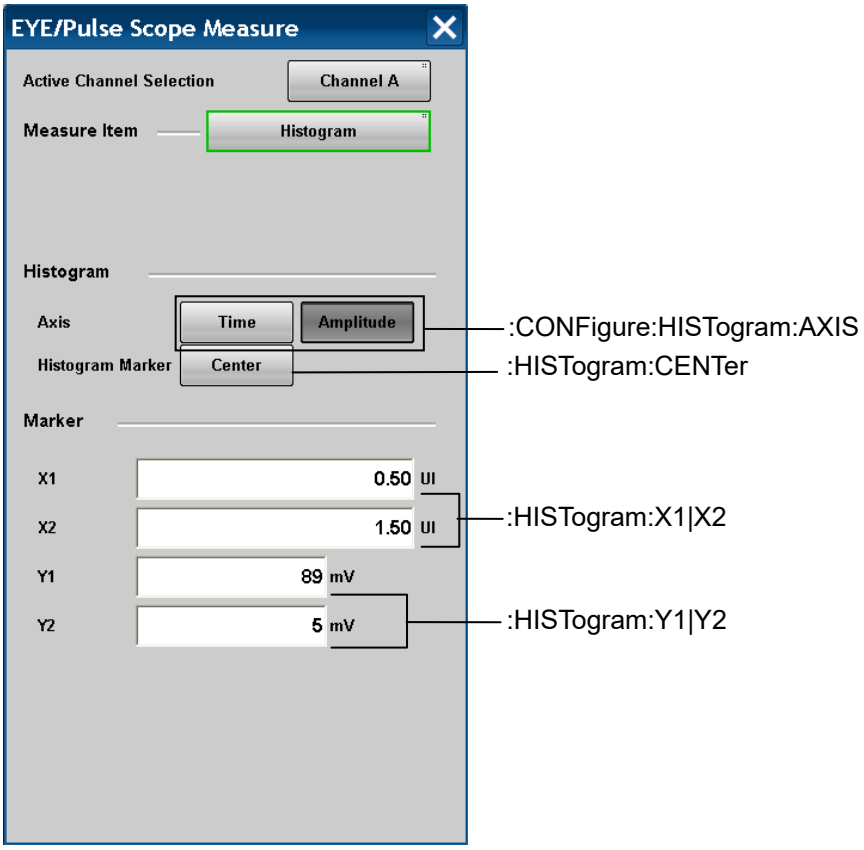


Figure 3.2.5.3-3 Messages Corresponding to Measure Dialog Box (MP210xA, MP2100B) (Histogram, Amplitude/Time&Histogram)

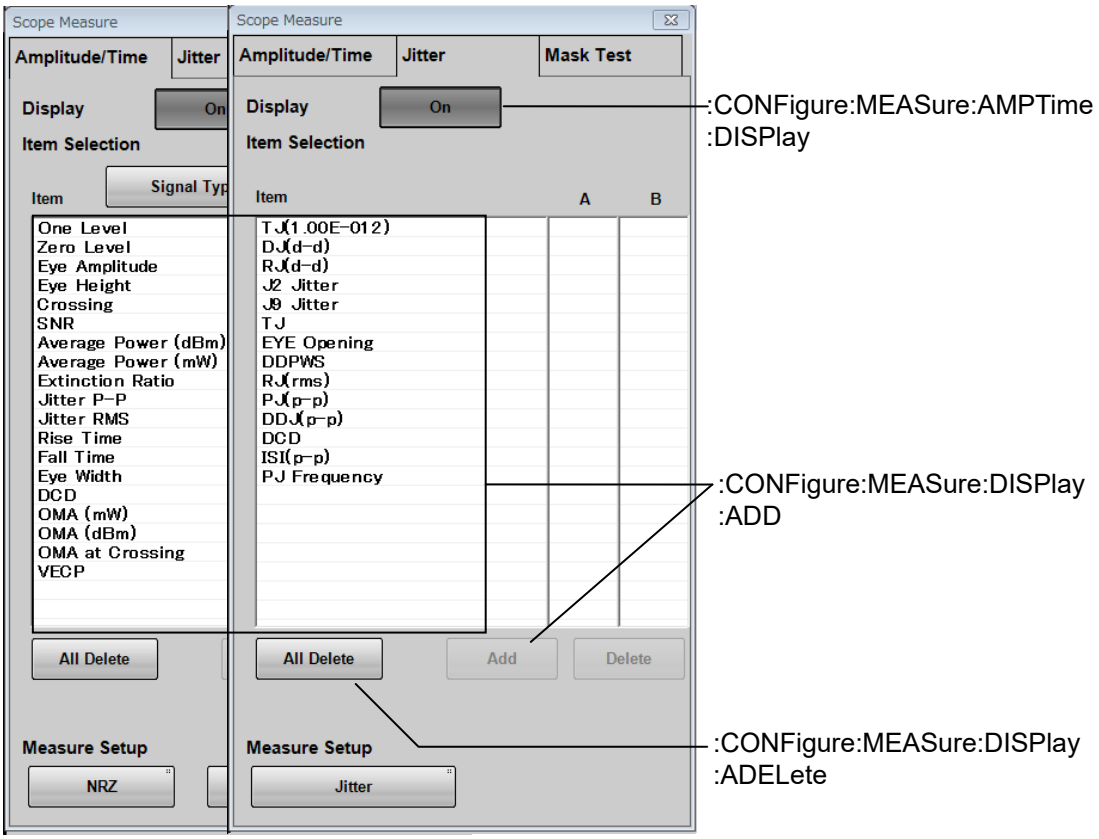


Figure 3.2.5.3-4 Messages Corresponding to Measure Dialog Box (MP2110A) (Amplitude/Time, Jitter)

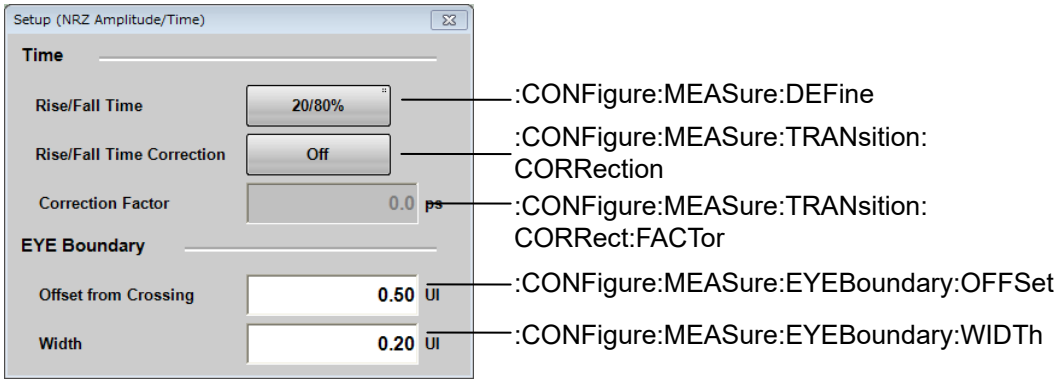


Figure 3.2.5.3-5 Messages Corresponding to Setup (NRZ Amplitude/Time) Dialog Box (MP2110A)

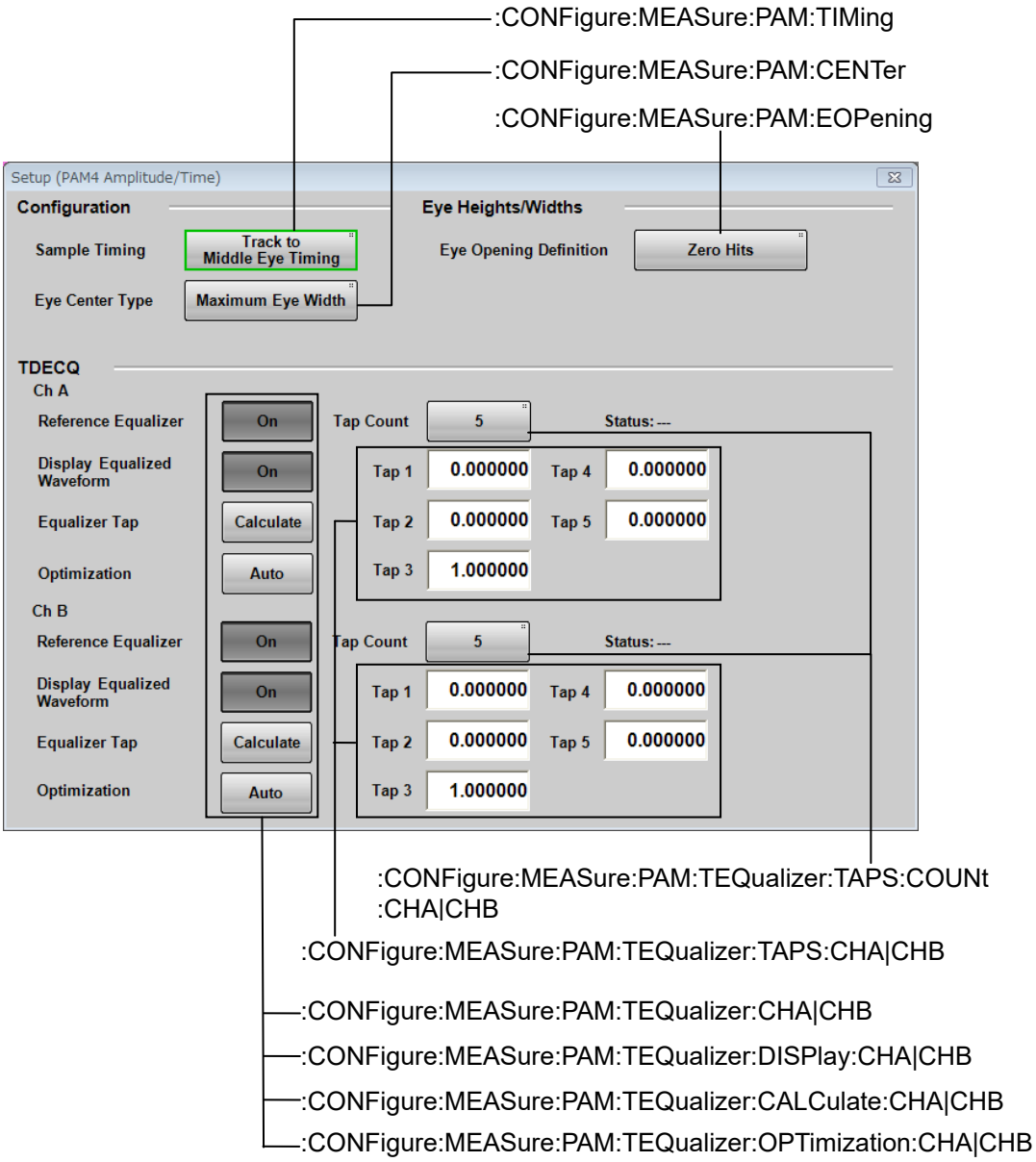


Figure 3.2.5.3-6 Messages Corresponding to Setup (PAM4 Amplitude/Time) Dialog Box (MP2110A)

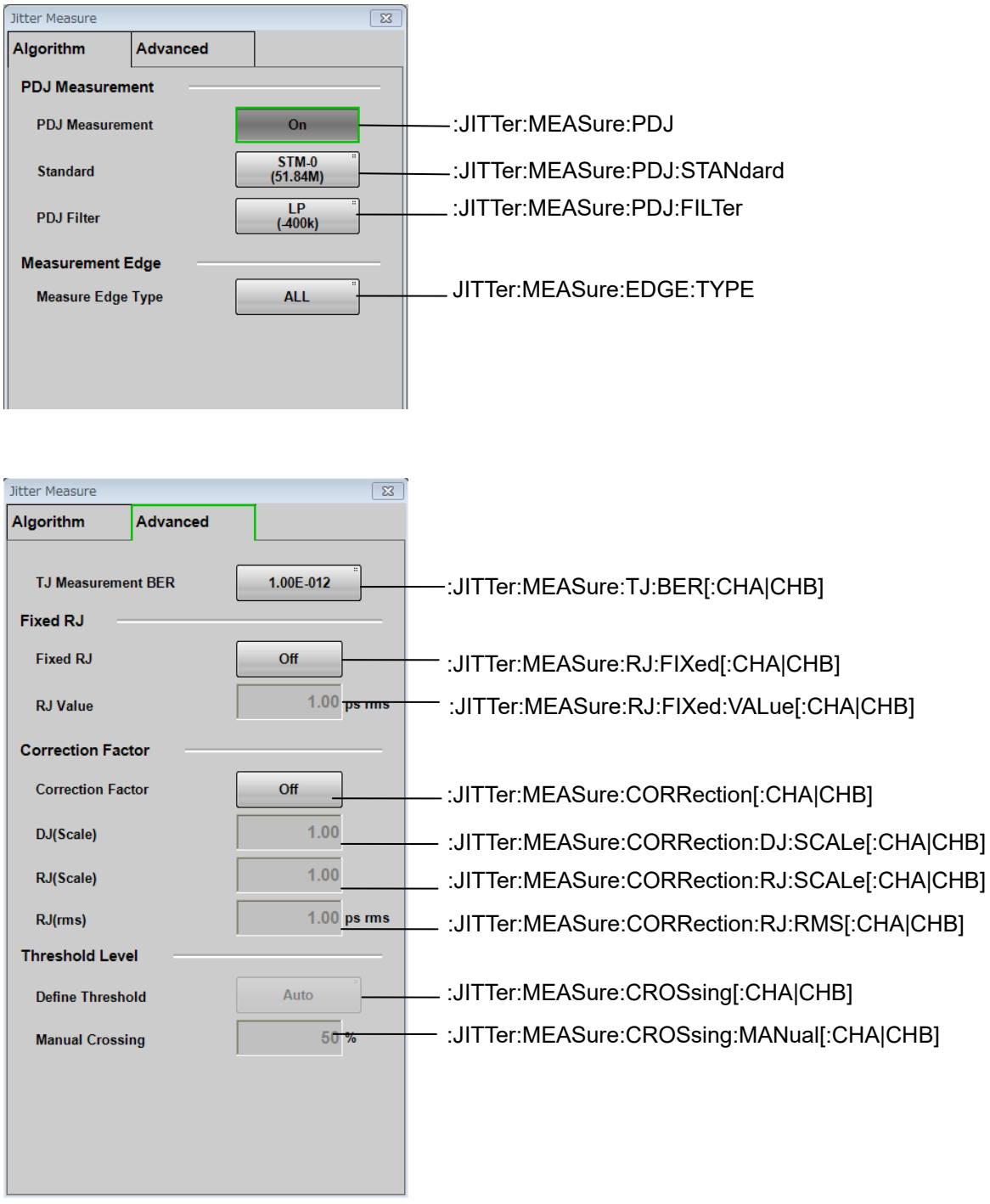


Figure 3.2.5.3-7 Messages Corresponding to Measure Dialog Box (MP2110A) (Jitter)

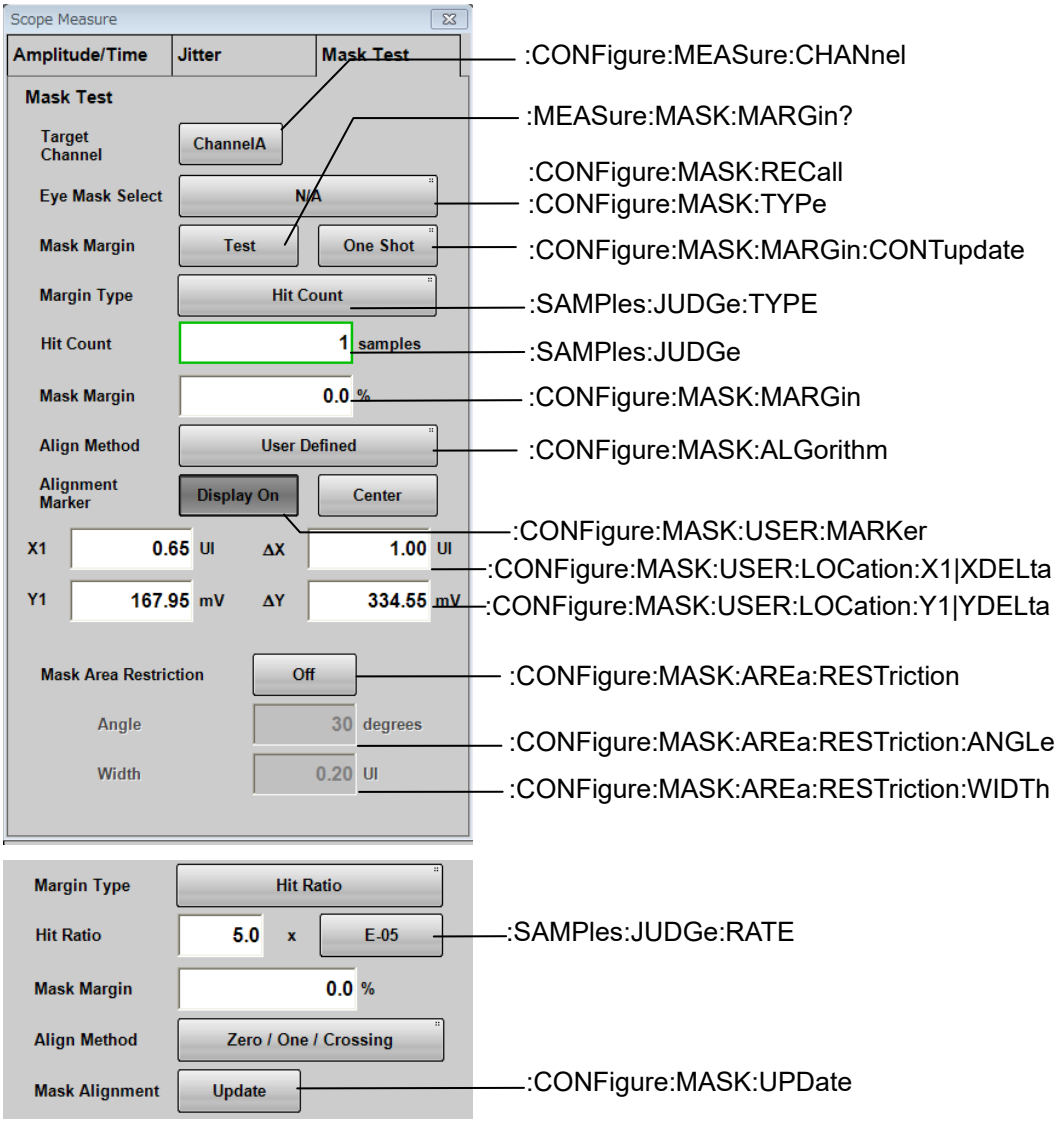


Figure 3.2.5.3-8 Messages Corresponding to Measure Dialog Box (MP2110A) (Mask Test)

3.2.5.4 Time

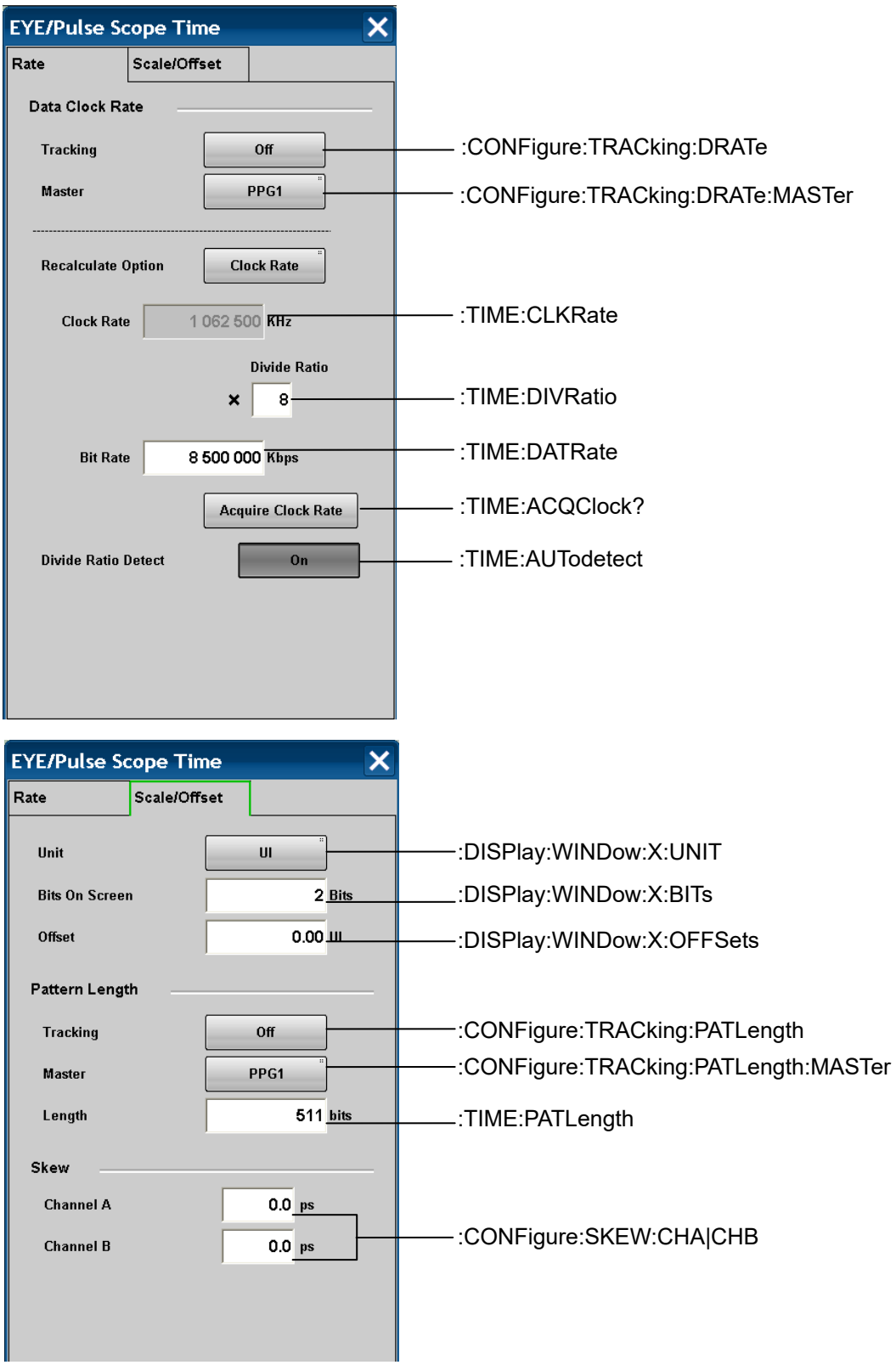


Figure 3.2.5.4-1 Messages Corresponding to Time Dialog Box (MP210xA, MP2100B)

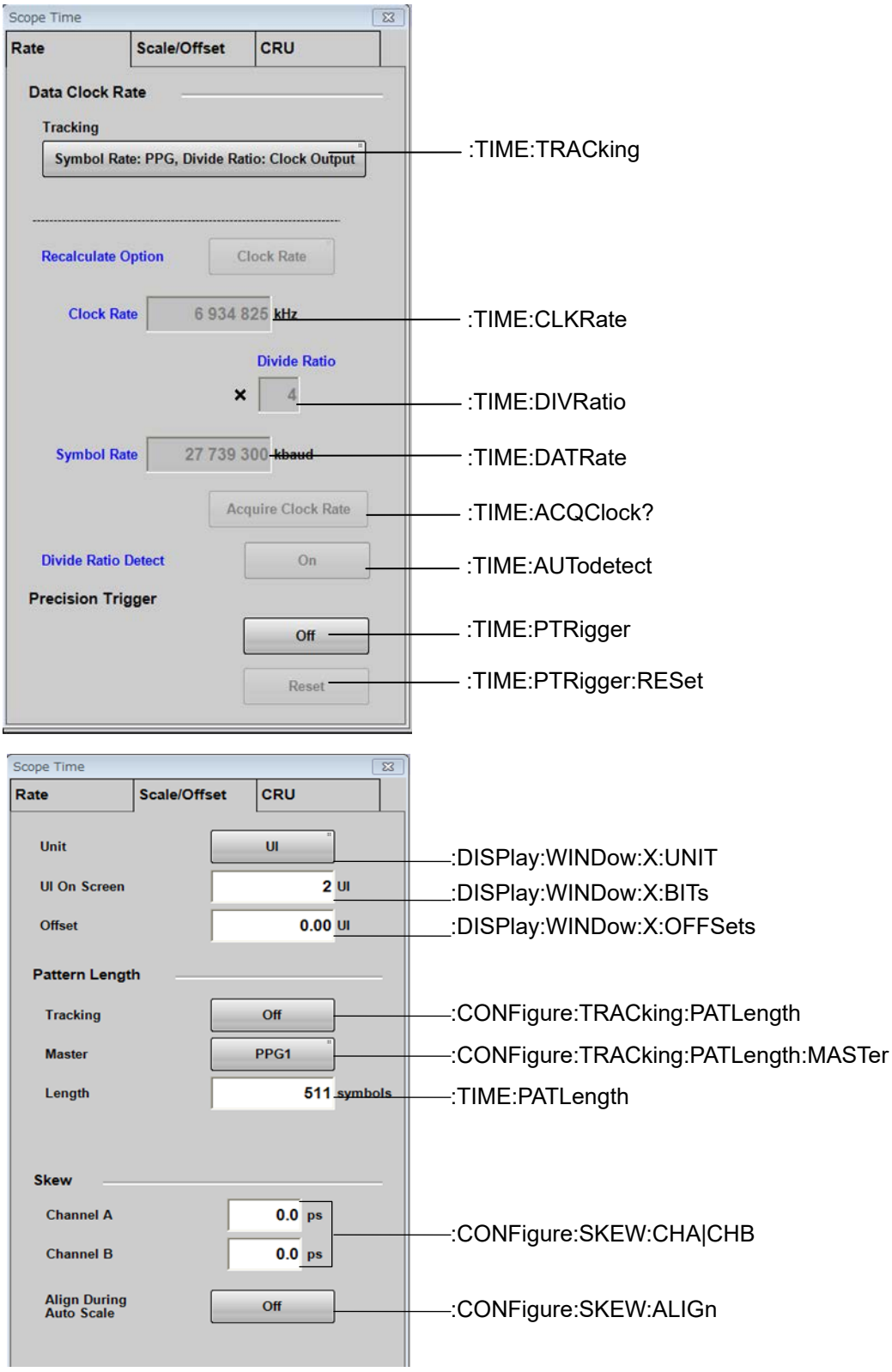


Figure 3.2.5.4-2 Messages Corresponding to Time Dialog Box (MP2110A) (Rate, Scale/Offset)

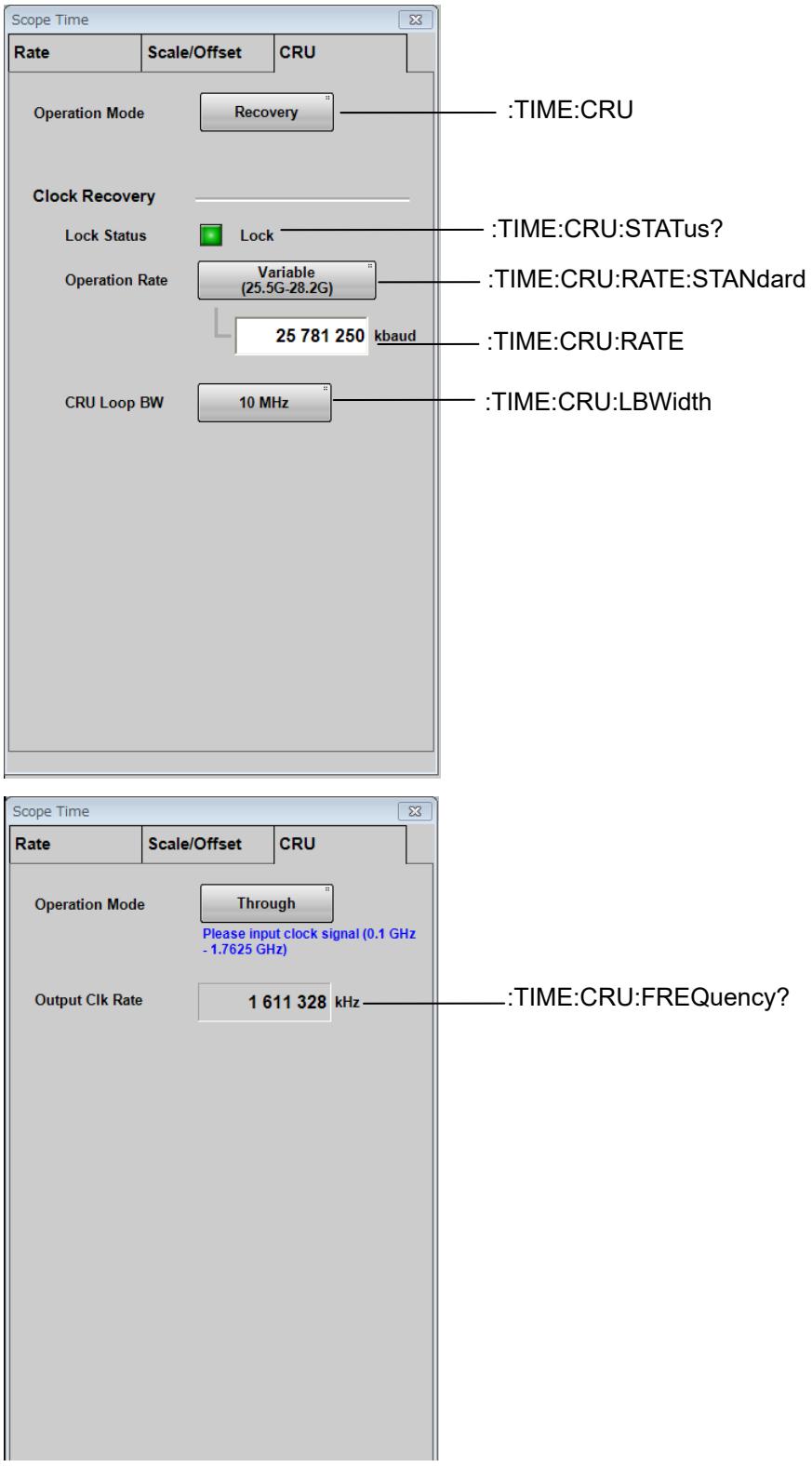


Figure 3.2.5.4-3 Messages Corresponding to Time Dialog Box (MP2110A) (CRU)

3.2.5.5 Amplitude, O/E

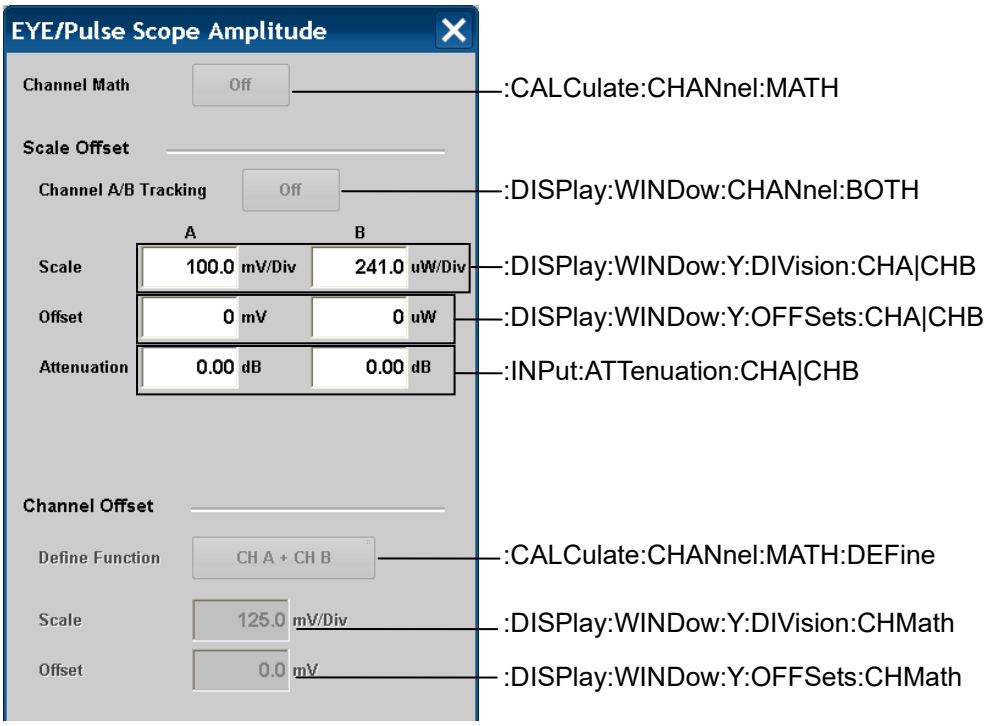


Figure 3.2.5.5-1 Messages Corresponding to Amplitude Dialog Box (MP210xA, MP2100B)

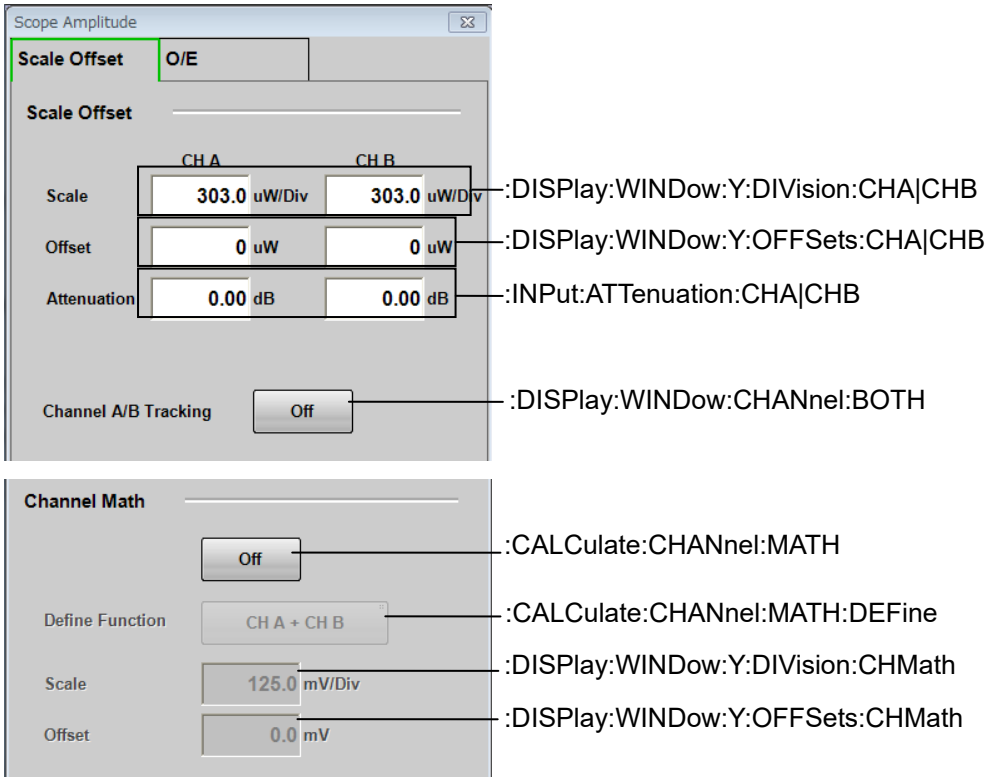


Figure 3.2.5.5-2 Messages Corresponding to Amplitude Dialog Box (MP2110A) (Scale Offset)

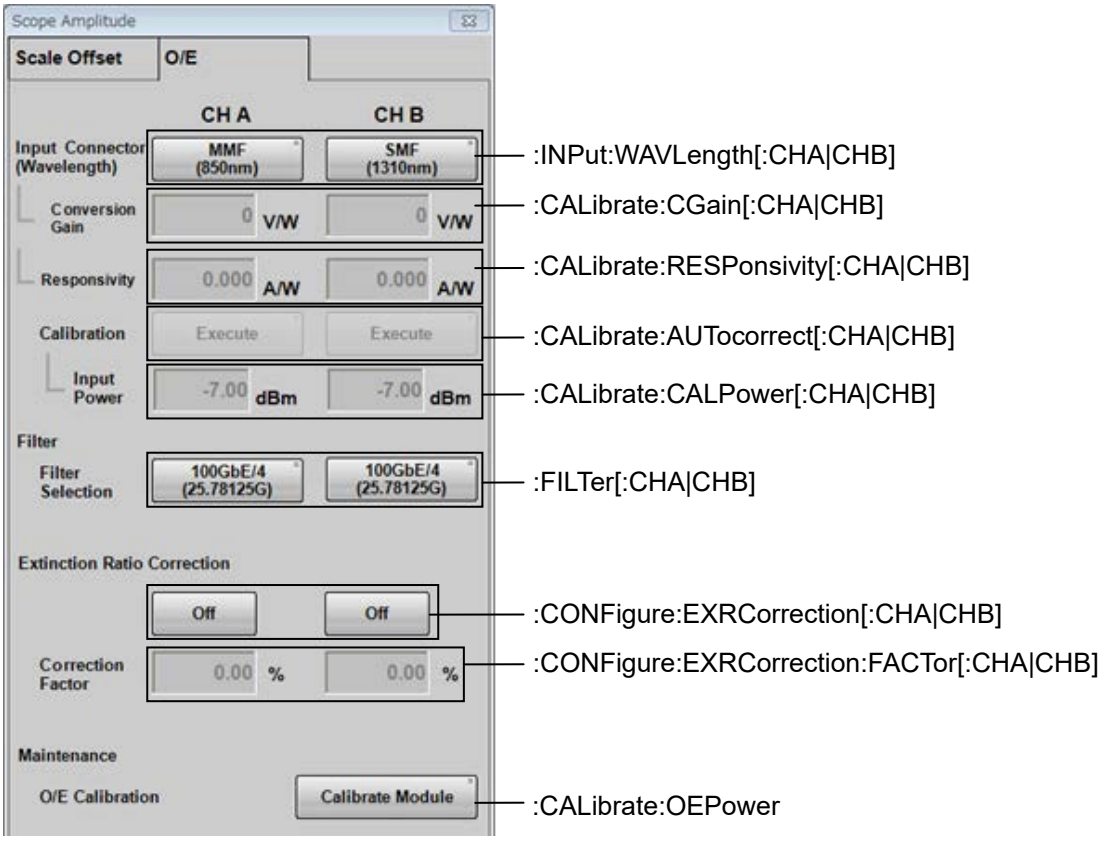


Figure 3.2.5.5-3 Messages Corresponding to Amplitude Dialog Box (MP2110A) (O/E)

