ADVANTEST®

ADVANTEST CORPORATION

D3371

Transmission Analyzer

Remote Programming Manual

MANUAL NUMBER FFE-8370629B00

© 2000 ADVANTEST CORPORATION
All rights reserved.

First printing October 30, 2000
Printed in Japan

Safety Summary

To ensure thorough understanding of all functions and to ensure efficient use of this instrument, please read the manual carefully before using. Note that Advantest bears absolutely no responsibility for the result of operations caused due to incorrect or inappropriate use of this instrument.

If the equipment is used in a manner not specified by Advantest, the protection provided by the equipment may be impaired.

- **Warning Labels**

  Warning labels are applied to Advantest products in locations where specific dangers exist. Pay careful attention to these labels during handling. Do not remove or tear these labels. If you have any questions regarding warning labels, please ask your nearest Advantest dealer. Our address and phone number are listed at the end of this manual.

  Symbols of those warning labels are shown below together with their meaning.

  DANGER: Indicates an imminently hazardous situation which will result in death or serious personal injury.

  WARNING: Indicates a potentially hazardous situation which will result in death or serious personal injury.

  CAUTION: Indicates a potentially hazardous situation which will result in personal injury or a damage to property including the product.

- **Basic Precautions**

  Please observe the following precautions to prevent fire, burn, electric shock, and personal injury.

  - Use a power cable rated for the voltage in question. Be sure however to use a power cable conforming to safety standards of your nation when using a product overseas.
  - When inserting the plug into the electrical outlet, first turn the power switch OFF and then insert the plug as far as it will go.
  - When removing the plug from the electrical outlet, first turn the power switch OFF and then pull it out by gripping the plug. Do not pull on the power cable itself. Make sure your hands are dry at this time.
  - Before turning on the power, be sure to check that the supply voltage matches the voltage requirements of the instrument.
  - Connect the power cable to a power outlet that is connected to a protected ground terminal. Grounding will be defeated if you use an extension cord which does not include a protected ground terminal.
  - Be sure to use fuses rated for the voltage in question.
  - Do not use this instrument with the case open.
  - Do not place anything on the product and do not apply excessive pressure to the product. Also, do not place flower pots or other containers containing liquid such as chemicals near this

product.

- When the product has ventilation outlets, do not stick or drop metal or easily flammable objects into the ventilation outlets.
- When using the product on a cart, fix it with belts to avoid its drop.
- When connecting the product to peripheral equipment, turn the power off.

- **Caution Symbols Used Within this Manual**

Symbols indicating items requiring caution which are used in this manual are shown below together with their meaning.

DANGER: Indicates an item where there is a danger of serious personal injury (death or serious injury).

WARNING: Indicates an item relating to personal safety or health.

CAUTION: Indicates an item relating to possible damage to the product or instrument or relating to a restriction on operation.

- **Safety Marks on the Product**

The following safety marks can be found on Advantest products.

 : ATTENTION - Refer to manual.

 : Protective ground (earth) terminal.

 : DANGER - High voltage.

 : CAUTION - Risk of electric shock.

- **Replacing Parts with Limited Life**

The following parts used in the instrument are main parts with limited life.
Replace the parts listed below before their expected lifespan has expired to maintain the performance and function of the instrument.
Note that the estimated lifespan for the parts listed below may be shortened by factors such as the environment where the instrument is stored or used, and how often the instrument is used.
The parts inside are not user-replaceable. For a part replacement, please contact the Advantest sales office for servicing.

Each product may use parts with limited life.
For more information, refer to the section in this document where the parts with limited life are described.

Main Parts with Limited Life

Part name	Life
Unit power supply	5 years
Fan motor	5 years
Electrolytic capacitor	5 years
LCD display	6 years
LCD backlight	2.5 years
Floppy disk drive	5 years
Memory backup battery	5 years

- **Hard Disk Mounted Products**

The operational warnings are listed below.

- Do not move, shock and vibrate the product while the power is turned on.
  Reading or writing data in the hard disk unit is performed with the memory disk turning at a high speed. It is a very delicate process.

- Store and operate the products under the following environmental conditions.
  An area with no sudden temperature changes.
  An area away from shock or vibrations.
  An area free from moisture, dirt, or dust.
  An area away from magnets or an instrument which generates a magnetic field.

- Make back-ups of important data.
  The data stored in the disk may become damaged if the product is mishandled. The hard disc has a limited life span which depends on the operational conditions. Note that there is no guarantee for any loss of data.

- **Precautions when Disposing of this Instrument**

When disposing of harmful substances, be sure dispose of them properly with abiding by the state-provided law.

Harmful substances: (1) PCB (polycarbon biphenyl)
(2) Mercury
(3) Ni-Cd (nickel cadmium)
(4) Other
Items possessing cyan, organic phosphorous and hexadic chromium and items which may leak cadmium or arsenic (excluding lead in solder).

Example: fluorescent tubes, batteries

Environmental Conditions

This instrument should be only be used in an area which satisfies the following conditions:

- An area free from corrosive gas
- An area away from direct sunlight
- A dust-free area
- An area free from vibrations
- Altitude of up to 2000 m

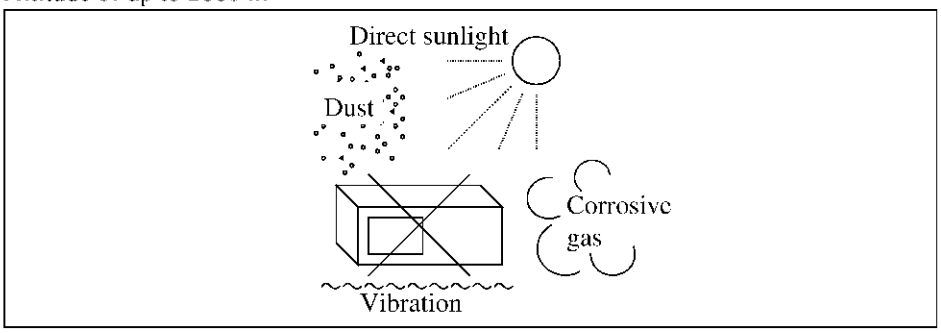


Figure-1 Environmental Conditions

- Operating position

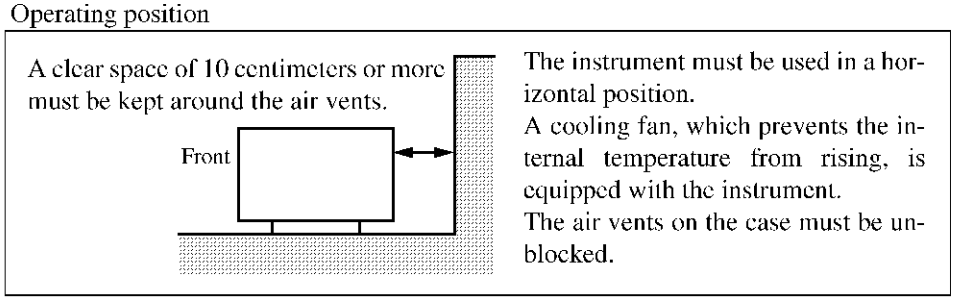


Figure-2 Operating Position

- Storage position

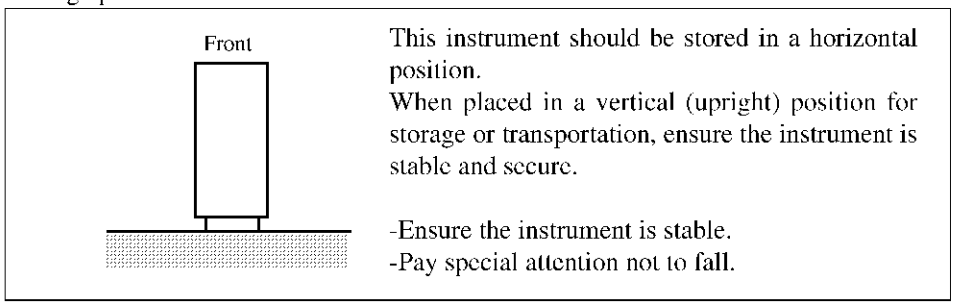


Figure-3 Storage Position

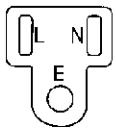
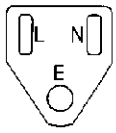
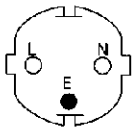
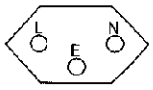
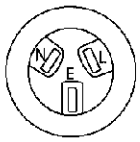
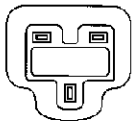
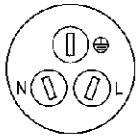
- The classification of the transient over-voltage, which exists typically in the main power supply, and the pollution degree is defined by IEC61010-1 and described below.

  Impulse withstand voltage (over-voltage) category II defined by IEC60364-4-443

  Pollution Degree 2

Types of Power Cable

Replace any references to the power cable type, according to the following table, with the appropriate power cable type for your country.

Plug configuration	Standards	Rating, color and length	Model number (Option number)
	PSE: Japan Electrical Appliance and Material Safety Law	125 V at 7 A Black 2 m (6 ft)	Straight: A01402 Angled: A01412
	UL: United States of America CSA: Canada	125 V at 7 A Black 2 m (6 ft)	Straight: A01403 (Option 95) Angled: A01413
	CEE: Europe DEMKO: Denmark NEMKO: Norway VDE: Germany KEMA: The Netherlands CEBEC: Belgium OVE: Austria FIMKO: Finland SEMKO: Sweden	250 V at 6 A Gray 2 m (6 ft)	Straight: A01404 (Option 96) Angled: A01414
	SEV: Switzerland	250 V at 6 A Gray 2 m (6 ft)	Straight: A01405 (Option 97) Angled: A01415
	SAA: Australia, New Zealand	250 V at 6 A Gray 2 m (6 ft)	Straight: A01406 (Option 98) Angled: ---------
	BS: United Kingdom	250 V at 6 A Black 2 m (6 ft)	Straight: A01407 (Option 99) Angled: A01417
	CCC:China	250 V at 10 A Black 2 m (6 ft)	Straight: A114009 (Option 94) Angled: A114109

PREFACE

This manual describes remote programming of the D3371 transmission analyzer.

The manuals of the analyzer consist of this volume and two other manuals. Additional information on the D3371 should be referred to as required.

D3371 Transmission Analyzer Operation Manual

This manual provides information on how to operate the D3371, the names for each part and the functional descriptions. To safely the analyzer, be sure to read this manual before using it (Separate volume).

1. Trademark

Microsoft Windows 98 is a trademark of Microsoft Corporation, registered in the U.S.

2. Organization

This manual consists of the following chapters:

Safety Summary	To use the analyzer safely, be sure to read this manual first.
1. INTRODUCTION	Outlines the GPIB interface and explains how it is connected.
2. GPIB BUS FUNCTIONS	Describes the GPIB bus functions of the D3371.
3. COMMAND SYNTAX	Describes the command syntax.
4. STATUS BYTES	Describes the operating models and event assignments of the status bytes.
5. COMMAND REFERENCE	Describes the configuration of subsystems, and gives precautions on the command syntax. Also, the commands are described in detail.
6. SAMPLE PROGRAMS	Presents sample programs used for remote controls over the GPIB used in the transmission analyzer.
A.1 LIST OF COMMANDS	Lists commands.
A.2 COMMANDS USED ON SETTING MENUS	Lists the commands used on setting menus.
A.3 DEFAULTS	Lists the default values.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

The D3371 transmission analyzer includes as standard the General Purpose Interface Bus (GPIB) interface, which complies with IEEE standard 488.1-1987 and 488.2-1987. The interface permits remote control from an external controller.

The chapter outlines the GPIB interface and explains how it is connected.

1.1 GPIB

The GPIB provides a high-performance bus for connecting computers and measurement instruments.

GPIB operation is defined by IEEE standard 488.1-1987. The GPIB, an interface with a bus structure, allows individual devices to have unique device addresses so that a particular device is specified. Up to 15 devices can be connected in parallel to a single bus. GPIB devices have at least one of the following functions:

- Talker: A device specified to send data to the bus is referred to as a talker. On the GPIB bus, only one device operates as an active talker.
- Listener: A device specified to receive data on the bus is referred to as a listener. There may be multiple active listener devices on the GPIB bus.
- Controller: A device that specifies a talker and a listener is referred to as a controller. On the GPIB bus, only one device operates as an active controller. In particular, a controller device that can control IFC and REN messages is referred to as a system controller.

  Only one system controller is permitted on the GPIB bus. At system startup, when more than one controller is connected to the bus, the system controller operates as the active controller, and the other devices that have controller capabilities operate as addressable devices.

  The active controller can use the Take Control (TCT) interface message to make another controller active. The controller that has used this message then becomes a non-active controller.

  A controller controls an overall system by sending interface messages or device messages to measurement instruments.

  These messages function as follows:

  - Interface messages, which control the GPIB
  - Device messages, which control a measurement instrument

1.2 Command Mode

The D3371 operates in the following command mode:

- Command mode defined by IEEE standard 488.2-1987

The 488.2-1987 standard expands the definitions of the following items appearing in 488.1-1987:

- Syntax for programming measurement instruments
- Communication protocol (procedure) for commands and data
- Common commands (*1)
- Status data structure
- System synchronization protocol

*1: Common commands are commands that work in the same way on any measurement instrument.

1.3 GPIB Setup

The following setup operations are required to use the GPIB interface:

- Connecting the GPIB
- Setting a GPIB address

1.3.1 Connecting the GPIB

A typical GPIB connection method is shown below. Secure the GPIB connector with two screws so that the connector does not come loose during use.

CAUTION: ***Before attaching the GPIB cable to the D3371, be sure to turn off the power.***

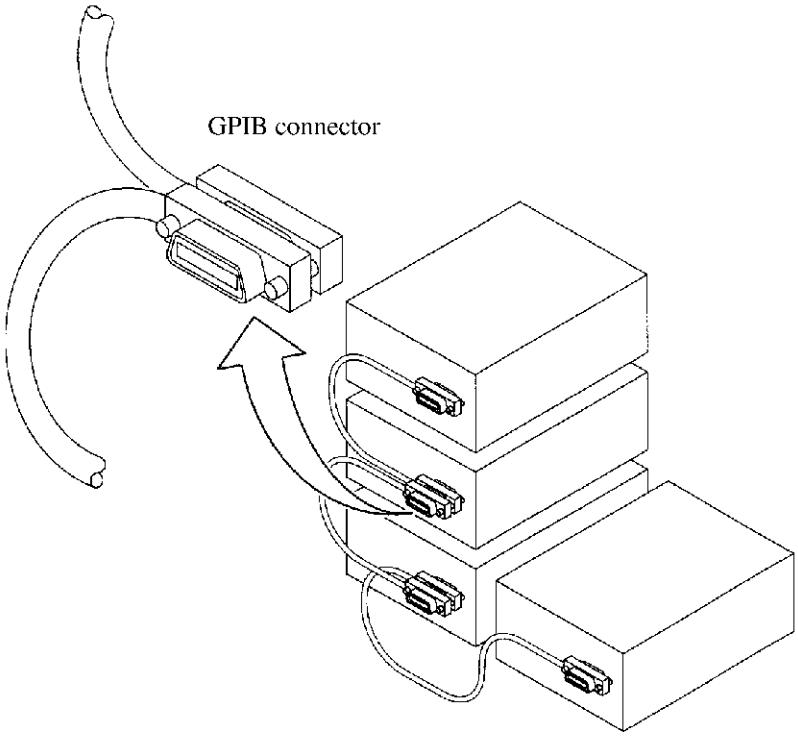


Figure 1-1 Connecting the GPIB cable

Observe the following when using the GPIB interface:

- The maximum total length of the GPIB cables used in one bus system is 2 m multiplied by the number of connected devices (each GPIB controller is also counted as one device). The maximum total cable length is 20 m.
- Up to 15 devices can be connected to one bus system.
- Although there are no restrictions on cable connections, do not stack four or more GPIB connectors on a single device. Stacking more than three GPIB connectors may apply excessive force to the mounting part of the connector, resulting in damage.