3.2.5.6 Marker

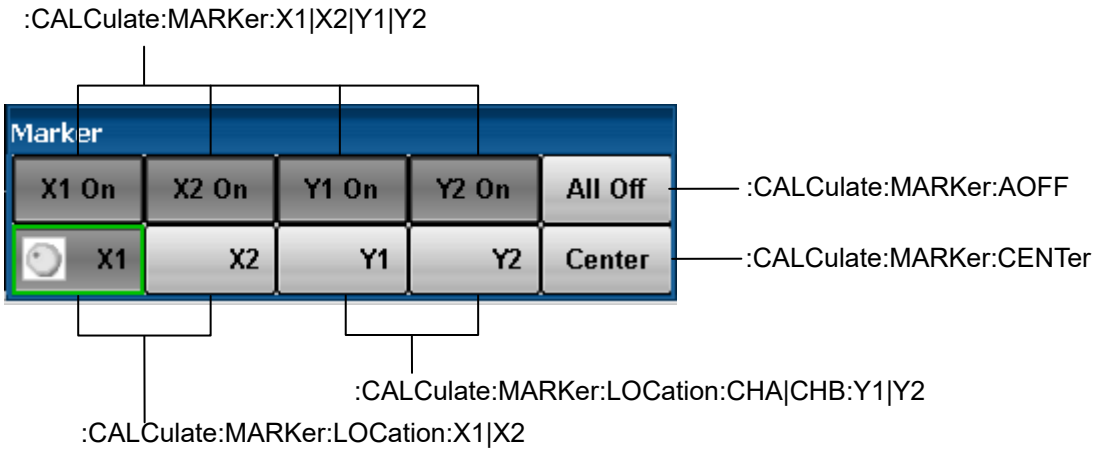


Figure 3.2.5.6-1 Messages Corresponding to Marker Dialog Box

3.2.5.7 Histogram

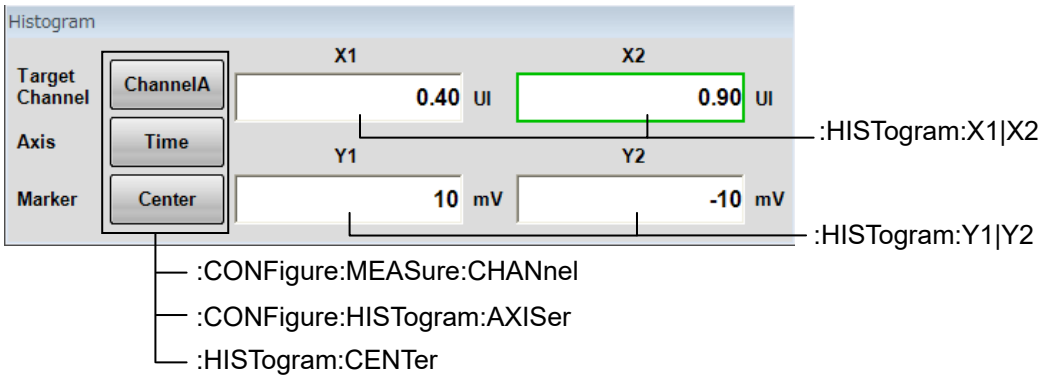


Figure 3.2.5.7-1 Messages Corresponding to Histogram Dialog Box (MP2110A)

3.2.6 Messages corresponding to Information

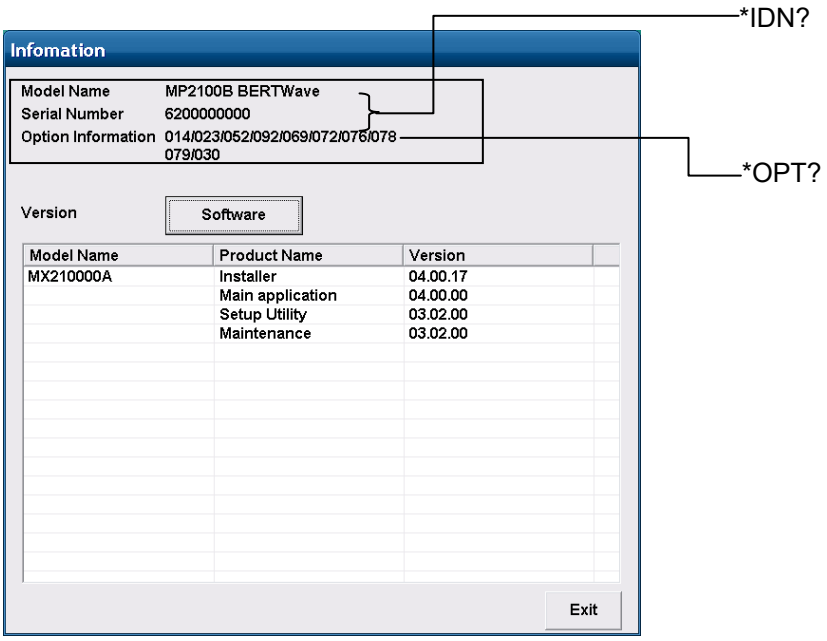


Figure 3.2.6-1 Messages Corresponding to Information Dialog Box from Setup Utility (MP210xA, MP2100B)

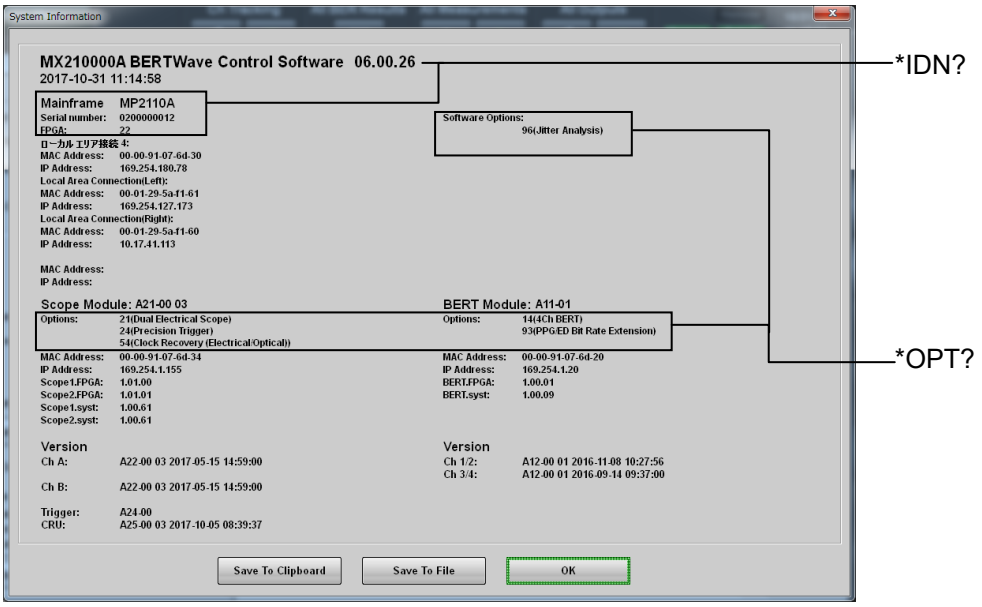


Figure 3.2.6-2 Messages Corresponding to Information Dialog Box from System Menu (MP2110A)

3.2.7 Messages with no corresponding panel operation

Command messages with no corresponding panel operation are listed below.

For messages corresponding to the status registers, refer to 2.8.4 “Operation Status Register” and 2.8.5 “Device Dependent Register”.

Table 3.2.7-1 Messages with No Corresponding Panel Operation (SCPI)

Command	Details
:SYSTem:DISPlay:DATA?	Queries the data of the screenshot image file.
:SYSTem:DISPlay:RESult	Sets and queries the On/Off state of the plotting processing of measurement results.
TRM	Sets and queries the terminator of response data.
GTL	Added in Version 5: Changes the status to Local.
:SYSTem:VERSion?	Queries the version of SCPI.
:TIME:TRACking:STATus?	Queries the tracking status of Scope.
:TRACe:CHANnelA|CHANnelB|CHANnels?	Queries the trace data of Scope.

3.3 Messages Corresponding to Common Operations

3.3.1 Setting system configuration

TRM

Function

This command sets and queries the type of terminator of the response data.

Syntax

```
{TRM|:SYSTem:TERMination} 0|1
{TRM|:SYSTem:TERMination}?
```

Parameter

0 LF+EOI (default)
1 CR+LF+EOI

LF (Line Feed) is 0x0A in ASCII.
CR (Carriage Return) is 0x0D in ASCII.
EOI (End or Identify) is a GPIB interface signal indicating the end of data.

Response Data

0|1

Example of Use

To set the terminator type to LF+EOI:

```
TRM 0
TRM?
>TRM 0
```

GTL (Go to local)

Function

Added in Version 5: This command releases the remote connection status and changes the status to Local.

Example of Use

```
GTL
```

:SYSTem:BEEPer:SET

Function

Only in MP210xA and MP2100B: This command sets and queries the buzzer ON/OFF.

Syntax

```
:SYSTem:BEEPer:SET <enable>
:SYSTem:BEEPer:SET?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

To set buzzer ON:

```
:SYST:BEEP:SET ON
:SYST:BEEP:SET?
>1
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

3.3.2 Obtaining system information

:SYSTem:VERSion?

Function

This command queries the SCPI version the software of the BERTWave is referring to.

Response Data

1999.0

Example of Use

```
:SYST:VERS?
>1999.0
```

*IDN? (Identification)

Function

This command queries product supplier name, model name, serial number, and installer version.

Syntax

```
*IDN?
```

Response Data

Anritsu, MP21{00{A | B} | 10A},<serial_number>,<version>

Example of Use

```
*IDN?
>Anritsu,MP2100A,6200123456,03.01.00
*IDN?
>Anritsu,MP2100B,6200123456,04.00.00
*IDN?
>Anritsu,MP2110A,6200123456,06.00.00
```

*OPT? (Option Identification Query)

Function

This command queries what options are installed.

Syntax

```
*OPT?
```

Response Data

<option_id>[,<option_id>]…

The installed options are returned in the OPT<number> format.

Table 3.3.2-1 option_id List

<option_id>	Option Name
OPT001	Dual Electrical Receiver
OPT003	Optical/Single-ended Electrical Receiver
OPT005	Extended PPG/ED Channel
OPT007	1ch Electrical BERT and Optical/Single-ended Electrical Scope
OPT011	1CH BERT
OPT012	2CH BERT
OPT014	4CH BERT
OPT021	Dual Electrical Scope
OPT022	Dual Optical Scope
OPT023	Optical and Single-ended Electrical Scope
OPT024	Precision Trigger
OPT025	Optical Scope for Singlemode
OPT026	Optical Scope for Multimode
OPT030	GPIB
OPT032	Dual Optical Scope Baseband Flat
OPT033	Optical and Single-ended Electrical Scope Baseband Flat
OPT035	Optical Scope for Singlemode Baseband Flat
OPT036	Optical Scope for Multimode Baseband Flat
OPT050	XFP Slot
OPT051	SFP+ Slot
OPT052	Full Rate Clock Output
OPT053	Clock Recovery (External Data)
OPT054	Clock Recovery (MP2110A: Electrical/Optical, MP2100B: Optical Data)
OPT055	Clock Recovery
OPT056	Low Pass Filter Bank (8.5G/10G/10.7G)
OPT057	Low Pass Filter Bank (2G/4G/8.5G/10G)
OPT058	Low Pass Filter Bank (1.2G/10G/10.7G)
OPT059	Low Pass Filter Bank (1.2G/2.5G/3.1G/6.2G)
OPT060	Low Pass Filter Bank (2G/3.1G/6.2G/10G)
OPT061	1 High Bit Rate Filter
OPT062	2 High Bit Rate Filter Bank
OPT063	4 High Bit Rate Filter Bank
OPT064	1 to 2 Low Bit Rate Filter Bank
OPT065	4 Low Bit Rate Filter Bank
OPT066	1 High Bit Rate and 1 to 2 Low Bit Rate Filter Bank
OPT067	1 to 2 High Bit Rate and 3 to 4 Low Bit Rate Filter Bank
OPT068	2 to 3 High Bit Rate and 1 to 2 Low Bit Rate Filter Bank
OPT069	3 High Bit Rate and 3 Low Bit Rate Filter Bank

Table 3.3.2-1 (Cont'd)

<option_id>	Option Name
OPT070	LPF for 156M (L)
OPT071	LPF for 622M (L)
OPT072	LPF for 1.0G (L)
OPT073	LPF for 1.2G (L)
OPT075	LPF for 2.5G (L)
OPT076	LPF for 2.1G (H)
OPT077	LPF for 2.5G (H)
OPT078	LPF for 2.6G (H)
OPT079	LPF for 3.1G (H)
OPT080	LPF for 4.2G (H)
OPT081	LPF for 5.0G (H)
OPT082	LPF for 6.2G (H)
OPT083	LPF for 8.5G (H)
OPT084	LPF for 9.9G to 10.3G (H)
OPT085	LPF for 10.5G to 11.3G (H)
OPT086	For MP210xA: LPF for Multi 10G (9.9G to 10.7G) (H) For MP2100B: LPF for Multi 10G (8.5G to 11.3G) (H)
OPT087	Filter Bank Set (622M/1.2G/2.5G/4.2G/6.2G/Multi 10G)
OPT088	Filter Bank Set (4.2G/5.0G/6.2G/Multi 10G)
OPT089	Filter Bank Set (156M/622M/1.2G/2.5G)
OPT090	Bit rate Extension for PPG/ED
OPT091	ED High Sensitivity
OPT092	PPG/ED Bit Rate Extension for 125M to 12.5G
OPT093	PPG/ED Bit Rate Extension
OPT095	PAM4 Analysis Software
OPT096	Jitter Analysis Software

Example of Use

```
*OPT?
>OPT001,OPT030,OPT050
```

:SYSTem:INFormation?

Function

This command queries the manufacturer, model name, serial number, and installed option(s) of the BERTWave.

Syntax

```
:SYSTem:INFormation?
```

Response Data

Anritsu,MP21{00{A | B} | 10A},<serial_number>[,<option_id>]…

For <option_id>, refer to "*OPT?".

Example of Use

```
:SYST:INF?
>Anritsu,MP2100A,6200123456,OPT001,OPT050
```

:SYSTem:{DATE|TIME}?

Function

This command queries the date or time of the BERTWave.

Syntax

```
:SYSTem:DATE?
:SYSTem:TIME?
```

Response Data

When querying the date: <year>,<month>,<day>
When querying the time: <hour>,<minute>,<second>

Example of Use

```
:SYST:DATE?
>2009,10,24
:SYST:TIME?
>9,50,39
```

:SYSTem:ERRor?

Function

This command queries the error code and error message.

Syntax

```
:SYSTem:ERRor?
```

Response Data

<integer>,<string>

<integer>
Range –32768 to 32767
0 is returned if no errors have occurred. For error codes returned by the BERTWave, refer to Appendix B “Error Codes”.

<string>
Error message corresponding to the value of <integer> (Up to 255 characters)

Example of Use

```
:SYST:ERR?
>0,"No error"
```

3.3.3 System alarm (MP210xA, MP2100B)

:SYSTem:ERRor:HCLear

Function

This command clears the system alarm history.

Syntax

```
:SYSTem:ERRor:HCLear
```

Example of Use

```
:SYST:ERR:HCL
```

:SYSTem:ERRor:HISTory?

Function

This command queries if there exists a system alarm history.

Syntax

```
:SYSTem:ERRor:HISTory?
```

Response Data

Not Occurred	No system alarms have occurred.
Occurred	A system alarm(s) has occurred.

Example of Use

```
:SYST:ERR:HIST?
>Not occurred
```

:SYSTem:DISPlay:ALARm

Function

This command sets and queries the On/Off state of the Auto Popup for system alarm occurrence.

Syntax

```
:SYSTem:DISPlay:ALARm <enable>
:SYSTem:DISPlay:ALARm?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:SYST:DISP:ALAR ON
:SYST:DISP:ALAR?
>1
```

:SYSTem:INFormation:ERRor?

Function

This command queries the system alarm information.

Syntax

```
:SYSTem:INFormation:ERRor?
```

Response Data

<error_code>[,<error_code>]…
0 is returned if no alarms have occurred.
If multiple alarms have occurred, all error codes are returned in ascending order.

1 PPG/ED Fatal Temperature
2 EYE/Pulse Scope Temperature
3 PPG/ED PLL Unlock (MP210xA)
PPG/ED Hardware Error (MP2100B)
4 Power
5 EYE/Pulse Scope Fatal Temperature
6 PPG/ED Illegal Mode (MP210xA)

Example of Use

```
:SYST:INF:ERR?
>1,2,3
```

3.3.4 Resetting/recalling settings

*RST (Reset)

Function

This command resets the settings to the factory defaults.

Syntax

```
*RST
```

Example of Use

```
*RST
```

Notes:

- Output from PPG/XFP/SFP+ is turned off.
- ED/Scope measurement is stopped; the data obtained so far in the stopped measurement is cleared.
- The time at which the system alarm occurred is cleared.

:SYSTem:MMEMory:RECall

Function

This command recalls the configuration file.

Syntax

```
:SYSTem:MMEMory:RECall <setup_file>
```

Parameter

<setup_file>

Specify the name and type of the configuration file you want to recall.

Table 3.3.4-1 Setting File Format

Contents of File	Format
Settings for the entire BERTWave	"<file_name>[.CND]",0,ALL
Settings for PPG/ED Ch1	"<file_name>.PE1",1,PE1
Settings for PPG/ED Ch2	"<file_name>.PE2",2,PE2
Settings for PPG/ED Ch3	"<file_name>.PE3",8,PE3
Settings for PPG/ED Ch4	"<file_name>.PE4",9,PE4
Settings for XFP	"<file_name>.XFP",3,XFP
Settings for SFP+	"<file_name>.SFP",3,SFP
Settings for O/E	"<file_name>.OES",4,OES
Settings for Scope	"<file_name>.WFS",5,WFS
Settings for Jitter	"<file_name>.JIT",6,JIT
Settings for Transmission	"<file_name>.TAS",7,TAS

Example of Use

```
:SYST:MMEM:REC "settings_all",0,ALL
:SYST:MMEM:REC "settings_ppged1.PE1",1,PE1
```

3.3.5 Storing settings and results

:SYSTem:MMEMory:STORe

Function

This command stores the settings or measurement results to a file.

Syntax

```
:SYSTem:MMEMory:STORe <setup_file>|<results_file>
```

Parameter

Specify the name and type of a file to which you want to store setting/results, as follows:

<setup_file>

Refer to :SYSTem:MMEMory:RECall

<results_file>

Table 3.3.5-1 File Format

Result	Format*1,*2
EDCh1 measurement	"<file_name>.{CSV|TXT}",1,ER1,{CSV|TXT}
EDCh2 measurement	"<file_name>.{CSV|TXT}",2,ER2,{CSV|TXT}
EDCh3 measurement	"<file_name>.{CSV|TXT}",8,ER3,{CSV|TXT}
EDCh4 measurement	"<file_name>.{CSV|TXT}",9,ER4,{CSV|TXT}
O/E measurement	"<file_name>.{CSV|TXT}",4,OE,{CSV|TXT}
Scope measurement	"<file_name>.{CSV|TXT}",5,WFR,{CSV|TXT}
Jitter measurement	"<file_name>.{CSV|TXT}",6,JIR,{CSV|TXT}
Transmission measurement (TransmissionAnalysis)	"<file_name>.{S2P|TXT}",7,TAR,{S2P|TXT}
Transmission measurement (WaveformEstimation)	"<file_name>.WFE",7,WER,WFE

*1: CSV, TXT, S2P and WFE indicate file formats.

*2: If CSV|TXT is omitted when storing Scope measurement results, the file is saved in binary format.

Example of Use

To store settings:

```
:SYST:MMEM:STOR "settings_all",0,ALL
:SYST:MMEM:STOR "settings_ppged1.PE1",1,PE1
:SYST:MMEM:STOR "settings_jitter.JIT",6,JIT
```

To store measurement results:

```
:SYST:MMEM:STOR "results_ed1.TXT",1,ER1,TXT
:SYST:MMEM:STOR "results_scope.CSV",5,WFR,CSV
:SYST:MMEM:STOR
"results_transmission_analysis.S2P",7,TAR,S2P
:SYST:MMEM:STOR
"results_waveform_estimation.WFE",7,WER,WFE
```

Note:

If the file extension of the saved configuration file is changed, it cannot be recalled.

3.3.6 Turning on/off plotting processing

:SYSTem:DISPlay:RESult

Function

This command sets and queries the On/Off state of the plotting processing of measurement results.

Syntax

```
:SYSTem:DISPlay:RESult {{0|OFF}[,ED]|{1|ON}}
:SYSTem:DISPlay:RESult?
```

Parameter

0|OFF	Plotting processing Off
ED	*Added in Version 4:*
	Plotting processing Off (Only ED measurement results)
1|ON	Plotting processing On

Response Data

0|1

Example of Use

To stop the plotting processing of ED/Scope measurement results:

```
:SYST:DISP:RES OFF
:SYST:DISP:RES?
>0
```

To stop the plotting processing of only ED measurement results:

```
:SYST:DISP:RES OFF,ED
:SYST:DISP:RES?
>0
```

To resume the plotting processing:

```
:SYST:DISP:RES ON
:SYST:DISP:RES?
>1
```

Note:

The response time for remote control can be reduced by setting the plotting processing to Off.

3.3.7 Screen Copy

:SYSTem:PRINt:COPY

Function

This command saves the full screenshot to a file.

Syntax

```
:SYSTem:PRINt:COPY [<file_name>,<directory>][,PNG|JPEG]
```

Parameter

<file_name>,<directory>
Specify the file name and folder to save the screenshot. If the specified folder is not found, it is created automatically.
If the file name and folder are omitted, the screenshot is saved as mmddyyyy_hhmmssmmm.png|jpeg in the following directory:

Storage location for MP210xA and MP2100B:
C:\Program Files\Anritsu\MP2100A\MX210000A\UserData\Screen Copy

Storage location for MP2110A:
C:\Users\Public\Documents\Anritsu\MX210000A\UserData\Screen Copy

PNG|JPEG
Specify the image file format. If omitted, it defaults to PNG.

Example of Use

```
:SYST:PRIN:COPY "screen_copy_full","C:\screen_copy"
```

Notes:

- If the file name is not specified, an image file is created newly each time this command is sent. Make sure there is sufficient disk space.
- To save the screenshot of the Scope screen, execute :EYEP:PRIN:COPY.
- To obtain image file data, execute :SYST:DISP:DATA?.

:SYSTem:DISPlay:DATA?

Function

This command queries the last screenshot image file saved by :{SYST|EYEP}:PRIN:COPY.

Syntax

```
:SYSTem:DISPlay:DATA?
```

Response Data

#<digit><data_size><binary_data><terminator>
<digit> is a one-digit number that indicates the number of digits of the value of <data_size>.
<data_size> indicates the data size of <binary_data>.
<binary_data> is image file data of the screenshot.
<terminator> indicates the terminator (LF or CR/LF) specified by the :SYST:TERM command.

Example of Use

To save the screenshot to an image file:

```
:SYST:PRIN:COPY "screen_copy_full","C:\screen_copy"
```

To query the screenshot image file:

```
:SYST:DISP:DATA?
>#541056Avdl-*;E4"as…
```

Note:

The end of response data cannot be detected by the terminator, because <binary_data> contains CR/LF. The procedure for obtaining the response data for this command is described below.

1. Disable the terminator detection function of the control interface.
2. Send :SYSTem:DISPlay:DATA?.
3. Read the first byte that follows "#" in the response data. It indicates the number of digits of the data size.
4. Read the bytes by the number of digits indicated. The bytes indicate the data size (byte).
5. Read the binary data of the indicated data size.
6. Read the terminator.
7. Enable the terminator detection function of the control interface.
8. Output the received binary data as-is to a file to create a screenshot image file.

Example: For binary data of 2002 bytes

#42002an%*qe4445+\...

1. Read the first byte “4” that follows “#”. It indicates the data size is a four-digit number.
2. Read the four-digit string “2002” that follows “4”. It indicates the data size is 2002 bytes.
3. Disable the terminator detection function of the control interface.
4. Read the binary data of 2002 bytes.
5. Output the read binary data of 2002 bytes to a file, which is called a screenshot image file.

3.3.8 Specifying screen display

:DISPlay:ACTive

Function

This command activates the screen of the specified function.

Syntax

```
:DISPlay:ACTive <module_id>
```

Parameter

<module_id>
Refer to ":MODule:ID".

Example of Use

To activate the PPG/ED Ch1 screen:

```
:DISP:ACT 1
```

Note:

The purpose of this command is only to activate the specified screen. To specify the module you want to remotely control, execute :MODule:ID.

:DISPlay:ACTive:ACResult

Function

Added in Version 4: This command displays all BER measurement results on the screen.

Syntax

```
:DISPlay:ACTive:ACResult
```

Example of Use

```
:DISP:ACT:ACR
```

3.4 Status Register

3.4.1 Clearing register

*CLS (Clear Status)

Function

This command clears the standard event status register and output queue.

Syntax

```
*CLS
```

Example of Use

```
*CLS
```

Notes:

*CLS common command clears the following registers.

- Standard event status register
- Output queue
  Therefore, bits 5 of status byte register became 0.

The setting value of each enable register does not vary depending on *CLS.

- Standard event status enable register
- Service request enable register
- Operation status register
- Device dependent status register

The *CLS common command clears the status byte register when sending *CLS command before the query after the program message terminator. All unread messages in the output queue are cleared at this time. The relevant message example indicates below.

```
SENS:BIT 8500000

*CLS;SENS:BIT?
```

When receiving SENS:BIT? after *CLS, the status byte register is cleared.

:STATus:PRESet

Function

This command presets the event registers and transition filters of the operation status register and device dependent (PPG/ED/XFP/SFP+/Scope) registers.

Syntax

```
:STATus:PRESet
```

Example of Use

```
:STAT:PRES
```

Note:

All bits in the event registers and negative transition filters are set to 0, and all bits in the positive transition filters are set to 1.

3.4.2 Status byte register

*STB? (Status Byte)

Function

This command queries the value of the status byte register.

Syntax

```
*STB?
```

Response Data

<integer> = bit0 + bit1 + bit2 + bit3 + bit4 + bit5 + bit6 + bit7

bit7 : 2^7 = 128	Operation status register
bit6 : 2^6 = 64	RQS service request
bit5 : 2^5 = 32	Standard event status register summary
bit4 : 2^4 = 16	MAV output queue
bit3 : 2^3 = 8	Not used
bit2 : 2^2 = 4	Error event queue
bit1 : 2^1 = 2	Not used
bit0 : 2^0 = 1	Not used

Example of Use

```
*STB?
>0
```

*SRE (Service Request Enable)

Function

This command sets and queries the value of the service request enable register.

Syntax

```
*SRE <integer>
*SRE?
```

Parameter

<integer> = 0 to 255
The mask bits of the status byte register are set to 0.
The meanings of the bits are the same as those of *STB?.

Response Data

<integer>

Example of Use

The following example shows how to mask bits 7,6,3,1 and 0 and permit bits 5, 4, and 2.

```
*SRE 52
*SRE?
>52
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

3.4.3 Standard event status register

*ESR? (Standard Event Status Register)

Function

This command queries the value of the standard event status register.

Syntax

```
*ESR?
```

Response Data

<integer> = bit0 + bit1 + bit2 + bit3 + bit4 + bit5 + bit6 + bit7

bit7: 2^7 = 128 Power-on
bit6: 2^6 = 64 Not used
bit5: 2^5 = 32 Command error
bit4: 2^4 = 16 Operation error
bit3: 2^3 = 8 Device Dependent error
bit2: 2^2 = 4 Not used
bit1: 2^1 = 2 Not used
bit0: 2^0 = 1 Completion of operation

For more information about errors, refer to Appendix B “Error Codes”.
This value (sum of standard event status register, 0 to 255) is obtained by ANDing with 8 bits set by *ESE.

Example of Use

To query the value of the standard event status register at the time the operation error and command error have occurred:

```
*ESR?
>48
```

Note:

The standard event status register is cleared by executing *ESR?.

*ESE (Event Status Enable)

Function

This command sets and queries the value of the standard event status enable register.

Syntax

```
*ESE <integer>
*ESE?
```

Parameter

<integer> = 0 to 255
Set 0 to the standard event status register bit(s) when masking it (them). The meanings of the bits are the same as those of *ESR?.

Response Data

<integer>

Example of Use

To mask bits 4, 5, 6 and 7 and enable bits 0, 1, 2 and 3:

```
*ESE 15
*ESE?
>15
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

3.4.4 Operation status register

:STATus:OPERation:CONDition?

Function

This command queries the details of the operation status condition register.

Syntax

```
:STATus:OPERation:CONDition?
```

Response Data

<integer> = bit4 + bit11

bit4: 2^4 = 16 Measurement (ED)

bit11: 2^{11} = 2048 Pattern Setting (PPG/ED)

Example of Use

```
:STAT:OPER:COND?
>16
```

Note:

1 is not returned as the response data of bit11 because USER cannot be set at Test Pattern in MP2110A.

:STATus:OPERation[:EVENt]?

Function

This command queries the operation status event register.

Syntax

```
:STATus:OPERation[:EVENt]?
```

Response Data

<integer> = bit4 + bit11 + bit12

The meanings of the bits are the same as those of :STATus:OPERation:CONDition?.

Example of Use

```
:STAT:OPER?
>16
```

:STATus:OPERation:ENABle

Function

This command sets and queries the operation status enable register.

Syntax

```
:STATus:OPERation:ENABle <integer>
:STATus:OPERation:ENABle?
```

Parameter

<integer> = 0 to 65535
Specify the value for the bit(s) you want to enable.
The meanings of the bits are the same as those
of :STATus:OPERation:CONDition?.

Response Data

<integer>

Example of Use

To set only bit 4 of the operation status event register to be queried:
At this time, $2^4 = 16$ is set in the operation status enable register.

```
:STAT:OPER:ENAB 16
```

To query the value of the operation status enable register:

```
:STAT:OPER:ENAB?
>16
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

:STATus:OPERation:NTRansition

Function

This command sets and queries the transition filter (negative transition) of the operation status register.

Syntax

```
:STATus:OPERation:NTRansition <integer>
:STATus:OPERation:NTRansition?
```

Parameter

<integer> = 0 to 65535
To set the event register to 1 when the condition register has changed from 0 to 1, set the bit to 1.
The meanings of the bits are the same as those of :STATus:OPERation:CONDition?.

Response Data

<integer>

Example of Use

To set bit 4 of operation status event register to bit 1 when bit 4 of operation status condition register changed from 1 to 0:
At this time, $2^4 = 16$ is set to the transition filter (negative transition).)

```
:STAT:OPER:NTR 16
```

To query transition filter (negative transition) of operation status register:

```
:STAT:OPER:NTR?
>16
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

:STATus:OPERation:PTRansition

Function

This command sets and queries the transition filter (positive transition) of the operation status register.

Syntax

```
:STATus:OPERation:PTRansition <integer>
:STATus:OPERation:PTRansition?
```

Parameter

<integer> = 0 to 65535
If the event register is set to 1 when the condition register is changed from 0 to 1, the bit is set to 1.
The meanings of the bits are the same as those of :STATus:OPERation:CONDition?.

Response Data

<integer>

Example of Use

To set bit 11 of operation status event register to bit 1 when bit 11 of operation status condition register changes from 0 to 1:
At this time, $2^{11} = 2048$ is set in the transition filter (positive transition).

```
:STAT:OPER:PTR 2048
```

To query transition filter (positive transition) of operation status register:

```
:STAT:OPER:PTR?
>2048
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

3.4.5 PPG/ED status register

:INSTrument:PE<ch>:RESet

Function

This command initializes the PPG/ED status event register.

Syntax

```
:INSTrument:PE<ch>:RESet
```

Parameter

<ch>

Channel number of PPG/ED

Range 1 to 4

Example of Use

```
:INST:PE1:RES
```

:INSTrument:PE<ch>:CONDition?

Function

This command queries the details of the PPG/ED condition register.

Syntax

```
:INSTrument:PE<ch>:CONDition?
```

Parameter

<ch>

Channel number of PPG/ED

Range 1 to 4

Response Data

<integer> = bit0 + bit1 + bit2 + bit3 + bit4 + bit5

bit5: $2^5 = 32$ Omission Error

bit4: $2^4 = 16$ Insertion Error

bit3: $2^3 = 8$ CR Unlock

bit2: $2^2 = 4$ Pattern Sync Loss

bit1: $2^1 = 2$ Total Error

bit0: $2^0 = 1$ PLL Unlock

Example of Use

```
:INST:PE1:COND?
>1
```

Note:

In MP2110A, 1 is not returned as the response data of bit3, bit4,

and bit5.
In MP2100B, 1 is not returned as the response data of bit3.

:INSTrument:PE<ch>[:EVENt]?

Function

This command queries the details of the PPG/ED event register.

Syntax

```
:INSTrument:PE<ch>[:EVENt]?
```

Parameter

<ch>
Channel number of PPG/ED
Range 1 to 4

Response Data

<integer> = bit0 + bit1 + bit2 + bit3 + bit4 + bit5
The meanings of the bits are the same as those of :INSTrument:PE<ch>:CONDition?.

Example of Use

```
:INST:PE1?
>1
```

:INSTrument:PE<ch>:NTRansition

Function

This command sets and queries the transition filter (negative transition) of the PPG/ED status.

Syntax

```
:INSTrument:PE<ch>:NTRansition <integer>
:INSTrument:PE<ch>:NTRansition?
```

Parameter

<ch>
Channel number of PPG/ED
Range 1 to 4
<integer> = 0 to 65535
To set the event register to 1 when the condition register has changed from 0 to 1, set the bit to 1.
The meanings of the bits are the same as those of :INSTrument:PE<ch>:CONDition?.

Response Data

<integer>

Example of Use

```
:INST:PE1:NTR 15
:INST:PE1:NTR?
>15
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

:INSTrument:PE<ch>:PTRansition

Function

This command sets and queries the transition filter (positive transition) of the PPG/ED status.

Syntax

```
:INSTrument:PE<ch>:PTRansition <integer>
:INSTrument:PE<ch>:PTRansition?
```

Parameter

<ch>
Channel number of PPG/ED
Range 1 to 4
<integer> = 0 to 65535
To set the event register to 1 when the condition register has changed from 0 to 1, set the bit to 1.
The meanings of the bits are the same as those of :INSTrument:PE<ch>:CONDition?.

Response Data

<integer>

Example of Use

```
:INST:PE1:PTR?
>3
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

3.4.6 Scope status register

:INSTrument:WAV:RESet

Function

This command initializes the Scope status event register.

Syntax

```
:INSTrument:WAV:RESet
```

Example of Use

```
:INST:WAV:RES
```

:INSTrument:WAV:CONDition?

Function

This command queries the details of the Scope status condition register.

Syntax

```
:INSTrument:WAV:CONDition?
```

Response Data

<integer> = bit0 + bit1 + bit2 + bit3 + bit4 + bit5 + bit6 + bit7 + bit8

bit8: $2^8 = 256$ Not used
bit7: $2^7 = 128$ CRU Unlock (*Only in MP2110A*)*1
bit6: $2^6 = 64$ PT phase unlock (*Only in MP2110A*)*2
bit5: $2^5 = 32$ Trigger setting wrong (*Only in MP2110A*)*3
bit4: $2^4 = 16$ Free Running (*Only in MP2110A*)*4
bit3: $2^3 = 8$ Not used
bit2: $2^2 = 4$ CAL Alarm (Orange)*5
bit1: $2^1 = 2$ CAL Alarm (Red)*6
bit0: $2^0 = 1$ CAL Alarm (Yellow)*7

*1: Check the input signal to CRU In and the CRU setting on the Time dialog box. If something is wrong, correct the setting so that the Lock Status lamp is lit green.

*2: Execute :TIME:PTRigger:RESet.

*3: Set the frequency of input trigger clock to 0.1 to 15.0 GHz (2.4 GHz or above when Precision Trigger is On,).

*4: Check that the signal is input to the trigger clock input connector

*5: *MP210xx: Added in Version 3.01.00*: Execute CALibrate: AMPLitude.

Sampling Rate is changed by 3% or more. This may occur when Sampling Mode (Eye/Pulse/Coherent Eye) is changed, or Bit Rate or Pattern Length is changed in Pulse/Coherent Eye Mode.

MP2110A: Added in Version 6: Execute CALibrate:OEPower or :CALibrate:AMPLitude. The O/E corrected value has an error. It is displayed when :INPut:WAVLength is changed.

*6: Execute CALibrate:AMPLitude. The temperature has a difference of ±5.0°C or more from the previous calibration.

*7: Execute CALibrate:AMPLitude. The temperature has a difference of ±2.5°C or more from the previous calibration.

Example of Use

```
:INST:WAV:COND?
>1
```

:INSTrument:WAV[:EVENt]?

Function

This command queries the details of the Scope status event register.

Syntax

```
:INSTrument:WAV:[EVENt]?
```

Response Data

<integer> = bit0 + bit1 + bit2 + bit3 + bit4 + bit5 + bit6 + bit7 + bit8
The meanings of the bits are the same as those of :INSTrument:WAV:CONDition?.

Example of Use

```
:INST:WAV?
>1
```

:INSTrument:WAV:NTRansition

Function

This command sets and queries the transition filter (negative transition) of the Scope status register.

Syntax

```
:INSTrument:WAV:NTRansition <integer>
:INSTrument:WAV:NTRansition?
```

Parameter

<integer> = 0 to 65535
To set the event register to 1 when the condition register has changed from 0 to 1, set the bit to 1.
The meanings of the bits are the same as those of :INSTrument:WAV:CONDition?.

Response Data

<integer>

Example of Use

```
:INST:WAV:NTR?
>1
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

:INSTrument:WAV:PTRansition

Function

This command sets and queries the transition filter (positive transition) of the Scope status register.

Syntax

```
:INSTrument:WAV:PTRansition <integer>
:INSTrument:WAV:PTRansition?
```

Parameter

<integer> = 0 to 65535
If the event register is set to 1 when the condition register is changed from 0 to 1, the bit is set to 1.
The meanings of the bits are the same as those of :INSTrument:WAV:CONDition?.

Response Data

<integer>

Example of Use

```
:INST:WAV:PTR?
>3
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

3.4.7 XFP/SFP+ status register

:INSTrument:XSFP:RESet

Function

This command initializes the XFP/SFP+ status event register.

Syntax

```
:INSTrument:XSFP:RESet
```

:INSTrument:XSFP:CONDition?

Function

This command queries the condition register details on the XFP/SFP+ status register.

Syntax

```
:INSTrument:XSFP:CONDition?
```

Response Data

<integer> = bit0 + bit1

bit1: $2^1 = 2$ LOS

bit0: $2^0 = 1$ Ready

Example of Use

```
:INST:XSFP:COND?
>0
```

:INSTrument:XSFP[:EVENt]?

Function

This command queries the details on the XFP/SFP+ status event register.

Syntax

```
:INSTrument:XSFP[:EVENt]?
```

Response Data

<integer> = bit0 + bit1

The meanings of the bits are the same as those of :INSTrument:XSFP:CONDition?.

Example of Use

```
:INST:XSFP?
>0
```

:INSTrument:XSFP:NTRansition

Function

This command sets and queries the transition filter (negative transition) of the XFP/SFP+ status.

Syntax

```
:INSTrument:XSFP:NTRansition <integer>
:INSTrument:XSFP:NTRansition?
```

Parameter

<integer> = 0 to 65535
To set the event register to 1 when the condition register has changed from 0 to 1, set the bit to 1.
The meanings of the bits are the same as those of :INSTrument:XSFP:CONDition?.

Response Data

<integer>

Example of Use

```
:INST:XSFP:NTR?
>3
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

:INSTrument:XSFP:PTRansition

Function

This command sets and queries the transition filter (positive transition) of the XFP/SFP+ status.

Syntax

```
:INSTrument:XSFP:PTRansition <integer>
:INSTrument:XSFP:PTRansition?
```

Parameter

<integer> = 0 to 65535
If the event register is set to 1 when the condition register is changed from 0 to 1, the bit is set to 1.
The meanings of the bits are the same as those of :INSTrument:XSFP:CONDition?.

Response Data

<integer>

Example of Use

```
:INST:XSFP:PTR?
>3
```

Note:

When the target register is “Not used”, enabling the register is not required in this command.

3.4.8 Register operation without dependence on module

*OPC (Operation Complete)

Function

*OPC sets bit 0 of the standard event status register to be changed from 0 to 1 when execution of the message being processed has been completed.
*OPC? queries OPC bit value.
OPC bit is "0" while the message is being processed and "1" when the processing is completed.
When one of the following happens, the operation completion waiting by *OPC and *OPC? become invalid.

- Power is turned on.
- DCL or SCL has been received on IEEE488.1 interface.
- *CLS command is received.
- *RST command is received.
- All running processes have been completed.

Syntax

```
*OPC
*OPC?
```

Response Data

1

Example of Use

```
*OPC?
>1
```

Note:

For the BERTWave, the response data for *OPC? is always 1.

Each of all the messages, including *OPC?, for the BERTWave cannot be executed, unless the previously sent message has been completed. Therefore, the response data for *OPC? is always 1 (Operation Complete).

*WAI (Wait to Continue)

Function

This command holds execution of the next message until processing of the message sent before *WAI is completed.

Syntax

```
*WAI
```

Example of Use

```
*WAI
```

Note:

It is not required to use *WAI, because the BERTWave sends a message after completing the processing of the previous message.

:SOURce:OUTPut:ASET

Function

This command sets and queries the On/Off state of signal outputting from all PPG channels and optical outputting from XFP/SFP+.

Syntax

```
:SOURce:OUTPut:ASET <enable>
:SOURce:OUTPut:ASET?
```

Parameter

0|OFF
1|ON

Response Data

0|1
0 All outputting (PPG/Optical) Off
1 At least one outputting (PPG/Optical) On

Example of Use

```
:SOUR:OUTP:ASET ON
:SOUR:OUTP:ASET?
>1
```

*TRG (Trigger)

Function

This command triggers the measurement for all modules (all ED channels and Scope).

Syntax

```
*TRG
```

Example of Use

```
*TRG
```

:SENSe:MEASure:ASTP

Function

This command stops the measurement for all modules (all ED channels and Scope).

Syntax

```
:SENSe:MEASure:ASTP
```

Example of Use

```
:SENS:MEAS:ASTP
```

:SENSe:MEASure:ASTate?

Function

This command queries the measurement status for all modules (all ED channels and Scope).

Syntax

```
:SENSe:MEASure:ASTate?
```

Response Data

0|1

0 Measurement stops for all modules.

1 At least one module, measurement in progress

Example of Use

```
:SENS:MEAS:AST?
>0
```

3.5 Specifying Module and Channel

There are two methods of specifying one of the function menu items of BERTWave, by using remote commands.

3.5.1 Specifying with command

This section describes the method of specifying a function menu item, by using the :MODule:ID command.
The commands sent after the :MODule:ID command will control items on the specified function menu.

:MODule:ID

Function

This command sets and queries the module that is subject to remote control.

Syntax

```
:MODule:ID <module_id>
:MODule:ID?
```

Parameter

<module_id> = {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

	Function menu items
1	PPG/ED Ch1
2	PPG/ED Ch2
3	XFP/SFP+
4	O/E
5	EYE/Pulse Scope
6	Jitter Analysis
7	Transmission Analysis
8	PPG/ED Ch3
9	PPG/ED Ch4

Response Data

<module_id>

Example of Use

To start/stop ED Ch1 measurement:

```
:MOD:ID 1
:SENSe:MEASure:STARt
:SENSe:MEASure:STOP
```

To start ED Ch3 measurement:

```
:MOD:ID 8
:SENSe:MEASure:STARt
```

To start Scope measurement:
```
:MOD:ID 5
:SAMPling:STATus RUN
```
To query the module ID:
```
:MOD:ID?
>5
```

Notes:

- The IDs for Ch3 and Ch4 of PPG/ED are respectively 8 and 9, not 3 and 4.
- In MP2110A, when O/E(4) is specified, Scope(5) is specified.
- If the ID is not specified correctly, an Undefined Header error occurs.

3.5.2 Specifying in header (Added in Version 3.02)

This section describes the method of specifying a function menu item, by adding a keyword for module/channel at the beginning of the header.
It is not required to send the :MODule:ID command described in 3.5.1 "Specifying with command".

Key Word	Function Menu Item
:BERT[<ch>]	PPG/ED Ch1 to Ch 4 When nothing is specified at <ch>, Ch1 is specified. Example: To start the ED channel 1 measurement: :BERT:SENSe:MEASure:STARtED To start the channel 3 measurement: :BERT3:SENSe:MEASure:STARt
:PMODule	XFP/SFP+ (Pluggable Module)
:OE	O/E In MP2110A, when :OE is specified, a command is sent to Scope.
:SCOPe	Scope Example: To start the Scope measurement: :SCOPe:SAMPling:STATus RUN
:JITTer	Jitter Analysis (MX210001A) Do not describe ":SENSe" that is the beginning of the command. Example: :JITTer:GRAPh:BATHtub:SAMPle
:VNA	Transmission Analysis (MX210002A) Do not describe ":SENSe" that is the beginning of the command. Example: :VNA:WE:SIGNal:SOURce

Notes:

- The above keywords cannot be used in native commands.
- This method is not available if the compatibility with the version 3.02 or earlier is required.

3.6 BERT(PPG/ED) messages

3.6.1 Window Operation

:DISPlay:RESult

Function

Only in MP2110A: This command sets and queries switching the Setup and Result panels of the BERT.

Syntax

```
:DISPlay:RESult <enable>
:DISPlay:RESult?
```

Parameter

0|OFF	Setup panel
1|ON	Result panel

Response Data

0|1

Example of Use

To switch the panel of the BERT to the Result panel.

```
:DISP:RES 1
:DISP:RES?
>1
```

3.6.2 Clock Input

:OUTPut:RCLock

Function

Added in Version 4: This command queries the Reference CLK for PPG/ED.

Syntax

```
:OUTPut:RCLock INT|EXT10M|EXT1_16|EXT1_40|CH1
:OUTPut:RCLock?
```

Parameter

In MP210xA and MP2100B:

INT	Internal 10MHz
EXT10M	10MHz In
EXT1_16	Ext 1/16 In
CH1	Dependent on the Reference CLK for PPG1 (Available if the remote control target is Ch2.)

In MP2110A:

INT	Internal 10MHz
EXT1_16	Ext 1/16 In
EXT1_40	Ext 1/40 In (When the Option 093 is installed)

Response Data

INT|EXT10M|EXT1_16| EXT1_40|CH1

Example of Use

```
:OUTP:RCL INT
:OUTP:RCL?
>INT
```

Note:

In MP210xA and MP2100B:
When Ch3 or Ch4 is selected at :MODule:ID, a setting error (–220 Parameter error) occurs because the clocks for Ch3 and Ch4 are dependent on the clocks for Ch1 and Ch2.
In MP2110A:
The setting can be changed regardless of the channel selection status.
When External is set, the settings using ":OUTPut:BITRate:STANdard" and ":OUTPut:BITRate:OFFSet" are ignored and an error (–220 Parameter error) occurs.

:OUTPut:RCLock:SELect

Function

Only in MP210xA and MP2100B: This command is compatible with version 4 or later, and sets and queries which clock (internal or external) is used as Reference CLK.

Syntax

```
:OUTPut:RCLock:SELect
INTernal|EXTernal|CH1External|CH2External|SYNChronize
:OUTPut:RCLock:SELect?
```

Parameter

INTernal	Ch1/2: Internal clock
EXTernal	Ch1/2: External clock
CH1External	Ch1: External clock, Ch2: Internal clock
CH2External	Ch1:Internal clock, Ch2: External clock
SYNChronize	Ch1: Internal clock, Ch2: Dependent on Reference CLK for PPG1

Response Data

INT | EXT | CH1E | CH2E | SYNC

Example of Use

```
:OUTP:RCL:SEL INT
:OUTP:RCL:SEL?
>INT
```

Notes:

- In MP210xA, SYNChronize is available only when the Option 052 is installed. (In MP2100B, it is available even when the Option 052 is not installed.)
- When Ch3 or Ch4 is selected at :MODule:ID, an error (–220 Parameter error) occurs because the bit rates of Ch3 and Ch4 are dependent on the bit rates of Ch1 and Ch2.
- When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:OUTPut:CMU:EXTClock

Function

Only in MP210xA and MP2100B: This command is compatible with version 4 or later, and sets and queries the connector to input Reference CLK when the external clock is used as Reference CLK.

Syntax

```
:OUTPut:CMU:EXTClock 10M|1_16
:OUTPut:CMU:EXTClock?
```

Parameter

10M Ext 10MHz In
1_16 Ext 1/16 In

Response Data

10M|1_16

Example of Use

To set the **Ext Clk In** connector for inputting an external clock:

```
:OUTP:CMU:EXTC 1_16
:OUTP:CMU:EXTC?
>1_16
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:OUTPut:RCLock:STATus?

Function

Only in MP2100B and MP2110A: This command queries the state of the Reference CLK status indicator.

Syntax

```
:OUTPut:RCLock:STATus?
```

Response Data

NONE	Reference clock was not detected.
NOT_READY	Reference clock was detected but is not synchronized with pattern data.
READY	Reference clock is synchronized with pattern data.

Example of Use

```
:OUTP:RCL:STAT?
>READY
```

:OUTPut:RCLock:APPLy

Function

Only in MP2100B and MP2110A: This command starts synchronization of the input clock when the external clock is used as Reference CLK.
This command is available when the response to :OUTPut:RCLock:STATus? is NOT_READY.

Syntax

```
:OUTPut:RCLock:APPLy
```

Example of Use

```
:OUTP:RCL:STAT?
>NOT_READY
:OUTP:RCL:APPL
```

Notes:

- In MP2100B:
  When Ch3 or Ch4 is selected at :MODule:ID, a setting error (–220 Parameter error) occurs because the clocks for Ch3 and Ch4 are dependent on the clocks for Ch1 and Ch2.
  In MP2110A:
  The setting can be changed regardless of the channel selection status.
- When Internal is selected at ":OUTPut:RCLock", the setting using the command is ignored and an error (–310,"System error") occurs.

3.6.3 Clock Output

:OUTPut:CLOCk:SOURce:CHANnel

Function

Only in MP2110A: This command sets and queries the clock source of the clock that is output to the Clock Output terminal.

Syntax

```
:OUTPut:CLOCk:SOURce:CHANnel 1|3
:OUTPut:CLOCk:SOURce:CHANnel?
```

Parameter

To set the clock source to Ch1/2: 1
To set the clock source to Ch3/4: 3

Response Data

1|3

Example of Use

To set the clock source to Ch1/2:

```
:OUTP:CLOC:SOUR:CHAN 1
:OUTP:CLOC:SOUR:CHAN?
>1
```

Note:

This command can be set only when the Option 014 is installed. When this command is executed while the Option 014 is not installed, an error (–113 Undefined Header) occurs.

:OUTPut:CLOCk:DIVRate?

Function

Only in MP2110A: This command queries the division rate (for operating bit rate) of the clock that is output to the Clock Output terminal.

Syntax

```
:OUTPut:CLOCk:DIVRate?
```

Response Data

1_2|1_4|OFF
1_2 1/2
1_4 1/4

Example of Use

To query the division rate of the clock that is output to the Clock Output terminal:

```
:OUTP:CLOC:DIVR?
>1_2
```

:OUTPut:SYNC:SOURce

Function

This command sets and queries the signal source to be output to the **Sync Out** connector.

Syntax

```
:OUTPut:SYNC:SOURce <character>
:OUTPut:SYNC:SOURce?
```

Parameter

<character>

In MP210xA and MP2100B:

For PPG Pattern Sync:	PPG{1|2|3|4}PATT
When using PPG as the signal source:	PPG{1|2}CLOC{1|2|4|8|16|64}
When using ED as the signal source:	ED{1|2|3|4}CLOC{4|8|16}

In MP2110A:

For PPG Pattern Sync:	PPG{1|2|3|4}PATT
When using PPG as the signal source:	PPGCLOC{8|16|40}

Response Data

<character>

Example of Use

To set the 1/16 divided clock that synchronizes with the data output from PPG1, as the signal to be output to the **Sync Out** connector:

```
:OUTP:SYNC:SOUR PPG1CLOC16
:OUTP:SYNC:SOUR?
>PPG1CLOC16
```

3.6.4 Tracking Setting

:BERT:ALL:PARam:TRACking

Function

Added in Version 4: This command sets and queries the On/Off state of Ch Tracking.

Syntax

```
:BERT:ALL:PARam:TRACking <enable>
:BERT:ALL:PARam:TRACking?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:BERT:ALL:PAR:TRAC ON
:BERT:ALL:PAR:TRAC?
>1
```

Notes:

- In MP210xA and MP2100B:
  When Ch Tracking is set to On and a channel other than Ch1 is set at Reference CLK of PPG2, an error (–220 Parameter error) occurs.
  In MP2110A:
  The setting can be changed regardless of the Ch Tracking status.
- If Ch Tracking is set to On, ED Tracking is set to On as well.

:SENSe:PARam:TRACking

Function

This command sets and queries the On/Off state of ED Tracking.

Syntax

```
:SENSe:PARam:TRACking <enable>
:SENSe:PARam:TRACking?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:SENS:PAR:TRAC ON
:SENS:PAR:TRAC?
>1
```

3.6.5 Bit Rate Setting

:OUTPut:BITRate:STANdard

Function

This command sets and queries the bit rate standard for the PPG.

Syntax

```
:OUTPut:BITRate:STANdard <bitrate_standard>
:OUTPut:BITRate:STANdard?
```

Parameter

<bitrate_standard> = <string>

The following strings can be used for <bitrate_standard>.

Table 3.6.5-1 Bit Rate Standards (MP210xA, MP2100B)

<string>	Standard	Bit Rate (bit/s)	Remarks
"VARIABLE"	Variable-1/1	6.25G to 12.5G	
"10G_FC_FEC"	10GFC FEC	11.3168G	
"10G_OTU2E"	OTU2e (10GbE FEC)	11.095728G	
"10G_OTU1E"	OTU1e (10GbE FEC)	11.049107G	
"OTU-2"	OTU2	10.709225G	
"OC-192FEC"	G.975 FEC	10.664228G	
"10G_FC"	10GFC	10.51875G	
"10G_LAN"	10GbE LAN/PHY	10.3125G	
"INF10G"	InfiniBand x4	10G	
"OC-192"	OC-192/STM-64	9.95328G	
"10G_WAN"	10GbE WAN/PHY	9.95328G	
"8G_FC"	8GFC	8.5G	
"VARIABLE-1/2"	Variable-1/2	6.25G to 3.125G	
"CPRI-10"	CPRI x10	6.144G	*2
"OBSAIRP3-8"	OBSAI RP3 x8	6.144G	*2
"INF5G"	InfiniBand x2	5G	
"4G_FC"	4GFC	4.25G	
"VARIABLE-1/4"	Variable-1/4	3.125G to 1.5625G	*1
"CPRI-5"	CPRI x5	3.072G	*1,*2
"OBSAIRP3-4"	OBSAI RP3 x4	3.072G	*1,*2
"OTU-1"	OTU1	2.666057G	*1
"2GBE"	2GbE	2.5G	*1
"INF"	InfiniBand	2.5G	*1
"OC-48"	OC-48/STM16	2.488G	*1

*1: If the Option 090 is not installed, the bit rate that is equal to or less than Variable-1/4 cannot be selected for the ED.

*2: If the Option 090/092 is not installed, the bit rate cannot be selected because the bit rate range is limited.

Table 3.6.5-1 Bit Rate Standards (MP210xA, MP2100B) (Cont'd)

<string>	Standard	Bit Rate	Remarks
"CPRI-4"	CPRI x4	2.4576G	*1
"2G_FC"	2GFC	2.125G	*1
"VARIABLE-1/8"	Variable-1/8	1.5625G to 781.25M	*1
"OBSAIRP3-2"	OBSAI RP3 x2	1.536G	*1,*2
"1GBE"	1GbE	1.25G	*1
"OC-24"	OC-24	1.244G	*1
"CPRI-2"	CPRI x2	1.2288G	*1
"1G_FC"	1GFC	1.0625G	*1
"VARIABLE-1/16"	Variable-1/16	781.25M to 390.625M	*1
"OBSAIRP3"	OBSAI RP3	768M	*1,*2
"OC-12"	OC-12/STM-4	622.08M	*1
"CPRI"	CPRI	614.4M	*1
"VARIABLE-1/32"	Variable-1/32	390.625M to 195.312M	*1
"VARIABLE-1/64"	Variable-1/64	195.312M to 125M	*1
"OC-3"	OC-3/STM-1	155.22M	*1

Table 3.6.5-2 Bit Rate Standards (MP2110A)

<string>	Standard	Bit Rate (bit/s)	Remarks
"VARIABLE"	Variable(24.3-28.2G)	24.3 to 28.2G	
"32G_FC"	32GFC	28.05G	
"OTU-4"	OTU4	27.952493G	
"100GE_4_FEC"	100GbE/4 FEC	27.7393G	
"100GE_4"	100GbE/4	25.78125G	
"INF_EDR"	InfiniBand EDR	25.78125G	
"INF_FDR"	InfiniBand FDR	25.78125G	*
"VARIABLE10G"	Variable (9.5-14.2G)	9.5 to 14.2G	*
"16G_FC"	16GFC	14.025G	*
"10G_FC_FEC"	10GFC FEC	11.3168G	*
"10G_OTU2E"	OTU2e (10GbE FEC)	11.095728G	*
"10G_OTU1E"	OTU1e (10GbE FEC)	11.049107G	*
"OTU-2"	OTU2	10.709225G	*
"OC-192FEC"	G.975 FEC	10.664228G	*
"10G_FC"	10GFC	10.51875G	*
"10G_LAN"	10GbE LAN/PHY	10.3125G	*
"INF10G"	InfiniBand x4	10G	*
"OC-192"	OC-192/STM-64	9.95328G	*
"10G_WAN"	10GbE WAN/PHY	9.95328G	*

*: This can be selected when the Option 093 is installed.

Response Data

<bitrate_standard> = <string>

Example of Use

```
:OUTP:BITR:STAN "10G_LAN"
:OUTP:BITR:STAN?
>"10G_LAN"
```

Notes:

- *Modified in Version 4:*
  In MP210xA and MP2100B:
  When bit rates are set to Ch3 and Ch4 using this command, a setting error (–220 Parameter error) occurs because the bit rates for Ch3 and Ch4 are dependent on the bit rates for Ch1 and Ch2.
  In MP2110A:
  The bit rate setting can be changed regardless of the selected channel.
- When External is set at Reference CLK, an error (–220 Parameter error) occurs.

:INPut:BITRate:STANdard

Function

This command is compatible with version 4 or later, and sets and queries the bit rate standard for the ED.

Syntax

```
:INPut:BITRate:STANdard <bitrate_standard>
:INPut:BITRate:STANdard?
```

Parameter

<bitrate_standard>
Refer to Parameter of ":OUTPut:BITRate:STANdard".

Response Data

<bitrate_standard>

Example of Use

```
:INP:BITR:STAN "10G_LAN"
:INP:BITR:STAN?
>"10G_LAN"
```

Notes:

- *Modified in Version 4:*
  In MP2100B:
  When this command is executed, an error (–220 Parameter error) occurs because the bit rate of ED is dependent on the bit rate of PPG.
  In MP2110A:
  The bit rates of PPG and ED are the common, so the bit rates can be set without error.
- When External is set at Reference CLK, an error (–220 Parameter error) occurs.

:OUTPut:BITRate

Function

This command sets and queries the bit rate of the PPG if the bit rate standard is Variable.

Syntax

```
:OUTPut:BITRate <bitrate>
:OUTPut:BITRate?
```

Parameter

<bitrate> = <numeric>

The setting ranges are limited as follows depending on the options (unit: 1 kbit/s steps).

Table 3.6.5-3 Bit Rate Range

In MP210xA and MP2100B:

<bitrate>	When the Option 092 is installed (kbit/s)	When the Option 090 is installed (kbit/s)	When the Option 090 and 092 are not installed (kbit/s)
"VARIABLE"	6250001 to 12500000	8000000 to 12500000	8500000 to 11320000
"VARIABLE-1/2"	3125001 to 6250000	4000000 to 6250000	4250000 to 5660000
"VARIABLE-1/4"	1562501 to 3125000	2000000 to 3125000	2125000 to 2830000
"VARIABLE-1/8"	781251 to 1562500	1000000 to 1562500	1062500 to 1415000
"VARIABLE-1/16"	390626 to 781250	500000 to 781250	531250 to 707500
"VARIABLE-1/32"	195313 to 390625	250000 to 390625	265625 to 353750
"VARIABLE-1/64"	125000 to 195312	125000 to 195312	132813 to 176875

In MP2110A:

<bitrate>	Setting Range
"VARIABLE"	24300000 to 28200000
"VARIABLE10G"	9500000 to 14200000 (When the Option 093 is installed)

Response Data

<bitrate> = <numeric>

Example of Use

```
:OUTP:BITR:STAN "VARIABLE"
:OUTP:BITR 8500000
:OUTP:BITR?
>8500000
```

Notes:

- In MP2100B of *Version 4*:
  When bit rates are set to Ch3 and Ch4 using this command, a

setting error (–220 Parameter error) occurs because the bit rates for Ch3 and Ch4 are dependent on the bit rates for Ch1 and Ch2.
In MP2110A:
The bit rate setting can be changed regardless of the selected channel.

- In MP2100B and MP2110A:
  <bitrate_standard> is automatically changed depending on the specified bit rate only when <bitrate_standard> is Variable.
- When External is set at Reference CLK, a setting error (–220 Parameter error) occurs.

:INPut:BITRate

Function

This command is compatible with version 4 or later, and sets and queries the bit rate of the ED if the bit rate standard is Variable.

Syntax

```
:INPut:BITRate <bitrate>
:INPut:BITRate?
```

Parameter

<bitrate> = <numeric>

Response Data

<bitrate> = <integer>

Example of Use

```
:INP:BITR:STAN "VARIABLE"
:INP:BITR 8500000
:INP:BITR?
>8500000
```

Notes:

- *Modified in Version 4:*
  In MP2100B:
  When this command is executed, an error (–220 Parameter error) occurs because the bit rate of ED is dependent on the bit rate of PPG.
- In MP2110A:
  The bit rates of PPG and ED are the common, so the bit rates can be set without error.

:OUTPut:BITRate:OFFSet

Function

This command sets and queries the bit rate offset for the PPG.

Syntax

```
:OUTPut:BITRate:OFFSet <numeric>
:OUTPut:BITRate:OFFSet?
```

Parameter

<numeric>

Range –100 to 100, 1 ppm step

Response Data

<integer>

Example of Use

```
:OUTP:BITR:OFFS 100
:OUTP:BITR:OFFS?
>100
```

Notes:

- In MP210xA and MP2100B:
  When Ch3 or Ch4 is selected at :MODule:ID and this command is used, a setting error (–220 Parameter error) occurs because the bit rates for Ch3 and Ch4 are dependent on the bit rates for Ch1 and Ch2.
  In MP2110A:
  The setting can be changed regardless of the selected channel.
- When External is set at Reference CLK, a setting error (–220 Parameter error) occurs.

:OUTPut:BITRate:DIVRate

Function

Only in MP210xA and MP2100B: This command sets and queries the clock divide ratio (1/n) of the PPG.

Syntax

```
:OUTPut:BITRate:DIVRate <character>
:OUTPut:BITRate:DIVRate?
```

Parameter

<character>
1_{1|2|4|8|16|32|64}

Response Data

1_{1|2|4|8|16|32|64}

Example of Use

To set the clock divide ratio of the PPG to 1/2:

```
:OUTP:BITR:DIVR 1_2
:OUTP:BITR:DIVR?
>1_2
```

Notes:

- This command is executed when the Reference CLK is Ext 1/16 In.
- When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:INPut:BITRate:DIVRate?

Function

Only in MP210xA and MP2100B: This command queries the clock divide ratio (1/n) of the error detector.

Syntax

```
:INPut:BITRate:DIVRate?
```

Response Data

1_{1|2|4|8|16|32|64}

Example of Use

```
:INP:BITR:DIVR?
>1_2
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

3.6.6 Test Pattern Setting

:SOURce:PATTern:TYPE

Function

This command sets and queries the test pattern of the PPG.

Syntax

```
:SOURce:PATTern:TYPE <character>
:SOURce:PATTern:TYPE?
```

Parameter

In MP210xA and MP2100B:

<character> = PRBS{7 | 9 | 15 | 23 | 31} | USER

In MP2110A:

<character> = PRBS{7 | 9 | 15 | 23 | 31} | CLOC{2 | 16}

Response Data

PRBS{7 | 9 | 15 | 23 | 31} | CLOC{2 | 16} | USER

Example of Use

To set the test pattern of the PPG to PRBS2^23–1:

```
:SOUR:PATT:TYPE PRBS23
:SOUR:PATT:TYPE?
>PRBS23
```

To set the test pattern of the PPG to 1/2 Clock Pattern:

```
:SOUR:PATT:TYPE CLOC2
:SOUR:PATT:TYPE?
>CLOC2
```

To set the test pattern of the PPG to ProgrammablePattern

```
:SOUR:PATT:TYPE USER
:SOUR:PATT:TYPE?
>USER
```

:SENSe:PATTern:TYPE

Function

This command sets and queries the test pattern of the ED.

Syntax

```
:SENSe:PATTern:TYPE <character>
:SENSe:PATTern:TYPE?
```

Parameter

In MP210xA and MP2100B:

<character> = PRBS{7|9|15|23|31}|USER

In MP2110A:

<character> = PRBS{7|9|15|23|31}

Response Data

PRBS{7|9|15|23|31}|USER

Example of Use

To set the test pattern of the ED to PRBS2^23–1:

```
:SENS:PATT:TYPE PRBS23
:SENS:PATT:TYPE?
>PRBS23
```

To set the test pattern of the ED to ProgrammablePattern

```
:SENS:PATT:TYPE USER
:SENS:PATT:TYPE?
>USER
```

Note:

The parameters that can be set in MP2110A and MP210xA, MP2100B are different.
When Ch Tracking is set to On:
In MP210xA and MP2100B:
When Ch3 or Ch4 is selected at :MODule:ID, a setting error (–220 Parameter error) occurs because the clocks for Ch3 and Ch4 are dependent on the clocks for Ch1 and Ch2.
In MP2110A:
The setting can be changed regardless of the channel selection status.
When ED Tracking is set to On, a setting error occurs.

:SOURce:PATTern:LOGic

Function

This command sets and queries the test pattern logic (positive/negative logic) of the PPG.

Syntax

```
:SOURce:PATTern:LOGic POSitive|NEGative
:SOURce:PATTern:LOGic?
```

Parameter

POSitive|NEGative

Response Data

POS|NEG

Example of Use

```
:SOUR:PATT:LOG POS
:SOUR:PATT:LOG?
>POS
```

Note:

When the test pattern for PPG is not PRBS, the setting using this command is ignored and an error (–220,"Parameter error") occurs.

:SENSe:PATTern:LOGic

Function

This command sets and queries pattern logic (negative/positive logic) of the ED.

Syntax

```
:SENSe:PATTern:LOGic POSitive|NEGative
:SENSe:PATTern:LOGic?
```

Parameter

POSitive|NEGative

Response Data

POS|NEG

Example of Use

```
:SENS:PATT:LOG POS
:SENS:PATT:LOG?
>POS
```

Note:

When the test pattern for ED is not PRBS, the setting using this command is ignored and an error (–220,"Parameter error") occurs.

:SOURce:MMEMory:PATTern:RECall

Function

Only in MP210xA and MP2100B: This command sets the programmable pattern file for the PPG.

Syntax

```
:SOURce:MMEMory:PATTern:RECall <file_name>,{BIN|TXT}
```

Parameter

<file_name>
Name of pattern file in the following folder (including file extension)
C:\Program Files\Anritsu\MP2100A\MX210000A\UserData\Pattern

BIN|TXT
BIN Binary file
TXT Text file

Example of Use

```
:SOUR:PATT:TYPE USER
:SOUR:MMEM:PATT:REC "10101010.dat",BIN
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SENSe:MMEMory:PATTern:RECall

Function

Only in MP210xA and MP2100B: This command sets the programmable pattern file for the ED.

Syntax

```
:SENSe:MMEMory:PATTern:RECall <file_name>,BIN|TXT
```

Parameter

Same as :SOURce:MMEMory:PATTern:RECall.

Example of Use

```
:SENS:PATT:TYPE USER
:SENS:MMEM:PATT:REC "10101010.dat",BIN
```

Notes:

- When ED Tracking is set to On, an error occurs.
- When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SOURce:PATTern:DATA:LENGth?

Function

This command queries the pattern length when the test pattern of the PPG is Programmable Pattern.

Syntax

```
:SOURce:PATTern:DATA:LENGth?
```

Response Data

<integer>

2 to 1305600

Example of Use

```
:SOUR:PATT:DATA:LENG?
>16384
```

:SENSe:PATTern:DATA:LENGth?

Function

This command queries the pattern length when the test pattern of the ED is Programmable Pattern.

Syntax

```
:SENSe:PATTern:DATA:LENGth?
```

Response Data

Same as :SOURce:PATTern:DATA:LENGth?.

Example of Use

```
:SENS:PATT:DATA:LENG?
>16384
```

3.6.7 PPG

:OUTPut:DATA:OUTPut

Function

This command sets and queries the On/Off state of PPG signal output.

Syntax

```
:OUTPut:DATA:OUTPut <enable>
:OUTPut:DATA:OUTPut?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:OUTP:DATA:OUTP ON
:OUTP:DATA:OUTP?
>1
```

Note:

To turn on and off the signal output from all channels, use :SOURce:OUTPut:ASET.

:OUTPut:DATA:AMPLitude

Function

This command sets and queries the amplitude voltage of the signal to be output from the **Data Out** and **$\overline{\text{Data}}$ Out** connectors of PPG.

Syntax

```
:OUTPut:DATA:AMPLitude DATA,<numeric>
:OUTPut:DATA:AMPLitude? DATA
```

Parameter

<numeric>

Range 0.10 to 0.80, 0.01 Vp-p step

Response Data

<numeric>

Example of Use

To set the output amplitude of the PPG to 0.5 Vp-p:

```
:OUTP:DATA:AMPL DATA,0.5
:OUTP:DATA:AMPL? DATA
>0.5
```

:OUTPut:DATA:ATTFactor

Function

This command sets and queries the External Attenuator Factor value of PPG.

Syntax

```
:OUTPut:DATA:ATTFactor DATA,<numeric>
:OUTPut:DATA:ATTFactor? DATA
```

Parameter

<numeric>

Range 0 to 30, 1 dB step

Response Data

<integer>

Example of Use

```
:OUTP:DATA:ATTF DATA,20
:OUTP:DATA:ATTF? DATA
>20
```

:OUTPut:DATA:RELative?

Function

This command queries the Relative value displayed on PPG panel.

Syntax

```
:OUTPut:DATA:RELative? DATA
```

Response Data

<numeric>

Range 0.00 to 0.80, 0.01 Vp-p step

Example of Use

```
:OUTP:DATA:REL? DATA
>0.4
```

:SOURce:PATTern:EADDition:SET

Function

Only in MP210xA and MP2100B: This command sets and queries whether to add a bit error(s) to the test pattern to be generated by PPG.

Syntax

```
:SOURce:PATTern:EADDition:SET <enable>
:SOURce:PATTern:EADDition:SET?
```

Parameter

0|OFF

1|ON

Response Data

0|1

Example of Use

```
:SOUR:PATT:EADD:SET ON
:SOUR:PATT:EADD:SET?
>1
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SOURce:PATTern:EADDition:VARiation

Function

Only in MP210xA and MP2100B: This command sets and queries the error addition mode (Repeat/Single) for PPG.