Example: When a system consists of five devices, the maximum total cable length is 10 m (five devices multiplied by 2 m/device = 10 m). You may vary the lengths individual cables if the total cable length is within the maximum allowable length. When connecting 10 or more devices, however, you need to connect some devices with 2 m or shorter cables so that the total cable length does not exceed 20 m.

1.3.2 Setting a GPIB Address

This section explains how to set a GPIB address for the D3371. For details, refer to "D3371 Transmission Analyzer Operation Manual."

1. Turn on the power.
   The measurement application of this device starts up.
2. Check that the D3371 is in the local state.
   To confirm the local state, check that the **REMOTE** lamp located on the left side of the front panel is off.
   If the D3371 is not in the local state, press the **LOCAL** key on the left side of the front panel to place the D3371 in the local state.
3. Click the **[Settings]** button on the standard toolbar.
   The Settings dialog box appears.
4. Click **[System]** on the module selection list bar.
5. Select **[GPIB]** from the **[Remote Interface]** drop-down list box.
6. In the **[Address]** text box, enter the GPIB address to be set.
7. Press the **[OK]** button.
   The Settings window closes, and the GPIB address is set.

2. GPIB BUS FUNCTIONS

This chapter describes the GPIB bus functions of the D3371.

2.1 GPIB Interface Functions

Code	Description
SH1	Source handshake function is available.
AH1	Acceptor handshake function is available.
T6	Basic talker function, serial polling function, turning the talker state off by the listener
TE0	Extended talker function is not available.
L4	Basic listener function, turning the listener state off by the talker
LE0	Extended listener function is not available.
SR1	Service request function is available.
RL1	Remote function, local function, local lockout function
PP0	Parallel poll function is not available.
DC1	Device clear function
DT0	Device trigger function is not available.
C0	Controller function is not available.
E1	Open collector bus driver is used.

2.2 D3371 Response to Interface Messages

This section describes how the D3371 responds to interface messages. The responses to interface messages are defined by IEEE standard 488.1-1987 and 488.2-1987.

For information on sending interface messages to the D3371, refer to the operation manual supplied with the controller used.

2.2.1 Interface Clear (IFC)

This message is sent directly to the D3371 over a signal line.

In response to this message, the D3371 stops operation of the GPIB bus. The D3371 stops all I/O operations, but does not clear the I/O buffers. (The I/O buffers are cleared by DCL.)

2.2.2 Remote Enable (REN)

This message is sent directly to the D3371 over a signal line. If this message is true, the D3371 is placed in the remote state when the D3371 is specified as a listener. This state is maintained until the D3371 receives GTL, REN becomes false, or the **LOCAL** key is pressed.

When the D3371 is in the local state, it ignores all data that it receives.

In the remote state, D3371 transmission analyzer applications ignore all key entries except the **LOCAL** key.

In the local lockout state (refer to Section 2.2.7, "Local Lockout (LLO)"), all key entries are ignored.

2.2.3 Serial Poll Enable (SPE)

When the D3371 receives this message from an external device, it enters serial polling mode.

When the D3371 is specified as the talker after it enters serial polling mode, the D3371 sends a status byte instead of ordinary messages. This mode is maintained until the D3371 receives a serial polling disable (SPD) message or IFC message.

While the D3371 is sending a service request (SRQ) message to the controller, bit 6 of the response data (RQS bit) is set to 1 (true). When the D3371 completes sending, the RQS bit becomes 0 (false).

The service request (SRQ) message is sent directly over a signal line.

2.2.4 Device Clear (DCL)

When the D3371 receives DCL, it performs the following operations:

- Clearing the input and output buffers
- Resetting the syntax analysis section, execution control section, and response data generation section
- Canceling any command that presents a remote command that should be executed next
- Canceling other commands that have been stopped temporarily to wait for parameters
- Canceling OPC and OPC?

The D3371 does not perform the following operations:

- Changing data that has been set or stored in the D3371
- Canceling the front panel operation
- Affecting or interrupting the current D3371 operation
- Changing the status byte except MAV (MAV is set to 0 as a result of the output buffer clear operation)

2.2.5 Selected Device Clear (SDC)

The D3371 responds in the same manner as when DCL is received. SDC is executed only when the D3371 functions as a listener.

In other cases, SDC is ignored.

2.2.6 Go To Local (GTL)

This message places the D3371 in the local state. In the local state, all front panel operations become valid.

2.2.7 Local Lockout (LLO)

This message places the D3371 in the local lockout state. When the D3371 enters the remote state after it is placed in the local lockout state, all front panel operations are disabled. (In the normal remote state, you can operate the front panel by pressing the LOCAL key.)

In this case, the D3371 can be placed in the local state with the following operations:

- Sending the GTL message to the D3371
- Making the REN message false (as a result, the local lockout state is also released)

2.3 Message Exchange Protocol

The D3371 receives program messages from controllers and other devices through the GPIB bus and generates response data. Program messages include commands, queries (commands inquiring response data are referred to as queries), and data. Such data is exchanged according to the protocol. This section explains the protocol.

The D3371 sends and receives messages according to a message exchange protocol that conforms to IEEE standard 488.2-1987.

2.3.1 GPIB Buffers

The D3371 has the following three types of buffers:

- Input buffer

  The input buffer is used to temporarily store data for command analysis.
  (The input buffer is 1024 bytes.)

  The input buffer is cleared by the following operations:

  - Turning on the power
  - Executing DCL or SDC

- Output buffer

  The output buffer is used to store data until it is read by the controller.
  (The output buffer capacity is 1024 bytes.)

  The output buffer is cleared by the following operations:

  - Turning on the power
  - Executing DCL or SDC

- Error queue

  The error queue is used to hold error messages for remote commands.
  (The queue has a depth of 10 levels.)

  Each time an error occurs during the analysis or execution of a remote command, a message is placed in the queue.

  Error messages can be read using the SYST:ERR? query. As one error message is read, the message is deleted from the queue.

  The error queue is cleared by the following operations:

  - Turning on the power
  - Executing *CLS

2.3.2 IEEE488.2-1987 Command Mode

The IEEE488.2-1987 command mode enables you to send and receive messages using a message exchange protocol which complies with IEEE standard 488.2-1987.

Note the following points when another controller or device receives a message from this device.

- Response data is generated when a query is received.
- Data is generated in the order that the queries are executed.

The parser receives command messages from the input buffer in the order that they are received, performs syntax analysis, and determines the operation to be performed by a received command.

During command syntax analysis, the parser also traces the command tree structure.

For analysis of the next command, the parser records information on the position in the tree structure from which the analysis should be started.

When the parser is cleared, this information points to the top of the tree structure.

The parser is cleared in one of the following four ways:

- Turning the power on
- Receiving DCL or SDC
- Receiving a colon (:) followed by a semicolon (;)
- Receiving the terminator or EOI

When the parser executes a query, data is generated in the output buffer as a response (i.e., a query must always be sent immediately before this device outputs data.)

This means that data generated by a query is not cleared unless the controller reads it.

Two conditions other than a read operation by the controller clear data cause a query error.

- Unterminated condition:

  The controller reads response data without terminating a query (LF code in ASCII or the END message of the GPIB), or without sending a query.

- Interrupted condition:

  When this device receives the next program message before the controller reads the response data.

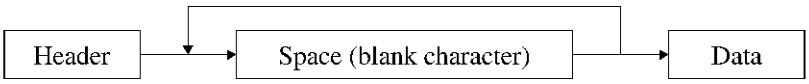
3. COMMAND SYNTAX

Except for character string data and block data, you can enter commands with either uppercase or lowercase characters.

This Chapter describes the command syntax.

3.1 Command Syntax

The following format defines the command syntax.



1. Header

   The header has a hierarchical structure consisting of multiple mnemonics delimited by colons (:). A mnemonic composed of four or more characters has an abbreviated short form consisting of four (or three) characters. (A non-abbreviated mnemonic is called a "long form.")

   Mnemonics in either form can be used in any combination.

2. Space (blank character)

   More than one space can be used. If a character other than a space is specified, an error occurs.

3. Data

   When a command requires more than one data item, the data items are delimited with a comma (,). A space (blank character) may be inserted before and after a comma (,).

   For information on data types, refer to Section 3.2, "Data Format."

4. Describing multiple commands

   More than one command can be described on one line by delimiting the commands with a semicolon (;).

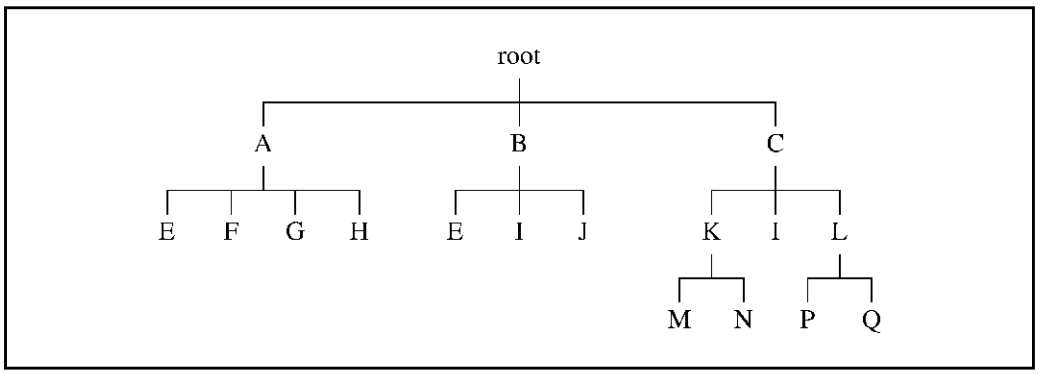
   In this case, commands are executed, moving through the current path according to the hierarchical structure contained in the header.

5. Current path movement

The current path is moved according to the following rules:

- At power-up: The current path is set to the root.
- Termination: The current path is set to the root.
- Colon (:): The current path is moved one level lower in a command tree. When a command begins with a colon (:), the colon sets the current path to the root.
- Semicolon (;): The current path does not change.
- Common command: Common commands can be executed regardless of the current path. Executing the *RST command sets the current path to the root.

Example: Assume the following header structure.



The current path operation is as follows.

- :A:E;:B:E

  The colon (:) in the second command moves the current path to the root, so A:E and B:E are both correct commands.

- :A:E<END>B:E

  <END> (terminator) moves the current path to the root, so A:E and B:E are both correct commands.

- :A:E;F;G;H

  A semicolon (;) does not move the current path, so the result of :A:E;F;G;H equals the result of the four commands A:E, A:F, A:G, and A:H.

- :C:I;K:N;M

  A colon (:) moves the current path, so K:N is viewed from the level of :C:. As a result, K:N is regarded as C:K:N. At the same time, K:N includes a colon (:), so the current path is changed to :C:K, and the last M is interpreted as C:K:M.

- :A:E;*ESE 16

  Common commands are not related to the current path, so *ESE 16 is executed correctly.

- :A:E;*ESE 16;F;G;H

  Common commands do not change the current path, so the third command F is searched for by the current path :A: set by the first command :A:E. As a result, F becomes A:F, G becomes A:G, and H becomes A:H.

  In the following examples, a syntax error occurs.

- :A:E;B:E

  A:E changes the current path to :A:.

  Therefore, B:E is searched for on the level of :A:, but because mnemonic B is not found, an error occurs.

- :C:K:M;L:P

  :C:K:M changes the current path to :C:K:.

  Therefore, L:P is searched for with :C:K:, and an error occurs because mnemonic L is not found.

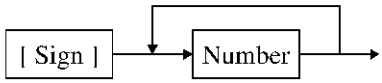
3.2 Data Format

The D3371 uses the data types shown below to input and output data.

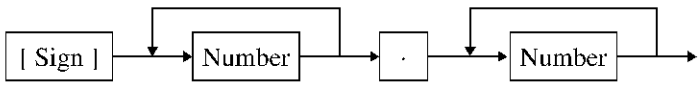
1. Numeric data

   There are three formats for numeric data as shown below. Any of the three formats may be used to input numeric values to the D3371 (the input data is rounded off as determined by the input data type).

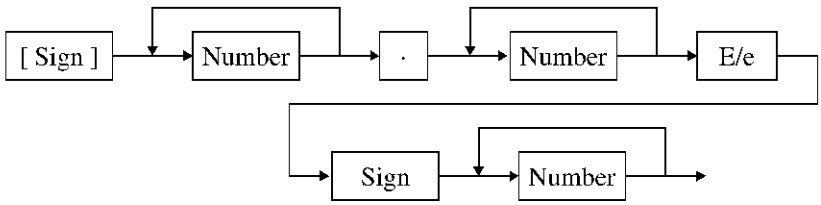
   - Integer type: NR1 format NOTE: The sign enclosed in brackets [] may be omitted.



   - Fixed-point type: NR2 format

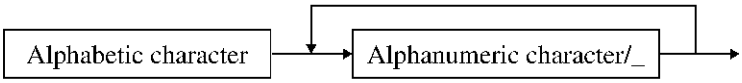


   - Floating-point type: NR3 format



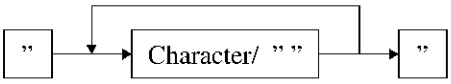
2. Character data

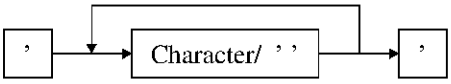
   The character data format is shown below.



3. Character-string data

   Character-string data has the following two formats.





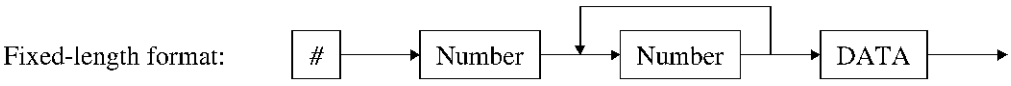
   In character-string data, characters can be used as ASCII 7 bit code characters.