Syntax

```
:SOURce:PATTern:EADDition:VARiation REPeat|SINGle
:SOURce:PATTern:EADDition:VARiation?
```

Parameter

REPeat|SINGle

Response Data

REP|SING

Example of Use

```
:SOUR:PATT:EADD:VAR REP
:SOUR:PATT:EADD:VAR?
>REP
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SOURce:PATTern:EADDition:SINGle

Function

This command generates a single error in the test pattern when the error addition mode for PPG is Single.

Syntax

```
:SOURce:PATTern:EADDition:SINGle
```

Example of Use

```
:SOUR:PATT:EADD:SING
```

:SOURce:PATTern:EADDition:RATE

Function

Only in MP210xA and MP2100B: This command sets and queries the rate of adding bit errors when the error addition mode for PPG is Repeat.

Syntax

```
:SOURce:PATTern:EADDition:RATE <character>[,1]
:SOURce:PATTern:EADDition:RATE?
```

Parameter

<character>
E_{2|3|4|5|6|7|8|9|10|11|12}
Exponent of error addition rate (2 to 12)

[,1]
Indicates a mantissa of the error addition rate is 1.

Response Data

<character>,1

Example of Use

To set the error addition rate to 1E–9:

```
:SOUR:PATT:EADD:RATE E_9,1
:SOUR:PATT:EADD:RATE?
>E_9,1
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

3.6.8 ED

3.6.8.1 Setting

:INPut:DATA:INTerface

Function

This command sets and queries the connector inputting signal into the ED.

Syntax

```
:INPut:DATA:INTerface DATA|DIFF|OPT|XDATa
:INPut:DATA:INTerface?
```

Parameter

DATA	Electrical Single-Ended Data (Inputting to the **Data In** connector)
DIFF	Differential 50 Ohm (Inputting to the **Data In** and $\overline{\text{Data}}$ **In** connectors)
OPT	Optical (Inputting to the **O/E Data In** connector)
XDATa	Electrical Single-Ended $\overline{\text{Data}}$ (Inputting to the $\overline{\text{Data}}$ **In** connector)

Response Data

DATA | DIFF | OPT | XDAT

Example of Use

```
:INP:DATA:INT DATA
:INP:DATA:INT?
>DATA
```

Notes:

- OPT is available if the Option 003 or 007 is installed.
- XDATa and DIFF cannot be selected for Ch1 of the Option 003/007.

:INPut:DATA:ATTFactor

Function

This command sets and queries the external attenuation factor for the ED in dB unit.

Syntax

```
:INPut:DATA:ATTFactor DATA,<numeric>
:INPut:DATA:ATTFactor? DATA
```

Parameter

<numeric>

Range 0 to 30, 1 dB step

Response Data

<integer>

Example of Use

```
:INP:DATA:ATTF DATA,10
:INP:DATA:ATTF? DATA
>10
```

:INPut:DATA:THReshold

Function

This command sets and queries the input threshold for ED, in mV units.

Syntax

```
:INPut:DATA:THReshold <numeric>
:INPut:DATA:THReshold?
```

Parameter

<numeric>

The range of the input threshold, assuming External ATT is A (dB), is as follows:

Range $-85*10^{\frac{A}{20}}$ to $85*10^{\frac{A}{20}}$, $10^{\frac{A}{20}}$ mV step

Response Data

<integer>

Notes:

- The setting needs to be made according to the specified step.
- The fractional portions of minimum and maximum values are truncated, and the step is rounded off to the nearest integer.

Example of Use

```
:INP:DATA:ATTF DATA,10
:INP:DATA:THR -270
:INP:DATA:THR?
>-270
```

:SENSe:PATTern:SYNC:ASYNc

Function

Only in MP210xA and MP2100B: This command sets and queries the On/Off state of Auto SYNC (auto pattern resynchronization processing) of ED.

Syntax

```
:SENSe:PATTern:SYNC:ASYNc <enable>
:SENSe:PATTern:SYNC:ASYNc?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:SENS:PATT:SYNC:ASYN ON
:SENS:PATT:SYNC:ASYN?
>1
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SENSe:PATTern:SYNC:THReshold

Function

Only in MP210xA and MP2100B: This command sets and queries the threshold for Auto SYNC of ED.

Syntax

```
:SENSe:PATTern:SYNC:THReshold <character>
:SENSe:PATTern:SYNC:THReshold?
```

Parameter

<character>
INT or E_{2|3|4|5|6|7|8} (1E–2 to 1E–8)

Response Data

<character>
INT|E_{2|3|4|5|6|7|8}

Example of Use

```
:SENS:PATT:SYNC:THR E_2
:SENS:PATT:SYNC:THR?
>E_2
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SENSe:PATTern:SYNC:PSMode

Function

Only in MP210xA and MP2100B: This command sets and queries the On/Off state of SYNC Control when the test pattern for ED is Programmable Pattern.

Syntax

```
:SENSe:PATTern:SYNC:PSMode FRAMe|NORMal
:SENSe:PATTern:SYNC:PSMode?
```

Parameter

FRAMe	SYNC Control On
NORMal	SYNC Control Off

Response Data

FRAM|NORM

Example of Use

To set SYNC Control to On:

```
:SENS:PATT:SYNC:PSM FRAM
:SENS:PATT:SYNC:PSM?
>FRAM
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SENSe:PATTern:SYNC:FPOSition

Function

Only in MP210xA and MP2100B: This command sets and queries the frame position when the test pattern for ED is Programmable Pattern and Sync Control is On.

Syntax

```
:SENSe:PATTern:SYNC:FPOSition <numeric>
:SENSe:PATTern:SYNC:FPOSition?
```

Parameter

<numeric>

Range 1 to <Data Length> –64, 1 bit step

Response Data

<integer>

Example of Use

```
:SENS:PATT:SYNC:FPOS 1
:SENS:PATT:SYNC:FPOS?
>1
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

3.6.8.2 Status-related commands

[:BERT:ALL]:DISPlay:RESult:EALarm:HRESet

Function

This command resets the error alarm history of ED.

Syntax

```
[:BERT:ALL]:DISPlay:RESult:EALarm:HRESet
```

Example of Use

To reset the error alarm history of the channel specified by :MODule:ID:

```
:DISP:RES:EAL:HRES
```

Added in Version 4: To clear the error alarm histories of all ED channels when specifying `:BERT:ALL`:

```
:BERT:ALL:DISP:RES:EAL:HRES
```

[:BERT:ALL]:CALCulate:DATA:MONitor?

Function

This command queries if the specified error/alarm has occurred at ED.

Syntax

```
[:BERT:ALL]:CALCulate:DATA:MONitor?
"BIT:TOTal"|"CRUNlock"|"PSLoss"
```

Parameter

"BIT:TOTal"	Bit Error
"CRUNlock"	CR Unlock
"PSLoss"	SYNC Loss

Response Data

"Occur"	Error/alarm has occurred.
"Not Occur"	Error/alarm has not occurred.

Example of Use

To query if Bit Error has occurred at the channel specified by :MODule:ID:

```
:CALC:DATA:MON? "BIT:TOT"
>"Occur"
```

Added in Version 4: When `:BERT:ALL` is specified, the data for all channels are returned sequentially from Ch1 (Ch1, Ch2, Ch3, Ch4).

```
:BERT:ALL:CALC:DATA:MON? "BIT:TOT"
>"Occur","Occur","Occur","Occur"
```

[:BERT:ALL]:CALCulate:DATA:MONitor:HISTory?

Function

Added in Version 5: This command queries whether there is the specified error/alarm history or not.

Syntax

```
[:BERT:ALL]:CALCulate:DATA:MONitor:HISTory?
"BIT:TOTal"|"PSLoss"
```

Parameter

"BIT:TOTal"	Bit Error
"PSLoss"	SYNC Loss

Response Data

"Occur"	Error/alarm has occurred.
"Not Occur"	Error/alarm has not occurred.

Example of Use

To query the Bit Error history of the channel specified by :MODule:ID:

```
:CALC:DATA:MON:HIST? "BIT:TOT"
>"Occur"
```

When `:BERT:ALL` is specified, the data for all channels are returned sequentially from Ch1 (Ch1, Ch2, Ch3, Ch4).

```
:BERT:ALL:CALC:DATA:MON:HIST? "BIT:TOT"
>"Occur","Occur","Occur","Occur"
```

3.6.8.3 Fast measurement command

[:BERT:ALL]:SENSe:MEASure:IMMediate?

Function

Added in Version 4: This command immediately executes BER measurement and returns results.

Syntax

```
[:BERT:ALL]:SENSe:MEASure:IMMediate? <time>[,<item>]
```

Parameter

<time>
Measurement time
Range 10 to 3000 step, 10 ms
<item>
Measurement items

"ER:TOTal"	Total Bit Error Rate
"EC:TOTal"	Total Bit Error Count
"PSLoss"	SYNC Loss

Response Data

If <item> is specified, only the specified measurement item(s) is returned.
If <item> is omitted, results of all the measurement items are returned in the order of Total ER, Total EC, and SYNC Loss, as a comma-separated string.
If :BERT:ALL is specified in the header, the data of all channels are returned in the order of Ch1, Ch2, Ch3, and Ch4.
Response data for each measurement item is returned in the following format:

- "ER:TOTal": "0.0001E-18" to "1.0000E-00" (Form2)
- "EC:TOTal": "0" to "9999999" or "1.0000E+07" to "9.9999E+17" (Form1)
- "PSLoss": "Not Occur" or "Occur"
- "-------" is returned if there is no data to return.

Example of Use

To perform 10 ms measurement on the specified channel, and query the result of Bit Error Rate:

```
:SENS:MEAS:IMM? 10,"ER:TOTal"
>"1.0000E-02"
```

To perform 1 s measurement on the specified channel, and query the results of all measurement items:

```
:SENS:MEAS:IMM? 1000
>"1.0000E-02","850001","Not Occur"
```

To perform 10 ms measurement on all channels, and query the results of Bit Error Rate (The response example below is for the BERTWave that consists of two channels.):

```
:ALL:SENS:MEAS:IMM? 10,"ER:TOTal"
>"1.0000E-02","1.0000E-02"
```

To perform 1 s measurement on all channels, and query the results of all measurement items (The response example below is for the BERTWave that consists of two channels.):

```
:ALL:SENS:MEAS:IMM? 1000
>"1.0000E-02","850001","Not
Occur","1.0000E-02","850001","Not Occur"
```

Notes:

- This command is effective in reducing the time required to perform a measurement sequence. (The shorter measurement period than GUI can be specified. The measurement sequence can automatically be started and stopped. Measurement results of multiple channels can be queried at a time.)
- This command is available in MP2100B and MP2110A, but not available in MP210xA.
- Execution of this command stops measurement on all ED channels.
- Measurement results of this command are not displayed in the PPG/ED screen, because fast processing is performed. The screen display of the scope is updated.

3.6.8.4 Measurement commands

[:BERT:ALL]:CALCulate:DATA:EALarm?

Function

This command queries the measurement results of the error detector (ED).

Syntax

```
[:BERT:ALL]:CALCulate:DATA:EALarm? "<period>:<item>"
```

Parameter

<period> = CURRent | LAST

CURRent	To query the current measurement results.
LAST	To query the results of the last measurement performed as specified by Gating Time.

<item>

Select a measurement item(s) from the following:

AINTerval:CRUNlock	CR Unlock Seconds
AINTerval:PSLoss	SYNC Loss Seconds
EC:TOTal	Bit Error Count Total
EC:INSertion	Bit Error Count Insertion
EC:OMIssion	Bit Error Count Omission
ER:TOTal	Bit Error Rate Total
ER:INSertion	Bit Error Rate Insertion
ER:OMIssion	Bit Error Rate Omission
CC:TOTal	Clock Count Total
FREQuency	FREQ(kHz)

Response Data

The response format varies, depending on the item specified for <item>, as follows.

AINTerval:{CRUNlock | PSLoss}

EC:{TOTal | INSertion | OMIssion}

CC:TOTal

Range: "0" to "9999999" or "1.0000E+07" to "9.9999E+17" (Form1)

ER:{TOTal | INSertion | OMIssion}

Range: "0.0001E-18" to "1.0000E-00" (Form2)

FREQuency

"<integer>" (Form3)

"-------" is returned if there is no data to return.

Example of Use

To query the Bit Error Rate with the channel specified by :MODule:ID:

```
:CALC:DATA:EAL? "CURR:ER:TOT"
>"0.0000E-12"
```

Added in Version 4: (When :BERT:ALL is specified)
To return the data of all channels in the order of Ch1, Ch2, Ch3, and Ch4:

```
:BERT:ALL:CALC:DATA:EAL? "CURR:ER:TOT"
>"0.0000E-12","0.0000E-12","0.0000E-12","0.0000E-12"
```

Notes:

- When CR Unlock or SYNC Loss is detected once or more in the 1 s interval, it is counted as the 1 second where an error has occurred.
- SYNC Loss is not counted while CR Unlock is detected.
- Bit Error is not counted while CR Unlock or SYNC Loss is detected in the 100 ms interval.
- In MP2110A, "-------" is returned for the query where INSertion, OMIssion, and CRUNlock are specified at <item>.
- In MP2100B, "-------" is returned for the query where CRUNlock is specified at <item>.

:SENSe:MEASure:EALarm:MODE

Function

This command sets and queries the gating cycle of ED.

Syntax

```
:SENSe:MEASure:EALarm:MODE REPeat|SINGle|UNTimed
:SENSe:MEASure:EALarm:MODE?
```

Parameter

REPeat	Repeatedly performs measurement for the period of time set for Gating Time.
SINGle	Performs single measurement for the period of time set for Gating Time.
UNTimed	Finishes measurement using panel operation or continues measurement until the :SENSe:MEASure:STOP is sent.

Response Data

REP|SING|UNT

Example of Use

```
:SENS:MEAS:EAL:MODE REP
:SENS:MEAS:EAL:MODE?
>REP
```

Note:

When Ch Tracking is set to ON:
In MP210xA and MP2100B:
When Ch3 or Ch4 is selected at :MODule:ID, a setting error (–220 Parameter error) occurs because the clocks for Ch3 and Ch4 are dependent on the clocks for Ch1 and Ch2.
In MP2110A:
The setting can be changed regardless of the channel selection status.

:SENSe:MEASure:EALarm:PERiod

Function

This command sets and queries the gating time of ED.

Syntax

```
:SENSe:MEASure:EALarm:PERiod
<days>,<hours>,<minutes>,<seconds>
:SENSe:MEASure:EALarm:PERiod?
```

Parameter

<days>,<hours>,<minutes>,<seconds>

Response Data

<days>,<hours>,<minutes>,<seconds>

Example of Use

To set the gating time to 1 minute:

```
:SENS:MEAS:EAL:PER 0,0,1,0
:SENS:MEAS:EAL:PER?
>0,0,1,0
```

Note:

When Ch Tracking is set to ON:
In MP210xA and MP2100B:
When Ch3 or Ch4 is selected at :MODule:ID, a setting error (–220 Parameter error) occurs because the clocks for Ch3 and Ch4 are dependent on the clocks for Ch1 and Ch2.
In MP2110A:
The setting can be changed regardless of the channel selection status.

:DISPlay:RESult:EALarm:MODE

Function

This command sets and queries the On/Off state of real-time update of ED measurement results.

Syntax

```
:DISPlay:RESult:EALarm:MODE <enable>
:DISPlay:RESult:EALarm:MODE?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:DISP:RES:EAL:MODE ON
:DISP:RES:EAL:MODE?
>1
```

Note:

If set to Off, measurement results are updated when the measurement progress reaches 100%.

[:BERT:ALL]:SENSe:MEASure:STARt

Function

This command starts the ED measurement.

Syntax

```
[:BERT:ALL]:SENSe:MEASure:STARt
```

Example of Use

To start measurement on the channel specified by :MODule:ID :

```
:SENS:MEAS:STAR
```

Added in Version 4: To start measurement on all ED channels:

```
:BERT:ALL:SENS:MEAS:STAR
```

Notes:

- If measurement is running, execution of this command clears the current data, and restarts measurement.
- To start measurement of all modules including Scope, use :SENS:MEAS:ASTR.

[:BERT:ALL]:SENSe:MEASure:STOP

Function

This command stops the ED measurement.

Syntax

```
[:BERT:ALL]:SENSe:MEASure:STOP
```

Example of Use

To stop measurement on the channel specified by :MODule:ID:

```
:SENS:MEAS:STOP
```

Added in Version 4: To stop measurement on all ED channels:

```
:BERT:ALL:SENS:MEAS:STOP
```

Note:

To stop measurement of all modules including Scope, use :SENSe:MEASure:ASTP.

[:BERT:ALL]:SENSe:MEASure:EALarm:STATe?

Function

This command queries measurement status of the ED.

Syntax

```
[:BERT:ALL]:SENSe:MEASure:EALarm:STATe?
```

Response Data

0|1

0 None of the channels is being measured.

1 At least one of the channels is being measured.

Example of Use

To query the measurement status of the channel specified by :MODule:ID:

```
:SENS:MEAS:EAL:STAT?
>0
```

Added in Version 4: When `:BERT:ALL` is specified, the measurement statuses for all ED channels are queried.

```
:BERT:ALL:SENS:MEAS:EAL:STAT?
>0
```

Note:

To query the measurement status of all modules including Scope, use :SENSe:MEASure:ASTate?.

:SENSe:MEASure:EALarm:STARt?

Function

This command queries the measurement start time of the ED.

Syntax

```
:SENSe:MEASure:EALarm:STARt?
```

Response Data

"<year>,<month>,<day>,<hour>,<minute>,<second>"

If time data is not available, "0,0,0,0,0,0" is returned.

Example of Use

To query the measurement start time of the ED:

```
:SENS:MEAS:EAL:STAR?
>"2009,10,05,16,25,40"
```

:SENSe:MEASure:EALarm:STOP?

Function

This command queries the measurement end time of the ED.

Syntax

```
:SENSe:MEASure:EALarm:STOP?
```

Response Data

"<year>,<month>,<day>,<hour>,<minute>,<second>"

If time data is not available, "0,0,0,0,0,0" is returned.

Example of Use

When Gating Cycle is Single or Repeat

```
:SENS:MEAS:EAL:STOP?
>"2009,10,05,16,25,40"
```

When Gating Cycle is Untimed and the measurement is performing:

```
:SENS:MEAS:EAL:STOP?
>"0,0,0,0,0,0"
```

:SENSe:MEASure:EALarm:ELAPsed?

Function

This command queries the period of time elapsed from the start time of the ED measurement.

Syntax

```
:SENSe:MEASure:EALarm:ELAPsed?
```

Response Data

"<days>,<hours>,<minutes>,<seconds>"

Example of Use

```
:SENS:MEAS:EAL:ELAP?
>"0,0,2,10"
```

:SENSe:MEASure:EALarm:TIMed?

Function

This command queries the measurement remaining time until the ED completes the measurement.

Syntax

```
:SENSe:MEASure:EALarm:TIMed?
```

Response Data

"<days>,<hours>,<minutes>,<seconds>"

Example of Use

```
:SENS:MEAS:EAL:TIM?
>"0,0,2,10"
```

3.7 XFP/SFP+ specific messages (MP210xA, MP2100B)

:CALCulate:OPTical:STATus?

Function

This command queries the status of the XFP/SFP+ slot optical transceiver.

Syntax

```
:CALCulate:OPTical:STATus? "READy"|"LOS"
```

Parameter

"READy" Detection of XFP/SFP+
"LOS" Occurrence of LOS

Response Data

If "READy" is specified:
"None" XFP/SFP+ not detected
"Occur" XFP/SFP+ detected
If "LOS" is specified:
"None" No LOS occurs or XFP/SFP+ not detected
"Occur" LOS occurs

Example of Use

```
:CALC:OPT:STAT? "LOS"
>"Occur"
```

:SOURce:OPTical:SIGNal:WLENgth?

Function

This command queries the wavelength of the optical transceiver (XFP/SFP+).

Syntax

```
:SOURce:OPTical:SIGNal:WLENgth?
```

Response Data

<string>
"xxxx" Four-digit wavelength (Unit: nm)
(Right-justified if it is less than four digits.)
"-------" XFP/SFP+ not detected

Example of Use

```
:SOUR:OPT:SIGN:WLEN?
>" 850"
```

:SOURce:OPTical:SIGNal:OUTPut

Function

This command sets and queries the optical output of the optical transceiver (XFP/SFP+).

Syntax

```
:SOURce:OPTical:SIGNal:OUTPut <enable>
:SOURce:OPTical:SIGNal:OUTPut?
```

Parameter

0|OFF
1|ON

Response Data

0|1

0	Optical output Off or XFP/SFP+ not detected
1	Optical output On

Example of Use

To set the optical output to On:

```
:SOUR:OPT:SIGN:OUTP ON
:SOUR:OPT:SIGN:OUTP?
>1
```

:SOURce:OPTical:XFP:REFClock

Function

This command sets and queries the reference clock of the optical transceiver (XFP).

Syntax

```
:SOURce:OPTical:XFP:REFClock <character>
:SOURce:OPTical:XFP:REFClock?
```

Parameter

<character>

ED1Sync	Sync with ED1
ED2Sync	Sync with ED2
PPG1Sync	Sync with PPG1
PPG2Sync	Sync with PPG2

Response Data

ED1Sync | ED2Sync | PPG1Sync | PPG2Sync

Example of Use

```
:SOUR:OPT:XFP:REF PPG1Sync
:SOUR:OPT:XFP:REF?
>PPG1Sync
```

3.8 O/E messages

Refer to Section 3.9.5.3, “O/E”.

3.9 Scope-specific messages

Modified in Version 6: Read “bitrate” in the explanation as “baud rate”.

3.9.1 Setting active channel and On/Off display

:INPut:CHA|CHB

Function

This command sets and queries the On/Off state of the Ch A/B waveform display of Scope.

Syntax

```
[:SENSe]:INPut:CHA|CHB 0|OFF|1|ON
[:SENSe]:INPut:CHA|CHB?
```

Parameter

0|OFF
1|ON

Response Data

OFF|ON

Example of Use

```
:INP:CHA ON
:INP:CHA?
>ON
```

:CONFigure:MEASure:CHANnel

Function

This command sets and queries the active channel for waveform measurement of Scope.

A channel on which the following operations are executed is called “active channel”.

1. Channel on which Auto Scale of X axis is executed.
   When both Channel A and B are On, Auto Scale is executed on X axis of the channel specified by this command.
   When one channel is On, Auto Scale is executed on that channel.
   Y axis of both channels are adjusted regardless of the setting of active channel.
2. Target channel when measurement results are acquired by the:FETCh*** and :MEASure*** commands.
3. Channel on which Mask Test and Histogram Measurement are executed.

Syntax

```
:CONFigure:MEASure:CHANnel A|B
:CONFigure:MEASure:CHANnel?
```

Parameter

A: Channel A
B: Channel B

Response Data

A|B

Example of Use

```
:CONF:MEAS:CHAN A
:CONF:MEAS:CHAN?
>A
```

Note:

The active channel is changed not only by this command, but also by the :INPut:CHA|CHB command. The channel that is turned On last by the :INPut:CHA|CHB command is the active channel.

:CONFigure:MEASure:TYPe

Function

This command sets and queries Measure Item of Scope.

Syntax

```
:CONFigure:MEASure:TYPe
AMPHistogram|AMPMask|AMPTIME|HISTogram|MASK|OFF
:CONFigure:MEASure:TYPe?
```

Parameter

AMPHistogram	Amplitude/Time & Histogram*
AMPMask	Amplitude/Time & Mask Test*
AMPTIME	Amplitude/Time*
HISTogram	Histogram
MASK	Mask Test
OFF	Off

*: Even when one of these values (including Amplitude/Time) is specified, the setting is ignored in the Pulse mode.

The functions are turned On/Off as the following table by the specified parameters.
The results of both Mask Test and Histogram cannot be displayed.

Table 3.9.1-1 Measure Item Settings and Measurement Result Display

Parameter	Amp/Time	MaskTest	Histogram
AMPHistogram	On	Off	On
AMPMask	On	On	Off
AMPTIME	On	Off	Off
HISTogram	Off	Off	On
MASK	Off	On	Off
OFF	Off	Off	Off

Response Data

AMPHistogram | AMPMask | AMPTIME | HISTogram | MASK | OFF

Example of Use

```
:CONF:MEAS:TYP AMPM
:CONF:MEAS:TYP?
>AMPMask
```

:CONFigure:MEASure:AMPTime:DISPlay

Function

Added in Version 6: This command sets and queries the On/Off state of the display of the Amplitude/Time results.
For screen display of measurement results, refer to the description of :CONFigure:MEASure:TYPE.

Syntax

```
:CONFigure:MEASure:AMPTime:DISPlay <enable>
:CONFigure:MEASure:AMPTime:DISPlay?
```

Parameter

0 | OFF	Measurement results are not displayed.
1 | ON	Measurement results are displayed.

Example of Use

```
:CONF:MEAS:AMPT:DISP ON
:CONF:MEAS:AMPT:DISP?
>1
```

3.9.2 Basic operation

:DISPlay:WINDow:GRAPhics:CLEar

Function

This command erases the trace on the Scope screen.

Syntax

```
:DISPlay:WINDow:GRAPhics:CLEar
```

Example of Use

```
:DISP:WIND:GRAP:CLE
```

:SAMPling:STATus

Function

This command runs and holds the sampling processing of Scope, and queries the state of the sampling processing.

Syntax

```
[:SENSe]:SAMPling:STATus RUN|HOLD
[:SENSe]:SAMPling:STATus?
```

Parameter

RUN Runs the sampling processing.
HOLD Holds the sampling processing.
Modified in Version 3.03/4.01: When the :SAMPling:STATus RUN command is sent while Sampling is running, the Sampling restarts.

Response Data

RUN|HOLD

Example of Use

To run the sampling processing:

```
:SAMP:STAT RUN
```

To query if the Sampling processing has been held (HOLD status):

```
:SAMP:STAT?
>HOLD
```

:DISPlay:WINDow:AUTOscale

Function

This command automatically adjusts the vertical and horizontal axes so that the waveform is displayed in the center of the Scope screen.

Syntax

```
:DISPlay:WINDow[:SCALe]:AUTOscale
[BOTH|HORIzontal|VERTical]
```

Parameter

In the Eye mode, the following parameters can be specified.
In the Coherent Eye mode, only BOTH can be specified.

BOTH	Scale on the vertical axis and offset on the horizontal axis auto-adjusted
HORizontal	Offset on the horizontal axis auto-adjusted
VERTical	Scale on the vertical axis auto-adjusted

Example of Use

```
:DISP:WIND:AUTO
```

Note:

If the parameter is specified when in Eye mode, execution of this command only adjusts the offset on the horizontal axis and the scale on the vertical axis, without measuring the waveform frequency. Additionally, when this parameter is specified, Acquire Clock (measuring frequency of the clock signal input to the **Trigger Clk In** connector) is not executed during Auto Scale, which provides an advantage of shorter duration of automatic adjustment by Auto Scale. If the clock frequency is already known, specify this parameter as needed.

:DISPlay:WINDow:ZOOM

Function

Added in Version 6: This command sets and queries the expanded/reduced state of waveform display area. It is used for setting 9 or more measurement items and for returning the reduced waveform display area to normal size (expanded).

Syntax

```
:DISPlay:WINDow:ZOOM <enable>
:DISPlay:WINDow:ZOOM?
```

Parameter

0|OFF	Waveform display area is in reduced size.
1|ON	Waveform display area is in normal size (expanded).

Example of Use

```
:DISPlay:WINDow:ZOOM ON
```

```
:DISPlay:WINDow:ZOOM?
>1
```

3.9.3 Setup

3.9.3.1 Signal Type

:DISPlay:SIGNal:CHA|CHB

Function

Added in Version 6: This command sets and queries the type of input signal (NRZ or PAM4). PAM4 can be set when MP2110A-095 is installed.

Syntax

```
:DISPlay:SIGNal:CHA|CHB NRZ|PAM4
:DISPlay:SIGNal:CHA|CHB?
```

Parameter

NRZ	NRZ
PAM4	PAM4

Example of Use

```
:DISP:SIGN:CHB NRZ
:DISP:SIGN:CHB?
>NRZ
```

3.9.3.2 Sampling

:DISPlay:MODE

Function

This command sets and queries the Sampling Mode of Scope.

Syntax

```
[:SENSe]:DISPlay:MODE AJITter|COHErenteye|EYE|PULSe
[:SENSe]:DISPlay:MODE?
```

Parameter

AJITter	AdvancedJitter mode (*Added in Version 6:* Can be set when OPT096 is installed.)
COHErenteye	Coherent eye mode
EYE	Eye mode
PULSe	Pulse mode

Response Data

AJITter | COHErenteye | EYE | PULSe

Example of Use

```
:DISP:MODE PULSe
:DISP:MODE?
>PULSe
```

:DISPlay:MODE:EYE:FAST

Function

Only in MP2100B: This command sets and queries the On/Off state of the Fast Sampling Mode of Scope.

Syntax

```
[:SENSe]:DISPlay:MODE:EYE:FAST <enable>
[:SENSe]:DISPlay:MODE:EYE:FAST?
```

Parameter

0 | OFF
1 | ON

Response Data

0 | 1

Example of Use

```
:DISP:MODE EYE
:DISP:MODE:EYE:FAST ON
```

```
:DISP:MODE:EYE:FAST?
>1
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:OPTion:MAX:SAMPles:NUMber

Function

This command sets and queries the Number Of Samples.

Syntax

```
[:SENSe]:OPTion:MAX:SAMPles:NUMber <integer>
[:SENSe]:OPTion:MAX:SAMPles:NUMber?
```

Parameter

<integer>

When Sampling mode is set to **EYE**:

In MP210xx: 509 | 1021 | 1350 | 2039 | 4093 | 8191 | 16381

In MP2110A: 1350 | 2048 | 4050

When Sampling mode is set to **Coherent Eye** or **Pulse**:

512 | 1024 | 2048 | 4096 | 8192 | 16384

Response Data

<integer>

Example of Use

```
:SENS:OPT:MAX:SAMP:NUM?
>8191
```

:ACCUmulation:TYPe

Function

This command sets and queries the data collection process of Scope.

Syntax

```
[:SENSe]:ACCUmulation:TYPe <character>
[:SENSe]:ACCUmulation:TYPe?
```

Parameter

<character>

NONe	The data collection is not overwritten. When the fresh data is collected, the displayed data is deleted.
INFinite	The data collection is overwritten. The acquired data does not go out of the screen.
LIMited	The data collection is limited by the number of samples and time. When it reaches the limited conditions, the data collection is ended. The acquired data does not go out of the screen.
PERSistency	The data collection is overwritten. After the fixed time, the acquired data goes out of the screen.
AVERaging	The average of the collected data is displayed. Only when Sampling Mode is set to **Pulse**, this can be used.

Response Data

NONe | INFinite | LIMited | PERSistency | AVERaging

Example of Use

```
:ACCU:TYP LIMited
:ACCU:TYP?
>LIMited
```

:ACCUmulation:LIMit

Function

This command sets and queries the limit (time/number) for the data collection, when Accumulation Type of Scope is Limited.

Syntax

```
[:SENSe]:ACCUmulation:LIMit
TIME|SAMPle|WAVeform|PATTern,<numeric>
[:SENSe]:ACCUmulation:LIMit?
```

Parameter

TIME | SAMPLe | WAVeform | PATTern

TIME The data collection is limited at time.

SAMPle	The data collection is limited by the number of samples.
WAVeform	The data collection is limited by the number of waveforms.
PATTern	*Added in Version 6:* The data collection is limited by the number of patterns. Valid in Advanced Jitter mode.

<numeric>
When specifying TIME, set the limit by the time in seconds.
When specifying SAMPle, set the limit by the number of samples in million units, in the range of 1 to 99999.
When specifying WAVeform, set the limit by the number of waveforms, in the range of 1 to 999999.
When specifying PATTern, set the limit by the number of patterns, in the range of 1 to 999999.

Response Data

TIME | SAMPLe | WAVeform | PATTern,<integer>

Example of Use

```
:ACCU:LIM SAMPLe,10
:ACCU:LIM?
>SAMPLe,10
```

Note:

The unit (second or million) does not need to be specified by the command. The unit is determined automatically depending on the specified limit type, as follows:

- second (when the limit type is Time)
- million (when the limit type is Sample)

When this message is transmitted while correcting the data (when Sampling of the screen is **RUN**), the displayed waveform is deleted and the data correction is done over again.

:ACCUmulation:PERSistency

Function

This command sets and queries the data display time when the data collection process of Scope is Persistency.

Syntax

```
[:SENSe]:ACCUmulation:PERSistency <numeric>
[:SENSe]:ACCUmulation:PERSistency?
```

Parameter

<numeric>
Time to display collected data (seconds)

Response Data

<integer>

Example of Use

```
:ACCU:PERS?
>10.0
```

:ACCUmulation:AVERaging

Function

This command sets and queries the averaging process count of Scope.

Syntax

```
[:SENSe]:ACCUmulation:AVERaging <integer>
[:SENSe]:ACCUmulation:AVERaging?
```

Parameter

<integer>
Range 1 to 9999

Response Data

<integer>
1 to 9999

Example of Use

```
:ACCU:AVER 1000
:ACCU:AVER?
>1000
```

3.9.3.3 Clock Recovery (MP2110A)

:TIME:CRU

Function

Added in Version 6: This command sets and queries the CRU mode.

Syntax

```
:TIME:CRU OFF|RECovery|THRough
:TIME:CRU?
```

Parameter

OFF
RECovery
THRough

Response Data

OFF|RECovery|THRough

Example of Use

```
:TIME:CRU RECovery
:TIME:CRU?
>RECovery
```

:TIME:CRU:RATE:STANdard

Function

Added in Version 6: This command sets and queries the operation rate for the CRU by standard name.

Syntax

```
:TIME:CRU:RATE:STANdard "<bitrate_standard>"
:TIME:CRU:RATE:STANdard?
```

Parameter

<bitrate_standard> = <string>
The following strings can be used for <bitrate_standard>.

Table 3.9.3-1 Symbol Rate Standards

<string>	Standard	Symbol Rate (baud)
"VARIABLE"	Variable (25.5-28.2G)	25.5 to 28.2G
"32G_FC"	32GFC	28.05G
"OTU-4"	OTU4	27.952493G
"100GE_4_FEC"	100GbE/4 FEC	27.7393G
"100GE_4"	100GbE/4	25.78125G
"400GE_8"	400GbE/8	26.5625G

Response Data

<bitrate_standard> = <string>

Example of Use

```
:TIME:CRU:RATE:STAN "100GE_4"
:TIME:CRU:RATE:STAN?
>"100GE_4"
```

:TIME:CRU:RATE

Function

Added in Version 6: This command sets and queries the CRU operation rate by symbol rate (kbaud).
If this command is sent when <bitrate_standard> is not VARIABLE, it is changed to VARIABLE automatically.

Syntax

```
:TIME:CRU:RATE <symbolrate>
:TIME:CRU:RATE?
```

Parameter

<symbolrate> = <numeric>
25500000 to 28200000

Response Data

<symbolrate> = <numeric>

Example of Use

```
:TIME:CRU:RATE 25781250
:TIME:CRU:RATE?
>25781250
```

:TIME:CRU:LBWidth

Function

Added in Version 6: This command sets and queries CRU Loop Band Width.

Syntax

```
:TIME:CRU:LBWidth 4M|10M|BITRATE_1667
:TIME:CRU:LBWidth?
```

Parameter

4M	4 MHz
10M	10 MHz
BITRATE_1667	<:TIME:CRU:RATE value> / 1667

Response Data

4M | 10M | BITRATE_1667

Example of Use

```
:TIME:CRU:LBW 10M
:TIME:CRU:LBW?
>10M
```

:TIME:CRU:STATus?

Function

Added in Version 6: This command queries the CRU lock status.

Syntax

```
:TIME:CRU:STATus?
```

Response Data

LOCK | UNLOCK

LOCK	CRU is locked. (The Status lamp is lit green
UNLOCK	CRU is not locked. (The Status lamp is lit red or black)

Example of Use

```
:TIME:CRU:STATus?
>LOCK
```

:TIME:CRU:FREQuency?

Function

Added in Version 6: This command queries the frequency (kHz) of CRU output clock signal.

Syntax

```
:TIME:CRU:FREQuency?
```

Response Data

<integer>

Example of Use

```
:TIME:CRU:FREQ?
>25781250
```

3.9.3.4 Clock Recovery (MP210xx)

:INPut:CLKRecovery

Function

This command sets and queries the clock recovery output mode of Scope.

Syntax

```
[:SENSe]:INPut:CLKRecovery OFF|LESS27|85
[:SENSe]:INPut:CLKRecovery?
```

Parameter

OFF	Sets the clock recovery output to Off
LESS27	Sets the clock recovery output to On and frequency to 0.1 to 2.7 GHz
85	Sets the clock recovery output to On and frequency to 8.5 to 12.5 GHz

Response Data

OFF|LESS27|85

Example of Use

```
:INP:CLKR 85
```

:CONFigure:CLKRecovery

Function

This command sets and queries the bandwidth of the clock recovery unit for Scope.

Syntax

```
:CONFigure:CLKRecovery 1|2|4|8 [MHz]
:CONFigure:CLRecovery?
```

Parameter

1|2|4|8 [MHz]

Response Data

1|2|4|8 MHz

Example of Use

To set the bandwidth of clock recovery unit to 4 MHz

```
:CONF:CLKR 4 MHz
:CONF:CLKR?
>4 MHz
```

3.9.3.5 Screen Copy

:EYEPulse:PRINt:COPY

Function

This command takes a screenshot of the Scope screen.

Syntax

```
[:SENSe]:EYEPulse:PRINt:COPY
[<file_name>,<directory>][,PNG|JPEG]
```

Parameter

Refer to :SYSTem:PRINt:COPY.

Example of Use

```
:EYEP:PRIN:COPY "screen_copy_eye","C:/screen_copy"
```

Notes:

- Specify a file name when capturing screen shots repeatedly. If the file name is not specified, an image file is created newly each time this command is sent. Make sure there is sufficient disk space.
- To take a screenshot of the entire screen, use :SYST:PRIN:COPY.
- To query the screenshot image file data, use :SYST:DISP:DATA?.
- In the software earlier than version 3.03/4.01.01, specify a module using :MOD:ID 5 before executing the COPY command.

:PRINt:GRATicule

Function

This command sets and queries the On/Off state for taking a screenshot of only the waveform area of Scope.

Syntax

```
[:SENSe]:PRINt:GRATicule <enable>
[:SENSe]:PRINt:GRATicule?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:PRIN:GRAT ON
:PRIN:GRAT?
>1
```

:PRINt:INVerse

Function

This command sets and queries the On/Off state of background color inversion for a screenshot of Scope.

Syntax

```
[:SENSe]:PRINt:INVerse <enable>
[:SENSe]:PRINt:INVerse?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:PRIN:INV OFF
:PRIN:INV?
>0
```

3.9.3.6 Waveform Color

:DISPlay:WAVeform:COLor

Function

Added in Version 5.01: This command sets the color of the waveform.

Syntax

```
:DISPlay:WAVeform:COLor CGRade|GSCale
:DISPlay:WAVeform:COLor?
```

Parameter

CGRade Color Grade
GSCale Gray Scale

Response Data

CGRade|GSCale

Example of Use

To set the waveform to color.

```
:DISP:WAV:COL CGRade
```

:DISPlay:MASK:COLor

Function

Added in Version 5.02: This command sets the color of the mask.

Syntax

```
:DISPlay:MASK:COLor PURPle|GRAY
:DISPlay:MASK:COLor?
```

Parameter

PURPle Purple
GRAY Gray

Response Data

PURPle|GRAY

Example of Use

To set the color of mask to purple.

```
:DISP:MASK:COL PURPle
```

3.9.3.7 Label

:DISPlay:INFormation

Function

Added in Version 5.02: This command sets and queries the On/Off state of the Preset Information display for Scope.

Syntax

```
:DISPlay:INFormation <enable>
:DISPlay:INFormation?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:DISP:INF OFF
:DISP:INF?
>0
```

:DISPlay:LABel

Function

This command sets a label to display in the Scope screen.

Syntax

```
:DISPlay:LABel "<label>"[,<pixel_x>,<pixel_y>]
:DISPlay:LABel?
```

Parameter

"<label>"
Specifies the character string of label (up to 1023 alphanumeric characters).
Added in Version 5.02: Starts a new line by inputting a line feed code (\n).

[,<pixel_x>,<pixel_y>]
Added in Version 5.02: Specifies the start position of label display (X, Y) (in the waveform display area, the upper left is (0, 0) and the lower right is (665, 497)).
*: This is enabled when **Preset Information** is set to **Off**.

Response Data

"<label>"[,<pixel_x>,<pixel_y>]

*: "<pixel_x>,<pixel_y>" is omitted when the default value is (0,0).

Example of Use

```
:DISP:LAB "BERTWave",50,10
:DISP:LAB?
> "BERTWave",50,10
:DISP:LAB "BERTWave",0,0
:DISP:LAB?
> "BERTWave"
```

Notes:

- Up to 1023 characters can be entered. However, all characters may not be displayed because the characters that can be displayed are limited.
- It takes a long time to actually display the label after completing this command processing during Sampling Run. Therefore, the wait time about 200 ms may be required before executing the COPY command that copies the screen (the required wait time varies depending on the length of the character string).

:DISPlay:LABel:DALL

Function

This command deletes the label displayed in the Scope screen.

Syntax

```
:DISPlay:LABel:DALL
```

Example of Use

```
:DISP:LAB:DALL
```

3.9.3.8 Trace Memory

:TMEMory:REFerence:SET

Function

This command saves the trace displayed on Scope as the reference trace.

Syntax

```
[:SENSe]:TMEMory:REFErence:SET
```

Example of Use

```
:TMEM:REF:SET
```

:TMEMory:REFerence:CLEar

Function

This command deletes the reference trace of Scope.

Syntax

```
[:SENSe]:TMEMory:REFerence:CLEar
```

Example of Use

```
:TMEM:REF:CLE
```

:TMEMory:CHANnel

Function

This command sets and queries the channel saved as a reference trace of Scope.
When saving the reference trace, use:TMEMory:REFerence:SET.

Syntax

```
[:SENSe]:TMEMory:CHANnel BOTH|CHA|CHB
[:SENSe]:TMEMory:CHANnel?
```

Parameter

BOTH	Channel A and Channel B
CHA	Channel A
CHB	Channel B

Response Data

Both | CHA | CHB

Example of Use

```
:TMEM:CHAN Both
:TMEM:CHAN?
>Both
```

3.9.3.9 Calibration

:CALibrate:TEMPerature?

Function

This command queries the current temperature and temperature during the calibration on the Scope module.

Syntax

```
:CALibrate:TEMPerature?
```

Response Data

<numeric>,<numeric>

Current temperature (°C), Temperature during calibration (°C)

Example of Use

```
:CAL:TEMP?
>39.6,24.4
```

:CALibrate:AMPLitude?

Function

This command initiates an amplitude calibration for Scope Channel A and B.

Syntax

```
:CALibrate:APPLitude[?]
```

For Version 3.2 or later, this command can be used as a query by appending "?".

Response Data

One of the following results is returned when the calibration ends.

"Calibration complete."	Calibration has successfully completed.
"Calibration Failed.(CHA)"	Calibration of channel A has failed.
"Calibration Failed.(CHB)"	Calibration of channel B has failed.
"Calibration Failed.(CHA&CHB)"	Calibration of channels A and B has failed.

Example of Use

```
:CAL:AMPL
```

(Time-waiting process of approx. 60 s)

```
>"Caribration complete."
```

Note:

In EYE mode, calibration takes about 50 s.
Therefore, the timeout for the interface must be set to 60 s. If

Timeout is set to less than 60 s, a calibration result may not be read due to a timeout error occurring before outputting the result to an output queue.

3.9.3.10 Selftest

:CALibrate:APPLication

Function

This command starts the self-test of the Scope.
Also, this command queries the self-test result of the Scope.

Syntax

```
:CALibrate:APPLication
:CALibrate:APPLication?
```

Response Data

When the self-diagnostic is completed successfully, "`Self Test Passed!`" is returned.
When the self-test error occurs, the response message is not returned.

Example of Use

```
:CAL:APPL?
>"Self Test Passed!"
```

3.9.4 Time

3.9.4.1 Trigger Clock Rate (MP2100B, MP2110A)

:TIME:TRACking

Function

Added in Version 5: This command sets and queries the tracking for Bit Rate and Divide Ratio of Scope.

Syntax

```
[:SENSe]:TIME:TRACking <master>
[:SENSe]:TIME:TRACking?
```

Parameter

<master>

For MP2100B

OFF	Tracking Off
PPG1_SYNCOUT	PPG1 Bit Rate and Sync Output Divide Ratio
PPG1_USER	PPG1 Bit Rate and User Defined Divide Ratio
PPG2_SYNCOUT	PPG2 Bit Rate and Sync Output Divide Ratio
PPG2_USER	PPG2 Bit Rate and User Defined Divide Ratio

For MP2110A

OFF	Tracking Off
PPG_CLOCKOUT	PPG Bit Rate and Clock Output Divide Ratio
PPG_SYNCOUT	PPG Bit Rate and Sync Output Divide Ratio
PPG_USER	PPG Bit Rate and User Defined Divide Ratio
CRU	*Added in Version 6:* Recovered Clock Rate and 1/2 Divide Ratio

Response Data

<master>

Example of Use

```
:TIME:TRAC PPG_CLOCKOUT
:TIME:TRAC?
>PPG_CLOCKOUT
```

:TIME:TRACking:STATus?

Function

Added in Version 5: This command queries the tracking statuses for Bit Rate and Divide Ratio of Scope.

Syntax

```
[:SENSe]:TIME:TRACking:STATus?
```

Response Data

NO_ERROR	No error or Tracking Off
ERR_REFCLC	Tracking error. The Reference CLK setting of PPG is External. Change it to Internal.
ERR_CLKOUT	Tracking error. The Clock Out output of PPG is OFF.
ERR_CRU	*Added in Version 6:* Tracking error. :TIME:CRU is not in RECovery mode.

Example of Use

```
:TIME:TRAC:STAT?
>NO_ERROR
```

3.9.4.2 Trigger Clock Rate (MP210xx)

:CONFigure:TRACking:DRATe

Function

Only in MP210xA and MP2100B: This command sets and queries the On/Off state of tracking of bit rate and clock rate of Scope.

Syntax

```
:CONFigure:TRACking:DRATe <enable>
:CONFigure:TRACking:DRATe?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:CONF:TRAC:DRAT ON
:CONF:TRAC:DRAT?
>1
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command. Use ":TIME:TRACking".

:CONFigure:TRACking:DRATe:MASTer

Function

Only in MP210xA and MP2100B: This command sets and queries the synchronization source for tracking of bit rate and clock rate of Scope.

Syntax

```
:CONFigure:TRACking:DRATe:MASTer 0|1|2|3
:CONFigure:TRACking:DRATe:MASTer?
```

Parameter

0 PPG1 Bit Rate and Sync Output Divide Ratio
1 ED1 Bit Rate and Sync Output Divide Ratio
2 PPG2 Bit Rate and Sync Output Divide Ratio
3 ED2 Bit Rate and Sync Output Divide Ratio

Response Data

0|1|2|3

Example of Use

```
:CONF:TRAC:DRAT:MAST 2
:CONF:TRAC:DRAT:MAST?
>2
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command. Use ":TIME:TRACking".

3.9.4.3 Setting

:TIME:ACQClock?

Function

This command sets Trigger Clock Rate automatically.

Syntax

```
[:SENSe]:TIME:ACQClock?
```

Response Data

<integer> MHz

Detected Clock Rate value

Example of Use

```
:TIME:ACQC?
>100.000 MHz
```

Note:

This command detects the frequency of the signal input to the Trigger Clk In connector and change the Clock Rate setting. To read the Clock Rate setting value only, use ":TIME:CLKRate" instead of this command.

:TIME:CLKRate

Function

This command sets and queries the clock rate of Scope.
When changing the clock rate, the bit rate is changed to the value multiplexed clock rate by divide ratio.

Syntax

```
[:SENSe]:TIME:CLKRate <numeric> [GHz|MHz|kHz]
[:SENSe]:TIME:CLKRate?
```

Parameter

<numeric>
Clock rate
GHz, MHz, or kHz can be used as the unit.
When omitting the unit, the unit is fixed to MHz.

Response Data

<numeric> MHz

Example of Use

To set the clock rate of the Scope to 10312.5 MHz

```
:TIME:CLKR 10312.5
:TIME:CLKR?
>10312.50 MHz
```

Note:

In MP2110A, when Precision Trigger is set to On, the lower limit of Clk Rate is 2.4 GHz.

:TIME:DATRate

Function

This command sets and queries the bit rate of Scope (symbol rate for MP2110A).
When changing the bit rate, the clock rate is changed to the value multiplexed bit rate by divide Ratio.

Syntax

```
[:SENSe]:TIME:DATRate <numeric>[Gbps|kbps|Mbps]
[:SENSe]:TIME:DATRate?
```

Parameter

<numeric>
Bit rate
Gbps, Mbps, or kbps can be used as the unit. When omitting the unit, the unit is fixed to Mbps.

Response Data

<numeric> Mbps

Example of Use

To set the bit rate of the Scope to 155220 kbit/s

```
:TIME:DATR 155220kbps
:TIME:DATR?
>155.220 Mbps
```

:TIME:DIVRatio

Function

This command sets and queries the clock divide ratio of Scope.
When changing the clock divide ratio, either of the bit rate (symbol rate for MP2110A) or clock frequency is changed.

Syntax

```
[:SENSe]:TIME:DIVRatio <integer>,{CLKR|DATA}
[:SENSe]:TIME:DIVRatio?
```

Parameter

<integer>

Divide ratio	1 to 99 When Sampling mode is **Pulse** or **CoherentEye**, it operates only when 1, 2, 4, 8, 16, 32, 40, 48, or 64 is set. (48 is added in Version 6)
DATR	The bit rate is re-calculated from the divide ratio and clock frequency.
CLKR	The clock frequency is re-calculated from the divide ratio and bit rate.

Response Data

<integer>

Divide ratio 1 to 99

Example of Use

To set the 1/16 value of the clock frequency to the bit rate:

```
:TIME:DIVR 16,CLKR
```

:TIME:AUTodetect

Function

This command sets and queries the On/Off state of auto detection of the clock divide ratio (1/n) input to the **Trigger Clk In** connector of Scope.

Syntax

```
[:SENSe]:TIME:AUTodetect ON|OFF
[:SENSe]:TIME:AUTodetect?
```

Parameter

0|OFF
1|ON

Response Data

ON|OFF

Example of Use

```
:TIME:AUT ON
:TIME:AUT?
>ON
```

:TIME:PTRigger

Function

Only in MP2110A: This command sets and queries the On/Off state of Precision Trigger for Scope.

Syntax

```
[:SENSe]:TIME:PTRigger <enable>
[:SENSe]:TIME:PTRigger?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:TIME:PTR 1
:TIME:PTR?
>1
```

Note:

This setting is enabled when the Option 024 is installed.

:TIME:PTRigger:RESet

Function

Only in MP2110A: This command resets Precision Trigger of Scope.

Syntax

```
[:SENSe]:TIME:PTRigger:RESet
```

Example of Use

```
:TIME:PTR:RES
```

Note:

Use this command when the PT phase unlock (bit6) is detected by :INSTrument:WAV:CONDition?.

3.9.4.4 Scale/Offset

:DISPlay:WINDow:X:UNIT

Function

This command sets and queries the unit of the horizontal scale of Scope.

Syntax

```
:DISPlay:WINDow:X[:SCALe]:UNIT PS|UI
:DISPlay:WINDow:X[:SCALe]:UNIT?
```

Parameter

PS: Sets the unit to picosecond (10^{-12} seconds).
UI: Sets the unit to Unit Interval.

Response Data

PS|UI

Example of Use

```
:DISP:WIND:X:UNIT UI
:DISP:WIND:X:UNIT?
>UI
```

:DISPlay:WINDow:X:BITs

Function

This command sets and queries the horizontal scale of Scope, by the number of bits.

Syntax

```
:DISPlay:WINDow:X[:SCALe]:BITs <integer>
:DISPlay:WINDow:X[:SCALe]:BITs?
```

Parameter

<integer>
Eye mode, Coherent Eye mode:
MP210xx 1 to 1000, 1 bit step
(If Fast Sampling Mode is On, the range is 1 to 100.)
MP2110A 1 to 100, 1 bit step
Pulse mode: 1 to 65535, 1 bit step

Response Data

<integer>

Example of Use

```
:DISP:WIND:X:BIT 2
```

```
:DISP:WIND:X:BIT?
>2
```

:DISPlay:WINDow:X:OFFSets

Function

This command sets and queries the offset value of the horizontal scale of the Scope.

Syntax

```
:DISPlay:WINDow:X[:SCALe]:OFFSets <numeric>
:DISPlay:WINDow:X[:SCALe]:OFFSets?
```

Parameter

<numeric>
Offset value
Unit UI or picosecond (ps)
The unit can be set using the :DISPlay:WINDow:X:UNIT.
When unit is UI: 0 to 32768
When unit is ps: Offset (UI) / Date-Rate (Tbps or 1000 Gbps)

Response Data

<numeric>

Example of Use

```
:DISP:WIND:X:OFFS 150
```

3.9.4.5 Test Pattern / Pattern Length

:TIME:PATTern:TYPE

Function

Added in Version 6: This command sets and queries input signal pattern type to be measured in Pulse CoherentEye mode.

Syntax

```
:TIME:PATTern:TYPE VARiable|PRBS{7|9|13|15}|SSPRQ
:TIME:PATTern:TYPE?
```

Parameter

VARiable
PRBS7
PRBS9
PRBS13
PRBS15
SSPRQ

If the desired pattern is not included in the options, select VARiable (Variable Length) and specify a pattern length by the :TIME:PATLength command.
However, the TDECQ measurement cannot be performed when VARiable is selected.
Also, if a type other than VARiable is selected, the sampling speed is faster.

Response Data

VARiable|PRBS{7|9|13|15}|SSPRQ

Example of Use

```
:TIME:PATT:TYPE PRBS15
:TIME:PATT:TYPE?
>PRBS15
```

:TIME:PATLength

Function

This command sets and queries the data pattern length used in the pulse or CoherentEye mode of the Scope.
Modified in Version 6: This setting is valid when TIME:PATTern:TYPE is VARiable.

Syntax

```
[:SENSe]:TIME:PATLength <numeric>
```

```
[:SENSe]:TIME:PATLength?
```

Parameter

<numeric>

Range	*Version 5 or older:*	1 to 16777216
	Version 6 or later:	2 to 32768

Response Data

<integer>

Version 5 or older:	1 to 16777216
Version 6 or later:	2 to 32768

Example of Use

```
:TIME:PATL 511
:TIME:PATL?
>511
```

:CONFigure:TRACking:PATLength

Function

This command sets and queries the On/Off state of pattern length tracking of Scope.

Syntax

```
:CONFigure:TRACking:PATLength <enable>
:CONFigure:TRACking:PATLength?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:CONF:TRAC:PATL OFF
:CONF:TRAC:PATL?
>0
```

:CONFigure:TRACking:PATLength:MASTer

Function

This command sets and queries the synchronization source for pattern length tracking of Scope.

Syntax

```
:CONFigure:TRACking:PATLength:MASTer 0|1|2|3|4|5|6|7
:CONFigure:TRACking:PATLength:MASTer?
```

Parameter

0 PPG1
1 ED1
2 PPG2
3 ED2
4 PPG3
5 ED3
6 PPG4
7 ED4

Response Data

0|1|2|3|4|5|6|7

Example of Use

```
:CONF:TRAC:PATL:MAST 1
:CONF:TRAC:PATL:MAST?
>1
```

3.9.4.6 Skew

:CONFigure:SKEW:CHA|CHB

Function

This command sets and queries the skew on Scope.

Syntax

```
:CONFigure:SKEW:CHA|CHB <numeric>
:CONFigure:SKEW:CHA|CHB?
```

Parameter

<numeric>
The range is "±BIT/DATR/2 (ps)" calculated using the scale of the X axis (:DISP:WIND:X:BIT) and BitRate (:TIME:DATR).

Response Data

<numeric>

Example of Use

```
:CONF:SKEW:CHA 6.4
:CONF:SKEW:CHA?
>6.4
```

:CONFigure:SKEW:ALIGn

Function

Added in Version 5.02: This command sets and queries the On/Off state for automatically adjusting the Skew setting value to display the waveforms of both Channel A and Channel B in the center when executing Auto Scale.

Syntax

```
:CONFigure:SKEW:ALIGn <enable>
:CONFigure:SKEW:ALIGn?
```

Parameter

0 | OFF
1 | ON

Response Data

0 | 1

Example of Use

```
:CONF:SKEW:ALIG ON
:CONF:SKEW:ALIG?
>1
```

3.9.5 Amplitude, O/E

3.9.5.1 Scale/Offset

:DISPlay:WINDow:CHANnel:BOTH

Function

This command sets and queries the On/Off state of Channel A/B Tracking of Scope.

Syntax

```
:DISPlay:WINDow:CHANnel:BOTH <enable>
:DISPlay:WINDow:CHANnel:BOTH?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:DISP:WIND:CHAN:BOTH ON
:DISP:WIND:CHAN:BOTH?
>1
```

:DISPlay:WINDow:Y:DIVision:CHA|CHB

Function

This command sets and queries the unit of the vertical scale of Scope.

Syntax

```
:DISPlay:WINDow:Y[:SCALe]:DIVision:CHA|CHB <numeric>
:DISPlay:WINDow:Y[:SCALe]:DIVision:CHA|CHB?
```

Parameter

<numeric>

Electrical channel:	1.0 to 200.0 × [10 (attenuation/20)] (mV)
Optical channel:	A value (μW) where the range of the electrical channel is divided by Conversion Gain (V/W)

Response Data

<numeric>

Example of Use

```
:DISP:WIND:Y:DIV:CHA?
>100.0
```

Notes:

- The settable maximum value varies depending on the attenuation set by :INPut:ATTenuation:CHA|CHB.
- For optical interface, the settable maximum value varies depending on the O/E conversion gain set by :CALibrate:CGain.

:DISPlay:WINDow:Y:OFFSets:CHA|CHB

Function

This command sets and queries the offset value for the vertical scale of Scope.

Syntax

```
:DISPlay:WINDow:Y[:SCALe]:OFFSets:CHA|CHB <numeric>
:DISPlay:WINDow:Y[:SCALe]:OFFSets:CHA|CHB?
```

Parameter

<numeric>

Electrical channel:	–500.0 to +500.0 (mV)
Optical channel:	A value (mW) where the range of the electrical channel is divided by Conversion Gain (V/W)

Response Data

<numeric>

Example of Use

```
:DISP:WIND:Y:OFFS:CHB?
>-50
```

Notes:

- The settable maximum value varies depending on the attenuation set by :INPut:ATTenuation:CHA|CHB.
- For optical interface, the settable maximum value varies depending on the O/E conversion gain set by :CALibrate:CGain.

:INPut:ATTenuation:CHA|CHB

Function

This command sets and queries the amount of attenuation for adjusting the Scope amplitude scale.

Syntax

```
[:SENSe]:INPut:ATTenuation:CHA|CHB <numeric>
[:SENSe]:INPut:ATTenuation:CHA|CHB?
```

Parameter

<numeric>

0.00 to 30.00 (dB)

Response Data

<numeric>

Example of Use

```
:INP:ATT:CHA 20.00
```

3.9.5.2 Channel Math

:CALCulate:CHANnel:MATH

Function

This command sets and queries the calculation between channels of Scope.

Syntax

```
:CALCulate:CHANnel:MATH <enable>
:CALCulate:CHANnel:MATH?
```

Parameter

0|OFF
1|ON

Response Data

1|0

Example of Use

```
:CALC:CHAN:MATH OFF
:CALC:CHAN:MATH?
>0
```

:CALCulate:CHANnel:MATH:DEFine

Function

This command sets and queries the calculation formula between channels of Scope.

Syntax

```
:CALCulate:CHANnel:MATH:DEFine 0|1|2
:CALCulate:CHANnel:MATH:DEFine?
```

Parameter

0: Channel A + Channel B
1: Channel A – Channel B
2: Channel B – Channel A

Response Data

0|1|2

Example of Use

```
:CALC:CHAN:MATH:DEF?
>1
```

:DISPlay:WINDow:Y:DIVision:CHMath

Function

This command sets and queries the value (mV) of the vertical scale of Scope when Channel Math is On.

Syntax

```
:DISPlay:WINDow:Y[:SCALe]:DIVision:CHMath <numeric>
:DISPlay:WINDow:Y[:SCALe]:DIVision:CHMath?
```

Parameter

<numeric>
Range 1 to 200 mV, 0.1 mV step

Response Data

<numeric>

Example of Use

```
:DISP:WIND:Y:DIV:CHM?
>100
```

Note:

The settable maximum value varies depending on the attenuation set by :INPut:ATTenuation:CHA|CHB.

:DISPlay:WINDow:Y:OFFSets:CHMath

Function

This command sets and queries the offset value (mV) of the vertical scale of Scope when Channel Math is On.

Syntax

```
:DISPlay:WINDow:Y[:SCALe]:OFFSets:CHMath <numeric>
:DISPlay:WINDow:Y[:SCALe]:OFFSets:CHMath?
```

Parameter

<numeric>

Range –1000.0 to +1000.0 mV, 0.1 mV step

Response Data

<numeric>

Example of Use

```
:DISP:WIND:Y:OFFS:CHM?
>-50
```

Note:

The settable maximum value varies depending on the attenuation set by :INPut:ATTenuation:CHA|CHB.

3.9.5.3 O/E

:CALibrate:OEPower[:JUDGe]

Function

This command judges if the O/E converter can be calibrated, and queries the result of judgment.

Syntax

```
:CALibrate:OEPower
:CALibrate:OEPower:JUDGe        Added in Version 3.01.09
:CALibrate:OEPower:JUDGe?       Added in Version 3.01.09
:CALibrate:OEPower?             Added in Version 5
```

Response Data

Pass | Fail

Pass	The O/E converter can be calibrated.
Fail	The O/E converter cannot be calibrated. (Optical input level > –30 dBm)

Example of Use

```
:CAL:OEP:JUDG
:CAL:OEP:JUDG?
>Pass
```

3.9.5.4 Conversion Gain/Responsivity

:INPut:WAVLength[:CHA|CHB]

Function

This command sets and queries the optical wavelength input to the O/E converter.

Syntax

```
[:SENSe]:INPut:WAVLength[:CHA|CHB] <character>
[:SENSe]:INPut:WAVLength[:CHA|CHB]?
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, CHB is specified.

Parameter

<character>

850	850 nm
USER_MMF	Defined by users (MMF) (*Only in MP2110A*)
1310	1310 nm
1550	1550 nm
USER	Defined by users (SMF) *Added in Version 3.01.13*

Response Data

850|USER_MMF|1310|1550|USER

Example of Use

To set the ChB O/E converter wavelength to 1550 nm

```
:INP:WAVL 1550
:INP:WAVL?
>1550
```

:CALibrate:CGain[:CHA|CHB]

Function

This command sets and queries the O/E Conversion Gain.

Syntax

```
:CALibrate:CGain[:CHA|CHB] <numeric>
:CALibrate:CGain[:CHA|CHB]?
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, CHB is specified.

Parameter

<numeric>

1 to 9999 (V/W)

Response Data

<integer>

Example of Use

```
:CAL:CG 320
:CAL:CG?
>320
```

Note:

Only in MP2110A: When Wavelength is not User and the O/E Conversion Gain is set using this command, the user is automatically changed to a user who is corresponding to the wavelength currently selected.

:CALibrate:SYSTem:CGain

Function

Only in MP210xA and MP2100B: This command sets and queries the System Conversion Gain for the O/E converter.

Syntax

```
:CALibrate:SYSTem:CGain <numeric>
:CALibrate:SYSTem:CGain?
```

Parameter

<numeric>
1 to 9999 (V/W)

Response Data

<integer>

Example of Use

```
:CAL:SYST:CG?
>160
```

:CALibrate:RESPonsivity[:CHA|CHB]

Function

This command sets and queries Responsivity of the O/E converter.

Syntax

```
:CALibrate:RESPonsivity[:CHA|CHB] <numeric>
:CALibrate:RESPonsivity[:CHA|CHB]?
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, CHB is specified.

Parameter

<numeric>
0.001 to 65.535 (A/W)

Response Data

<numeric>

Example of Use

```
:CAL:RESP?
>0.853
```

Note:

Only in MP2110A: When Wavelength is not User and Responsivity is set using this command, the user is automatically changed to a user who is corresponding to the wavelength currently selected.

:CALibrate:AUTocorrect[:CHA|CHB]

Function

Added in Version 3.01.13: This command automatically adjusts the values of Conversion Gain, Responsivity, and System Conversion Gain so that they become equal to the values measured by the optical power meter.

Syntax

```
:CALibrate:AUTocorrect[:CHA|CHB]
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, CHB is specified.

Example of Use

To set User at Wavelength:

```
:INP:WAVL USER
```

To measure the unmodulated optical signal power by the optical power meter and set the measured value:

```
:CAL:CALP -7.00
:CAL:CALP?
>-7.00
```

To input the optical signal to the optical input connector and automatically adjust the values:

```
:CAL:AUT
```

Note:

Perform the following settings before adjusting the values automatically using this command.

- Set User at Wavelength.
- Set the result of measuring the unmodulated optical signal power by the optical power meter to :CALibrate:CALPower.

:CALibrate:CALPower[:CHA|CHB]

Function

Added in Version 3.01.13: This command sets and queries the optical power measurement value used for adjusting the values automatically by `:CALibrate:AUTocorrect`.

Syntax

```
:CALibrate:CALPower[:CHA|CHB] <numeric>
:CALibrate:CALPower[:CHA|CHB]?
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, `CHB` is specified.

Parameter

<numeric>

–10 to –2 dBm

Example of Use

Refer to “Example of Use” of :CALibrate:AUTocorrect.

3.9.5.5 Filter

:FILTer[:CHA|CHB]

Function

Added in Version 5: This command sets and queries the internal low-pass filter.

Syntax

```
[:SENSe]:FILTer[:CHA|CHB] {"NO_FILTER"|"<standard>"}
[:SENSe]:FILTer[:CHA|CHB]?
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, CHB is specified.

Parameter

Table 3.9.5.5-1 Filter Settings List (MP2110A)

"<standard>"	Standard	Bit Rate (bit/s)	Remarks
"100GE_4"	100GbE/4	25.78125G	
"100GE_4_FEC"	100GbE/4 FEC	27.7393G	
"OTU-4"	OTU4	27.952493G	
"32G_FC"	32GFC	28.05G	
"400GE_8"	400GbE/8	19.34GHz	*
"400GE_8_SMF"	400GbE/8(SMF)	13.3GHz	*
"400GE_8_MMF"	400GbE/8(MMF)	12.6GHz	*

*: *Added in Version 6:* Digital filter for TDECQ measurement. It is selectable under the following conditions.

- Signal Type: PAM4
- Sampling Mode: Coherent Eye
- Pattern Length Type: Other than Variable

Table 3.9.5.5-2 Filter Settings List (MP2100B)

<string>	Standard	Bit Rate (bit/s)	Remarks
"10G_FC_FEC"	10GFC FEC	11.3168G	
"10G_OTU2E"	OTU2e (10GbE FEC)	11.095728G	
"OTU-2"	OTU2	10.709225G	
"OC-192FEC"	G.975 FEC	10.664228G	
"10G_FC"	10GFC	10.51875G	

Table 3.9.5.5-2 Filter Settings List (MP2100B) (Cont'd)

<string>	Standard	Bit Rate (bit/s)	Remarks
"10G_LAN"	10GbE LAN/PHY	10.3125G	
"INF10G"	InfiniBand x4	10G	
"OC-192"	OC-192/STM-64	9.95328G	
"10G_WAN"	10GbE WAN/PHY	9.95328G	
"8G_FC"	8GFC	8.5G	
"6_3G"	fc = 6.3GHz	6.3G	
"XAUI-2"	XAUI Optical x2	6.25G	
"CPRI-10"	CPRI x10	6.144G	
"INF5G"	InfiniBand x2	5G	
"CPRI-8"	CPRI x8	4.9515G	
"4G_FC"	4GFC	4.25G	
"10G_FC_LX4"	10GFC-LX4	3.1875G	
"10GBASE_LX4"	10GBASE-LX4	3.125G	
"CPRI-5"	CPRI x5	3.072G	
"OTU-1"	OTU1	2.666057G	
"2GBE"	2GbE	2.5G	
"INF"	InfiniBand	2.5G	
"OC-48"	OC-48/STM16	2.488G	
"CPRI-4"	CPRI x4	2.4576G	
"2G_FC"	2GFC	2.125G	
"1GBE"	1GbE	1.25G	
"OC-24"	OC-24	1.244G	
"CPRI-2"	CPRI x2	1.2288G	
"1G_FC"	1GFC	1.0625G	
"OC-12"	OC-12/STM-4	622.08M	
"CPRI"	CPRI	614.4M	
"OC-3"	OC-3/STM-1	155.22M	

Response Data

{"NO_FILTER"|"<standard>"}

Example of Use

To set 100GbE/4 to the filter:

```
:FILT "100GE_4"
:FILT?
>"100GE_4"
```

:INPut:FILTer:ENABle

Function

Only in MP210xA and MP2100B: Added in Version 3.01.13: This command sets and queries the On/Off state of the Option 086 filter.

Syntax

```
[:SENSe]:INPut:FILTer:ENABle 0|1
[:SENSe]:INPut:FILTer:ENABle?
```

Parameter

0 Sets the Filter to Off.
1 Sets the Filter to On.

Response Data

0|1

Example of Use

To set the filter to Off.

```
:INPut:FILTer:ENABle 0
:INPut:FILTer:ENABle?
>0
```

Note:

This command can be used only when the Option 086 of MP210xA or MP2100B is installed. In MP2110A, this command cannot be used, so use the command :FILTer[:CHA|CHB].
Perform Amplitude Calibration before using this command.

:INPut:FILTer

Function

Only in MP210xA and MP2100B: This command sets and queries the internal low-pass filter.

Syntax

```
[:SENSe]:INPut:FILTer <integer>
[:SENSe]:INPut:FILTer?
```

Parameter

Table 3.9.5.5-3 Filter Settings List (MP210xA, MP2100B)

<integer>	Standards	Bit Rate (bit/s)
0	None	-
1	2GFC	2.125G
2	4GFC	4.25G
3	fc = 6.3GHz	6.3G
4	10GFC	10.51875G
5	10GbE WAN	9.95328G
6	10GbE LAN/PHY	10.3125G
7	OC192/STM-64	9.95328G
8	G.975 FEC	10.664228G
9	OTU-2	10.709225G
10	1GFC	1.0625G
11	10GFC FEC	11.3168G
12	1GbE	1.25G
13	2GbE	2.5G
14	10GbE FEC	11.095728G
15	Infiniband Optical	2.5G
16	Infiniband Optical x2	5G
17	Infiniband Optical x4	10G
18	OC3/STM-1	155.52M
19	OC12/STM-4	622.08M
20	OC24	1.244G
21	OC48/STM-16	2.488G
22	OTU-1	2.66648G
23	CPRI	614.4M
24	CPRI x2	1.2288G
25	CPRI x4	2.4576G
26	CPRI x5	3.072G
27	CPRI x8	4.9515G
28	CPRI x10	6.144G
29	10GBASE-LX4	3.125G

Table 3.9.5.5-3 Filter Settings List (MP210xA, MP2100B) (Cont'd)

<integer>	Standards	Bit Rate (bit/s)
30	10GFC-LX4	3.1875G
31	XAUI Optical x2	6.25G
32	8GFC	8.5G

Response Data

<integer>

Example of Use

To set the filter to OC192/STM-64:

```
:INP:FILT 7
:INP:FILT?
>7
```

Note:

In MP2110A, this command cannot be used, so use the command :FILTer[:CHA|CHB].

3.9.5.6 Extinction Ratio Correction

:CONFigure:EXRCorrection[:CHA|CHB]

Function

This command sets and queries the On/Off state of the extinction ratio correction of the O/E converter.

Syntax

```
:CONFigure:EXRCorrection[:CHA|CHB] 0|1
:CONFigure:EXRCorrection[:CHA|CHB]?
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, CHB is specified.

Parameter

0 Off

1 On

Response Data

0|1

Example of Use

```
:CONF:EXRC 1
:CONF:EXRC?
>1
```

:CONFigure:EXRCorrection:FACTor[:CHA|CHB]

Function

This command sets and queries the extinction ratio correction factor for the O/E converter.

Syntax

```
:CONFigure:EXRCorrection:FACTor[:CHA|CHB] <numeric>
:CONFigure:EXRCorrection:FACTor[:CHA|CHB]?
```

[:CHA|CHB] can be specified only in MP2110A. When nothing is specified, CHB is specified.

Parameter

<numeric>
–9.99 to 9.99%, 0.01 step

Response Data

<numeric>

Example of Use

To set Ch B extinction ratio correction factor to 1.2%

```
:CONF:EXRC:FACT 1.20
:CONF:EXRC:FACT?
>1.20
```

3.9.6 Measure

3.9.6.1 Setting Measurement Items

:CONFigure:MEASure:DISPlay:ADD

Function

Added in Version 6: This command adds the measurement items for the Amplitude/Time measurement.

Syntax

```
:CONFigure:MEASure:DISPlay:ADD CHA|CHB,<meas_item>
```

Parameter

CHA: Channel A
CHB: Channel B
<meas_item>

For NRZ

```
<meas_item>={AVEPower{:DBM|MW}|CROSsing|EXTRatio|EYEAmpl
itude|EYEHeight|LEVel:ONE|LEVel:ZERO|OMA:DBM|OMA:MW|OMAX
p|SNR|VECP|DCD|EYEWidth|FTIMe|JITTer:PPeak|JITTer:RMS|TR
ISe}
```

For PAM4

```
<meas_item>=[:TEQualizer]:{AVEPower|EYE:HEIGht|EYE:LEVEL
|LEVEL|LEVEL:PPeak|LEVEL:RMS|LINearity|OER|OOMA|TDECQ|EY
E:SKEW|EYE:WIDTh|LEVEL:SKEW }
```

For Jitter (OPT096)

```
<meas_item>=[:AJITter]:{DJ|EOPening|J2|J9|RJ|TJ}|:AJITte
r:{DCD|DDJ|DDPWS|ISI|PJ[:FREQuency]|RJ[:RMS]}
```

Example of Use

```
:CONF:MEAS:DISP:ADD CHA,LEVel:ZERO
```

:CONFigure:MEASure:DISPlay:ADELete

Function

Added in Version 6: This command deletes all measurement items for the Amplitude/Time measurement.