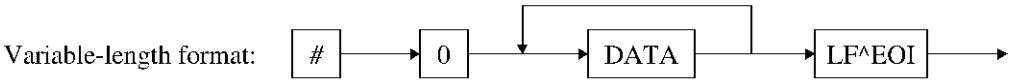
NOTE: ***In character-string data beginning with a double quotation mark ("), the double quotation mark (") must be represented by two consecutive double quotation marks (" ").***
In character-string data beginning with a single quotation mark ('), the single quotation mark (') must be represented by two consecutive single quotation marks (' ').

   When response data is character-string data, the character-string data to be output must always begin with a double quotation mark (").

4. Block data

   Block data has two formats. Either can be used to input block data to the D3371.





   In fixed-length format, the numeric character following # indicates the number of bytes in the subsequent data. 0 cannot be used (0 indicates variable-length format).

   Example: Block data #3128 <data byte>
   The number 3 following # indicates the number of the digits of the subsequent character string (128), and the number 128 indicates the number of the bytes of the subsequent <data byte>.

3.3 Delimiter (Terminator)

1. When entering a command
   When a GPIB command is read, it is always terminated with LF or EOI.
2. When responding to a query
   When the result of a query is output, EOI is always output together with LF.

NOTE: ***LF = Line feed (0A in hexadecimal)***
EOI = Single-line signal (End Or Identify)

---

4. STATUS BYTES

The D3371 has a hierarchical status register structure that complies with IEEE standard 488.2-1987, and can send various device statuses to the controller. This chapter describes the status byte operation model and event assignment.

4.1 Status Registers

4.1.1 Status Register Structure

The D3371 employs the status register model defined by IEEE standard 488.2-1987. This model consists of a condition register, transition filter, event register, and enable register.

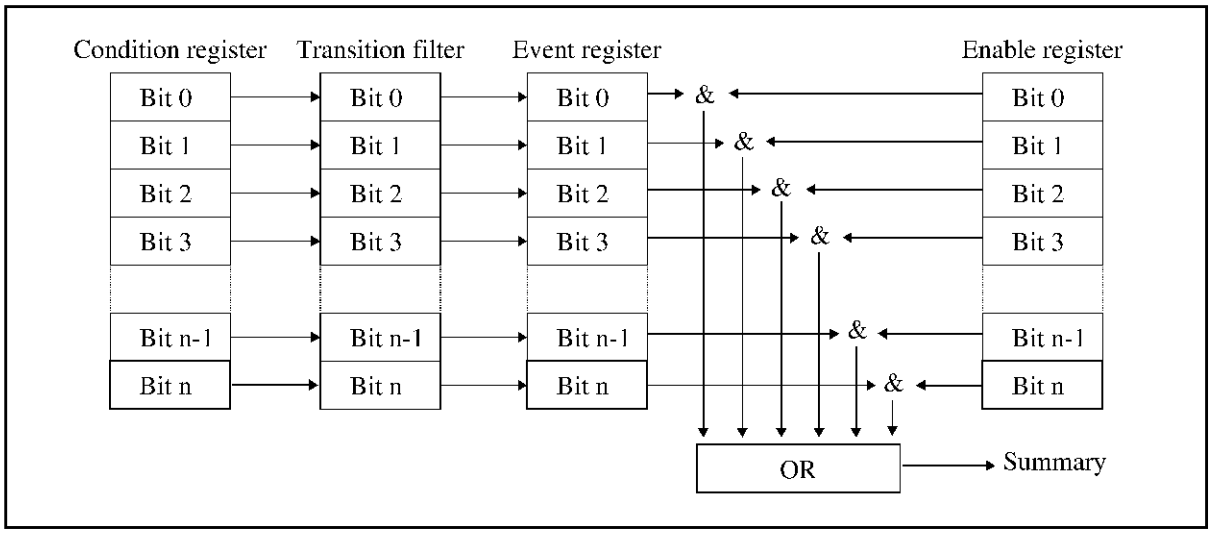


Figure 4-1 Status Register Structure

1. Condition register

   The condition register monitors the latest status of a device for each condition bit. This register does not save the status in memory but changes in real time. If an event corresponding to a bit occurs, the bit is set to 1. No data can be written to the condition register.

2. Transition filter

   The transition filter specifies whether to set an event bit to 1 if a condition bit changes in the positive or negative direction. To define an event bit as 1, set a bit corresponding to the event to 1. Data can be written to the transition filter.

3. Event register

   If a change specified in the transition filter occurs in the condition register, the corresponding event bit of the event register is set to 1. If the condition register or transition filter does not exist or is invalid, the event bit is set to 1 when the corresponding event occurs. This register is cleared if it is read using a query or if *CLS is executed. No data can be written to the event register.

4. Enable register

   The enable register specifies which event bit of the event register is assumed to be valid when a summary is generated. Each bit of the enable register is ANDed with each bit of the event register, and the results are ORed to generate the summary. The summary is written to the next status register. To generate a summary for an event bit, set the corresponding bit to 1. Data can be written to the enable register.

4.1.2 Types of Status Registers

The D3371 has seven types of status registers as follows.

1. Status byte register: Refer to Section 4.2.
2. Standard event status register: Refer to Section 4.3.
3. Operation status register: Refer to Section 4.4.
4. Questionable status register: Refer to Section 4.5.
5. Settling status register: Refer to Section 4.6.
6. Clock loss status register: Refer to Section 4.7.
7. Device status register: Refer to Section 4.8.

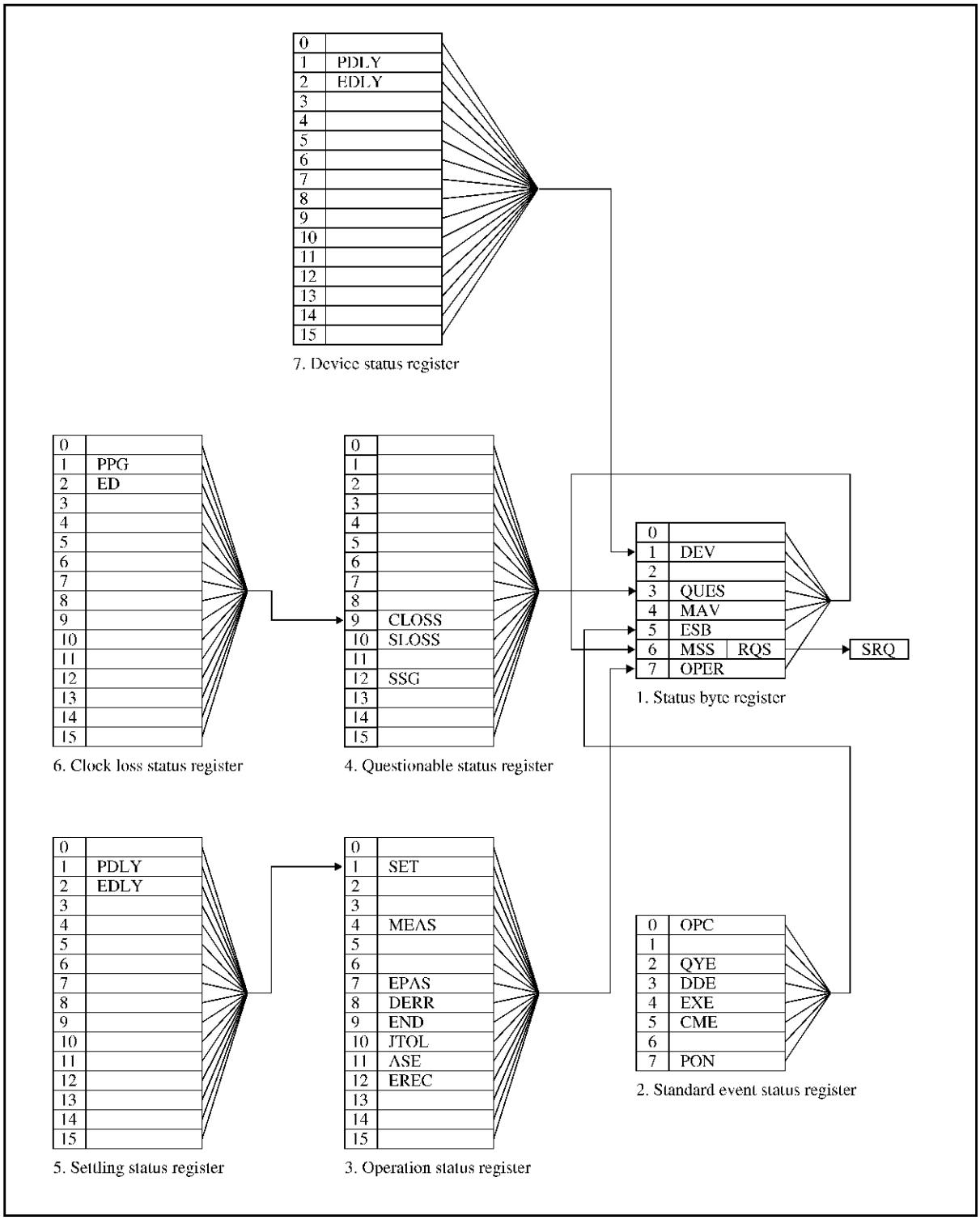


Figure 4-2 Detailed Status Register Configuration

4.1.2 Types of Status Registers

The status register configuration for the transmission analyzer is shown in Figure 4-3.

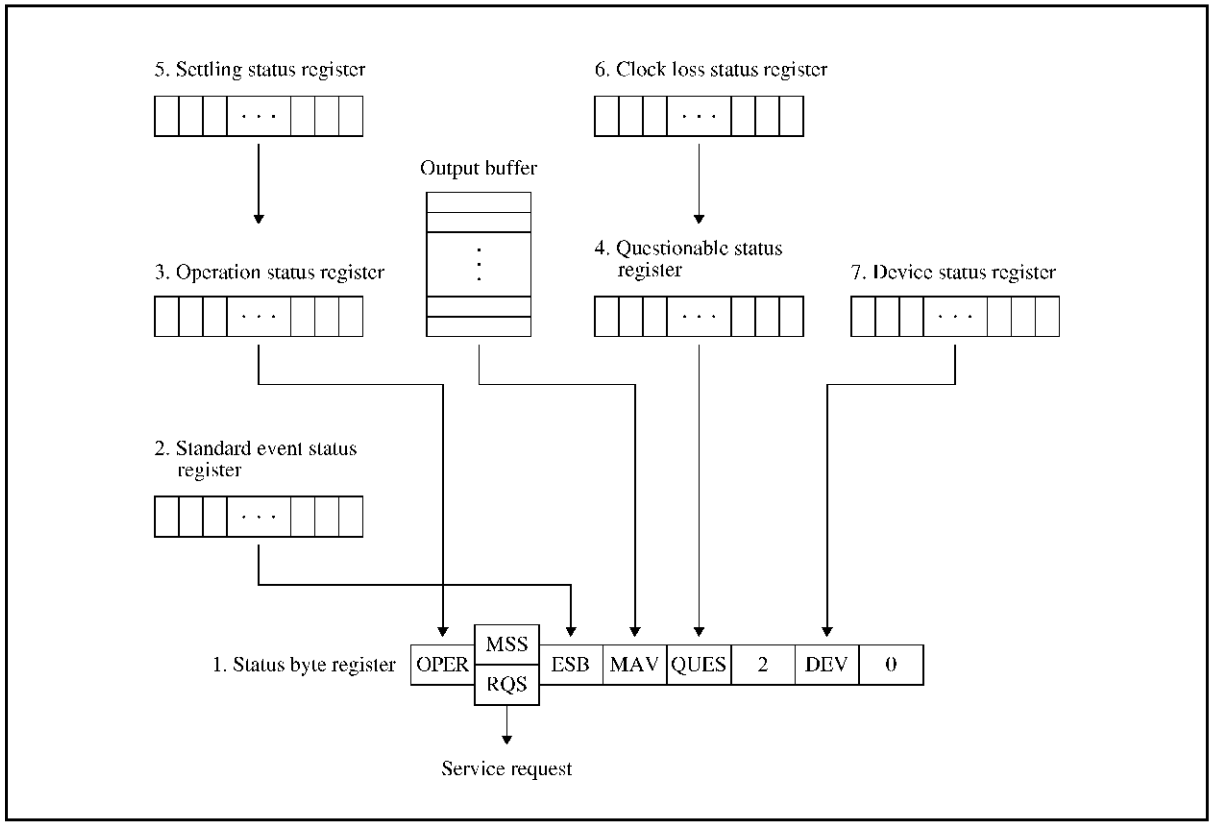


Figure 4-3 Outline of the Status Register Configuration

4.2 Status Byte Register

The status byte register summarizes the information from the status registers (refer to Section 4.1.1). The summary of the status byte register is sent as a service request to the controller. The status byte register therefore operates in a way with a slightly different structure than the status register. This section describes the status byte register.

The following diagram shows the structure of the status byte register.

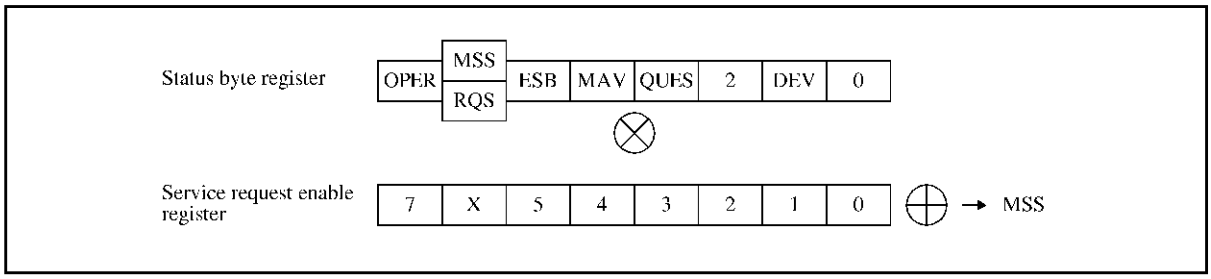


Figure 4-4 Status Byte Register Structure

Except for the following three points, the status byte register has the same structure as the status register structure described in Section 4.1.1.

1. The summary of the status byte registers is written to bit 6 of the status byte register.
2. Bit 6 of the enable register is always valid and cannot be changed.
3. Bit 6 (MSS) of the status byte register is written to service request RQS.

The status byte register responds to serial polling from the controller. When a response to a serial polling is made, bits 0 to 5 and bit 7 of the status byte register and RQS are read, and then RQS is reset to 0. The other bits are not cleared unless the corresponding sources are set to 0.

The status byte register, RQS, and MSS are cleared by executing *CLS.

The bit assignments of the standard status byte register are shown below.

This register provides summaries of other registers.

Bit	Mnemonic	Description
7	OPER	OPER is a summary of the operation status register.
6	RQS/MSS	RQS is true when MSS of the status byte register is set to 1. MSS is a summary bit of the entire status data structure.
		MSS is not read by serial polling (MSS is set to 1 if RQS is 1).
		To read MSS, use common command *STB?. *STB? reads bit 0 to 5, and bit 7 of the status byte register and MSS. *STB? does not clear the status byte register and MSS.
		MSS is not set to 0 until all unmasked sources in the status register structure are cleared.
5	ESB	ESB is a summary of the standard event status register.
4	MAV	MAV is a summary bit of the output buffer. This bit is set to 1 while the output buffer has output data and set to 0 after data has been read.
3	QUES	QUES is a summary of the questionable status register.
2		Always set to 0.
1	DEV	DEV is a summary of the device status register.
0		Always set to 0.