Syntax

```
:CONFigure:MEASure:DISPlay:ADELete
```

Example of Use

```
:CONF:MEAS:DISP:ADEL
```

:CONFigure:MEASure:AMPTIME{1|2|3|4}

Function

Only in MP210xA and MP2100B: This command sets and queries the amplitude/time measurement items to be displayed in the Scope screen.

Syntax

```
:CONFigure:MEASure:AMPTIME{1|2|3|4} {CHA|CHB},<integer>
:CONFigure:MEASure:AMPTIME{1|2|3|4}?
```

Parameter

1|2|3|4

CHA: Channel A
CHB: Channel B

<integer> Measurement item

0	One Level
1	Zero Level
2	Eye Amplitude
3	Eye Height
4	Crossing
5	SNR
6	Average Power (dBm)
7	Average Power (mW)
8	Extinction Ratio
9	Jitter p-p
10	Jitter RMS
11	Rise Time
12	Fall Time
13	Eye Width
14	DCD
15	OMA (mW)
16	OMA (dBm)

Response Data

{CHA|CHB|N/A},{<integer>|N/A}

N/A No measured items

Example of Use

To display the following measurement results on the screen:

- Channel A jitter (p-p)
- Channel A jitter (RMS)
- Channel A Crossing
- Channel A eye amplitude

```
:CONF:MEAS:AMPTIME1 CHA,9
:CONF:MEAS:AMPTIME2 CHA,10
:CONF:MEAS:AMPTIME3 CHA,4
:CONF:MEAS:AMPTIME4 CHA,2
```

To query the measurement result displayed on the screen:

```
:CONF:MEAS:AMPTIME1?
>CHA,9
:CONF:MEAS:AMPTIME2?
>CHA,10
:CONF:MEAS:AMPTIME3?
>CHA,4
:CONF:MEAS:AMPTIME4?
>CHA,2
```

:CONFigure:MEASure:AREa:DISPlay

Function

This command sets and queries the Amplitude/Time measurement area display of Scope.

Syntax

```
:CONFigure:MEASure:AREa:DISPlay <enable>
:CONFigure:MEASure:AREa:DISPlay?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:CONF:MEAS:ARE:DISP ON
:CONF:MEAS:ARE:DISP?
>1
```

:CONFigure:MEASure:AREa:ITEM

Function

This command sets and queries the measurement item number displayed in the Amplitude/Time measurement area of Scope.

Syntax

```
:CONFigure:MEASure:AREa:ITEM <integer>
:CONFigure:MEASure:AREa:ITEM?
```

Parameter

<integer>

MP210xA, MP2100B: 1 to 4

MP2110A: 1 to 32

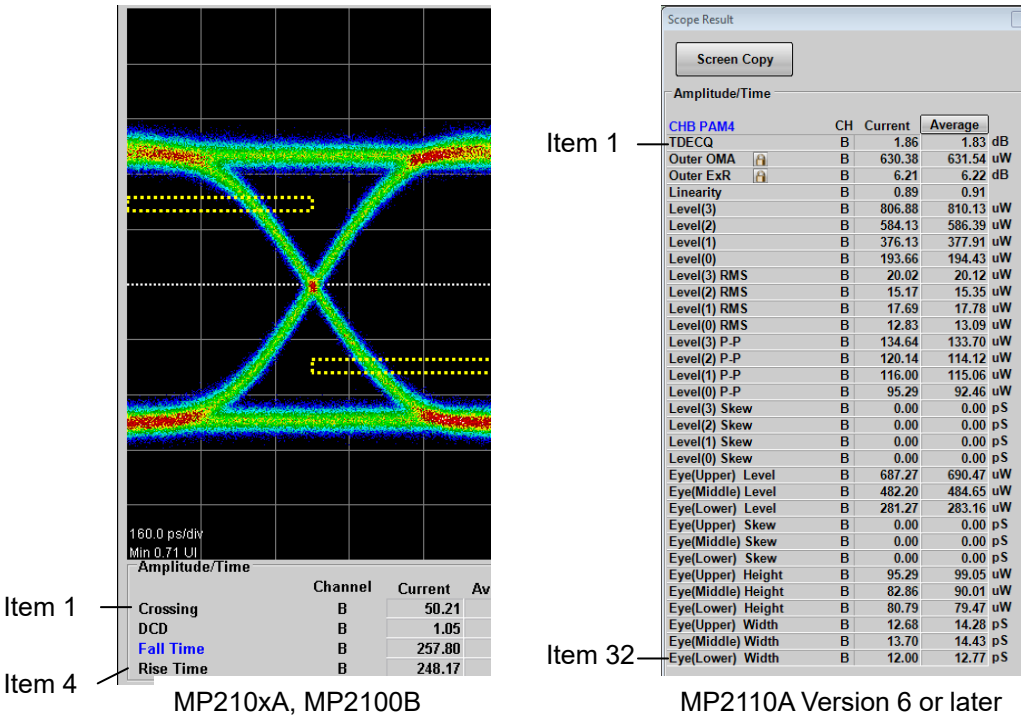


Figure 3.9.6.1-1 Measurement Item Number

Response Data

<integer>

Example of Use

```
:CONF:MEAS:ARE:ITEM 4
:CONF:MEAS:ARE:ITEM?
>4
```

3.9.6.2 Amplitude/Time (PAM4)

:CONFigure:MEASure:PAM:TIMing

Function

Added in Version 6: This command sets and queries the phase reference for each Eye of PAM4 signal.

Syntax

```
:CONFigure:MEASure:PAM:TIMing TRACk|INDependent
:CONFigure:MEASure:PAM:TIMing?
```

Parameter

TRACk	Track to Middle Eye Timing
INDependent	Independent Timing

Response Data

TRACk|INDependent

Example of Use

```
:CONF:MEAS:PAM:TIM TRACk
:CONF:MEAS:PAM:TIM?
>TRACk
```

:CONFigure:MEASure:PAM:CENTer

Function

Added in Version 6: This command sets and queries the reference for Eye Center Type of PAM4 signal.

Syntax

```
:CONFigure:MEASure:PAM:CENTer HEIGht|WIDTh
:CONFigure:MEASure:PAM:CENTer?
```

Parameter

HEIGht	Maximum Eye Height
WIDTh	Maximum Eye Width

Response Data

HEIGht|WIDTh

Example of Use

```
:CONF:MEAS:PAM:CENT HEIGht
:CONF:MEAS:PAM:CENT?
>HEIGht
```

:CONFigure:MEASure:PAM:EOPening

Function

Added in Version 6: This command sets and queries Eye Opening Definition in Eye Heights/Widths of PAM4 signal.

Syntax

```
:CONFigure:MEASure:PAM:EOPening ZERO|E_{1|2|3|4|5|6}
:CONFigure:MEASure:PAM:EOPening?
```

Parameter

ZERO	Zero Hits
E_1	1E–01
E_2	1E–02
E_3	1E–03
E_4	1E–04
E_5	1E–05
E_6	1E–06

Response Data

ZERO|E_{1|2|3|4|5|6}

Example of Use

```
:CONF:MEAS:PAM:EOP ZERO
:CONF:MEAS:PAM:EOP?
>ZERO
```

:CONFigure:MEASure:PAM:TEQualizer:CHA|CHB

Function

Added in Version 6: This command sets and queries the On/Off state of TDECQ Equalizer of PAM4 signal.

Syntax

```
:CONFigure:MEASure:PAM:TEQualizer:CHA|CHB <enable>
:CONFigure:MEASure:PAM:TEQualizer:CHA|CHB?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:CONF:MEAS:PAM:TEQ:CHA ON
```

```
:CONF:MEAS:PAM:TEQ:CHA?
>1
```

:CONFigure:MEASure:PAM:TEQualizer:DISPlay:CHA|CHB

Function

Added in Version 6: This command sets and queries Display Equalized Waveform after TDECQ Equalizer is applied.

Syntax

```
:CONFigure:MEASure:PAM:TEQualizer:DISPlay:CHA|CHB
<enable>
:CONFigure:MEASure:PAM:TEQualizer:DISPlay:CHA|CHB?
```

Parameter

0|OFF
1|ON

Response Data

0|1
0 Displays the waveform before TDECQ Equalizer is applied.
1 Displays the waveform after TDECQ Equalizer is applied.

Example of Use

```
:CONF:MEAS:PAM:TEQ:DISP:CHB OFF
:CONF:MEAS:PAM:TEQ:DISP:CHB?
>0
```

:CONFigure:MEASure:PAM:TEQualizer:CALCulate:CHA|CHB

Function

Added in Version 6: This command sets and queries the optimal Tap value for TDECQ Equalizer of PAM4 signal.

Syntax

```
:CONFigure:MEASure:PAM:TEQualizer:CALCulate:CHA|CHB
```

:CONFigure:MEASure:PAM:TEQualizer:OPTimization:CHA|CHB

Function

Added in Version 6.00.45: Sets and queries how to calculate the optimal Tap coefficient for PAM4 TDECQ Equalizer.

Syntax

```
:CONFigure:MEASure:PAM:TEQualizer:OPTimization:CHA|CHB
AUTO|FAST
:CONFigure:MEASure:PAM:TEQualizer:OPTimization:CHA|CHB?
```

Parameter

AUTO | FAST

Response Data

AUTO | FAST

Example of Use

```
:CONF:MEAS:PAM:TEQ:OPT:CHB AUTO
:CONF:MEAS:PAM:TEQ:OPT:CHB?
>AUTO
```

:CONFigure:MEASure:PAM:TEQualizer:TAPS:COUNt:CHA|CHB

Function

Added in Version 6: This command sets and queries the Tap number of TDECQ Equalizer of PAM4 signal.

Syntax

```
:CONFigure:MEASure:PAM:TEQualizer:TAPS:COUNt:CHA|CHB
5|7|9
:CONFigure:MEASure:PAM:TEQualizer:TAPS:COUNt:CHA|CHB?
```

Parameter

5 | 7 | 9

Response Data

5 | 7 | 9

Example of Use

```
:CONF:MEAS:PAM:TEQ:TAPS:COUN:CHB 5
:CONF:MEAS:PAM:TEQ:TAPS:COUN:CHB?
>5
```

:CONFigure:MEASure:PAM:TEQualizer:TAPS:CHA|CHB

Function

Added in Version 6: This command sets and queries the Tap value of TDECQ Equalizer of PAM4 signal.

Syntax

```
:CONFigure:MEASure:PAM:TEQualizer:TAPS:CHA|CHB <string>
:CONFigure:MEASure:PAM:TEQualizer:TAPS:CHA|CHB?
```

Parameter

<string>

Response Data

<string>

Tap values up to six decimal places are returned separated by commas.

Example of Use

```
:CONF:MEAS:PAM:TEQ:TAPS:CHB "0,0.2,0.4,0.6,0.8"
:CONF:MEAS:PAM:TEQ:TAPS:CHB?
>"0.000000,0.200000,0.400000,0.600000,0.800000"
```

3.9.6.3 Amplitude/Time (Setting Rise/Fall Time Measurement)

:CONFigure:MEASure:DEFine

Function

This command sets and queries the level measuring the Rise/Fall time on Scope.

Syntax

```
:CONFigure:MEASure:DEFine 0|1
:CONFigure:MEASure:DEFine?
```

Parameter

0 20/80%
1 10/90%

Response Data

0|1

Example of Use

```
:CONF:MEAS:DEF 1
:CONF:MEAS:DEF?
>1
```

:CONFigure:MEASure:TRANsition:CORRection

Function

This command sets and queries the On/Off state of use of the correction factor for rise/fall time of Scope.

Syntax

```
:CONFigure:MEASure:TRANsiton:CORRection <enable>
:CONFigure:MEASure:TRANsiton:CORRection?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

To use the correction factor:

```
:CONF:MEAS:TRAN:CORR ON
:CONF:MEAS:TRAN:CORR?
>1
```

:CONFigure:MEASure:TRANsition:CORRect:FACTor

Function

This command sets and queries the correction factor for rise/fall time of Scope.

Syntax

```
:CONFigure:MEASure:TRANsiton:CORRect:FACTor <numeric>
:CONFigure:MEASure:TRANsiton:CORRect:FACTor?
```

Parameter

<numeric>
Correction factor
Range 0.0 to 9999.9

Response Data

<numeric>

Example of Use

```
:CONF:MEAS:TRAN:CORR:FACT 0.0
:CONF:MEAS:TRAN:CORR:FACT?
>0.0
```

3.9.6.4 Amplitude/Time (Setting Zero/One Level Measurement)

:CONFigure:MEASure:EYEBoundary:OFFSet

Function

This command sets and queries the horizontal position for 1 and 0 level measurement performed by Scope.

Syntax

```
:CONFigure:MEASure:EYEBoundary:OFFSet <numeric>
:CONFigure:MEASure:EYEBoundary:OFFSet?
```

Parameter

<numeric>
Position measuring level (UI)
Range 0.00 to 1.00

Response Data

<numeric>

Example of Use

```
:CONF:MEAS:EYEB:OFFS 0.3
:CONF:MEAS:EYEB:OFFS?
```

```
>0.3
```

:CONFigure:MEASure:EYEBoundary:WIDTh

Function

This command sets and queries the horizontal width measuring 1 level and 0 level on Scope.

Syntax

```
:CONFigure:MEASure:EYEBoundary:WIDTh <numeric>
:CONFigure:MEASure:EYEBoundary:WIDTh?
```

Parameter

<numeric>
Width measuring level (UI)
Range 0.00 to 1.00

Response Data

<numeric>

Example of Use

```
:CONF:MEAS:EYEB:WIDT 0.20
:CONF:MEAS:EYEB:WIDT?
>0.20
```

3.9.6.5 Mask Test

:CONFigure:MASK:RECall

Function

Added in Version 5: This command sets and queries the mask used for Scope.

Syntax

```
:CONFigure:MASK:RECall "<file_name>"
:CONFigure:MASK:RECall?
```

Parameter

"<file_name>"
Mask file names: The following files are pre-installed in MP2110A.
"100GbE-CLR4-FEC.txt"
"100GbE-CLR4.txt"
"100GbE-CWDM4.txt"
"100GbE-ER4_Tx.txt"
"100GbE-LR4_Tx.txt"
"100GbE-SR4_Rx.txt"
"100GbE-SR4_Tx.txt"
"32GFC_MM.txt"
"32GFC_SM.txt"
"8GFC_Elect_Rx.txt"
"8GFC_Elect_Tx.txt"
"HDMI_TP1.txt"
"HDMI_TP2.txt"
"InfiniBand_EDR _Cable_In_Limiting.txt"
"InfiniBand_EDR _Cable_Out_Limiting.txt"
"InfiniBand_EDR_Host_Out_Limiting.txt"
"InfiniBand_EDR_Stressed_In_Limiting.txt"
"OTU-4.txt"

Response Data

"<file_name>"

Example of Use

```
:CONF:MASK:REC "100GbE-LR4_Tx.txt"
:CONF:MASK:REC?
>"100GbE-LR4_Tx.txt"
```

:CONFigure:MASK:TYPe

Function

Only in MP210xA and MP2100B: This command sets and queries the type of the mask used for Scope.

Syntax

```
:CONFigure:MASK:TYPe <integer>[,<file_name>]
:CONFigure:MASK:TYPe?
```

Parameter

<integer>	Mask
–1	User Defined
0	1GFC
1	2GFC
2	4GFC
3	8GFC
4	8GFC_Elect_Rx
5	8GFC_Elect_Tx
6	10GFC
7	10GbE FEC
8	1GbE
9	2GbE
10	10GbE WAN
11	10GbE LAN/PHY
12	10GFC FEC
13	OC48/STM16
14	OTU-1
15	OC192/STM64
16	OC192/STM64 FEC(G.975)
17	OTU-2 1310nm
18	OTU-2 1550nm
19	OTU-2 1550nm Expand
20	OTU-2 Amplified

If –1 is set to <integer>, specify the file name in <file_name>.

Response Data

<integer>,"<file_name>"

Example of Use

To select Mask other than User Mask:

```
:CONF:MASK:TYP?
>8,""
```

To select "test.txt" as a user mask file:

```
:CONF:MASK:TYP -1,"test.txt"
:CONF:MASK:TYP?
```

```
>-1,"test.txt"
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command. Use ":CONFigure:MASK:RECall".

3.9.6.6 Mask Test (Mask Alignment)

:CONFigure:MASK:ALGorithm

Function

This command sets and queries the mask alignment method of Scope.

Syntax

```
:CONFigure:MASK:ALGorithm 0|2
:CONFigure:MASK:ALGorithm?
```

Parameter

0 Zero/One/Crossing: Mask alignment by detecting the intersection of Zero Level and One level

2 User Defined

Response Data

0|2

Example of Use

```
:CONF:MASK:ALG 2
:CONF:MASK:ALG?
>2
```

:CONFigure:MASK:UPDate

Function

This command updates the mask alignment of the mask test for Scope when Align Method is Zero/One/Crossing.

Syntax

```
:CONFigure:MASK:UPDate
```

Example of Use

```
:CONF:MASK:UPD
```

:CONFigure:MASK:USER:MARKer

Function

This command sets and queries the On/Off state of the Alignment Marker display when Align Method is User Defined.

Syntax

```
:CONFigure:MASK:USER:MARKer <enable>
:CONFigure:MASK:USER:MARKer?
```

Parameter

<enable>

0 Display Off

1 Display On

Response Data

0|1

Example of Use

```
:CONF:MASK:USER:MARK 1
:CONF:MASK:USER:MARK?
>1
```

:CONFigure:MASK:USER:LOCation:X1|XDELta

Function

This command sets and queries the position for the horizontal direction of the mask when Align Method is User Defined.

Syntax

```
:CONFigure:MASK:USER:LOCation:X1|XDELta <numeric>
:CONFigure:MASK:USER:LOCation:X1|XDELta?
```

Parameter

<numeric>

When X1 is specified:	Position of X1 (UI)
When XDELta is specified:	Difference between X1 and X2, Positive number (UI)

Response Data

<numeric>

Example of Use

To set the user alignment marker X1 to the position 0.25 UI and the difference between the user alignment markers X1 and X2 to the position 1 UI:

```
:CONF:MASK:USER:LOC:X1 0.25
:CONF:MASK:USER:LOC:X1?
>0.25
:CONF:MASK:USER:LOC:XDEL 1
:CONF:MASK:USER:LOC:XDEL?
>1.00
```

:CONFigure:MASK:USER:LOCation:Y1|YDELta

Function

This command sets and queries the position for the vertical direction of the mask when Align Method is User Defined.

Syntax

```
:CONFigure:MASK:USER:LOCation:Y1|YDELta <numeric>
:CONFigure:MASK:USER:LOCation:Y1|YDELta?
```

Parameter

<numeric>

When `Y1` is specified:	Position of Y1 (mV)
When `YDELta` is specified:	Difference between Y1 and Y2, Positive number (mV)

Response Data

<numeric>

Example of Use

To set the user adjustment marker Y1 to 10 mV and the user adjustment marker Y2 to –10 mV, respectively

```
:CONF:MASK:USER:LOC:Y1 10
:CONF:MASK:USER:LOC:Y1?
>10.00
:CONF:MASK:USER:LOC:YDEL 20
:CONF:MASK:USER:LOC:YDEL?
>20.00
```

3.9.6.7 Mask Test (Mask Margin)

:CONFigure:MASK:MARGin:CONTupdate

Function

This command sets and queries the mask margin updating method for the mask test of Scope.

Syntax

```
:CONFigure:MASK:MARGin:CONTupdate 0|1
:CONFigure:MASK:MARGin:CONTupdate?
```

Parameter

0 Updates the mask margin only once.
1 Updates the mask margin whenever measuring

Response Data

0|1

Example of Use

```
:CONF:MASK:MARG:CONT 0
:CONF:MASK:MARG:CONT?
>0
```

:CONFigure:MASK:MARGin

Function

This command sets and queries the mask margin for the test mask of the Scope.

Syntax

```
:CONFigure:MASK:MARGin <mask_margin>[%]
:CONFigure:MASK:MARGin?
```

Parameter

<mask_margin>[%]
Mask Margin
Range –100 to 100%
In version 5.00 or earlier, the value is an integer (<integer>). In version 5.01 or later, the value is a number up to first decimal place (<numeric>).

Response Data

<mask_margin>%

Example of Use

```
:CONF:MASK:MARG 10%
```

```
:CONF:MASK:MARG?
>10%
```

:MEASure:MASK:MARGin?

Function

This command queries the results after performing the mask margin measurement using **One Shot**.

Syntax

```
:MEASure:MASK:MARGin?
```

Response Data

<integer>
Range –100 to 100 %
In version 5.00 or earlier, the value is an integer (<integer>). In version 5.01 or later, the value is a number up to first decimal place (<numeric>).

Example of Use

```
:MEAS:MASK:MARG?
>12
```

:SAMPles:JUDGe:TYPE

Function

Added in Version 3.03/4.01: This command sets and queries the Margin Type for Mask Margin measurement.

Syntax

```
[:SENSe]:SAMPles:JUDGe:TYPE COUNt|RATE
[:SENSe]:SAMPles:JUDGe:TYPE?
```

Parameter

COUNt	Hit Count
RATE	Hit Ratio

Response Data

COUNt|RATE

Example of Use

To set the threshold for Mask Margin measurement to Hit Count 10:

```
:SAMP:JUDG:TYPE COUNt
:SAMP:JUDG 10
:SAMP:JUDG?
>10
```

To set the threshold for Mask Margin measurement to Hit Ratio 1E–5:

```
:SAMP:JUDG:TYPE RATE
:SAMP:JUDG:RATE E_5,1
:SAMP:JUDG:RATE?
>E_5,1
```

:SAMPles:JUDGe

Function

This command sets and queries the Hit Count for Mask Margin measurement.

Syntax

```
[:SENSe]:SAMPles:JUDGe <numeric>
[:SENSe]:SAMPles:JUDGe?
```

Parameter

<numeric>
Sample point counts in the mask area

Response Data

<integer>

Example of Use

See “Example of Use” of :SAMPles:JUDGe:TYPE

:SAMPles:JUDGe:RATE

Function

Added in Version 3.03/4.01: This command sets and queries the Hit Ratio for Mask Margin measurement.

Syntax

```
[:SENSe]:SAMPles:JUDGe:RATE E_<exponent>[,<mantissa>]
[:SENSe]:SAMPles:JUDGe:RATE?
```

Parameter

Parameter	Description	Range
<exponent>	Exponent of Hit Ratio	1 to 12
<mantissa>	Mantissa of Hit Ratio (1 if omitted)	1 to 9*1 1.0 to 9.9*2

*1: In version 5.00 or earlier, the value is an integer. (<integer>)

*2: In version 5.01 or later, the value is a number up to first decimal place. (<numeric>)

To set the Hit Ratio to 2×10^{-6}, specify the parameter as "E_6,2".

Response Data

E_<exponent>,<mantissa>

Example of Use

See "Example of Use" of :SAMPles:JUDGe:TYPE.

3.9.6.8 Mask Test (Mask Margin Area Restriction)

:CONFigure:MASK:AREa:RESTriction

Function

This command sets and queries the mask area restriction of Scope.

Syntax

```
:CONFigure:MASK:AREa:RESTriction <enable>
:CONFigure:MASK:AREa:RESTriction?
```

Parameter

0 | OFF
1 | ON

Response Data

0 | 1

Example of Use

```
:CONF:MASK:ARE:REST ON
:CONF:MASK:ARE:REST?
>1
```

:CONFigure:MASK:AREa:RESTriction:ANGLe

Function

This command sets and queries the angle restricting the mask area of Scope.

Syntax

```
:CONFigure:MASK:AREa:RESTriction:ANGLe <integer>
:CONFigure:MASK:AREa:RESTriction:ANGLe?
```

Parameter

<integer>
Angle restriction mask area
Range –90 to 90 degrees

Response Data

<integer>

Example of Use

```
:CONF:MASK:ARE:REST:ANGL -30
:CONF:MASK:ARE:REST:ANGL?
>-30
```

:CONFigure:MASK:AREa:RESTriction:WIDTh

Function

This command sets and queries the width restricting the mask area of Scope.

Syntax

```
:CONFigure:MASK:AREa:RESTriction:WIDTh <numeric>
:CONFigure:MASK:AREa:RESTriction:WIDTh?
```

Parameter

<numeric>
Width restricting mask area
Range 0.01 to 1.00 UI

Response Data

<numeric>

Example of Use

```
:CONF:MASK:ARE:REST:WIDT 0.15
:CONF:MASK:ARE:REST:WIDT?
>0.15
```

3.9.6.9 Histogram

:CONFigure:HISTogram:AXIS

Function

This command sets and queries the axis for histogram measurement of Scope.

Syntax

```
:CONFigure:HISTogram:AXIS TIME|AMPLitude
:CONFigure:HISTogram:AXIS?
```

Parameter

TIME	Time direction histogram
AMPLitude	Amplitude direction histogram

Response Data

TIME|AMPLitude

Example of Use

```
:CONF:HIST:AXIS AMPLitude
:CONF:HIST:AXIS?
>AMPLitude
```

:HISTogram:CENTer

Function

This command moves the marker position of the histogram measurement of Scope to the center of the screen.

Syntax

```
[:SENSe]:HISTogram:CENTer
```

Example of Use

```
:HIST:CENT
```

:HISTogram:X1|X2

Function

This command sets and queries the position of the marker X1 or X2 for setting the histogram measurement area of Scope.

Syntax

```
[:SENSe]:HISTogram:X1|X2 <numeric>
[:SENSe]:HISTogram:X1|X2?
```

Parameter

X1|X2 Marker

<numeric>
Marker position (Time)
Unit UI or ps
Range ps: 0 to 1271001
UI: 0 to 32768

Response Data

<numeric>

Example of Use

```
:HIST:X1 10050
:HIST:X1?
>10050
```

:HISTogram:Y1|Y2

Function

This command sets and queries the position of the marker Y1 or Y2 for setting the histogram measurement area of Scope.

Syntax

```
[:SENSe]:HISTogram:Y1|Y2 <numeric>
[:SENSe]:HISTogram:Y1|Y2?
```

Parameter

Y1|Y2 Marker

<numeric>
Marker position (Amplitude)
Unit mV (Electrical input)
μW (Optical input)
Range mV: Between minimum and maximum values for display area
μW: Between minimum and maximum values for display area
Minimum value: Offset – Scale × 5
Maximum value: Offset + Scale × 5

Response Data

<numeric>

Example of Use

```
:HIST:Y2 -60
:HIST:Y2?
>-60
```

3.9.7 Acquiring results

:FETCh:AMPTime:QUEStionableeye?

Function

This command queries if "EYE?", "NRZ? ", or "PAM4?" is displayed on the Scope amplitude measurement screen.

Syntax

```
:FETCh:AMPLitude:QUEStionableeye?
```

Response Data

0|1|N/A

0: Not display

1: Display

N/A Not amplitude measurement screen

Example of Use

```
:FETC:AMPT:QUES?
>1
```

:FETCh:AMPLitude:<meas_item>[:CHA|CHB]?

Function

This command queries the results of all measurement items for the amplitude measurement function of Scope.

Syntax

```
:FETCh:AMPLitude<meas_item>[:CHA|CHB]?
```

For NRZ

```
<meas_item>=:{AVEPower|CROSsing|EXTRatio|EYEAmplitude|EYEHeight|LEVel:ONE|LEVel:ZERO|OMA:DBM|OMA:MW|OMAXp|SNR|VECP}
```

Added in Version 6: For PAM4

```
<meas_item>=[:TEQualizer]:{AVEPower|EYE[0-2]:HEIGht|EYE[0-2]:LEVEL|LEVEL[0-3]|LEVEL[0-3]:PPeak|LEVEL[0-3]:RMS|LINearity|OER|OOMA|TDECQ}
```

For Version 6 or later, this command can be used as a query by appending CHA|CHB.

Response Data

<current>,<average>,<std_dev>,<min>,<max>

The measurement results of the items specified at <meas_item> are returned at <numeric> in the following sequence. When there is no measurement result, N/A is returned.

Current value, average value, standard deviation value,
Minimum value, Maximum value

The items in Table 3.9.7-1 are measured when :DISPlay:SIGNal is NRZ and :DISPlay:MODE is EYE or COHErenteye.

Table 3.9.7-1 Response Data of Amplitude Measurement (NRZ)

Header	Contents of Response
:FETCh:AMPLitude:LEVel:ONE?	One Level*1
:FETCh:AMPLitude:LEVel:ZERO?	Zero Level*1
:FETCh:AMPLitude:EYEAmplitude?	Eye Amplitude*1
:FETCh:AMPLitude:EYEHeight?	Eye Height*1
:FETCh:AMPLitude:CROSsing?	Crossing (%)
:FETCh:AMPLitude:SNR?	SNR
:FETCh:AMPLitude:AVEPower?	Combination of the mW value and the dBm value of Average Power*2,*3
:FETCh:AMPLitude:EXTRatio?	Extinction Ratio*2
:FETCh:AMPLitude:OMA:MW?	OMA (mW)*2
:FETCh:AMPLitude:OMA:DBM?	OMA (dBm)*2
:FETCh:AMPLitude:MEASurement?	Combination of all above items (11 items)*4
:FETCh:AMPLitude:OMAXp?	OMA Cross Point*2,*4
:FETCh:AMPLitude:VECP?	VECP*2,*4

*1: The unit for electrical channel is mV and the unit for optical channel is μW.

*2: When the optical channel is used, the value is returned. When the electrical channel is used, N/A is returned.

*3: The values are output in the sequence listed below.

Table 3.9.7-2 Average Power Data Order

Item	<current>	<average>	<std_dev>	<min>	<max>
Average Power (mW)	1	3	5	7	9
Average Power (dBm)	2	4	6	8	10

*4: The value is output in version 6 or later of MP2110A.

Added in Version 6: The items in Table 3.9.7-3 are measured when :DISPlay:SIGNal is PAM4 and :DISPlay:MODE is EYE or COHErenteye.
To query the results when TDECQ Equalizer is applied, specify :TEQualizer.

Table 3.9.7-3 Response Data of Amplitude Measurement (PAM4)

Header	Contents of Response
:FETCh:AMPLitude[:TEQualizer]:LINearity[:CHA|CHB]?	Linearity
:FETCh:AMPLitude[:TEQualizer]:LEVEL[0-3][:CHA|CHB]?	Levels*1
:FETCh:AMPLitude[:TEQualizer]:LEVEL[0-3]:RMS[:CHA|CHB]?	Levels RMS*1
:FETCh:AMPLitude[:TEQualizer]:LEVEL[0-3]:PPeak[:CHA|CHB]?	Levels Peak-Peak*1
:FETCh:AMPLitude[:TEQualizer]:EYE[0-2]:LEVEL?	Eye Levels*1
:FETCh:AMPLitude[:TEQualizer]:EYE[0-2]:HEIGht?	Eye Heights*1
:FETCh:AMPLitude:AVEPower?	Combination of the mW value and the dBm value of Average Power *2, *3
:FETCh:AMPLitude[:TEQualizer]:TDECQ?	TDECQ*2
:FETCh:AMPLitude[:TEQualizer]:OOMA?	Outer OMA*2
:FETCh:AMPLitude[:TEQualizer]:OER?	Outer Extinction Ratio (Outer ExR) *2

*1: The unit for electrical channel is mV and the unit for optical channel is µW.

*2: When the optical channel is used, the value is returned. When the electrical channel is used, N/A is returned.

*3: For the data output order, refer to Table 3.9.7-2"Average Power Data Order".

Example of Use

```
:FETC:AMPL:LEV:ONE?
>35.16,35.11,0.11,34.78,35.44
:FETC:AMPL:LEV:ZERO?
>-15.12,-15.20,0.05,-15.35,-15.05
:FETC:AMPL:EYEA?
>55.22,54.89,0.12,54.53,55.25
:FETC:AMPL:EYEH?
>45.81,45.77,0.08,45.53,46.01
:FETC:AMPL:CROS?
>46.01,45.80,0.19,45.27,46.41
:FETC:AMPL:SNR?
>10.08,10.11,0.19,9.55,10.70
:FETC:AMPL:AVEP?
>25.00,-16.02,25.50,-15.93,0.02,0.05,24.86,-16.05,26.12,
-15.83
:FETC:AMPL:EXTR?
>6.82,6.77,0.13,6.38,7.16
:FETC:AMPL:OMA:MW?
```

```
>0.15,0.16,0.03,0.06,0.25
:FETC:AMPL:OMA:DBM?
>-8.22,-8.24,0.21,-8.85,-7.59
```

Note:

When Measure Item is not Amplitude/Time, N/As are returned as all response data.

:FETCh:TIME:<meas_item>[:CHA|CHB]?

Function

This command queries the results obtained by the time measurement function of Scope.

Syntax

```
:FETCh:TIME<meas_item>[:CHA|CHB]?
```

For NRZ

```
<meas_item>=:{DCD|EYEWidth|FTIMe|JITTer:PPeak|JITTer:RMS
|TRISe}
```

For PAM4

```
<meas_item>=[:TEQualizer]:{EYE[0-2]:SKEW|EYE[0-2]:WIDTh|
LEVEL[0-3]:SKEW}
```

For Jitter (OPT096)

```
<meas_item>=[:AJITter]:{DJ|EOPening|J2|J9|RJ|TJ}|:AJITte
r:{DCD|DDJ|DDPWS|ISI|PJ[:FREQuency]|RJ[:RMS]}
```

For Version 6 or later, this command can be used as a query by appending CHA|CHB.
However, CHA|CHB cannot be appended when :DISPlay:MODE is AJITter.
Also, CHA|CHB cannot be appended in version older than 6. Active Channel result can be acquired.

Response Data

<current>,<average>,<std_dev>,<min>,<max>
The measurement results of the items specified at <meas_item> are returned at <numeric> in the following sequence. When there is no measurement result, N/As are returned.

Current value, average value, standard deviation value,
Minimum value, Maximum value

The items in Table 3.9.7-4 are measured when :DISPlay:SIGNal is NRZ and :DISPlay:MODE is EYE or COHErenteye.

Table 3.9.7-4 Response Data of Time Measurement (NRZ)

Header	Contents of Response
:FETCh:TIME:JITTer:PPeak?	Jitter p-p
:FETCh:TIME:JITTer:RMS?	Jitter RMS
:FETCh:TIME:TRISe?	Rise Time
:FETCh:TIME:FTIMe?	Fall Time
:FETCh:TIME:EYEWidth?	Eye Width
:FETCh:TIME:DCD?	DCD (%)

Added in Version 6: The items in Table 3.9.7-5 are measured when :DISPlay:SIGNal is PAM4 and :DISPlay:MODE is EYE or COHErenteye.
To query the results when TDECQ Equalizer is applied, specify :TEQualizer.

Table 3.9.7-5 Response Data of Time Measurement (PAM4)

Header	Contents of Response
:FETCh:TIME[:TEQualizer]:LEVEL[0-3]:SKEW?	Level Skews
:FETCh:TIME[:TEQualizer]:EYE[0-2]:SKEW?	Eye Skews
:FETCh:TIME[:TEQualizer]:EYE[0-2]:WIDTh?	Eye Widths

Added in Version 6: The items in Table 3.9.7-6 are measured when MP2110A-096 is installed.

Table 3.9.7-6 Response Data of Time Measurement (Jitter)

Header	Contents of Response
:FETCh:TIME[:AJITter]:TJ?	TJ (TJ Measurement BER) *1
:FETCh:TIME[:AJITter]:J2?	J2 Jitter*1
:FETCh:TIME[:AJITter]:J9?	J9 Jitter*1
:FETCh:TIME[:AJITter]:EOPening?	EYE Opening*1
:FETCh:TIME[:AJITter]:RJ?	RJ (d-d) *1
:FETCh:TIME[:AJITter]:DJ?	DJ (d-d) *1
:FETCh:TIME:AJITter:RJ:RMS?	RJ (rms) *2
:FETCh:TIME:AJITter:PJ?	PJ (p-p) *2
:FETCh:TIME:AJITter:PJ:FREQuency?	PJ Frequency*2 of PJ vs Freq
:FETCh:TIME:AJITter:DDPWS?	DDPWS*2
:FETCh:TIME:AJITter:DDJ?	DDJ (p-p) *2
:FETCh:TIME:AJITter:DCD?	DCD*2
:FETCh:TIME:AJITter:ISI?	ISI (p-p) *2

*1: Specify :AJITter when :DISPlay:MODE is set to AJITter.
Do not specify :AJITter when :DISPlay:MODE is set to EYE.

*2: Measured when :DISPlay:MODE is AJITter.

Example of Use

```
:FETC:TIME:JITT:PP?
>66.25,65.89,0.98,63.95,68.83
:FETC:TIME:JITT:RMS?
>15.31,15.52,0.26,14.74,16.30
:FETC:TIME:TRIS?
>128.22,130.11,1.52,125.55,134.67
:FETC:TIME:FTIM?
>133.66,129.96,2.59,122.19,137.75
:FETC:TIME:EYEW?
>208.60,206.15,3.32,216.11,196.19
:FETC:TIME:DCD?
>47.2,45.22,1.22,41.56,48.88
```

Note:

The response data is N/A when the Measure Item is not set to Amplitude/Time.