4.3 Standard Event Status Register

The bit assignments of the standard event status register are shown below.

This register provides standard information on a measuring instrument.

Bit	Mnemonic	Description
7	PON	Set to 1 at power-up.
6		Always set to 0.
5	CME	Set to 1 when the parser finds a syntax error.
4	EXE	Set to 1 when the execution of an instruction received as a GPIB command fails for some reason (parameter not within the allowable range, and so on).
3	DDE	Set to 1 when an error except a command error, execution error, or query error occurs.
2	QYE	Set to 1 when the controller attempts to read data from the D3371, but the data is not present or has been lost.
1		Always set to 0.
0	OPC	Set to 1 when there is no command being executed by the D3371 after reception of the *OPC command.

4.4 Operation Status Register

The bit assignments of the operation status register are shown below.

This 16-bit register provides information on the processing status of the measurement instrument.

DERR and END are event bits, and the condition register and transition filter are ignored.

Bit	Mnemonic	Description
15 to 13		Always set to 0.
12	EREC	Indicates that an error is being recorded. The period where synchronization and/or clock are lost is included in the period that an error is being recorded. (Error phase analysis option)
11	ASE	Indicates that an automatic search is being executed.
10	JTOL	Indicates that jitter tolerance is being measured. (Jitter tolerance option)
9	END	Indicates that the measurement period has ended.
8	DERR	Indicates that an error was detected during basic measurement.
7	EPAS	Indicates that error records are being saved on a file (Error phase analysis option).
6 to 5		Always set to 0.
4	MEAS	Indicates that measurement is in progress.
3 to 2		Always set to 0.
1	SET	SET is a summary of the settling status register.
0		Always set to 0.

4.5 Questionable Status Register

The bit assignments of the questionable status register are shown below.

This 16-bit register provides information on the error status of the measurement instrument.

Bit	Mnemonic	Description
15 to 13		Always set to 0.
12	SSG	Indicates that an error is occurring in the SSG.
11		Always set to 0.
10	SLOSS	Indicates that a sync loss occurred.
9	CLOSS	CLOSS is a summary of the clock loss status register.
8 to 0		Always set to 0.

4.6 Settling Status Register

The bit assignments of the settling status register are shown below.

This 16-bit register provides information about the status of setting the measurement instrument.

Bit	Mnemonic	Description
15 to 13		Always set to 0.
2	EDLY	Indicates that the ED module delay is being set.
1	PDLY	Indicates that the PPG module delay is being set.
0		Always set to 0.

4.7 Clock Loss Status Register

The bit assignments of the clock loss status register are shown below.

This 16-bit register provides information on the clock loss status of the measurement instrument.

Bit	Mnemonic	Description
15 to 13		Always set to 0.
2	ED	Indicates that a clock loss is occurring in the ED module.
1	PPG	Indicates that a clock loss is occurring in the PPG module.
0		Always set to 0.

4.8 Device Status Register

The bit assignments of the device status register are shown below.

This 16-bit register provides information about failures of the measurement instrument.

This register has no condition register or transition filter. The enable register is always set as on and this register cannot be changed.

Bit	Mnemonic	Description
15 to 13		Always set to 0.
2	EDLY	Indicates that the delay circuit of the ED module has malfunctioned.
1	PDLY	Indicates that the delay circuit of the PPG module has malfunctioned.
0		Always set to 0.

4.9 Clearing and Resetting the Status Registers

The following table shows how to clear and reset the status registers:

Register		*CLS	After startup	STATus: PREset
Status byte register		Cleared	Cleared	Unchanged
Service request enable register		Unchanged	Cleared	Unchanged
Standard event status register		Cleared	Cleared	Unchanged
Standard event status enable register		Unchanged	Cleared	Unchanged
Operation status register	Event register	Cleared	Cleared	Unchanged
	Enable register	Unchanged	All 0s	All 0s
	Positive transition filter	Unchanged	All 1s	All 1s
	Negative transition filter	Unchanged	All 0s	All 0s
Questionable status register	Event register	Cleared	Cleared	Unchanged
	Enable register	Unchanged	All 0s	All 0s
	Positive transition filter	Unchanged	All 1s	All 1s
	Negative transition filter	Unchanged	All 0s	All 0s
Clock loss status register	Event register	Cleared	Cleared	Unchanged
	Enable register	Unchanged	All 1s	All 1s
	Positive transition filter	Unchanged	All 1s	All 1s
	Negative transition filter	Unchanged	All 0s	All 0s
Settling status register	Event register	Cleared	Cleared	Unchanged
	Enable register	Unchanged	All 1s	All 1s
	Positive transition filter	Unchanged	All 1s	All 1s
	Negative transition filter	Unchanged	All 0s	All 0s
Device status register	Event register	Cleared	Cleared	Unchanged

5. COMMAND REFERENCE

The first part of this chapter describes the configuration of subsystems and lists command syntax precautions. The second part describes each command in more detail.

CAUTION:

- ***When using a command, note that part of the command mnemonic can be omitted.***

  (Example) ***Although expressed differently, the following two commands have the same meaning:***
  SOURCE1:PATTERN:SELECT PROGRAM
  SOUR:PATT PROG

- ***If you cannot find information on SOUR:PATT in this form, refer to Appendix A.1, "LIST OF COMMANDS," for the full form of the command to get information on it. The commands are listed in the full form in the alphabetical index and you can get information on them.***

5.1 GPIB Command Index

This GPIB command index is used as the index for Chapter 5.

5.2 Preface to Commands Descriptions

This section describes the configuration of subsystems and lists some precautions for the command syntax.

5.2.1 Subsystem Configuration

The analyzer's commands are configured from the subsystems listed in the table below.

The modules corresponding to subsystems that have the same name are specified with suffixes.

Subsystems	Description	Corresponding modules
Common commands	Command whose operation is the same on all measuring instruments	
SOURce[1]	Command related to PPG generation signal Includes setting of patterns and error addition	PPG
OUTPut[1]	Command related to DATA output Includes setting of the DATA output signal and termination voltage	PPG
OUTPut3	Command related to $\overline{\text{DATA}}$ output Includes setting of the $\overline{\text{DATA}}$ output signal and termination voltage	PPG
SOURce2	Command related to the clock signal Includes setting of a clock delay	PPG
OUTPut2	Command related to CLOCK output Includes setting of the CLOCK output signal and termination voltage	PPG
OUTPut4	Command related to $\overline{\text{CLOCK}}$ output Includes setting of the $\overline{\text{CLOCK}}$ output signal and termination voltage	PPG
SOURce5	Command related to BURST OUTPUT Includes setting of the signal source and burst signal	PPG
SOURce6	Command related to the trigger output Includes the type and location of the trigger output	PPG
SOURce7	Command related to the AUX output Includes setting the AUX output type	PPG
SENSe[1]	Command related to measurement Includes setting of patterns, synchronization, automatic search, and basic measurement	ED
INPut[1]	Command related to DATA input Includes setting of DATA input polarity, threshold voltage, and termination voltages	ED

Subsystems	Description	Corresponding modules
INPut2	Command related to CLOCK input Includes setting of CLOCK input polarity, amount of delay, and termination voltages	ED
SOURce8	Command related to the trigger output Includes setting for the trigger output type	ED
SOURce9	Command related to the AUX output Includes setting for the AUX output type	ED
FETCh	Inquires about the error measurement result, measurement time, and error monitor information	ED
CFETch	Inquires about the progress of the current error measurement	ED
DISPlay	Command related to different types of display Includes setting of the display format	
HCOPy	Command that prints the measurement result	
MMEMory	Command related to reading and saving of measurement data and setting information	
SYSTem	Command related to the system Includes setting of audible alarm, date, and time	
STATus	Command related to status registers. Includes setting and querying of the operation status register and questionable status register	
SOURce10	Command related to SSG Includes setting of frequency	SSG
JTOLerance	Command related to making specifications for jitter tolerance measurement. Used to specify the parameter table for jitter tolerance, clock frequency, measurement mode, and evaluation criteria.	SSG, ED

5.2.2 Notes on Command Descriptions

1. Commands and response data formats are described using the following symbols:

   < >: Indicates a syntactical element
   A description of the element follows the element.

   |: Indicates that one, and only one, of multiple items should be selected
   (Example) A | B | C Select A, B, or C.

   [ ]: Indicates that the item enclosed is optional and can be omitted

   { }: Encloses items that represent a group One of the items separated by | must be selected.

2. Availability of a command and a query is indicated as follows:

   Command/Query: Both a command and a query are available.

   Command: Only a command is available.

   Query: Only a query is available.

3. A mnemonic of 4 or more characters has a short form, which is indicated by uppercase in the text.

   (Example) SOURce1:PATTern:SELect PROGram

   Short form SOUR1:PATT:SEL PROG
   Long form SOURCE1:PATTERN:SELECT PROGRAM

4. In a query, append a question mark (?) to the command header. If the query requires parameters, also code the parameter in query format.

5. The parameter format used in command explanations is common to all commands and is shown below.

   <int>: Numeric data that can be entered in NR1, NR2, or NR3 format.
   When the analyzer receives this numeric data, the data is rounded off to an integer.

   <real>: Numeric data that can be entered in NR1, NR2, or NR3 format.
   When the analyzer receives this numeric data, the data is rounded off to a real number for the significant number of digits.

   <bool>: 0 | 1 switch
   0 corresponds to OFF and 1 to ON.

   <str>: Character string
   Indicates an alphanumeric symbol enclosed by double quotation marks (") or apostrophes (').

   <block>: Block data type
   Block data is an 8-bit binary data string.
   For more information on format, refer to Section 3.2, "Data Format."

5.2.3 Synchronization between the D3371 and Controller

The following commands overlap commands defined in IEEE488.2.

SOURce2:DELay <int> (PPG module clock delay setting)

INPut2:DELay <int> (ED module clock delay setting)

SENSe[1]:ASEarch[:STATe] ON (auto-search execution)

JTOLerance[:STATe] ON (Executes the jitter tolerance measurement)

MMEMory:STORe:ED:EPANalysis:TSERies:TEXT <cn>,<sp1>,<sp2>,<sp3>,<nr>,<str>
(Saves the error records in a file in the time-series format.)

MMEMory:STORe:ED:EPANalysis:STATistics:TEXT <sp1>,<sp2>,<sp3>,<range>,<str>
(Saves the error records in a file in the statistical format.)

Use *OPC, *OPC?, and *WAI when an operation is completed so that the D3371 is synchronized with the controller from the next operation onward. All commands other than those listed above are sequential commands.

5.2.4 GPIB Commands According to the System Versions

Table 5-1 GPIB Commands According to the System Version

GPIB Commands	A00/A02	B00
SENS[1]:SYNC:THR:MEM:GAIN <real>	1E-11 and 1E-12 can be used in <real>.	1E-11 and 1E-12 cannot be used in <real>.
SENS[1]:SYNC:THR:MEM:LOSS <real>	1E-11 and 1E-12 can be used in <real>.	1E-11 and 1E-12 cannot be used in <real>.
SOUR<n>:TRIG[:MODE] SYNC	SOUR8:TRIG[:MODE] SYNC SYNC trigger signal is generated from the front terminal.	SOUR9:AUX[:MODE] SYNC SYNC trigger signal is generated from the rear terminal.

5.3 Common Commands

1. `*CLS`

- Function — Clears status bytes and related data
- Availability of command and query — Command
- Command — *CLS
- Description — *CLS clears the status data structure and forcibly cancels *OPC and *OPC?. It also clears the error queue. However, since this command itself does not clear the output buffer, the MAV bit is not cleared if there is output data. If this command is executed at the beginning of a line, however, all the statuses, including the MAV bit, are cleared, because the data is cleared.

2. `*ESE`

- Function — Sets the standard event status enable register
- Availability of command and query — Command/Query
- Command — *ESE <int>
- Parameter — <int>
- Response format — NR1 (integer type)
- Description — Sets the standard event status enable register. The standard event status register corresponding to the bit set to 1 in this register is reflected in the status byte register as the effective bit.

  For details, refer to Chapter 4, "STATUS BYTES."
- Example — If the DDE bit (bit 3) and the OPC bit (bit 0) are set to enable, *ESE 9 is set because 9 is the result of the expression "$2^3 + 2^0 = 8 + 1 = 9$."

3. `*ESR?`

- Function — Reads the standard event status register
- Availability of command and query — Query
- Query — *ESR?
- Response format — NR1 (integer type)
- Description — Reads the contents of the standard event status register
  When read, the standard event status register is cleared and the bit (bit 5) in the corresponding status byte is also cleared.
  For details, refer to Chapter 4, "STATUS BYTES."

4. `*IDN?`

- Function — Queries the equipment
- Availability of command and query — Query
- Query — *IDN?
- Response format — <manufacturer>, <model>, <serial number>, <firmware level>
  <manufacturer> = ADVANTEST
  <model> = model-name
  <serial number> = serial-number
  <firmware level> = system-version
- Description — In quires about information identifying this device and outputs the four items in the above response format in character string format.

5. `*OPC`

- Function — Reports termination of all commands being executed
- Availability of command and query — Command/Query
- Command — *OPC
- Response format — 1
- Description — *OPC sets the OPC bit of the standard event status register to 1 when all the commands being executed have terminated. If it receives the next command before all the commands being executed have terminated, the analyzer also waits for termination of the next command. In other words, when the analyzer executes nothing after it receives *OPC, the standard event status register is set.
  *OPC? writes 1 instead of the OPC bit set by *OPC to the output buffer. It can therefore terminate commands according to the timing at which the analyzer receives a response from the controller.
  Both *OPC and *OPC? are released by the DCL interface message, *CLS, and *RST.
  Also refer to *WAI.

6. `*RST`

- Function — Resets the equipment
- Availability of command and query — Command
- Query — *RST
- Description — *RST resets the analyzer. Specifically, it performs the following processing:
  1. Initializes the analyzer (refer to Appendix A.3, "DEFAULTS.")
  2. Invalidates *OPC and *OPC?.

  Reset has no effect on the following items:
  1. GPIB bus status
  2. GPIB address
  3. Output buffer
  4. Status data structure
  5. Device calibration data

7. `*SRE`

- Function — Sets the service request enable register
- Availability of command and query — Command/Query
- Command — *SRE <int>
- Response format — NRI (integer type)
- Desciption — Sets the service request enable register.
  The status byte register corresponding to the bit set to 1 in this register is reflected in the MSS as the effective bit.
  For a query, response data bit 6 is always 0.

  For details, refer to Chapter 4, "STATUS BYTES."
- Example — If the OPER bit (bit 7), ESB bit (bit 5), and MAV bit (bit 4) are set to enable, *SRE 176 is set because 176 is the result of the expression "$2^7 + 2^5 + 2^4 = 128 + 32 + 16 = 176$."

8. `*STB?`

- Function — Reads the status byte register
- Availability of command and query — Query
- Query — *STB?
- Response format — NR1 (integer type)
- Description — Reads the contents of the status byte register
  The request summary bit read here is MSS.
  This register and MSS are not cleared when they are read.

  For details, refer to Chapter 4, "STATUS BYTES."

9. `*TST?`

- Function — Query about self-test result
- Availability of command and query — Query
- Query — *TST?
- Response format — 0 | error-code
- Description — *TST? causes the analyzer to execute the self-test and return the result.
  Response 0 indicates that the self-test is successful. Other responses indicate error codes. The analyzer returns only response 0 for *TST?.

10. *WAI

- Function — Waits for termination of all commands being executed
- Availability of command and query — Command
- Command — *WAI
- Description — *WAI waits for termination of all the commands that are being executed.
  Executing this command causes all subsequent commands to wait until the command being executed terminates.
  *WAI is cancelled by the DCL interface message.

11. *OPT?

- Function — Queries about options.
- Availability of command and query — Query
- Query — *OPT?
- Response format — <opt>, <opt>, ...
  <opt> = Option number

<opt>	Description
10	PPG (2 V) module
11	PPG (3 V) module
12	ED module
13	SSG module
70	Jitter tolerance option
71	Pattern option
72	Error phase analysis option

- Description — *OPT? inquires about the options installed in this device.

5.4 PPG Module

5.4.1 SOURce[1] Subsystem

1. `SOURce[1]:PATTern[:SELect]`

- Function — Sets a pattern type
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern[:SELect] <type>
- Parameter — <type> = {PRBS<n> | ZSUBstitut<n> | PROGram | STM<n> | FLEXible}
- Query response format — PRBS<n> | ZSUB<n> | PROG | STM<n> | FLEX
- Description — Sets a generation pattern type for the PPG module

Command parameter	Pattern type
PRBS<n>	Pseudo random pattern (PRBS) with pattern length (2^n-1) bits <n> = 7, 9, 10, 11, 15, 23, 31
ZSUB<n>	Zero substitution pattern (ZSUB) with pattern length 2^n bits <n> = 7, 9, 10, 11, 15
PROG	Programmable pattern (PROG)
STM<n>	STM pattern (STM) <n> = 4,16 This setting is available when the pattern option is installed.
FLEX	Flexible pattern (FLEX) This setting is available when the pattern option is installed.

2. `SOURce[1]:PATTern:PRBS:MRATio`

- Function — Sets a PRBS pattern mark ratio
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:PRBS:MRATio <ratio>
- Parameter — <ratio> = {MRATio<n> | MINVerted4}
- Query response format — MRAT<n> | MINV4
- Description — Sets a mark ratio if the PRBS pattern is used

Command parameter	Mark ratio
MRAT<n>	<n> = 0: Mark ratio 0/8 <n> = 1: Mark ratio 1/8 <n> = 2: Mark ratio 1/4 <n> = 4: Mark ratio 1/2 <n> = 6: Mark ratio 3/4 <n> = 7: Mark ratio 7/8 <n> = 8: Mark ratio 8/8
MINV4	Mark ratio 1/2B

3. `SOURce[1]:PATTern:ZSUBstitut[:ZLENgth]`

- Function — Sets a consecutive zero bit length
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:ZSUBstitut[:ZLENgth] <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets a consecutive zero bit length if the ZSUB pattern is used
  Setting range N to (2^N-1) bits (N = 7, 9, 10, 11, 15)
  Setting resolution 1 bit

4. `SOURce[1]:PATTern:PROGram[:LENGth]`

- Function — Sets a PROG pattern length
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:PROGram[:LENGth] <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets a pattern length if the PROG pattern is used

Setting range	Setting resolution
1 bit to 262,144 bits	1 bit
262,146 bits to 524,288 bits	2 bits
524,292 bits to 1,048,576 bits	4 bits
1,048,584 bits to 2,097,152 bits	8 bits
2,097,168 bits to 4,194,304 bits	16 bits
4,194,336 bits to 8,388,608 bits	32 bits

5. `SOURce[1]:PATTern:PROGram:DATA`

- Function: Sets the PROG pattern using character string data
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:PROGram:DATA <start-bit>, <length>, <str>
- Query: SOURce[1]:PATTern:PROGram:DATA? <start-bit>, <length>
- Parameter:
  - <start-bit>: Address at which pattern setting starts
    - Setting range: 0 bit to (pattern length - 1) bit
    - Setting resolution: 1 bit
  - <length>: Number of bits to be set
  - <str>: Data of the pattern to be set
    - "B--------": Pattern data specification in binary numbers
      - --------: 0, 1 (A maximum of 512 characters)
    - "H--------": Pattern data specification in hexadecimal numbers
      - --------: 0 to 9, a to f, A to F (A maximum of 512 characters)
- Query response format:
  - <str>: Response pattern
    - "H--------": Response to pattern data in hexadecimal numbers
      - --------: 0 to 9, A to F (A maximum of 512 characters)
- Description:
  A portion of the PROG pattern is overwritten by the contents of <str> from <start_bit> up to the bit length defined by <length>.
  Any data exceeding the range of the PROG pattern is ignored.
  If the number of data bits specified in <str> is smaller than <length>, the data up to the number of bits specified in <str> is set.
  If <length> is smaller than the number of data bits specified in <str>, the data up to the number of bits specified in <length> is set.
  In a query, data that exceeds the STM pattern range or 512 characters (512 × 4 = 2,048 bits) cannot be read.
- Example of use:
  When the 16-bit data 4142 (hexadecimal number) is set, starting at address 0 bit
  SOUR1:PATT:PROG:DATA 0, 16, "H4142"

6. `SOURce[1]:PATTern:PROGram:BDATa`

- Function — Sets the PROG pattern using block data
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:PROGram:BDATa <start-bit>, <length>, <block>
- Query — SOURce[1]:PATTern:PROGram:BDATa? <start-bit>, <length>
- Parameter
  - <start-bit>: Address at which pattern setting starts
    - Setting range: 0 bit to (pattern length - 1) bit
    - Setting resolution: 1 bit
  - <length>: Number of bits to be set
  - <block>: Data of the pattern to be set
- Query response format — <block>: Response pattern
- Description — A portion of the PROG pattern is overwritten by the contents of <block> from <start_bit> up to the bit length defined by <length>. Any data exceeding the range of the PROG pattern is ignored.
  If the number of data bits specified in <block> is smaller than <length>, the data up to the number of bits specified in <block> is set.
  If <length> is smaller than the number of data bits specified in <block>, the data up to the number of bits specified in <length> is set.
  In a query, data that exceeds the range of the PROG pattern cannot be read.
- Example of use — When the 16-bit data 4142 (hexadecimal number) is set, starting at address 0 bit
  SOUR1:PATT:PROG:BDAT 0, 16, #12AB

7. `SOURce[1]:PATTern:STM:NFRame` (Pattern option)

- Function: Specifies the number of frames in the STM pattern
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:STM:NFRame <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: Specifies the number of frames in the STM patterns

STM pattern	Setting range
STM4	1 to 107
STM16	1 to 26

Setting resolution: 1

8. `SOURce[1]:PATTern:STM:DATA` (Pattern option)

- Function: Sets the STM pattern using character string data
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:STM:DATA <start_frm>,<start_row>, <start_col>,<length>,<str>
- Query: SOURce[1]:PATTern:STM:DATA? <start_frm>,<start_row>, <start_col>,<length>
- Parameter:

<start_frm> = The frame number where the pattern setting starts
Setting range: 1 to the number of frames
Setting resolution: 1

<start_row> = The row where the pattern setting starts
Setting range: 1 to 9
Setting resolution: 1

<start_col> = The column where the pattern setting starts

STM pattern	Setting range
STM4	1 byte to 1080 byte
STM16	1 byte to 4320 byte

Setting resolution: 1 byte

<length> = The number of bits to be set
<str> = A pattern data to be set
"B--------" Specifies pattern data in binary numbers
-------: 0,1
(Maximum of 512 characters)
"H--------" Specifies pattern data in hexadecimal numbers
-------: 0 to 9, a to f, A to F
(Maximum of 512 characters)

- Query response format

<str> = A response pattern
"H---------" Pattern data in hexadecimal numbers is returned
----: 0 to 9, a to f, A to F
(Maximum of 512 characters)

- Description

A portion of the STM pattern is overwritten by the contents of <str> from <start_frm>, <start_row>, and <start_col> up to the bit length specified by <length>.

Data exceeding the STM pattern range is ignored.

If the number of bits in the data in <str> is less than <length>, the STM pattern up to the bit length specified by <str> is set.

If the bit length specified by <length> is less than the bit length specified by <str>, the STM pattern up to the bit length specified by <length> is set.

When the PRBS pattern is inserted in a payload, the number of bits in the payload portion is the same as the number stored in the pattern memory.

In a query, data that exceeds the STM pattern range or 512 characters (512 × 4 = 2,048 bits) cannot be read. Data in the pattern memory is always read.

- Example

To set 4142 (hexadecimal) in the area starting from the first row and the first column of the first frame, use the following:
SOUR1:PATT:STM:DATA 1,1,1,16,"H4142"

9. `SOURce[1]:PATTern:STM:BDATa` (Pattern option)

- Function: Sets the STM pattern using block data
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:STM:BDATa <start_frm>,<start_row>, <start_col>,<length>,<block>
- Query: SOURce[1]:PATTern:STM:BDATa? <start_frm>,<start_row>, <start_col>,<length>
- Parameter:

<start_frm> = The frame number where the pattern setting starts
Setting range: 1 to the number of frames
Setting resolution: 1

<start_row> = The row where the pattern setting starts
Setting range: 1 to 9
Setting resolution: 1

<start_col> = The column where the pattern setting starts

STM pattern	Setting range
STM4	1 byte to 1080 byte
STM16	1 byte to 4320 byte

Setting resolution:1 byte

<length> = The number of bits to be set
<block> = A pattern data to be set

- Query response format: <block> = A response pattern
- Description:

A portion of the STM pattern is overwritten by the content of <block> from <start_frm>, <start_row>, and <start_col> up to the bit length specified with <length> .

Data exceeding the STM pattern range is ignored.

If the number of bits of data in <block> is smaller than <length>, data up to the bit length specified in <block> is set.

If <length> is smaller than the number of bits of data in <block>, data up to the bit length specified in <length> is set.

When the PRBS pattern is inserted in a payload, the number of bits in the payload portion is the same as the number stored in the pattern memory.

In a query, data that exceeds the STM pattern range cannot be read. Data in the pattern memory is always read.

- Example: To set 4142 (16-bit hexadecimal data) in the area starting at the first row and the first column of the first frame, use the following:
SOUR1:PATT:STM:BDAT1,1,1,16,#12AB.

10. `SOURce[1]:PATTern:STM:PAYLoad:PRBS` (Pattern option)

- Function — Sets the function which inserts a PRBS pattern into an STM pattern payload to on or off
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:STM:PAYLoad:PRBS <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Whether a PRBS pattern is inserted (ON) or not inserted (OFF) into an STM pattern payload is specified.

11. `SOURce[1]:PATTern:STM:PAYLoad:PRBS:SELect` (Pattern option)

- Function — Sets the PRBS pattern length to be inserted into an STM pattern payload
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:STM:PAYLoad:PRBS:SELect <type>
- Parameter — <type> = PRBS<n>
  <n> = 7,9,10,11,15,23,31
- Query response format — PRBS<n>
  <n> = 7,9,10,11,15,23,31
- Description — The PRBS pattern length to be inserted into an STM pattern payload is specified.

12. `SOURce[1]:PATTern:STM:PAYLoad:PRBS:MRATio` (Pattern option)

- Function: Specifies the mark ratio of a PRBS pattern to be inserted into an STM pattern payload
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:STM:PAYLoad:PRBS:MRATio <ratio>
- Parameter: <ratio> = {MRATio<n> | MINVerted4}
- Query response format: MRAT<n> | MINV4
- Description: The mark ratio of a PRBS pattern to be inserted into an STM pattern payload is specified.

Command parameter	Mark ratio
MRAT<n>	<n> = 0 Mark ratio 0/8 <n> = 1 Mark ratio 1/8 <n> = 2 Mark ratio 1/4 <n> = 4 Mark ratio 1/2 <n> = 6 Mark ratio 3/4 <n> = 7 Mark ratio 7/8 <n> = 8 Mark ratio 8/8
MINV4	Mark ratio 1/2B

13. `SOURce[1]:PATTern:STM:SCRamble` (Pattern option)

- Function: Sets the STM pattern scrambling function to on or off
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:STM:SCRamble <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Whether an STM pattern is scrambled (ON) or not scrambled (OFF) is specified.

14. `SOURce[1]:PATTern:STM:B1` (Pattern option)

- Function: Sets the function which inserts B1 byte into an STM pattern to on or off
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:STM:B1 <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Whether B1 byte is inserted into an STM pattern (ON) or not (OFF) is specified.

15. `SOURce[1]:PATTern:FLEXible:STABle` (Pattern option)

- Function: Edits patterns in a FLEX pattern sequence table
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:FLEXible:STABle <index>,<patt>, <length>,<trig_signal>
- Query: SOURce[1]:PATTern:FLEXible:STABle? <index>
- Parameter:
  <index> = The index to specify a pattern in a FLEX pattern sequence table
  Setting range: 1 to the number of patterns
  Setting resolution: 1
  <patt> = Pattern to be selected{<prog_no>| PRBS}
  <length> = The pattern length when <patt> is PRBS
  Setting range: 128 bits to 2,097,152 bits
  Setting resolution: 64 bits
  <trig_signal> = Output level of FLEX Trigger: {HIGH| LOW}
- Query response format: <patt>,<length>,<trig_signal>

- Description

Patterns in a FLEX pattern sequence table are edited.
The setting for <length> is effective only if the selected pattern is a PRBS pattern.
Specify a value if a PROG pattern is selected.
The PROG pattern length used in FLEX patterns is specified by the following:

SOUR1:PATT:FLEX:PROG[:LENG]

In a query, the PROG pattern length can be read using <length> if the selected pattern is a PROG pattern.

<patt>	Pattern to be selected
<prog_no>	PROG pattern Specifies a PROG pattern number Setteing range: 1 to 127 Setteing resolution: 1
PRBS	PRBS pattern

<trig_signal>	Output levels
HIGH	HIGH level Generates a HIGH level signal in FLEX Trigger
LOW	LOW level Generates a LOW level signal in FLEX Trigger

16. `SOURce[1]:PATTern:FLEXible:STABle:COUNt?` (Pattern option)

- Function — Sends a query for the number of patterns in a FLEX pattern sequence table
- Availability of command and query — Query
- Query — SOURce[1]:PATTern:FLEXible:STABle:COUNt?
- Query response format — NR1 (integer type)
- Description — A query for the number of patterns in a FLEX pattern sequence table is sent.