:FETCh:MASK:<meas_item>?

Function

This command queries the results for the mask measurement of the Scope.

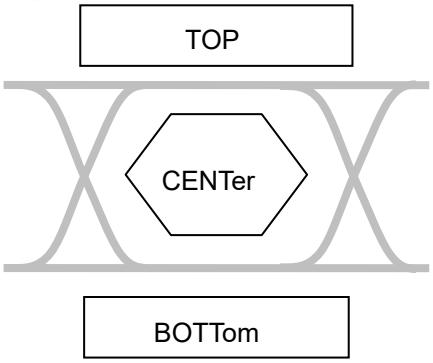
Syntax

```
:FETCh:MASK:<meas_item>?
<meas_item>=MEASurement|SAMPles:{FAILed[:{BOTTom|CENTer|
TOP}]|TOTal}
```

Response Data

The measurement results of the items specified at <meas_item> are returned. When there is no measurement result, N/A(s) is (are) returned.

Table 3.9.7-7 Response Data of Mask Measurement

Header	Contents of Response
:FETCh:MASK:MEASurement?	<integer>,<integer> Total Samples, Total Failed Samples
:FETCh:MASK:SAMPles:TOTal?	<integer> Total Samples
:FETCh:MASK:SAMPles:FAILed?	<integer> Total Failed Samples
:FETCh:MASK:SAMPles:FAILed:BOTTom? :FETCh:MASK:SAMPles:FAILed:CENTer? :FETCh:MASK:SAMPles:FAILed:TOP?	<integer> Bottom/Center/Top Mask Failed Samples 

Example of Use

```
:FETC:MASK:MEAS?
>16831,30
:FETC:MASK:SAMP:TOT?
>16831
:FETC:MASK:SAMP:FAIL?
>30
:FETC:MASK:SAMP:FAIL:BOTT?
>0
:FETC:MASK:SAMP:FAIL:CENT?
>20
:FETC:MASK:SAMP:FAIL:TOP?
>10
```

Note:

When Measure Item is not Mask Test, N/As are returned as all response data.

:FETCh:HISTogram:AMPLitude:<meas_item>?

Function

This command queries the histogram measurement result of the amplitude axis for Scope.

Syntax

```
:FETCh:HISTogram:AMPLitude:<meas_item>?
<meas_item>=HITS|MEAN|MEASurement|PPeak|STDDeviation
```

Response Data

The measurement results of the items specified at <meas_item> are returned. When there is no measurement result, N/A(s) is (are) returned.

Table 3.9.7-8 Response Data of Histogram Measurement (Amplitude)

Header	Contents of Response
:FETCh:HISTogram:AMPLitude:MEAN?	<numeric> Mean*
:FETCh:HISTogram:AMPLitude:STDDeviation?	<numeric> Std Dev*
:FETCh:HISTogram:AMPLitude:PPeak?	<numeric> P-P (Peak to Peak)*
:FETCh:HISTogram:AMPLitude:HITS?	<integer> Hits (Number of data within area)
:FETCh:HISTogram:AMPLitude:MEASurement?	Returns the above four results separated by commas.

*: The unit for electrical channel is mV and the unit for optical channel is μW.

Example of Use

```
:FETC:HIST:AMPL:MEAN?
>32.1
:FETC:HIST:AMPL:STDD?
>4.53
:FETC:HIST:AMPL:PP?
>28.1
:FETC:HIST:AMPL:HITS?
>89632
:FETC:HIST:AMPL:MEAS?
>32.1,4.53,28.1,89632
```

Notes:

- This message queries data that the measurement ends.
- When Measure Item is not Histogram or is the time axis histogram, N/A(s) is (are) returned as the response data.

:FETCh:HISTogram:TIME:<meas_item>?

Function

This command queries the time axis histogram measurement result of Scope.

Syntax

```
:FETCh:HISTogram:TIME:<meas_item>?
<meas_item>=HITS|MEAN|MEASurement|PPeak|STDDeviation
```

Response Data

The measurement results of the items specified at <meas_item> are returned. When there is no measurement result, N/A(s) is (are) returned.

Table 3.9.7-9 Response Data of Histogram Measurement (Time)

Header	Contents of Response
:FETCh:HISTogram:TIME:MEAN?	<numeric> Mean (ps)
:FETCh:HISTogram:TIME:STDDeviation?	<numeric> Std Dev (ps)
:FETCh:HISTogram:TIME:PPeak?	<numeric> P-P (ps)
:FETCh:HISTogram:TIME:HITS?	<integer> Hits (Number of data within area)
:FETCh:HISTogram:TIME:MEASurement?	Returns the above four results separated by commas.

Example of Use

```
:FETC:HIST:TIME:MEAN?
>1.53
:FETC:HIST:TIME:STDD?
>0.022
:FETC:HIST:TIME:PP?
>0.081
:FETC:HIST:TIME:HITS?
>6831
:FETC:HIST:TIME:MEAS?
>1.53,0.022,0.081,6831
```

Notes:

- This message queries data that the measurement ends.
- When Measure Item is not Histogram or is the amplitude axis histogram, N/A(s) is (are) returned as the response data.

:TRACe:CHANnelA|CHANnelB|CHANnels?

Function

This command queries the trace data of Scope that is in eye pattern mode.

Syntax

```
:TRACe[:DATA]:{CHANnelA|CHANnelB|CHANnels}?
```

Parameter

:TRACe:CHANnelA

This command requests sending of trace data for Channel A. Before sending this command, always set Channel A to On and Channel B to Off.

:TRACe:CHANnelB

This command requests sending of trace data for Channel B. Before sending this command, always set Channel B to On and Channel A to Off.

:TRACe:CHANnels

This command requests sending of trace data for Channel A and Channel B. Set both Channel A and Channel B to On.
When the display of the channel querying the trace data is set to Off, the data cannot be returned.

Response Data

- When channel to read not displayed: "Channel Off"
- When reading one channel:
  {CHA | CHB}-<integer>(<numeric>,<numeric>)[,(<numeric>,<numeric>)]…
- When reading one channel:
  CHA-<integer>(<numeric>,<numeric>)[,(<numeric>,<numeric>)]…,CHB-<integer>(<numeric>,<numeric>)[,(<numeric>,<numeric>)]…

CHA | CHB
CHA: Channel A
CHB: Channel B

<integer>
Trace data score

(<numeric>,<numeric>)
Each time and amplitude

Example of Use

To query the trace data for the Channel A:

```
:TRAC:CHANA?
```

```
>CHA-2039(86.0,39.97),(86.0,167.13)…
```

To query the trace data for the Channel A and B

```
:TRAC:CHAN?
>CHA-2039(86.0,39.97),(86.0,167.13)…,(285.9,-3.92),CHB-2
039(86.0,152.10)…
```

Note:

This message is invalid when Scope is in Pulse or Coherent Eye mode.

Before sending this message, send :TRACe:PREPare to stop updating the Scope screen and make the Scope get ready for receiving trace data query.

If this message is sent after :TRACe:PREPare is sent, the Scope starts data acquisition and creates trace data. Once trace data is created, this message can be used.

To release this status, send :TRACe:END after completing reading of the trace data.

The following changes are made for the MX210000A of Version 3.00 or later.

- This message can be used when Scope is in pulse mode. However, this message cannot be used when in coherent eye.
- Sending of the command :TRACe:PREPare is not required before sending the data.
- Sending of the command :TRACe:END is not required after sending the data.

:TRACe:PREPare

Function

This command sets the instrument to the status (readout mode) in which the Scope trace data can be read via the remote interface.

Syntax

```
:TRACe[:DATA]:PREPare CHA|CHB|BOTH
```

Parameter

CHA:

Sets status for reading only Channel A data
Before using this command, Channel A is set to On and Channel B is set to Off.
Send :TRACe:CHANnelA? after this parameter is set.

CHB:

Sets status for reading only Channel B data
Before using this command, Channel B is set to On and Channel A is set to Off.
Send :TRACe:CHANnelB? after this parameter is set.

BOTH:

Sets status for reading Channel A and B data
Before using this command, Both Channel A and B is set to On.
Send :TRACe:CHANnels? after this parameter is set.

Example of Use

To set the Channel A waveform display to On.

```
:INPut:CHA ON
:INPut:CHB OFF
```

To activate the data reading status of Channel A.

```
:TRACe:PREPare CHA
```

To read out the data of Channel A.

```
:TRACe:CHANnelA?
```

To release the data reading status of Channel A.

```
:TRACe:END
```

Notes:

- When this command is sent, updating of the Scope waveform screen is stopped.
- Send :TRACe:PREPare before sending :TRACe:CHANnelA | CHANnelB | CHANnels? to query the trace data.
- The status set by this command is released by :TRACe:END.

:TRACe:END

Function

This command terminates the status (readout mode) set by the :TRACe:PREPare command.

Syntax

```
:TRACe[:DATA]:END
```

Example of Use

```
:TRAC:END
```

Notes:

- When executing this command, the normal screen-update mode is returned.
- Send this command after reading all the trace data after sending the trace data query.
- This command is used in combination with the :TRACe:PREPare command.

3.9.8 Marker

:CALCulate:MARKer:AOFF

Function

This command deletes the marker displayed in the Scope screen.

Syntax

```
:CALCulate:MARKer:AOFF
```

Example of Use

```
:CALC:MARK:AOFF
```

:CALCulate:MARKer:CENTer

Function

This command displays all markers of Scope at the center of the screen.

Syntax

```
:CALCulate:MARKer:CENTer
```

Example of Use

```
:CALC:MARK:CENT
```

:CALCulate:MARKer:X1|X2|Y1|Y2

Function

This command sets and queries the On/Off state of the amplitude marker X1/X2/Y1/Y2 display of Scope.

Syntax

```
:CALCulate:MARKer:X1|X2|Y1|Y2 <enable>
:CALCulate:MARKer:X1|X2|Y1|Y2?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:CALC:MARK:X1 ON
:CALC:MARK:X1?
>1
```

:CALCulate:MARKer:LOCation:X1|X2

Function

This command sets and queries the position of the time marker X1/X2 of Scope.

Syntax

```
:CALCulate:MARKer:LOCation:X1|X2 <numeric>
:CALCulate:MARKer:LOCation:X1|X2?
```

Parameter

<numeric>
Unit UI or ps

Response Data

<numeric>
When the marker is not displayed, N/A is returned.

Example of Use

To display the marker X2 to the 1.5 UI position

```
:CALC:MARK:LOC:X2 1.5
:CALC:MARK:LOC:X2?
>1.500
```

:CALCulate:MARKer:LOCation:XDELta?

Function

This command queries the difference between X1 marker and X2 marker.

Syntax

```
:CALCulate:MARKer:LOCation:XDELta?
```

Response Data

<numeric>
Unit: UI or ps
The response data when both or either of markers are not displayed is "N/A".
Set the unit (UI or ps) by :DISPlay:WINDow:X:UNIT.

Example of Use

```
:CALC:MARK:LOC:XDEL?
>152.330
```

:CALCulate:MARKer:LOCation:CHA|CHB:Y1|Y2

Function

This command sets and queries the position of the Y1/Y2 marker of Scope.

Syntax

```
:CALCulate:MARKer:LOCation:CHA|CHB:Y1|Y2 <numeric>
:CALCulate:MARKer:LOCation:CHA|CHB:Y1|Y2?
```

Parameter

<numeric>

The range and step vary depending on the scale of the Y axis (:DISP:WIND:Y:DIV:CHA|CHB).

Response Data

<numeric>

The response data when the marker is not displayed is "N/A".

Example of Use

To query the value of the marker Y2 for the Channel A waveform:

```
:CALC:MARK:LOC:CHA:Y2 55.35
:CALC:MARK:LOC:CHA:Y2?
>55.35
```

:CALCulate:MARKer:LOCation:CHA|CHB:YDELta?

Function

This command queries the difference between Y1 marker and Y2 marker.

Syntax

```
:CALCulate:MARKer:LOCation:CHA|CHB:YDELta?
```

Response Data

<numeric>

The unit for electrical channel is mV and the unit for optical channel is μW.

The response data when both or either of markers are not displayed is "N/A".

Example of Use

```
:CALC:MARK:LOC:CHB:YDEL?
>15.3
```

3.9.9 Jitter

:JITTer:RESult:ERRor?

Function

Added in Version 6: This command queries an error(s) that occurred during the AdvancedJitter measurement.

Syntax

```
:JITTer:RESult:ERRor?
```

Response Data

<integer>: Total of the values that are given to each error type

Error Type	Value
EYE?	1
TIE Error	2
Pattern Lost	4
Illegal Error	32768

When more than one error occurs at the same time, the values of all errors are totalized.
For example, when Pattern Lost and Illegal Error happen at the same time, the response data is 4+32768=32772.

Example of Use

When Pattern Lost occurs

```
:JITT:RES:ERR?
> 4
```

:JITTer:MEASure:PDJ

Function

Added in Version 6: This command sets and queries the On/Off state of the PDJ measurement.

Syntax

```
:JITTer:MEASure:PDJ <enable>
:JITTer:MEASure:PDJ?
```

Parameter

0|OFF
1|ON

Response Data

0|1

0 PDJ Off

1 PDJ On

Example of Use

```
:JITT:MEAS:PDJ ON
:JITT:MEAS:PDJ?
>1
```

:JITTer:MEASure:PDJ:STANdard

Function

Added in Version 6: This command sets and queries the standard for the PDJ measurement.

Syntax

```
:JITTer:MEASure:PDJ:STANdard 0|1|2|3|4|5
:JITTer:MEASure:PDJ:STANdard?
```

Parameter

0: STM-0

1: STM-1

2: STM-4

3: STM-16

4: STM-64

5: STM-256

Response Data

0|1|2|3|4|5

Example of Use

```
:JITT:MEAS:PDJ:STAN 5
:JITT:MEAS:PDJ:STAN?
>5
```

:JITTer:MEASure:PDJ:FILTer

Function

Added in Version 6: This command sets and queries the filter for the PDJ measurement.

Syntax

```
:JITTer:MEASure:PDJ:FILTer 0|1|2|3|4|5|6|7|8
:JITTer:MEASure:PDJ:FILTer?
```

Parameter

0:	LP
1:	HP0+LP
2:	HP1+LP
3:	HP1'+LP
4:	HP2+LP
5:	HP+LP
6:	HP'+LP
7:	LP'
8:	HP0+LP'

Response Data

0|1|2|3|4|5|6|7|8

Example of Use

```
:JITT:MEAS:PDJ:FILT 0
:JITT:MEAS:PDJ:FILT?
>0
```

:JITTer:MEASure:EDGE:TYPE

Function

Added in Version 6: This command sets and queries the edge detection method of pattern data for the Jitter measurement.

Syntax

```
:JITTer:MEASure:EDGE:TYPE ALL|FALL|RISE
:JITTer:MEASure:EDGE:TYPE?
```

Parameter

ALL:	All
FALL:	Fall
RISE:	Rise

Response Data

ALL|FALL|RISE

Example of Use

```
:JITT:MEAS:EDGE:TYPE ALL
:JITT:MEAS:EDGE:TYPE?
>ALL
```

:JITTer:MEASure:TJ:BER[:CHA|CHB]

Function

Added in Version 6: This command sets and queries the bit error rate for the TJ and Eye Opening measurement.

Syntax

```
:JITTer:MEASure:TJ:BER[:CHA|CHB] <character>
:JITTer:MEASure:TJ:BER[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**. When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

<character>: Set the bit error rate referring to the table below.

<character>	Bit Error Rate	<character>	Bit Error Rate
E_1	10^{-1}	E_10	10^{-10}
E_2	10^{-2}	E_11	10^{-11}
E_3	10^{-3}	E_12	10^{-12}
E_4	10^{-4}	E_13	10^{-13}
E_5	10^{-5}	E_14	10^{-14}
E_6	10^{-6}	E_15	10^{-15}
E_7	10^{-7}	E_16	10^{-16}
E_8	10^{-8}	E_17	10^{-17}
E_9	10^{-9}	E_18	10^{-18}

Response Data

E_1|E_2|E_3|E_4|E_5|E_6|E_7|E_8|E_9|E_10|E_11|E_12|E_13|E_14|E_15|E_16|E_17|E_18

Example of Use

```
:JITT:MEAS:TJ:BER E_9
:JITT:MEAS:TJ:BER?
>E_9
```

:JITTer:MEASure:RJ:FIXed[:CHA|CHB]

Function

Added in Version 6: This command sets and queries Fixed RJ of the Jitter measurement.

Syntax

```
:JITTer:MEASure:RJ:FIXed[:CHA|CHB] <enable>
:JITTer:MEASure:RJ:FIXed[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**, When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

0|OFF Fixed RJ Off
1|ON Fixed RJ On

Response Data

0|1

Example of Use

```
:JITT:MEAS:RJ:FIX ON
:JITT:MEAS:RJ:FIX?
>1
```

:JITTer:MEASure:RJ:FIXed:VALue[:CHA|CHB]

Function

Added in Version 6: This command sets and queries RJ Value of Fixed RJ.

Syntax

```
:JITTer:MEASure:RJ:FIXed:VALue[:CHA|CHB] <numeric>
:JITTer:MEASure:RJ:FIXed:VALue[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**, When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

<numeric>: 0.01 to 999.99

Response Data

<numeric>: 0.01 to 999.99

Example of Use

```
:JITT:MEAS:RJ:FIX:VAL 2.50
:JITT:MEAS:RJ:FIX:VAL?
>2.50
```

:JITTer:MEASure:CORRection[:CHA|CHB]

Function

Added in Version 6: This command sets and queries Correction Factor of the Jitter measurement.

Syntax

```
:JITTer:MEASure:CORRection[:CHA|CHB] <enable>
```

```
:JITTer:MEASure:CORRection[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**. When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

0|OFF	Correction Factor Off
1|ON	Correction Factor On

Response Data

0|1

Example of Use

```
:JITT:MEAS:CORR ON
:JITT:MEAS:CORR?
>1
```

:JITTer:MEASure:CORRection:DJ:SCALe[:CHA|CHB]

Function

Added in Version 6: This command sets and queries DJ (scale) of Correction Factor.

Syntax

```
:JITTer:MEASure:CORRection:DJ:SCALe[:CHA|CHB] <numeric>
:JITTer:MEASure:CORRection:DJ:SCALe[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**. When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

<numeric>: 0.01 to 999.99

Response Data

<numeric>: 0.01 to 999.99

Example of Use

```
:JITT:MEAS:CORR:DJ:SCAL 5.00
:JITT:MEAS:CORR:DJ:SCAL?
>5.00
```

:JITTer:MEASure:CORRection:RJ:SCALe[:CHA|CHB]

Function

Added in Version 6: This command sets and queries RJ (scale) of Correction Factor.

Syntax

```
:JITTer:MEASure:CORRection:RJ:SCALe[:CHA|CHB] <numeric>
:JITTer:MEASure:CORRection:RJ:SCALe[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**. When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

<numeric>: 0.01 to 999.99

Response Data

<numeric>: 0.01 to 999.99

Example of Use

```
:JITT:MEAS:CORR:RJ:SCAL 20.00
:JITT:MEAS:CORR:RJ:SCAL?
>20.00
```


:JITTer:MEASure:CORRection:RJ:RMS[:CHA|CHB]

Function

Added in Version 6: This command sets and queries RJ (rms) of Correction Factor.

Syntax

```
:JITTer:MEASure:CORRection:RJ:RMS[:CHA|CHB] <numeric>
:JITTer:MEASure:CORRection:RJ:RMS[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**. When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

<numeric>: 0.01 to 999.99

Response Data

<numeric>: 0.01 to 999.99

Example of Use

```
:JITT:MEAS:CORR:RJ:RMS 2.50
:JITT:MEAS:CORR:RJ:RMS?
>2.50
```

:JITTer:MEASure:CROSsing[:CHA|CHB]

Function

Added in Version 6: This command sets and queries the edge detection method of pattern data for the Jitter measurement.

Syntax

```
:JITTer:MEASure:CROSsing[:CHA|CHB] AUTO|MANual
:JITTer:MEASure:CROSsing[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**. When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

AUTO: Auto
MANual: Manual

Response Data

AUTO|MANual

Example of Use

```
:JITT:MEAS:CROS AUTO
:JITT:MEAS:CROS?
> AUTO
```

:JITTer:MEASure:CROSsing:MANual[:CHA|CHB]

Function

Added in Version 6: This command sets and queries Manual Crossing of Threshold Level for the Jitter measurement.

Syntax

```
:JITTer:MEASure:CROSsing:MANual[:CHA|CHB] <integer>
:JITTer:MEASure:CROSsing:MANual[:CHA|CHB]?
```

A channel can be specified when Sampling Mode is **Eye**. When Sampling Mode is **Advanced Jitter**, a channel cannot be specified.

Parameter

<integer>: 30 to 70%

Response Data

<integer>: 30 to 70

Example of Use

```
:JITT:MEAS:CROS:MAN 55
```

```
:JITT:MEAS:CROS:MAN?
>55
```

:JITTer:GRAPh:ESTimate:RJDJ:CHA|CHB

Function

Added in Version 6: This command sets and queries Estimated RJ/DJ Histogram on the TJ Histogram screen.

Syntax

```
:JITTer:GRAPh:ESTimate:RJDJ:CHA|CHB <enable>
:JITTer:GRAPh:ESTimate:RJDJ:CHA|CHB?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:JITT:GRAP:EST:RJDJ OFF
:JITT:GRAP:EST:RJDJ?
>0
```

:JITTer:GRAPh:ESTimate:RJPJ

Function

Added in Version 6: This command sets and queries Estimated RJ/PJ Histogram on the RJ/PJ Histogram screen.

Syntax

```
:JITTer:GRAPh:ESTimate:RJPJ <enable>
:JITTer:GRAPh:ESTimate:RJPJ?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:JITT:GRAP:EST:RJPJ 1
:JITT:GRAP:EST:RJPJ?
>1
```

:JITTer:GRAPh:DDJ:ALL|FALL|RISE

Function

Added in Version 6: This command sets and queries the On/Off state of DDJ Histogram graph display (All Edge, Fall Edge, Rise Edge).

Syntax

```
:JITTer:GRAPh:DDJ:ALL|FALL|RISE <enable>
:JITTer:GRAPh:DDJ:ALL|FALL|RISE?
```

Parameter

0|OFF
1|ON

Response Data

0|1

Example of Use

```
:JITT:GRAP:DDJ:FALL 0
:JITT:GRAP:DDJ:FALL?
>0
:JITT:GRAP:DDJ:RISE ON
:JITT:GRAP:DDJ:RISE?
>1
```

:JITTer:MEASure:PJ:FREQuency:CALCulate

Function

Added in Version 6: This command executes PJ Calculation for PJ vs Frequency graph.

Syntax

```
:JITTer:MEASure:PJ:FREQuency:CALCulate
```

Appendix A Command Compatibility With Existing Products

Appendix A explains the command compatibility between BERTWaves with different models, and the compatible commands that can be used as the command messages for both conventional products and BERTWave.

A.1 Command Compatibility Between MP2100A and MP2100B A-2
A.2 Command Compatibility Between MP2100B and MP2110A A-3
A.3 Command Compatibility Tables with Products Other Than BERTWave A-5
A.4 Introduction of Native Commands A-7
A.5 Compatible Commands and Descriptions A-8

A.1 Command Compatibility Between MP2100A and MP2100B

If one of the following bit rate setting commands is used for PPG Ch3 or PPG Ch4 of MP2100B, an error occurs:

```
:OUTPut:BITRate:STANdard
:OUTPut:CLOCk:OPERation
:OUTPut:BITRate
:OUTPut:CLOCk:FREQuency
:OUTPut:CMU:FREQuency
:OUTPut:BITRate:OFFSet
:OUTPut:CLOCk:OFFset:PPM
```

If one of the following ED bit rate setting commands is used for MP2100B, an error occurs:

```
:INPut:BITRate:STANdard
:INPut:BITRate
```

The following changes are made to the Reference CLK setting commands for MP2100B:

Addition:
`:OUTPut:RCLock`
Changed into compatible commands:
`:OUTPut:CMU:REFClock`
`:OUTPut:RCLock:SELect`

For MP2100B, Ch Tracking and Ch2 Reference CLK are set to On and Ch1, by default. Therefore, if your BERTWave is MP2100A, execution of a sequence results in an error, in case where the following items for Ch1 and Ch2 are set to different values, respectively.

- Bitrate
- Test Pattern
- Gating Cycle/Period

A.2 Command Compatibility Between MP2100B and MP2110A

In MP2110A, the following commands that were dependent on Ch2 to 4 (Master was Ch1) are changed to the common setting. So the settings of these commands can be changed even when any channel is specified.

```
:OUTPut:RCLock
:OUTPut:RCLock:APPLy
:OUTPut:BITRate:STANdard
:INPut:BITRate:STANdard
:OUTPut:BITRate
:INPut:BITRate
:OUTPut:BITRate:OFFSet
```

In MP2110A, the following commands that were dependent on Ch2 to 4 (Master was Ch1) when Ch Tracking was ON are changed to the common setting. So the settings of these commands can be changed even when any channel is specified.

```
:SOURce:PATTern:TYPE
:SENSe:MEASure:EALarm:MODE
:SENSe:MEASure:EALarm:PERiod
```

In MP2110A, the following commands cannot be used. When the command is used, an error occurs.

```
:SYSTem:BEEPer:SET
:SYSTem:ERRor:HCLear
:SYSTem:ERRor:HISTory?
:SYSTem:DISPlay:ALARm
:SYSTem:INFormation:ERRor?
:OUTPut:RCLock:SELect
:OUTPut:CMU:EXTClock
:INPut:BITRate:DIVRate?
:OUTPut:BITRate:DIVRate
:SOURce:MMEMory:PATTern:RECall
:SENSe:MMEMory:PATTern:RECall
:SOURce:PATTern:EADDition:SET
:SOURce:PATTern:EADDition:VARiation
:SOURce:PATTern:EADDition:RATE
:SENSe:PATTern:SYNC:ASYNc
:SENSe:PATTern:SYNC:THReshold
:SENSe:PATTern:SYNC:PSMode
```

```
:SENSe:PATTern:SYNC:FPOSition
:DISPlay:MODE:EYE:FAST
:CALibrate:SYSTem:CGain
```

In MP2110A, the following commands are changed.

```
:CONFigure:TRACking:DRATe
:CONFigure:TRACking:DRATe:MASTer
```

→ This command is changed to "`:TIME:TRACking`".
And "`:TIME:TRACking:STATus?`" was added.

```
:INPut:FILTer:ENABle
:INPut:FILTer
```

→This command is changed to "`:FILTer[:CHA|CHB]`".

```
:CONFigure:MASK:TYPe
```

→This command is changed to "`:CONFigure:MASK:RECall`".

In MP2110A, the following commands were added.

```
:OUTPut:CLOCk:DIVRate?
:OUTPut:CLOCk:SOURce:CHANnel
:DISPlay:RESult
[:BERT:ALL]:CALCulate:DATA:MONitor:HISTory?
:TIME:PTRigger
:TIME:PTRigger:RESet
```

In MP2110A, specifying ChA or ChB was added to the following commands. When nothing is specified, ChB is specified. When ChA or ChB is specified in MP210xA and MP2100B, an error (–113 Undefined Header) occurs.

```
:INPut:WAVLength[:CHA|CHB]
:CALibrate:CGain[:CHA|CHB]
:CALibrate:RESPonsivity[:CHA|CHB]
:CALibrate:AUTocorrect[:CHA|CHB]
:CALibrate:CALPower[:CHA|CHB]
:CONFigure:EXRCorrection[:CHA|CHB]
:CONFigure:EXRCorrection:FACTor[:CHA|CHB]
```

In MP2110A, the default value of Number of Samples (:OPTion:MAX:SAMPles:NUMber) of Scope is different. The default value in MP210xx is 8191 and the default value in MP2110A is 2048.

A.3 Command Compatibility Tables with Products Other Than BERTWave

✓: Compatible.

*: Partly compatible; occasional errors when sending commands for previous hardware to this instrument.

(Blank): Incompatible; sending commands for previous hardware to this instrument always causes errors.

Table A.3-1 Command Compatibility

SCPI Command	Native Command	MP1800A	MP1632C	MP1776A
:CALCulate:DATA:EALarm	END?	*		
	ER?	✓	✓	
	EC?	✓	✓	
	CC?	✓	✓	
	FRQ?	✓	✓	
:CALCulate:DATA:MONitor	MTR?	✓		
	ERS?	✓	✓	
:CALCulate:OPTical:STATus		*		
:DISPlay:RESult:EALarm:HRESet	HRE	✓	✓	✓
:DISPlay:RESult:EALarm:MODE	CUR	✓		
:INPut:DATA:INTerface	DSD	*		
:MODule:ID		✓		
:OUTPut:CLOCk:FREQuency		✓	✓	
:OUTPut:CLOCk:OFFset:PPM	COP	✓		
:OUTPut:CMU:EXTClock	CEC			
:OUTPut:CMU:FREQuency	CRF	✓		
:OUTPut:CMU:REFClock	CRE	✓		
:OUTPut:CMU:RESolution	CRS	✓		
:OUTPut:DATA:AMPLitude	DAP	✓	✓	
:OUTPut:DATA:ATTFactor	DAT	✓		
:OUTPut:DATA:OUTPut	DON	✓		
:OUTPut:RCLock:SELect	RFC	*	*	
:SENSe:MEASure:ASTRt	SAT	✓		
:SENSe:MEASure:ASTP	SOT	✓		
:SENSe:MEASure:ASTate?	STT?	✓		
:SENSe:MEASure:EALarm:MODE	MOD	✓	✓	✓
:SENSe:MEASure:EALarm:ELAPsed	PRD	✓	✓	✓
:SENSe:MEASure:STARt	STA	✓	✓	✓
:SENSe:MEASure:STOP	STO	✓	✓	✓
:SENSe:PATTern:LOGic	LGC	✓		

Table A.3-1 Command Compatibility (Cont'd)

SCPI Command	Native Command	MP1800A	MP1632C	MP1776A
:SENSe:PATTern:TYPE	PTS	*		
:SENSe:PATTern:SYNC:ASYNc	SYN	✓	✓	✓
:SENSe:PATTern:SYNC:THReshold	SYE	✓		✓
:SENSe:PATTern:SYNC:PSMode	SYM	*	*	*
:SENSe:PATTern:SYNC:FPOSition	FPS	✓		
:SENSe:MEASure:EALarm:STATe?	MSR?	✓	✓	✓
:SENSe:MEASure:EALarm:STARt?	MSA?	✓	✓	✓
:SENSe:MEASure:EALarm:STOP?	MSO?	✓	✓	✓
:SENSe:MEASure:EALarm:ELAPsed?	MLP?	✓	✓	✓
:SENSe:MEASure:EALarm:TIMed?	ETI?	✓	✓	✓
:SOURce:OPTical:SIGNal:WLENgth?		✓		
:SOURce:OPTical:SIGNal:OUTPut		✓		
:SOURce:OUTPut:ASET	OON	✓	✓ (OON)	
:SOURce:PATTern:EADDition:RATE	ERT	✓	*	
:SOURce:PATTern:EADDition:SET	EAD	✓	✓	
:SOURce:PATTern:EADDition:SINGle	ESI	✓	✓	
:SOURce:PATTern:EADDition:VARiation	EAV	✓		
:SOURce:PATTern:LOGic	LGC	✓		
:SOURce:PATTern:TYPE	PTS	*		
:STATus:OPERation:CONDition?		✓	✓	
:STATus:OPERation:ENABle		✓	✓	
:STATus:OPERation[:EVENt]?		✓	✓	
:STATus:OPERation:NTRansition		✓	✓	
:STATus:OPERation:PTRansition		✓	✓	
:STATus:PRESet		✓	✓	
:SYSTem:ERRor?		✓	✓	✓
:SYSTem:{DATE|TIME}?	RTM?	✓		
:SYSTem:DISPlay:RESult		✓		
:SYSTem:INFormation:ERRor?	INF?	*		
:SYSTem:MEMory:INITialize	INI	✓	✓	✓
:SYSTem:PRINt:COPY	HCP	✓	✓	✓
:SYSTem:TERMination	TRM	✓	✓	
:SYSTem:VERSion?		✓	✓	✓

A.4 Introduction of Native Commands

Native Command Usage Purpose

The purpose of the Native command is to allow use of remote interface software for previous Anritsu pulse pattern generator and error detector models with the BERTWave.

Native Command Format

The Native command is configured by the string of ASCII code.
For the Native command format, refer to 2.7 “Message Format”.
The word string that indicates the Native command end is LF (line feed) or CR (carriage return) +LF.

Note:

Native commands must end with LF or CR+LF; otherwise, a timeout error results because the communication is not able to end.

Channel Setting Method

When using the pulse pattern generator/ error detector with two or more channels, the channel to be controlled is set using :MODule:ID command before executing the Native command.

Example of Use

When controlling the PPG/ED_1ch:

```
:MOD:ID 1
```

When controlling the PPG/ED_2ch:

```
:MOD:ID 2
```

When querying the currently controlled channel:

```
:MOD:ID?
```

A.5 Compatible Commands and Descriptions

RTM?

Function

Compatible command: This command has the same function as :SYSTem:{DATE|TIME}?.

Response Data

RTM <year>,<month>,<day>,<hour>,<minute>,<second>
Each of the parameters in the response data is a two-digit number. The lower two digits of the year is output as <year>.

Example of Use

```
RTM?
>RTM 09,10,24,09,51,13
```

INF?

Function

Compatible command: This command has the same function as :SYSTem:INFormation:ERRor?.

Response Data

INF <error_code>
The meanings of <error_code> are the same as those of :SYSTem:INFormation:ERRor?.

Example of Use

```
INF?
>INF 1
```

:SYSTem:MEMory:INITialize

Function

This command has the same function as *RST.

Example of Use

```
:SYST:MEM:INIT
```

INI

Function

Compatible command: This command has the same function as :SYSTem:MEMory:INITialize.

Example of Use

```
INI
```

HCP

Function

Compatible command: This command has the same function as :SYSTem:PRINt:COPY.

Example of Use

```
HCP
```

OON

Function

This command sets/queries the On/Off state of signal outputting from all PPG channels and optical outputting from XFP/SFP+.

Syntax

```
OON 0|1
OON?
```

Parameter

0 PPG output and optical output OFF
1 PPG output and optical output ON

Response Data

OON 0|1

Example of Use

```
OON 1
OON?
>OON 1
```

:SENSe:MEASure:ASTRt

Function

This command has the same function as *TRG.

Example of Use

```
:SENS:MEAS:ASTR
```

SAT

Function

Compatible command: This command has the same function as :SENSe:MEASure:ASTRt.

Example of Use

```
SAT
```

SOT

Function

Compatible command: This command has the same function as :SENSe:MEASure:ASTP.

Example of Use

```
SOT
```

STT?

Function

Compatible command: This command has the same function as :SENSe:MEASure:ASTate?.

Response Data

STT 0|1

Example of Use

```
STT?
>STT 1
```

:OUTPut:CMU:REFClock

Function

Compatible command: This command has the same function as :OUTPut:RCLock:SELect.

Syntax

```
:OUTPut:CMU:REFClock
INTernal|EXTernal|CH1External|CH2External|SYNChronize
:OUTPut:CMU:REFClock?
```

Example of Use

```
:OUTP:CMU:REFC INT
:OUTP:CMU:REFC?
>INT
```

RFC

Function

Compatible command: This command has the same function as :OUTPut:RCLock:SELect.

Syntax

```
RFC 0|1|2|3|4
RFC?
```

Parameter

0 INTernal
1 EXTernal
2 Added in Version 4: CH1External
3 Added in Version 4: CH2External
4 Added in Version 4: SYNChronize

Response Data

RFC 0|1|2|3|4

Example of Use

```
RFC 0
RFC?
>RFC 0
```

CRE

Function

Compatible command: This command has the same function as RFC.

Syntax

```
CRE 0|1|2|3|4
CRE?
```

Example of Use

```
CRE 0
CRE?
>CRE 0
```

CEC

Function

Compatible command: This command has the same function as :OUTPut:CMU:EXTClock.

Syntax

```
CEC 0|1
CEC?
```

Parameter

0 Ext 10MHz In
1 Ext 1/16 In

Response Data

CEC 0|1

Example of Use

```
CEC 1
CEC?
>CEC 1
```

SOP

Function

Compatible command only for MP210xA, MP2100B: This command has the same function as :OUTPut:SYNC:SOURce.