17. `SOURce[1]:PATTern:FLEXible:STABle:NEW` (Pattern option)

- Function: Adds patterns to a FLEX pattern sequence table
- Availability of command and query: Command
- Command: SOURce[1]:PATTern:FLEXible:STABle:NEW <index>,<patt>, <length>,<trig_signal>
- Parameter:

<index> = The index to specify a pattern in a FLEX pattern sequence table
Setting range: 1 to (the number of the patterns + 1)
Setting resolution: 1

<patt> = Pattern to be selected{<prog_no>| PRBS}
<length> = The pattern length in the condition that <patt> is PRBS.
Setting range: 128 bits to 2,097,152 bits
Setting resolution: 64 bits

<trig_signal> = Output level of FLEX Trigger: {HIGH| LOW}

- Description:

Patterns are added to a FLEX pattern sequence table. A pattern is added to the position specified with <index>.
The maximum number of patterns in a pattern sequence table is 1024. No more patterns can be added if 1024 patterns already exist in the table.
The setting <length> is effective only if the selected pattern is a PRBS pattern.
Specify a value if a PROG pattern is selected.

The PROG pattern length used in FLEX patterns is specified by the following:
SOUR1:PATT:FLEX:PROG [:LENG]

<patt>	Pattern to be selected
<prog_no>	PROG pattern Specifies a PROG pattern number Setting range: 1 to 127 Setting resolution: 1
PRBS	PRBS pattern

<trig_signal>	Output level
HIGH	HIGH level Generates a HIGH level signal in FLEX Trigger
LOW	LOW level Generates a LOW level signal in FLEX Trigger

18. `SOURce[1]:PATTern:FLEXible:STABle:REMove` (Pattern option)

- Function: Deletes patterns in a FLEX pattern sequence table
- Availability of command and query: Command
- Command: SOURce[1]:PATTern:FLEXible:STABle:REMove <index>
- Parameter: <index> = The index to specify a pattern in a FLEX pattern sequence table
  Setting range: 1 to the number of patterns
  Setting resolution: 1
- Description: Patterns are deleted from a FLEX pattern sequence table.
  All of the patterns cannot be deleted. An index-1 pattern cannot be deleted if the index-2 pattern is a PRBS pattern.

19. `SOURce[1]:PATTern:FLEXible:STABle:DEFault` (Pattern option)

- Function: Sets the FLEX pattern sequence table default setting
- Availability of command and query: Command
- Command: SOURce[1]:PATTern:FLEXible:STABle:DEFault
- Description: Restores a FLEX pattern sequence table to the default setting

20. `SOURce[1]:PATTern:FLEXible:PROGram[:LENGth]` (Pattern option)

- Function: Sets a PROG pattern length used in a FLEX pattern.
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:FLEXible:PROGram[:LENGth] <prog_no>,<length>
- Query: SOURce[1]:PATTern:FLEXible:PROGram[:LENGth]? <prog_no>
- Parameter: <prog_no> = The PROG pattern number used in a FLEX pattern.
  Setting range: 1 to 127
  Setting resolution: 1
  <length> = A PROG pattern length
  Setting range: 128 bits to 65,536 bits
  Setting resolution: 64 bits
- Query response format: <length>= NR1 (integer type)
- Description: A PROG pattern length used in a FLEX pattern is specified.

21. `SOURce[1]:PATTern:FLEXible:PROGram:DATA` (Pattern option)

- Function: Sets the PROG pattern used in a FLEX pattern using character string
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:FLEXible:PROGram:DATA <prog_no>, <start_bit>,<length>,<str>
- Query: SOURce[1]:PATTern:FLEXible:PROGram:DATA? <prog_no>, <start_bit>,<length>
- Parameter:

  <prog_no> = The PROG pattern number used in a FLEX pattern.
  Setting range: 1 to 127
  Setting resolution: 1

  <start_bit> = The address where the pattern setting starts.
  Setting range: 0 bit to (the pattern length -1) bit
  Setting resolution: 1 bit

  <length> = The number of bits to be specified

  <str> = A pattern data to be set
  "B--------" Specifying pattern data using binary numbers
  -------: 0,1 (A maximum of 512 characters)
  "H--------" Specifying pattern data using hexadecimal numbers
  -------: 0 to 9, a to f, A to F (A maximum of 512 characters)
- Query response format: <str> = A response pattern
  "H--------" Specifying a pattern data using hexadecimal numbers
  -------: 0 to 9, a to f, A to F (A maximum of 512 characters)
- Description:

  A portion of the PROG pattern specified by <prog_no> is overwritten by the contents of <str> from <start_bit> up to the bit length defined by <length>.
  Data exceeding the PROG pattern range specified by <prog_no> is ignored.
  If the number of data bits in <str> is smaller than <length>, the data up to the number of bits specified in <str> is set.
  If <length> is smaller than the number of data bits in <str>, the data up to the number of bits specified in <length> is set.
  In a query, data that exceeds the PROG pattern range specified by <prog_no> or 512 characters (512 × 4 = 2,048 bits) cannot be read.

- Example — To set 4142 (16-bit hexadecimal data) in the area starting at address 0 of PROG pattern 1, use the following:

  SOUR1:PATT:FLEX:PROG:DATA 1,0,16, "H4142"

22. `SOURce[1]:PATTern:FLEXible:PROGram:BDATa` (Pattern option)

- Function — Specifies the PROG pattern used in a FLEX pattern using block data
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:FLEXible:PROGram:BDATa <prog_no>, <start_bit>,<length>,<block>
- Query — SOURce[1]:PATTern:FLEXible:PROGram:BDATa? <prog_no>, <start_bit>,<length>
- Parameter — <prog_no> = The PROG pattern number used in a FLEX pattern.
  Setting range: 1 to 127
  Setting resolution: 1

  <start_bit> = The address where the pattern setting starts.
  Setting range: 0 bit to (the pattern length -1) bit
  Setting resolution: 1 bit

  <length> = The number of bits to be specified
  <block> = A pattern data to be set
- Query response format — <block> = A response pattern
- Description — A portion of the PROG pattern specified by <prog_no> is overwritten by the contents of <block> from <start_bit> up to the bit length defined by <length>.
  Data exceeding the PROG pattern range specified by <prog_no> is ignored.
  If the number of data bits in <block> is smaller than <length>, the data up to the number of bits specified in <block> is set.
  If <length> is smaller than the number of data bits in <block>, the data up to the number of bits specified in <length> is set.
  In a query, data that exceeds the PROG pattern range specified by <prog_no> cannot be read.
- Example — To set 4142 (16-bit hexadecimal data) in the area starting at address 0 of PROG pattern 1, use the following:

  SOUR1:PATT:FLEX:PROG:BDATA 1,0,16, #12AB

23. `SOURce[1]:PATTern:FLEXible:PRBS:SELect` (Pattern option)

- Function: Sets the PRBS pattern length used in a FLEX pattern
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:FLEXible:PRBS:SELect <type>
- Parameter: <type> = PRBS <n>
  <n> = 7,9,10,11,15,23,31
- Query response format: PRBS <n>
  <n> = 7,9,10,11,15,23,31
- Description: The PRBS pattern length used in a FLEX pattern is specified.

24. `SOURce[1]:PATTern:FLEXible:PRBS:MRATio` (Pattern option)

- Function: Sets the mark ratio of a PRBS pattern used in a FLEX pattern
- Availability of command and query: Command/Query
- Command: SOURce[1]:PATTern:FLEXible:PRBS:MRATio <ratio>
- Parameter: <ratio> = {MRATio<n> | MINVerted4}
- Query response format: MRAT<n> | MINV4
- Description: The mark ratio of a PRBS pattern used in a FLEX pattern is specified.

Command parameter	Mark ratio
MRAT<n>	<n> = 0 Mark ratio 0/8 <n> = 1 Mark ratio 1/8 <n> = 2 Mark ratio 1/4 <n> = 4 Mark ratio 1/2 <n> = 6 Mark ratio 3/4 <n> = 7 Mark ratio 7/8 <n> = 8 Mark ratio 8/8
M1NV4	Mark ratio 1/2B

25. `SOURce[1]:PATTern:POLarity`

- Function — Sets the pattern polarity
- Availability of command and query — Command/Query
- Command — SOURce[1]:PATTern:POLarity <type>
- Parameter — <type> = {NORMal | INVerted}
- Query response format — NORM | INV
- Description — Sets the pattern polarity

Command parameter	Pattern polarity
NORM	Positive logic
INV	Negative logic

26. `SOURce[1]:EADDition`

- Function — Sets error addition on or off
- Availability of command and query — Command/Query
- Command — SOURce[1]:EADDition <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Specifies whether an error is added (ON) or not (OFF)

27. `SOURce[1]:EADDition:SINGle`

- Function — Adds a 1-bit error
- Availability of command and query — Command
- Command — SOURce[1]:EADDition:SINGle
- Description — Adds a 1-bit error
  This command is valid only when error addition is ON and the error addition mode is in the single mode.

28. `SOURce[1]:EADDition:MODE`

- Function — Sets an error addition mode
- Availability of command and query — Command/Query
- Command — SOURce[1]:EADDition:MODE <type>
- Parameter — <type> = {SINGle | REPeat | EXTernal}
- Query response format — SING | REP | EXT
- Description — Sets an error addition mode

Command parameter	Error addition mode
SING	Single mode Adds a 1-bit error manually
REP	Repeat mode. Adds an addition rate error
EXT	External control mode. Adds an error using external control

29. `SOURce[1]:EADDition:RATE`

- Function — Sets an error addition rate
- Availability of command and query — Command/Query
- Command — SOURce[1]:EADDition:RATE <real>
- Parameter — <real> = 1E-2, 1E-3, 1E-4, 1E-5, 1E-6, 1E-7, 1E-8, 1E-9
- Query response format — NR3 (floating-point type)
- Description — Sets an error addition rate
  This command is valid only when the error addition mode is in the repeat mode.

30. `SOURce[1]:EADDition:ROUTe`

- Function — Sets an error addition route
- Availability of command and query — Command/Query
- Command — SOURce[1]:EADDition:ROUTe <int>
- Parameter — <int> = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
- Query response format — NR1 (integer type)
- Description — Sets an error addition route

5.4.2 OUTPut[1] Subsystem

1. `OUTPut[1][:STATe]`

- Function: Sets DATA output on or off
- Availability of command and query: Command/Query
- Command: OUTPut[1][:STATe] <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Sets whether the DATA output is enabled (ON) or disabled (OFF)

2. `OUTPut[1]:VOLTage[:LEVel][:IMMediate][:AMPLitude]`

- Function: Sets DATA output amplitude
- Availability of command and query: Command/Query
- Command: OUTPut[1]:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point type)
- Description: Sets DATA output amplitude

Termination type	Setting range
LVPECL (+1.3 V)	0.60 to 1.00 V_{P-P}
to GND (0 V)	0.30 to 2.00 V_{P-P} (2 V output module) 0.30 to 3.00 V_{P-P} (3 V output module)
ECL (-2 V)	0.60 to 1.00 V_{P-P}
CML	0.30 V_{P-P} to 1.00 V_{P-P}

Setting resolution: 0.01 V_{P-P}
Setting unit: V_{P-P}

3. `OUTPut[1]:VOLTage[:LEVel][:IMMediate]:HIGH`

- Function: Sets the HIGH level offset value of DATA output
- Availability of command and query: Command/Query
- Command: OUTPut[1]:VOLTage[:LEVel][:IMMediate]:HIGH <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point type)
- Description: Sets the offset value of DATA output as the HIGH level

Termination type	Setting range
LVPECL (+1.3 V)	+2.30 V to +2.70 V
to GND (0 V)	-2.00 V to +2.00 V *1
ECL (-2 V)	-1.00 V to -0.60 V
CML	(Vcc-0.20) V to (Vcc+0.20) V *2

*1 If an amplitude exceeding 2 V_{P-P} is defined on a 3 V output module, the offset setting range is limited to -1.00 V to +1.00 V.

*2 Vcc indicates DATA output termination voltage.

Setting resolution: 0.01 V
Setting unit: V

4. `OUTPut[1]:VOLTage[:LEVel][:IMMediate]:MIDDle`

- Function: Sets the intermediate level offset value of DATA output
- Availability of command and query: Command/Query
- Command: OUTPut[1]:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point type)
- Description: Sets the offset value of DATA output as the intermediate value between the HIGH and LOW levels
  Set the offset value of DATA output so that the offset value for HIGH level will stay within the range
  Setting resolution: 0.01 V
  Setting unit: V

5. `OUTPut[1]:VOLTage[:LEVel][:IMMediate]:LOW`

- Function — Sets the LOW level offset value of DATA output
- Availability of command and query — Command/Query
- Command — OUTPut[1]:VOLTage[:LEVel][:IMMediate]:LOW <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of DATA output as the LOW level
  Set the offset value of DATA output so that the offset value for HIGH level will stay within the range.
  Setting resolution: 0.01 V
  Setting unit: V

6. `OUTPut[1]:CROSs`

- Function — Sets the cross point of DATA output
- Availability of command and query — Command/Query
- Command — OUTPut[1]:CROSs <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the cross point of DATA output
  Setting range: 20% to 80%
  Setting resolution: 1%

7. `OUTPut[1]:TERMination`

- Function — Sets a DATA output termination type
- Availability of command and query — Command/Query
- Command — OUTPut[1]:TERMination <type>
- Parameter — <type> = {P1R3V | GND | M2V | CML}
- Query response format — P1R3V | GND | M2V | CML
- Description — Sets a DATA output termination type

Command parameter	Termination type
P1R3V	LVPECL (+1.3 V)
GND	to GND (0 V)
M2V	ECL (-2 V)
CML	CML

8. `OUTPut[1]:TERMination:VOLTage`

- Function — Sets DATA output termination voltage
- Availability of command and query — Command/Query
- Command — OUTPut[1]:TERMination:VOLTage <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point-type)
- Description — Sets DATA output termination voltage
  This command is effective only if the termination type is CML.
  Setting range: 0.00 V to 3.50 V
  Setting resolution: 0.05 V
  Setting unit: V

5.4.3 OUTPut3 Subsystem

1. `OUTPut3[:STATe]`

- Function — Sets $\overline{\text{DATA}}$ output on or off
- Availability of command and query — Command/Query
- Command — OUTPut3[:STATe] <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether the $\overline{\text{DATA}}$ output is enabled (ON) or disabled (OFF)

2. `OUTPut3:VOLTage[:LEVel][:IMMediate][:AMPLitude]`

- Function: Sets $\overline{\text{DATA}}$ output amplitude
- Availability of command and query: Command/Query
- Command: OUTPut3:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point type)
- Description: Sets $\overline{\text{DATA}}$ output amplitude

Termination type	Setting range
LVPECL (+1.3 V)	0.60 to 1.00 V_{P-P}
to GND (0 V)	0.30 to 2.00 V_{P-P} (2 V output module) 0.30 to 3.00 V_{P-P} (3 V output module)
ECL (-2 V)	0.60 to 1.00 V_{P-P}
CML	0.30 V_{P-P} to 1.00 V_{P-P}

Setting resolution: 0.01 V_{P-P}
Setting unit: V_{P-P}

3. `OUTPut3:VOLTage[:LEVel][:IMMediate]:HIGH`

- Function: Sets the HIGH level offset value of $\overline{\text{DATA}}$ output
- Availability of command and query: Command/Query
- Command: OUTPut3:VOLTage[:LEVel][:IMMediate]:HIGH <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point type)
- Description: Sets the offset value of $\overline{\text{DATA}}$ output as the HIGH level

Termination type	Setting range
LVPECL (+1.3 V)	+2.30 V to +2.70 V
to GND (0 V)	-2.00 V to +2.00 V *1
ECL (-2 V)	-1.00 V to -0.60 V
CML	(Vcc-0.20) V to (Vcc+0.20) V *2

*1 If an amplitude exceeding 2 V_{P-P} is defined on a 3 V output module, the offset setting range is limited to -1.00 V to +1.00 V.
*2 Vcc indicates $\overline{\text{DATA}}$ output termination voltage.

Setting resolution: 0.01 V
Setting unit: V

4. `OUTPut3:VOLTage[:LEVel][:IMMediate]:MIDDle`

- Function — Sets the intermediate level offset value of $\overline{\text{DATA}}$ output
- Availability of command and query — Command/Query
- Command — OUTPut3:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of $\overline{\text{DATA}}$ output as the intermediate value between the HIGH and LOW levels
  Set the offset value of $\overline{\text{DATA}}$ output so that the offset value for HIGH level will stay within the range.
  Setting resolution: 0.01 V
  Setting unit: V

5. `OUTPut3:VOLTage[:LEVel][:IMMediate]:LOW`

- Function — Sets the LOW level offset value of $\overline{\text{DATA}}$ output
- Availability of command and query — Command/Query
- Command — OUTPut3:VOLTage[:LEVel][:IMMediate]:LOW <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of $\overline{\text{DATA}}$ output as the LOW level
  Set the offset value of $\overline{\text{DATA}}$ output so that the offset value for HIGH level will stay within the range.
  Setting resolution: 0.01 V
  Setting unit: V

6. `OUTPut3:CROSs`

- Function — Sets the cross point of $\overline{\text{DATA}}$ output
- Availability of command and query — Command/Query
- Command — OUTPut3:CROSs <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the cross point of $\overline{\text{DATA}}$ output
  Setting range: 20% to 80%
  Setting resolution: 1%

7. `OUTPut3:TERMination`

- Function — Sets $\overline{\text{DATA}}$ output termination type
- Availability of command and query — Command/Query
- Command — OUTPut3:TERMination <type>
- Parameter — <type> = {P1R3V | GND | M2V | CML}
- Query response format — P1R3V | GND | M2V | CML
- Description — Sets $\overline{\text{DATA}}$ output termination type

Command parameter	Termination type
P1R3V	LVPECL (+1.3 V)
GND	to GND (0 V)
M2V	ECL (-2 V)
CML	CML

8. `OUTPut3:TERMination:VOLTage`

- Function — Sets $\overline{\text{DATA}}$ output termination voltage
- Availability of command and query — Command/Query
- Command — OUTPut3:TERMination:VOLTage <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point-type)
- Description — Sets $\overline{\text{DATA}}$ output termination voltage
  This command is effective only if the termination type is CML.
  Setting range: 0.00 V to 3.50 V
  Setting resolution: 0.05 V
  Setting unit: V

9. `OUTPut3:TRACk`

- Function — Sets the tracking function of data output on or off
- Availability of command and query — Command/Query
- Command — OUTPut3:TRACk <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether data output is tracked (ON) or not (OFF)
  If ON is selected, the $\overline{\text{DATA}}$ output uses the same setting as the DATA output.

5.4.4 SOURce2 Subsystem

1. `SOURce2:DELay`

- Function — Sets the clock delay amount
- Availability of command and query — Command/Query
- Command — SOURce2:DELay <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets an amount for the clock delay
  Setting range: -1000 to 1000 ps
  Setting resolution: 1 ps
  Setting unit: ps

5.4.5 OUTPut2 Subsystem

1. `OUTPut2[:STATe]`

- Function — Sets CLOCK output on or off
- Availability of command and query — Command/Query
- Command — OUTPut2[:STATe] <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether the CLOCK output is enabled (ON) or disabled (OFF)

2. `OUTPut2:VOLTage[:LEVel][:IMMediate][:AMPLitude]`

- Function — Sets the CLOCK output amplitude
- Availability of command and query — Command/Query
- Command — OUTPut2:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — A CLOCK output amplitude is set.

Termination type	Setting range
LVPECL (+1.3 V)	0.60 to 1.00 V_{P-P}
to GND (0 V)	0.30 to 2.00 V_{P-P}
ECL (-2 V)	0.60 to 1.00 V_{P-P}
CML	0.30 V_{P-P} to 1.00 V_{P-P}

Setting resolution: 0.01 V_{P-P}
Setting unit: V_{P-P}

3. `OUTPut2:VOLTage[:LEVel][:IMMediate]:HIGH`

- Function — Sets the HIGH level offset value of CLOCK output
- Availability of command and query — Command/Query
- Command — OUTPut2:VOLTage[:LEVel][:IMMediate]:HIGH <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of CLOCK output as the HIGH level

Termination type	Setting range
LVPECL (+1.3 V)	+2.30 to +2.70 V
to GND (0 V)	-2.00 to +2.00 V
ECL (-2 V)	-1.00 to -0.60 V
CML	(Vcc-0.20) V to (Vcc+0.20) V*1

*1 Vcc indicates CLOCK output termination voltage.

Setting resolution: 0.01 V
Setting unit: V

4. `OUTPut2:VOLTage[:LEVel][:IMMediate]:MIDDle`

- Function — Sets the intermediate level offset value of CLOCK output
- FunctionAvailability of command and query — Command/Query
- Command — OUTPut2:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of CLOCK output as the intermediate value between the HIGH and LOW levels
  Set the offset value of CLOCK output so that the offset value for HIGH level will stay within the range.
  Setting resolution: 0.01 V
  Setting unit: V

5. `OUTPut2:VOLTage[:LEVel][:IMMediate]:LOW`

- Function — Sets the LOW level offset value of CLOCK output
- Availability of command and query — Command/Query
- Command — OUTPut2:VOLTage[:LEVel][:IMMediate]:LOW <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — The offset value of CLOCK output as the LOW level is set
  Set the offset value of CLOCK output so that the offset value for HIGH level will stay within the range.
  Setting resolution: 0.01 V
  Setting unit: V

6. `OUTPut2:TERMination`

- Function — Sets a CLOCK output termination type
- Availability of command and query — Command/Query
- Command — OUTPut2:TERMination <type>
- Parameter — <type> = {P1R3V | GND | M2V | CML}
- Query response format — P1R3V | GND | M2V | CML
- Description — A CLOCK output termination type is set.

Command parameter	Termination type
P1R3V	LVPECL (+1.3 V)
GND	to GND (0 V)
M2V	ECL (-2 V)
CML	CML

7. `OUTPut2:TERMination:VOLTage`

- Function — Sets a CLOCK output termination voltage
- Availability of command and query — Command/Query
- Command — OUTPut2:TERMination:VOLTage <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point-type)
- Description — A CLOCK output termination voltage is set.
  This command is effective only if the termination type is CML.
  Setting range: 0.00 V to 3.50 V
  Setting resolution: 0.05 V
  Setting unit: V

5.4.6 OUTPut4 Subsystem

1. `OUTPut4[:STATe]`

- Function: Sets $\overline{\text{CLOCK}}$ output on or off
- Availability of command and query: Command/Query
- Command: OUTPut4[:STATe] <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Sets whether the $\overline{\text{CLOCK}}$ output is enabled (ON) or disabled (OFF)

2. `OUTPut4:VOLTage[:LEVel][:IMMediate][:AMPLitude]`

- Function: Sets a $\overline{\text{CLOCK}}$ output amplitude
- Availability of command and query: Command/Query
- Command: OUTPut4:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point type)
- Description: Sets a $\overline{\text{CLOCK}}$ output amplitude

Termination type	Setting range
LVPECL (+1.3 V)	0.60 to 1.00 V_{P-P}
to GND (0 V)	0.30 to 2.00 V_{P-P}
ECL (-2 V)	0.60 to 1.00 V_{P-P}
CML	0.30 V_{P-P} to 1.00 V_{P-P}

Setting resolution: 0.01 V_{P-P}
Setting unit: V_{P-P}

3. `OUTPut4:VOLTage[:LEVel][:IMMediate]:HIGH`

- Function — Sets the HIGH level offset value of $\overline{\text{CLOCK}}$ output
- Availability of command and query — Command/Query
- Command — OUTPut4:VOLTage[:LEVel][:IMMediate]:HIGH <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of $\overline{\text{CLOCK}}$ output as the HIGH level

Termination type	Setting range
LVPECL (+1.3 V)	+2.30 to +2.70 V
to GND (0 V)	-2.00 to +2.00 V
ECL (-2 V)	-1.00 to -0.60 V
CML	(Vcc-0.20) V to (Vcc+0.20) V*1

*1 Vcc indicates $\overline{\text{CLOCK}}$ output termination voltage.

Setting resolution: 0.01 V
Setting unit: V

4. `OUTPut4:VOLTage[:LEVel][:IMMediate]:MIDDle`

- Function — Sets the intermediate level offset value of $\overline{\text{CLOCK}}$ output
- Availability of command and query — Command/Query
- Command — OUTPut4:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of $\overline{\text{CLOCK}}$ output as the intermediate value between the HIGH and LOW levels
  Set the offset value of $\overline{\text{CLOCK}}$ output so that the offset value for HIGH level will stay within the range.
  Setting resolution: 0.01 V
  Setting unit: V

5. `OUTPut4:VOLTage[:LEVel][:IMMediate]:LOW`

- Function — Sets the LOW level offset value of $\overline{\text{CLOCK}}$ output
- Availability of command and query — Command/Query
- Command — OUTPut4:VOLTage[:LEVel][:IMMediate]:LOW <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the offset value of $\overline{\text{CLOCK}}$ output as the LOW level
  Set the offset value of $\overline{\text{CLOCK}}$ output so that the offset value for HIGH level will stay within the range.
  Setting resolution: 0.01 V
  Setting unit: V

6. `OUTPut4:TERMination`

- Function — Sets a $\overline{\text{CLOCK}}$ output termination type
- Availability of command and query — Command/Query
- Command — OUTPut4:TERMination <type>
- Parameter — <type> = {P1R3V | GND | M2V | CML}
- Query response format — P1R3V | GND | M2V | CML
- Description — Sets a $\overline{\text{CLOCK}}$ output termination type

Command parameters	Termination types
P1R3V	LVPECL (+1.3 V)
GND	to GND (0 V)
M2V	ECL (-2 V)
CML	CML

7. `OUTPut4:TERMination:VOLTage`

- Function — Sets a $\overline{\text{CLOCK}}$ output termination voltage
- Availability of command and query — Command/Query
- Command — OUTPut4:TERMination:VOLTage <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point-type)
- Description — A $\overline{\text{CLOCK}}$ output termination voltage is specified.
  This command is effective only if the termination type is CML.
  Setting range: 0.00 V to 3.50 V
  Setting resolution: 0.05 V
  Setting unit: V

8. `OUTPut4:TRACk`

- Function — Sets the tracking function of clock output to on or off
- Availability of command and query — Command/Query
- Command — OUTPut4:TRACk <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether clock output is tracked (ON) or not (OFF)
  If ON is selected, the $\overline{\text{CLOCK}}$ output uses the same setting as the CLOCK output.

5.4.7 SOURce5 Subsystem

1. `SOURce5:BOUTput[:STATe]`

- Function: Sets the Burst Mode to on or off
- Availability of command and query: Command/Query
- Command: SOURce5:BOUTput[:STATe] <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Sets whether the burst mode is enabled (ON) or disabled (OFF)

2. `SOURce5:BOUTput:SOURce`

- Function: Sets a BURST OUTPUT signal source
- Availability of command and query: Command/Query
- Command: SOURce5:BOUTput:SOURce <type>
- Parameter: <type> = {INTernal | EXTernal}
- Query response format: INT | EXT
- Description: Sets a BURST OUTPUT signal source

Command parameter	Signal source
INT	Internal signal
EXT	External input

3. `SOURce5:BOUTput:CYCLe`

- Function — Sets a burst cycle
- Availability of command and query — Command/Query
- Command — SOURce5:BOUTput:CYCLe <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets a burst cycle
  Sets a value larger than the OFF time of the burst cycle
  Setting range: 2 to 50000 µs
  Setting resolution: 1 µs
  Setting unit: µs

4. `SOURce5:BOUTput:OTIMe`

- Function — Sets the OFF Time in the burst cycle
- Availability of command and query — Command/Query
- Command — SOURce5:BOUTput:OTIMe <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the OFF Time in the burst cycle
  Set a value less than the value used in the burst cycle as the OFF Time.
  Setting range: 1 to 49999 µs
  Setting resolution: 1 µs
  Setting unit: µs

5.4.8 SOURce6 Subsystem

1. `SOURce6:TRIGger[:MODE]`

- Function — Specifies the output source of a trigger
- Availability of command and query — Command/Query
- Command — SOURce6:TRIGger[:MODE] <type>
- Parameter — <type> = {CLOCk8 | CLOCk32 | PATTern | FLEXible}
- Query response format — CLOC8 | CLOC32 | PATT | FLEX
- Description — Specifies the output source of a trigger

Command parameter	Trigger signal output source
CLOC8	1/8 frequency clock of the output source clock
CLOC32	1/32 frequency clock of the output source clock
PATT	Trigger signal synchronized with the pattern
FLEX	The level specified for FLEX Trigger in the pattern sequence table Setting is possible if the pattern option is installed and the pattern type is FLEX.

2. `SOURce6:TRIGger:PRBS:POSition`

- Function — Sets the trigger signal output position when a PRBS pattern is generated
- Availability of command and query — Command/Query
- Command — SOURce6:TRIGger:PRBS:POSition <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the trigger signal output position when a PRBS pattern is generated
  Setting range: 0 to (pattern length - 1) bit
  Setting resolution: 1 bit

3. `SOURce6:TRIGger:ZSUBstitut:POSition`

- Function — Sets the trigger signal output position when a ZSUB pattern is generated
- Availability of command and query — Command/Query
- Command — SOURce6:TRIGger:ZSUBstitut:POSition <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the trigger signal output address when a ZSUB pattern is generated
  Setting range: 0 bit to (pattern length - 1) bit
  Setting resolution: 16 bits

4. `SOURce6:TRIGger:PROGram:POSition`

- Function — Sets the trigger signal output position when a PROG pattern is generated
- Availability of command and query — Command/Query
- Command — SOURce6:TRIGger:PROGram:POSition <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the trigger signal output position when a PROG pattern is generated
  Setting range: 0 bit to (pattern length - 1) bit
  Setting resolution: 16 bits

5. `SOURce6:TRIGger:STM:FNUMber` (Pattern option)

- Function — Sets the trigger output frame number for the STM patterns
- Availability of command and query — Command/Query
- Command — SOURce6:TRIGger:STM:FNUMber <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — The trigger output frame number for the STM patterns is specified.
  Setting range: 1 to the number of frames
  Setting resolution: 1

6. `SOURce6:TRIGger:STM:ROW` (Pattern option)

- Function: Sets the trigger output row for the STM patterns
- Availability of command and query: Command/Query
- Command: SOURce6:TRIGger:STM:ROW <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The trigger output row for the STM patterns is specified.
  Setting range: 1 to 9
  Setting resolution: 1

7. `SOURce6:TRIGger:STM:COLumn` (Pattern option)

- Function: Sets the trigger output column for the STM patterns
- Availability of command and query: Command/Query
- Command: SOURce6:TRIGger:STM:COLumn <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The trigger output column for the STM patterns is specified.