Syntax

```
SOP <integer>
SOP?
```

Parameter

<integer>

0	PPG1_1/1Clk
1	PPG1_1/2Clk
2	PPG1_1/4Clk
3	PPG1_1/8Clk
4	PPG1_1/16Clk
5	PPG1_1/64Clk
6	PPG2_1/1Clk
7	PPG2_1/2Clk
8	PPG2_1/4Clk
9	PPG2_1/8Clk
10	PPG2_1/16Clk
11	PPG2_1/64Clk
13	ED1_1/16Clk
15	ED2_1/16Clk
16	PPG1_Pattern Sync
17	PPG2_Pattern Sync
19	ED1_1/4Clk
20	ED1_1/8Clk
22	ED2_1/4Clk
23	ED2_1/8Clk
24	ED3_1/4Clk
25	ED3_1/8Clk
26	ED4_1/4Clk
27	ED4_1/8Clk
28	ED4_1/16Clk
29	PPG3_Pattern Sync
30	PPG4_Pattern Sync

Response Data

SOP <integer>

Example of Use

To set the 1/16 divide clock synchronized with the data output of the PPG Channel 1 to the output signal to the Sync Output connector.

```
SOP 4
SOP?
>SOP 4
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:SENSe:PARam:AEXecute

Function

Compatible command only for MP210xA, MP2100B (Version 4 or later):
This command is compatible with version 4 or later, and performs batch setting of PPG/ED of Ch1 and Ch2.

Syntax

```
:SENSe:PARam:AEXecute
<ppg1>,<ppg2>,<ed1>,<ed2>,<bitrate_standard>,<bitrate>,<
ppm>,<pattern>,<amplitude>
```

Parameter

<ppg1>,<ppg2>,<ed1>,<ed2>

Specify the object device(s) to be set by the command.

Specify 1 for the object device(s) to be set by the command; specify 0 otherwise.

<bitrate_standard> = <string>

Select a standard for the bit rate of PPG/ED. The parameter values are the same as for :{OUTPut|INPut}:BITRate:STANdard.

<bitrate> = <integer>

Specify/query the bit rate (kbit/s) if "Variable" is selected as the bit rate standard for PPG/ED. The parameter values are the same as for :{OUTPut|INPut}:BITRate.

<ppm> = <integer>

Specify/query the bit rate offset (ppm) for PPG. The parameter values are the same as for :OUTPut:BITRate:OFFSet.

<pattern> = PRBS{7|9|15|23|31}|USER

Specify the test pattern for PPG/ED. The parameter values are the same as for :{SOURce|SENSe}:PATTern:TYPE.

<amplitude> = <numeric>

Specify the signal amplitude (Vpp) of PPG. The parameter values are the same as for :OUTPut:DATA:AMPLitude.

Example of Use

To set PPG/ED of Ch1/2 to the bit rate of 10.3125 Gbit/s, offset of 100 ppm, test pattern of PRBS 2^31–1, and amplitude of 0.8 Vpp:

```
:SENS:PAR:AEX 1,1,1,1,"VARIABLE",10312500,100,PRBS31,0.80
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:OUTPut:CLOCk:OPERation

Function

Compatible command only for MP210xA, MP2100B: This command has the same function as :OUTPut:BITRate:STANdard.

Syntax

```
:OUTPut:CLOCk:OPERation <bitrate_standard>
:OUTPut:CLOCk:OPERation?
```

Parameter

<bitrate_standard> = <string>
The parameter values are the same as
for :{OUTPut|INPut}:BITRate:STANdard.

Example of Use

```
:OUTP:CLOC:OPER "10G_LAN"
:OUTP:CLOC:OPER?
>"10G_LAN"
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

OPE

Function

Compatible command only for MP210xA, MP2100B: This command has the same function as :OUTPut:BITRate:STANdard.

Syntax

```
OPE <bitrate_standard>
OPE?
```

Parameter

<bitrate_standard> = <integer>

Table A.5-1 Bit Rate Standards

<integer>	Standard	Bit Rate (bit/s)	Remarks
0	Variable-1/1	6.25G to 12.5G	
1	1GFC	1.0625G	*1
2	2GFC	2.125G	*1
3	4GFC	4.25G	
4	8GFC	8.5G	
5	10GFC	10.51875G	
6	10GFC FEC	11.3168G	
7	1GbE	1.25G	*1
8	2GbE	2.5G	*1
9	Infiniband	2.5G	*1
10	10GbE WAN/PHY	9.95328G	
11	10GbE LAN/PHY	10.3125G	
12	OTU1e (10GbE FEC)	11.049107G	
13	OTU2e (10GbE FEC)	11.095728G	

*1: If the Option 090 is not installed, the bit rate that is equal to or less than Variable-1/4 cannot be selected for the ED.

Table A.5-1 Bit Rate Standards (Cont'd)

<integer>	Standard	Bit Rate	Remarks
14	OC-3/STM-1	155.22M	*1
15	OC-12/STM-4	622.08M	*1
16	OC-48/STM16	2.488G	*1
17	OTU-1	2.666057G	*1
18	OC-192/STM-64	10.664228G	
19	G.975 FEC	10.709225G	
20	OTU2	10.709225G	
21	Variable-1/2	6.25G to 3.125G	
22	Variable-1/4	3.125G to 1.5625G	*1
23	Variable-1/8	1.5625G to 781.25M	*1
24	Variable-1/16	781.25M to 390.625M	*1
25	Variable-1/32	390.625M to 195.312M	*1
26	Variable-1/64	195.312M to 125M	*1
27	Infiniband x2	5G	
28	Infiniband x4	10G	
29	OC-24	1.244G	*1
30	CPRI	614.4M	*1
31	CPRI x2	1.2288G	*1
32	CPRI x4	2.4576G	*1
33	CPRI x5	3.072G	*1, *2
34	CPRI x10	6.144G	*2
35	OBSAI RP3	768M	*1, *2
36	OBSAI RP3 x2	1.536G	*1, *2
37	OBSAI RP3 x4	3.072G	*1, *2
38	OBSAI RP3 x8	6.144G	*2
39	Variable (9.5-14.2G)	9.5G to 14.2G	

*2: If the Option 090/092 is not installed, the bit rate cannot be selected because the bit rate range is limited.

Response Data

OPE <bitrate_standard> = <integer>

Example of Use

```
OPE 3
OPE?
>OPE 3
```

Note:

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

:OUTPut:CLOCk:FREQuency

Function

Compatible command: This command has the same function as :OUTPut:BITRate.

Syntax

```
:OUTPut:CLOCk:FREQuency <bitrate>
:OUTPut:CLOCk:FREQuency?
```

Example of Use

```
:OUTP:CLOC:FREQ 8500000
:OUTP:CLOC:FREQ?
>8500000
```

:OUTPut:CMU:FREQuency

Function

Compatible command: This command has the same function as :OUTPut:BITRate.

Syntax

```
:OUTPut:CMU:FREQuency <bitrate>
:OUTPut:CMU:FREQuency?
```

Example of Use

```
:OUTP:CMU:FREQ 8500000
:OUTP:CMU:FREQ?
>8500000
```

CRF

Function

Compatible command: This command has the same function as :OUTPut:BITRate.

Syntax

```
CRF <bitrate>
CRF?
```

Response Data

CRF <bitrate>

Example of Use

```
CRF 8500000
CRF?
>CRF 8500000
```

:OUTPut:CMU:RESolution

Function

Compatible command: This command is compatible with existing products, and sets and queries the unit of bit rate specified by the :OUTPut:CMU:FREQuency command.

Syntax

```
:OUTPut:CMU:RESolution KHZ|MHZ
:OUTPut:CMU:RESolution?
```

Parameter

KHZ|MHZ

Response Data

KHZ|MHZ

Example of Use

```
:OUTP:CMU:RES KHZ
:OUTP:CMU:RES?
>KHZ
```

Note:

Even if MHZ is specified by this command, the Bit Rate display in the screen will not change.

CRS

Function

Compatible command: This command has the same function as :OUTPut:CMU:RESolution.

Syntax

```
CRS 0|1
CRS?
```

Parameter

0 kHz
1 MHz

Response Data

CRS 0|1

Example of Use

```
CRS 0
CRS?
>CRS 0
```

:OUTPut:CLOCk:OFFSet:PPM

Function

Compatible command: This command has the same function as :OUTPut:BITRate:OFFSet.

Syntax

```
:OUTPut:CLOCk:OFFSet:PPM <numeric>
:OUTPut:CLOCk:OFFSet:PPM?
```

Example of Use

```
:OUTP:CLOC:OFFS:PPM 100
:OUTP:CLOC:OFFS:PPM?
>100
```

COP

Function

Compatible command: This command has the same function as :OUTPut:BITRate:OFFSet.

Syntax

```
COP <numeric>
COP?
```

Response Data

COP <integer>

Example of Use

```
COP 100
COP?
>COP 100
```

PTS

Function

Compatible command: This command has the same function as :SOURce:PATTern:TYPE and :SENSe:PATTern:TYPE.

Syntax

```
PTS <ppg_ed>,<pattern>
PTS? <ppg_ed>
```

Parameter

<ppg_ed>

0 PPG

1 ED

<pattern>

0 PRBS 2^7–1

1 PRBS 2^9–1

2 PRBS 2^15–1

3 PRBS 2^23–1

4 PRBS 2^31–1

5 Programmable Pattern

6 1/2 Clock Pattern

7 1/16 Clock Pattern

Response Data

PTS <pattern>

Example of Use

To set the ED test pattern to PRBS 2^15–1:

```
PTS 1,2
PTS? 1
>PTS 2
```

LGC

Function

Compatible command: This command has the same function as :SOURce:PATTern:LOGic and :SENSe:PATTern:LOGic.

Syntax

```
LGC <ppg_ed>,<pos_neg>
LGC? <ppg_ed>
```

Parameter

<ppg_ed>

0 PPG

1 ED

<pos_neg>

0 Positive

1 Negative

Response Data

LGC <pos_neg>

Example of Use

To set the test pattern logic of the PPG to the negative logic:

```
LGC 0,1
LGC? 0
>LGC 1
```

DLN?

Function

Compatible command: This command has the same function as :SOURce:PATTern:DATA:LENGth? and :SENSe:PATTern:DATA:LENGth?.

Syntax

```
DLN? <ppg_ed>
```

Parameter

<ppg_ed>

0 PPG

1 ED

Response Data

DLN <integer>

Example of Use

```
DLN? 1
>16384
```

DON

Function

Compatible command: This command has the same function as :OUTPut:DATA:OUTPut.

Syntax

```
DON 0|1
DON?
```

Parameter

0 Off

1 On

Response Data

DON 0|1

Example of Use

```
DON 1
DON?
>DON 1
```

DAP

Function

Compatible command: This command has the same function as :OUTPut:DATA:AMPLitude.

Syntax

```
DAP <numeric>
DAP?
```

Parameter

The parameter values are the same as for :OUTPut:DATA:AMPLitude.

Response Data

DAP <numeric>

Example of Use

```
DAP 0.5
DAP?
>DAP 0.5
```

DAT

Function

Compatible command: This command has the same function as :OUTPut:DATA:ATTFactor.

Syntax

```
DAT <numeric>
DAT?
```

Parameter

The parameter values are the same as for :OUTPut:DATA:ATTFactor.

Response Data

DAT <integer>

Example of Use

```
DAT 20
DAT?
>DAT 20
```

PRO?

Function

Compatible command: This command has the same function as :OUTPut:DATA:RELative?.

Response Data

PRO <numeric>

Example of Use

```
PRO?
>PRO 0.4
```

EAD

Function

Compatible command: This command has the same function as :SOURce:PATTern:EADDition:SET.

Syntax

```
EAD 0|1|7
EAD?
```

Parameter

0 Does not generate error
1 Generates error
7 Sets to generate single error

Response Data

EAD 0|1

Example of Use

```
EAD 1
EAD?
>EAD 1
```

EAV

Function

Compatible command: This command has the same function as :SOURce:PATTern:EADDition:VARiation.

Syntax

```
EAV 0|1
EAV?
```

Parameter

0 Repeat
1 Single

Response Data

EAV 0|1

Example of Use

```
EAV 0
EAV?
>EAV 0
```

ESI

Function

Compatible command: This command has the same function as :SOURce:PATTern:EADDition:SINGle.

Example of Use

```
ESI
```

ERT

Function

Compatible command: This command has the same function as :SOURce:PATTern:EADDition:RATE.

Syntax

```
ERT 1,<integer>
ERT?
```

Parameter

1: Indicates a fixed-point part of the error addition rate is 1.

<integer>

Exponent part of error addition rate

Range 2 to 12

Response Data

ERT 1,<integer>

<integer>

2 to 12

Example of Use

To set the error addition rate to 1E–9:

```
ERT 1,9
ERT?
>ERT 1,9
```

DSD

Function

Compatible command: This command has the same function as :INPut:DATA:INTerface.

Syntax

```
DSD 0|1|2|3
DSD?
```

Parameter

0 Electrical Single-Ended Data
1 Electrical Single-Ended XData
2 Differential 50 Ohm
3 Optical

Response Data

DSD 0|1|2|3

Example of Use

```
DSD 0
DSD?
>DSD 0
```

DTH

Function

Compatible command: This command has the same function as :INPut:DATA:THReshold.

Syntax

```
DTH <numeric>
DTH?
```

Parameter

The parameter values are the same as for :INPut:DATA:THReshold.

Response Data

DTH <integer>

Example of Use

```
DTH -85
DTH?
>DTH -85
```

SYN

Function

Compatible command: This command has the same function as :SENSe:PATTern:SYNC:ASYNc.

Syntax

```
SYN 0|1
SYN?
```

Response Data

SYN 0|1

Example of Use

```
SYN 1
SYN?
>SYN 1
```

SYE

Function

Compatible command: This command has the same function as :SENSe:PATTern:SYNC:THReshold.

Syntax

```
SYE <integer>
SYE?
```

Parameter

<integer>

0	1E–2
1	1E–3
2	1E–4
3	1E–5
4	1E–6
5	1E–7
6	1E–8
8	Internal

Response Data

SYE 0|1|2|3|4|5|6|8

Example of Use

```
SYE 0
SYE?
>SYE 0
```

SYM

Function

Compatible command: This command has the same function as :SENSe:PATTern:SYNC:PSMode.

Syntax

```
SYM 0|1
SYM?
```

Parameter

0 SYNC Control Off
1 SYNC Control On

Response Data

SYM 0|1

Example of Use

```
SYM 0
SYM?
>SYM 0
```

FPS

Function

Compatible command: This command has the same function as :SENSe:PATTern:SYNC:FPOSition.

Syntax

```
FPS <numeric>
FPS?
```

Parameter

<numeric>

Response Data

<integer>

Example of Use

```
FPS 1
FPS?
>FPS 1
```

HRE

Function

Compatible command: This command has the same function as :DISPlay:RESult:EALarm:HRESet.

Example of Use

```
HRE
```

MTR?

Function

Compatible command: This command has the same function as :CALCulate:DATA:MONitor?.

Syntax

```
MTR? 0|3|4
```

Parameter

0 Bit Error
3 CR Unlock
4 SYNC Loss

Response Data

MTR 0|1
0 When an alarm occurs
1 When no alarm occurs

Example of Use

```
MTR? 0
>MTR 0
```

ERS?

Function

Compatible command: This command is compatible with existing products and queries if a bit error has occurred at ED.

Response Data

ERS 0|1

0 Indicates a bit error has not occurred.

1 Indicates a bit error has occurred.

Example of Use

```
ERS?
>ERS 0
```

:SENSe:PARam:AEXecute?

Function

Compatible command only for MP210xA, MP2100B (version 4 or later):
This command is compatible with version 4 or later, and performs BER measurement of 10 ms duration at Ch1 and Ch2.

Syntax

```
:SENSe:PARam:AEXecute?
```

Response Data

<ed1_er>,<ed1_ec>,<ed1_unlock>,<ed1_sync>,<ed2_er>,<ed2_ec>,<ed2_unlock>,<ed2_sync>

<ed1_er>/<ed2_er>

Total bit error rate

Range: 0.0001E–18 to 1.0000E–00

(Form2)

<ed1_ec>/<ed2_ec>

Total bit error count

Range: 0 to 9999999 or 1.0000E+07 to 9.9999E+17

(Form1)

<ed1_unlock>/<ed2_unlock>

CR Unlock status

Range: "Not Occur" or "Occur"

<ed1_sync>/<ed2_sync>

SYNC Loss status

Range: "Not Occur" or "Occur"

Example of Use

To stop the re-plotting processing, and query the measurement results with a gating time of 10 ms:

```
:SYST:DISP:RES OFF
:SENS:PAR:AEX?
>"0.0000E-07","0","Not Occur","Not
Occur","1.0000E-02","850007","Not Occur","Not Occur"
```

Note:

If the measurement is performed with a gating time of 10 ms under the following conditions, an error occurs:

- When the re-plotting processing for the Scope screen is on
  Turn off the re-plotting processing for the Scope screen, and then send this command.

- When BER measurement is in progress
  Before executing this command, make sure that the measurement has completed, by using :SENSe:MEASure:EALarm:STATe? for example.

- When the user pattern is being recalled
  If one of the following pattern file recall commands is executed, read the status register to make sure the recall processing is completed.

  :SOURce:MMEMory:PATTern:RECall
  :SENSe:MMEMory:PATTern:RECall

When this command is used in MP2110A, an error (–113 Undefined Header) occurs because MP2110A does not support this command.

END?

Function

Compatible command: This command has the same function as :CALCulate:DATA:EALarm?.

Syntax

```
END? {0|1|2},{1|2|3|4}
```

Parameter

Specify the measurement data item to query, combining the first and second parameters.
Though CURRent or LAST can be specified for :CALCulate:DATA:EALarm?, this command queries only the results of the last measurement performed as specified by Gating Time.

0,1	Measurement Start Time
0,2	Measurement End Time
0,3	Measurement elapsed time
0,4	Measurement remaining time
1,3	SYNC Loss Seconds
1,4	CR Unlock Seconds
2,1	Bit Error Rate Total
2,2	Bit Error Count Total
2,3	Clock Count Total
2,6	Bit Error Rate Insertion
2,7	Bit Error Rate Omission
2,8	Bit Error Count Insertion
2,9	Bit Error Count Omission

Response Data

END <result>
Count, Seconds: 0 to 9999999 or 1.0000E+07 to 9.9999E+17 (Form1)
Rate: 0.0000E–18 to 1.0000E–00 (Form2)
"-------" if there is no data to return.

Start/end time:YY-MM-DD HH:MM:SS format
Elapsed/remaining time: D HH:MM:SS format
"ERR" if there is no data to return.

Example of Use

To query the measurement elapsed time:

```
END? 0,3
>END 0 00:00:25
```

To query the error rate:

```
END? 2,1
>END 1.0253E-06
```

ER?

Function

Compatible command: This command queries the bit error rate measured by the ED.

Response Data

ER <form2>|-------

<form2>: 0.0000E–18 to 1.0000E–00

-------: If there is no data.

Example of Use

```
ER?
>ER 3.8938E-02
```

EC?

Function

Compatible command: This command queries the bit error count measured by the ED.

Response Data

EC <form1>|-------

<form1>: 0 to 9999999 or 1.0000E+07 to 9.9999E+17

-------: If there is no data.

Example of Use

```
EC?
>EC 1.9469E+08
```

CC?

Function

Compatible command: This command queries the clock count measured by the ED.

Response Data

CC <form1>|-------

<form1>: 0 to 9999999 or 1.0000E+07 to 9.9999E+17

-------: If there is no data.

Example of Use

```
CC?
>CC 1.0000E-09
```

FRQ?

Function

Compatible command: This command queries the clock frequency measured by the ED.

Response Data

FRQ <integer>

Example of Use

```
FRQ?
>FRQ 8500000
```

MOD

Function

Compatible command: This command has the same function as :SENSe:MEASure:EALarm:MODE.

Syntax

```
MOD 0|1|2
MOD?
```

Parameter

0 Repeat
1 Single
2 Untimed

Response Data

MOD 0|1|2

Example of Use

To set the gating cycle of ED to Repeat:

```
MOD 0
MOD?
>MOD 0
```

PRD

Function

Compatible command: This command has the same function as :SENSe:MEASure:EALarm:PERiod.

Syntax

```
PRD <days>,<hours>,<minutes>,<seconds>
PRD?
```

Parameter

<days>,<hours>,<minutes>,<seconds>

Response Data

PRD <days>,<hours>,<minutes>,<seconds>
Each of the parameters in the response data is a two-digit number.

Example of Use

To set the gating time to 1 hour:

```
PRD 0,1,0,0
PRD?
>PRD 00,01,00,00
```

CUR

Function

Compatible command: This command has the same function as :DISPlay:RESult:EALarm:MODE.

Syntax

```
CUR 0|1
CUR?
```

Parameter

0 Off
1 On

Response Data

CUR 0|1

Example of Use

```
CUR 1
CUR?
>CUR 1
```

STA

Function

Compatible command: This command has the same function as :SENSe:MEASure:STARt.

Example of Use

```
STA
```

STO

Function

Compatible command: This command has the same function as :SENSe:MEASure:STOP.

Example of Use

```
STO
```

MSR?

Function

Compatible command: This command has the same function as :SENSe:MEASure:EALarm:STATe?.

Response Data

MSR 0|1

0 Measurement stops

1 During measurement

Example of Use

```
MSR?
>MSR 1
```

MSA?

Function

Compatible command: This command has the same function as :SENSe:MEASure:EALarm:STARt?.

Response Data

MSA <year>,<month>,<day>,<hour>,<minute>,<second>

0000,00,00,00,00,00 is returned if there is no time data to return.

Example of Use

```
MSA?
>MSA 2009,10,05,16,25,40
```

MSO?

Function

Compatible command: This command has the same function as :SENSe:MEASure:EALarm:STOP?.

Response Data

MSO <year>,<month>,<day>,<hour>,<minute>,<second>
0000,00,00,00,00,00 is returned if there is no time data to return.

Example of Use

```
MSO?
>MSO 2009,10,05,16,25,40
MSO?
>MSO 0000,00,00,00,00,00
```

MLP?

Function

Compatible command: This command has the same function as :SENSe:MEASure:EALarm:ELAPsed?.

Response Data

MLP <days>,<hours>,<minutes>,<seconds>

Example of Use

```
MLP?
>MLP 00,00,02,10
```

ETI?

Function

Compatible command: This command has the same function as :SENSe:MEASure:EALarm:TIMed?.

Response Data

ETI <days>,<hours>,<minutes>,<seconds>

Example of Use

```
ETI?
>ETI 00,00,02,10
```

:FETCh:AMPLitude:MEASurement?

Function

Compatible command (Version 6 or later): This command queries the results of all measurement items for the amplitude measurement function of Scope.

Syntax

```
:FETCh:AMPLitude:MEASurement?
```

Response Data

<current>,<average>,<std_dev>,<min>,<max>

The measurement results of the items specified at <meas_item> are returned at <numeric> in the following sequence.

Current value, average value, standard deviation value,
Minimum value, Maximum value

The response consists of 55 items and is output in the following order.

Table A.5-2 Data Order

Item	<current>	<average>	<std_dev>	<min>	<max>
One Level*1	1	12	23	34	45
Zero Level*1	2	13	24	35	46
Eye amplitude*1	3	14	25	36	47
Eye height*1	4	15	26	37	48
Crossing (%)	5	16	27	38	49
SNR	6	17	28	39	50
Average Power (dBm)*2	7	18	29	40	51
Average Power (mW)* 2	8	19	30	41	52
Extinction Ratio* 2	9	20	31	42	53
OMA (mW)* 2	10	21	32	43	54
OMA (dBm)* 2	11	22	33	44	55

*1: The unit for electrical channel is mV and the unit for optical channel is µW.

*2: When the optical channel is used, the value is returned. When the electrical channel is used, N/A is returned.

Example of Use

```
:FETC:AMPL:MEAS?
>35.12,-15.01,50.53,46.29,46,10.5,N/A,N/A,N/A,N/A,N/A,
36.98,-14.85,52.66,40.44,43.09,10.22,N/A,N/A,N/A,N/A,N/A,
0.23,0.15,1.46,1.25,0.69,0.38,N/A,N/A,N/A,N/A,N/A,
36.29,-15.3,48.28,36.39,41.02,9.08,N/A,N/A,N/A,N/A,N/A,
37.67,-14.4,57.04,44.19,45.16,11.36,N/A,N/A,N/A,N/A,N/A
```

Notes:

- *Only in MP210xA and MP2100B:* When Measure Item is not Amplitude/Time, N/As are returned as all response data.
- *Only in MP2110A:* The results of **OMA at Crossing** and **VECP** cannot be obtained.

:FETCh:TIME:MEASurement?

Function

Compatible command (Version 6 or later): This command queries the results obtained by the time measurement function of Scope.

Syntax

```
:FETCh:TIME:MEASurement?
```

Response Data

<current>,<average>,<std_dev>,<min>,<max>
The measurement results of the items specified at <meas_item> are returned at <numeric> in the following sequence.
Current value, average value, standard deviation value,
Minimum value, Maximum value
The response consists of 30 items and is output in the following order.

Table A.5-3 Data Order

Item	<current>	<average>	<std_dev>	<min>	<max>
Jitter p-p (ps)	1	7	13	19	25
Jitter RMS (ps)	2	8	14	20	26
Rise Time (ps)	3	9	15	21	27
Fall Time (ps)	4	10	16	22	28
Eye Width (ps)	5	11	17	23	29
DCD (%)	6	12	18	24	30

Example of Use

```
:FETC:TIME:MEAS?
>66.29,15.03,128.26,133.69,208.61,47.22,68.03,15.33,125.
99,134.01,203.98,47.01,1.99,0.31,2.21,1.86,3.19,0.55,62.
06,14.40,119.36,128.43,194.41,45.36,74.00,16.26,132.62,1
39.59,213.55,48.66
```

Notes:

- *Only in MP210xA and MP2100B:* When Measure Item is not **Amplitude/Time**, N/As are returned as all response data.
- *Only in MP2110A:* The jitter measurement results of MP2110A-096 cannot be obtained.

:MEASure:AMPLitude[:CHA|CHB]?

Function

Compatible command (Version 6 or later): This command sets **Eye** at Sampling Mode and **Amplitude/Time** at Measure Item of Scope and queries the result of the amplitude measurement.

Syntax

```
:MEASure:AMPLitude[:CHA|CHB]?
```

For Version 6 or later, this command can be used as a query by appending CHA|CHB.
For version 6 or older, CHA|CHB cannot be appended. The results of the active channel are obtained.

Response Data

The measurement results of the items in the following table are returned separated by commas. The results are output in the following sequence.

Table A.5-4 Response Data of Amplitude Measurement

One Level*1
Zero Level*1
Eye Amplitude*1
Eye Height*1
Crossing (%)
SNR
Average Power (mW)*2
Average Power (dBm)*2
Extinction Ratio (dB)*2
OMA (mW)*2
OMA (dBm)*2

*1: The unit for electrical channel is mV and the unit for optical channel is µW.

*2: When the optical channel is used, the value is returned. When the electrical channel is used, N/A is returned.

Example of Use

```
:MEAS:AMPL?
>35.2,-15.1,50.3,46.2,46,10.5,N/A,N/A,N/A,N/A,N/A
```

Notes:

- Before using this command, sets the Accumulation Type of the Setup dialog box to **Limited** and the measurement time to **Limit Type**.

- The response does not return during the set measurement time. To prevent occurrence of timeout errors, set the sufficiently-long timeout for remote interface, with respect to the measurement time.
- Use :FETCh:AMPLitude:<meas_item>? to obtain the current measurement results without performing the measurement.
- *Only in MP2110A:* The measurement results of **OMA at Crossing** and **VECP** cannot be obtained.

:MEASure:TIME?

Function

Compatible command (Version 6 or later): This command sets **Eye** at Sampling Mode and **Amplitude/Time** at Measure Item of Scope and queries the result of the time measurement.

Syntax

```
:MEASure:TIME?
```

Response Data

The measurement results of <numeric> for the 6 items separated by commas are output in the following sequence.

Jitter P-P (ps), Jitter RMS (ps), Rise Time (ps), Fall Time (ps), Eye Width (ps), DCD (%)

Example of Use

```
:MEAS:TIME?
>66.2,15.3,128.2,133.6,208.6,47.2
```

Notes:

- Before using this command, sets the Accumulation Type of the Setup dialog box to **Limited** and the measurement time to **Limit Type**.
- The response does not return during the set measurement time. To prevent occurrence of timeout errors, set the sufficiently-long timeout for remote interface, with respect to the measurement time.
- Use :FETCh:TIME:<meas_item>? to obtain the current measurement results without performing the measurement.

:MEASure:MASK?

Function

Compatible command (Version 6 or later): This command sets **Eye** at Sampling Mode and **Mask Test** at Measure Item of Scope and queries the result of the measurement.

Syntax

```
:MEASure:MASK?
```

Response Data

The measurement results of <integer> for the 2 items separated by commas are returned in the following sequence.

Total Samples, Total Failed Samples

Example of Use

```
:MEAS:MASK?
>16831,0
```

Notes:

- Before using this command, sets the Accumulation Type of the Setup dialog box to **Limited** and the measurement time to **Limit Type**.
- Each query does not return a response until the specified period of measurement time has elapsed. To prevent occurrence of timeout errors, set the sufficiently-long timeout for remote interface, with respect to the measurement time.
- Use :FETCh:MASK:<meas_item>? to obtain the current measurement results without performing the measurement.

:MEASure:HISTogram:AMPLitude?

Function

Compatible command (Version 6 or later): This command sets **Eye** at Sampling Mode and **Histogram** at Measure Item of Scope and queries the result of the amplitude axis histogram measurement.

Syntax

```
:MEASure:HISTogram:AMPLitude?
```

Response Data

<numeric>,<numeric>,<numeric>,<integer>
The response data is output in the following sequence:
Mean (mV/μW), Std Dev (mV/μW), P-P (mV/μW), Hits

Example of Use

```
:MEAS:HIST:AMPL?
>32.1,4.53,28.1,89632
```

Notes:

- Before using this command, sets the Accumulation Type of the Setup dialog box to **Limited** and the measurement time to **Limit Type**. And set the histogram measurement axis as the amplitude by :CONF:HIST:AXIS AMPLitude.
- The response does not return during the set measurement time. To prevent occurrence of timeout errors, set the sufficiently-long timeout for remote interface, with respect to the measurement time.
- Use :FETCh:HISTogram:AMPLitude:<meas_item>? to obtain the current measurement result without performing the measurement.

:MEASure:HISTogram:TIME?

Function

Compatible command (Version 6 or later): This command sets **Eye** at Sampling Mode and **Histogram** at Measure Item of Scope and queries the result of the time axis histogram measurement.

Syntax

```
:MEASure:HIStogram:TIME?
```

Response Data

<numeric>,<numeric>,<numeric>,<integer>
The response data is output in the following sequence:
Mean (ps), Std Dev (ps), P-P (ps), Hits

Example of Use

```
:MEAS:HIST:TIME?
>1.53,0.022,0.081,6831
```

Notes:

- Before using this command, sets the Accumulation Type of the Setup dialog box to **Limited** and the measurement time to **Limit Type**. And set the histogram measurement axis as the time by :CONF:HIST:AXIS TIME.
- The response does not return during the set measurement time. To prevent occurrence of timeout errors, set the sufficiently-long timeout for remote interface, with respect to the measurement time.
- Use :FETCh:HISTogram:TIME:<meas_item>? to obtain the current measurement result without performing the measurement.

Appendix B Error Codes

This appendix explains the code and message responses to the `SYSTem:ERRor?` query command.

- Command error
- Execution error
- Device unique error

When these errors occur, the standard event status register bit becomes 1. A service request can be generated when an error occurs depending on the setting of the standard event status enable register bit.

When an error occurs, the standard event status register bit that becomes 1 is listed in the table below.

Table B-1 Relationship between Error Number and Standard Event Register

Error Code	Message	Error Name	Standard Event Register Bit
–113	Undefined header	Command error	5
–220	Parameter error	Execution error	4
–310	System error	Device Dependent error	3

Command error

The corresponding bit is set to 1 if the received program message is undefined, does not conform to the syntax or has a misspelling. Bit 5 of the standard event status register is set when the following errors occur. The errors are generated when the following events occur.

- When sending message not in conformance with syntax described in section 2.7 “Message Format”
  Example: At typographical error in header
  Header includes 2-byte character

- When sending message not in conformance with Common Commands or Device Unique Commands described in Chapter 3 “Message Details” and Appendix A “Command Compatibility With Existing Products”

- When sending a module command, without specifying the module with :Module:ID

- When sending a message not described in Chapter 3
  Examples:
  - The message contains a typing error(s).
  - The message contains a double-byte character(s).
  - The space that should separate the message and parameter is missing.
  - The semicolon that should separate messages is missing.
  - The upper-case (mandatory) character(s) in the message has been omitted.

- When the number of parameters in the message is incorrect
  Examples:
  - Two parameters have been added to the message requiring one parameter when sending the message.
  - The comma(s) that should separate parameters is missing.

- When the parameter format is incorrect
  Examples:
  - Character-string data has been specified in the message requiring a numeric value as a parameter, when sending the message.
  - Character-string data is not enclosed by quotation marks.

Execution error

Bit 4 of the standard event status register is set when the following errors occur. The errors are generated when the following events occur.

- When header continuation parameter value out of setting range
  Example: When 850000 set when bit rate setting range is 8500000 to 11320000

- When message cannot be executed in current equipment status
  Example: When sending message for setting Scope to instrument without Scope function

Device Dependent error

Bit 3 of the standard event status register is set when an error occurs internally.

Appendix C Sample Program

Appendix C explains the sample program for controlling BERTWave remotely.
In MP2100B and MP2110A, the sample program is stored in the BERTWave built-in drive.

Copy the sample program to your PC to use it.

Storage location:
C:\Program Files\Anritsu\MP2100A\MX210000A\Examples\C#_SCPI_Sample

Refer to "Readme_en.pdf" (SCPI Sample Program Quick Reference Guide) stored in the above directory for details of the setup and method to execute the sample program.

Appendix D Bibliography

(1) IEEE488.1-1987 *IEEE Standard Digital Interface for Programmable Instrumentation -Description*

(2) IEEE488.2-1992 *IEEE Standard Codes, Formats, Protocols, and Common Commands for Use With IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation -Description*

(3) IVI Foundation *SCPI 1999*

(4) IEEE Std 802.3-2015 *IEEE Standard for Ethernet*

(5) Anritsu Corporation *SCPI Sample Program*
https://www.anritsu.com/en-US/test-measurement/support/downloads/manuals/dwl16980

(6) Anritsu Corporation *MX180000A Signal Quality Analyzer Control Software Remote Control Operation Manual*
https://www.anritsu.com/en-US/test-measurement/support/downloads/manuals/dwl16651

(7) Anritsu Corporation *Ideal Remote-Control Sequences*
https://www.anritsu.com/en-US/test-measurement/support/downloads/application-notes/dwl15661

Index

References are to page numbers.

A

Abbreviation 1-5

B

binary data..... 2-14
Bit Meaning of Scope Status Register 2-27
Bit Rate Standards..... 3-93

C

Checking Connection 2-10
Command..... 2-12
Command messages with no corresponding panel operation 3-40
Commands for Confirming Execution of operation at OperationStatus Register 2-25
Common Commands 2-16
Connecting Ethernet..... 2-5
Connecting GPIB..... 2-6

D

data parts..... 2-13
Device Dependent Commands..... 2-16
device dependent register 2-26

E

Ethernet Interface..... 2-4

G

GPIB Cable Connection 2-6
GPIB Interface 2-4

H

header 2-13

L

LAN crossover cable..... 2-5
LAN straight cable..... 2-5

M

Meaning of PPG/ED Ch1 to Ch4 Status Register 2-27
Meaning of Status Byte Register 2-20
Meaning of XFP/SFP+ Status Bit 2-27
Message Configuration 2-13
Message Format 2-12
Message Types..... 2-12
Messages Corresponding to Amplitude Dialog Box 3-36
Messages Corresponding to Amplitude/Time Measurement Result 3-17
Messages Corresponding to Amplitude/Time&Histogram Measurement Result..... 3-21
Messages Corresponding to Amplitude/Time&Mask Measurement Result 3-20
Messages Corresponding to Common Operations..... 3-4
Messages Corresponding to ED Panel 3-7
Messages Corresponding to Histogram Dialog Box 3-38
Messages Corresponding to Histogram Measurement Result 3-19
Messages Corresponding to Marker Dialog Box 3-37
Messages Corresponding to Marker Display3-19
Messages Corresponding to Mask Test Measurement Result 3-18
Messages Corresponding to Measure Dialog Box 3-26, 3-28
Messages Corresponding to Measure Dialog Box (Mask Test, Amplitude/Time&Mask)..... 3-27
Messages Corresponding to O/E Panel 3-14
Messages Corresponding to PPG Panel..... 3-6
Messages Corresponding to Scope Panel 3-15, 3-16
Messages Corresponding to Setup Dialog Box 3-24
Messages Corresponding to System Alarm Dialog Box 3-5
Messages Corresponding to Time Dialog Box 3-33
Messages Corresponding to XFP/SFP+ Panel 3-13

N

numeric data..... 2-14

O

operation status register 2-23
Operation Status Register 2-24

P

Program Messages 2-12

Q

Query 2-12

R

Register Bit Decimal Conversion Values 2-18
Register Structure 2-17
Response Messages 2-12
Rules for Describing Messages 3-3

S

Safety Symbols ii
Setting Interface 2-8
Standard event status enable register 2-21
Standard Event Status Register 2-22
Status Byte Register 2-19
String data 2-13