STM pattern	Setting range
STM4	1 byte to 1080 bytes
STM16	1 byte to 4320 bytes

Setting resolution: 2 bytes

8. `SOURce6:TRIGger:STM:MODE` (Pattern option)

- Function: Sets the trigger output mode for the STM patterns
- Availability of command and query: Command/Query
- Command: SOURce6:TRIGger:STM:MODE <type>
- Parameter: <type> = {PATTern | FRAMe}
- Query response format: PATT | FRAM
- Description: The trigger output mode for the STM patterns is specified

Command parameter	STM pattern trigger output mode
PATT	Pattern mode A trigger is output for every pattern.
FRAM	Frame mode A trigger is output for every frame.

9. `SOURce6:TRIGger:FLEXible:INDex` (Pattern option)

- Function: Sets the trigger output index for the FLEX patterns
- Availability of command and query: Command/Query
- Command: SOURce6:TRIGger:FLEXible:INDex <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The trigger output index for the FLEX patterns is specified.
  Setting range: 1 to the number of the patterns
  Setting resolution: 1

10. `SOURce6:TRIGger:FLEXible:ADDRess` (Pattern option)

- Function: Sets the trigger output address for the FLEX patterns
- Availability of command and query: Command/Query
- Command: SOURce6:TRIGger:FLEXible:ADDRess <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The trigger output address for the FLEX patterns is specified.
  Setting range: 0 bits to (the pattern length of a trigger output index - 1) bits
  Setting resolution: 16 bits

5.4.9 SOURce7 Subsystem

The current version does not have any commands and queries for the SOURce7 subsystem.

5.5 ED Module

5.5.1 SENSe[1] Subsystem

1. `SENSe[1]:PATTern[:SELect]`

- Function — Sets a pattern type
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern[:SELect] <type>
- Parameter — <type> = {PRBS<n> | ZSUBstitut<n> | PROGram | STM<n> | FLEXible}
- Query response format — PRBS<n> | ZSUB<n> | PROG | STM<n> | FLEX
- Description — Specifies a measurement pattern type for the ED module

Command parameter	Pattern type
PRBS<n>	Pseudo random pattern (PRBS) with pattern length (2^n-1) bit <n> = 7, 9, 10, 11, 15, 23, 31
ZSUB<n>	Zero substitution pattern (ZSUB) with pattern length 2^n bit <n> = 7, 9, 10, 11, 15
PROG	Programmable pattern (PROG)
STM<n>	STM pattern (STM) <n> = 4,16 Can be set if the pattern option is installed
FLEX	Flexible pattern (FLEX) Can be set if the pattern option is installed

2. `SENSe[1]:PATTern:PRBS:MRATio`

- Function — Sets a PRBS pattern mark ratio
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:PRBS:MRATio <ratio>
- Parameter — <ratio> = {MRATio<n> | MINVerted4}
- Query response format — MRAT<n> | MINV4
- Description — Sets the mark ratio of the PRBS pattern

Command parameter	Mark ratio
MRAT<n>	<n> = 0 Mark ratio 0/8 <n> = 1 Mark ratio 1/8 <n> = 2 Mark ratio 1/4 <n> = 4 Mark ratio 1/2 <n> = 6 Mark ratio 3/4 <n> = 7 Mark ratio 7/8 <n> = 8 Mark ratio 8/8
MINV4	Mark ratio 1/2B

3. `SENSe[1]:PATTern:ZSUBstitut[:ZLENgth]`

- Function — Sets the consecutive zero bit length
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:ZSUBstitut[:ZLENgth] <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the consecutive zero bit length of the ZSUB pattern
  Setting range — N to (2^N-1) bits (N=7, 9, 10, 11, 15)
  Setting resolution — 1 bit

4. `SENSe[1]:PATTern:PROGram[:LENGth]`

- Function — Sets the PROG pattern length
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:PROGram[:LENGth] <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the pattern length of the PROG pattern

Setting range	Setting resolution
1 bit to 262,144 bits	1 bit
262,146 bits to 524,288 bits	2 bits
524,292 bits to 1,048,576 bits	4 bits
1,048,584 bits to 2,097,152 bits	8 bits
2,097,168 bits to 4,194,304 bits	16 bits
4,194,336 bits to 8,388,608 bits	32 bits

5. `SENSe[1]:PATTern:PROGram:DATA`

- Function: Sets the PROG pattern using character string data
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:PROGram:DATA <start-bit>, <length>, <str>
- Query: SENSe[1]:PATTern:PROGram:DATA? <start-bit>, <length>
- Parameter:
  - <start-bit>: Address at which pattern setting starts
    - Setting range: 0 bit to (pattern length - 1) bit
    - Setting resolution: 1 bit
  - <length>: Number of bits to be set
  - <str>: Pattern data to be set
    - "B--------": Specifies pattern data in binary numbers
      - --------: 0, 1 (A maximum of 512 characters)
    - "H--------": Specifies pattern data in hexadecimal numbers
      - --------: 0 to 9, a to f, A to F (A maximum of 512 characters)
- Query response format: <str>: Response pattern
  - "H--------": Response to pattern data in hexadecimal numbers
    - --------: 0 to 9, A to F (A maximum of 512 characters)
- Description:
  A portion of the PROG pattern is overwritten by the contents of <str> from <start_bit> up to the bit length defined by <length>.
  Any data exceeding the range of the PROG pattern is ignored.
  If the number of data bits specified in <str> is smaller than <length>, the data up to the number of bits specified in <str> is set.
  If <length> is smaller than the number of data bits specified in <str>, the data up to the number of bits specified in <length> is set.
  In a query, data that exceeds the range of the PROG pattern or 512 characters (512 × 4 = 2048 bits) cannot be read.
- Example:
  When the 16-bit data 4142 (hexadecimal number) is set from address 0, use the following:
  SENS1:PATT:PROG:DATA 0, 16, "H4142"

6. `SENSe[1]:PATTern:PROGram:BDATa`

- Function — Sets the PROG pattern using block data
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:PROGram:BDATa <start-bit>, <length>, <block>
- Query — SENSe[1]:PATTern:PROGram:BDATa? <start-bit>, <length>
- Parameter
  - <start-bit>: Address at which pattern setting starts
    - Setting range: 0 bit to (pattern length - 1) bit
    - Setting resolution: 1 bit
  - <length>: Number of bits to be set
  - <block>: Data of the pattern to be set
- Query response format — <block>: Response pattern
- Description — A portion of the PROG pattern is overwritten by the contents of <block> from <start_bit> up to the bit length defined by <length>. Any data exceeding the range of the PROG pattern is ignored.
  If the number of data bits specified in <block> is smaller than <length>, the data up to the number of bits specified in <block> is set.
  If <length> is smaller than the number of data bits specified in <block>, the data up to the number of bits specified in <length> is set.
  A query cannot read data exceeding the range of the PROG pattern.
- Example of use — When the 16-bit data 4142 (hexadecimal number) is set from address 0, use the following:
  SENS1:PATT:PROG:BDAT 0, 16, #12AB

7. `SENSe[1]:PATTern:STM:NFRame` (Pattern option)

- Function — Sets the number of frames for the STM pattern
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:STM:NFRame <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Specifies the number of frames for the STM pattern

STM pattern	Setting range
STM4	1 to 107
STM16	1 to 26

Setting resolution:1

8. `SENSe[1]:PATTern:STM:DATA` (Pattern option)

- Function: Sets the STM pattern using character string data
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:STM:DATA <start_frm>,<start_row>, <start_col>,<length>,<str>
- Query: SENSe[1]:PATTern:STM:DATA? <start_frm>,<start_row>, <start_col>,<length>
- Parameter:

<start_frm> = The frame number where the pattern setting starts.
Setting range: 1 to the number of the frames
Setting resolution: 1

<start_row> = The row where the pattern setting starts.
Setting range: 1 to 9
Setting resolution: 1

<start_col> = The column where the pattern setting starts.

STM patterns	Setting ranges
STM4	1 byte to 1080 byte
STM16	1 byte to 4320 byte

Setting resolution:1 byte

<length> = The number of bits to be set
<str> = A pattern data to be set
"B--------": Specifying pattern data using binary numbers
--------: 0, 1
(A maximum of 512 characters)
"H--------": Specifying pattern data using a hexadecimal numbers
--------: 0 to 9, a to f, A to F
(A maximum of 512 characters)

- Query response format: <str> = A response pattern
"H--------": Specifying pattern data using hexadecimal numbers
--------: 0 to 9, a to f, A to F
(A maximum of 512 characters)

- Description — A portion of the PROG pattern is overwritten by the contents of <str> from <start_frm>, <start_row>, and <start_col> up to the bit length defined by <length>.
  Any data exceeding the range of the STM pattern is ignored.
  If the number of data bits specified in <str> is smaller than <length>, the data up to the number of bits specified in <str> is set.
  If <length> is smaller than the number of data bits specified in <str>, the data up to the number of bits specified in <length> is set.
  When the PRBS pattern is inserted in a payload, the number of bits in the payload portion is the same as the number stored in the pattern memory.
  In a query, data that exceeds the STM pattern range or 512 characters (512 × 4 = 2,048 bits) cannot be read. Data in the pattern memory is always read.
- Example — When 4142 (16-bit hexadecimal data) is set from the first row and the first column of the first frame, use the following:

  SENS 1: PATT: STM: DATA 1,1,1,16, "H4142"

9. `SENSe[1]:PATTern:STM:BDATa` (Pattern option)

- Function — Sets the STM pattern using block data
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:STM:BDATa <start_frm>,<start_row>, <start_col>,<length>,<block>
- Query — SENSe[1]:PATTern:STM:BDATa? <start_frm>,<start_row>, <start_col>,<length>
- Parameter — <start_frm> = The frame number where the pattern setting starts
  Setting range: 1 to the number of frames
  Setting resolution: 1

  <start_row> = The row where the pattern setting starts.
  Setting range: 1 to 9
  Setting resolution: 1

  <start_col> = The column where the pattern setting starts.

STM pattern	Setting range
STM4	1 byte to 1080 byte
STM16	1 byte to 4320 byte

  Setting resolution: 1 byte

  <length> = The number of bits to be set
  <block> = A pattern data to be set
- Query response format — <block> = A response pattern

- Description — A portion of the PROG pattern is overwritten by the contents of <block> from the <start_frm>, <start_row>, and <start_col> up to the bit length defined by <length>.
  Any data exceeding the range of the STM pattern is ignored.
  If the number of data bits specified in <block> is smaller than <length>, the data up to the number of bits specified in <block> is set.
  If <length> is smaller than the number of data bits specified in <block>, the data up to the number of bits specified in <length> is set.
  When the PRBS pattern is inserted in a payload, the number or bits in the payload portion is the same as the number stored in the pattern memory.
  In a query, data that exceeds the STM pattern range can not be read. Data in the pattern memory is always read.
- Example — When 4142 (16-bit hexadecimal data) is set from the first row and the first column of the first frame, use the following:

  SENS1:PATT:STM:BDAT 1,1,1,16,#12AB.

10. `SENSe[1]:PATTern:STM:PAYLoad:PRBS` (Pattern option)

- Function — Sets the function which inserts the PRBS pattern in an STM pattern payload to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:STM:PAYLoad:PRBS <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — The function which inserts a PRBS pattern in an STM pattern payload is enabled (ON)/disabled (OFF).

11. `SENSe[1]:PATTern:STM:PAYLoad:PRBS:SELect` (Pattern option)

- Function — Sets the PRBS pattern length to be inserted in an STM pattern payload
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:STM:PAYLoad:PRBS:SELect <type>
- Parameter — <type> = PRBS<n>
  <n> = 7,9,10,11,15,23,31
- Query response format — PRBS<n>
  <n> = 7,9,10,11,15,23,31
- Description — The PRBS pattern length which is inserted into an STM pattern payload is specified.

12. `SENSe[1]:PATTern:STM:PAYLoad:PRBS:MRATio` (Pattern option)

- Function: Sets the mark ratio of a PRBS pattern to be inserted in an STM pattern payload
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:STM:PAYLoad:PRBS:MRATio <ratio>
- Parameter: <ratio> = {MRATio<n> | MINVerted4}
- Query response format: MRAT<n> | MINV4
- Description: The mark ratio of a PRBS pattern which is inserted in an STM pattern payload is specified.

Command parameter	Mark ratio
MRAT<n>	<n> = 0 Mark ratio 0/8 <n> = 1 Mark ratio 1/8 <n> = 2 Mark ratio 1/4 <n> = 4 Mark ratio 1/2 <n> = 6 Mark ratio 3/4 <n> = 7 Mark ratio 7/8 <n> = 8 Mark ratio 8/8
MINV4	Mark ratio 1/2B

13. `SENSe[1]:PATTern:STM:SCRamble` (Pattern option)

- Function: Sets the STM pattern scrambling function to on or off
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:STM:SCRamble <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Whether an STM pattern is scrambled (ON) or not scrambled (OFF) is specified.

14. `SENSe[1]:PATTern:STM:B1` (Pattern option)

- Function: Sets the function which inserts B1 byte in an STM pattern on or off
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:STM:B1 <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Whether B1 byte is inserted into an STM pattern (ON) or not (OFF) is specified.

15. `SENSe[1]:PATTern:FLEXible:STABle` (Pattern option)

- Function: Edits patterns in a FLEX pattern sequence table
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:FLEXible:STABle <index>,<patt>,<length>, <trig_signal>
- Query: SENSe[1]:PATTern:FLEXible:STABle? <index>
- Parameter:
  <index> = The index to specify a pattern in a FLEX pattern sequence table
  Setting range: 1 to the number of the patterns
  Setting resolution: 1
  <patt> = Pattern to be selected {<prog_no>| PRBS}
  <length> = The pattern length when <patt> is PRBS.
  Setting range: 128 bits to 2,097,152 bits
  Setting resolution: 64 bits
  <trig_signal> = FLEX Trigger output level{HIGH | LOW}
- Query response format: <patt>,<length>,<trig_signal>
- Description:
  Patterns in a FLEX pattern sequence table are edited.
  The <length> setting is effective only if the selected pattern is a PRBS pattern.
  Set a value if a PROG pattern is selected.
  The PROG pattern length used in FLEX patterns is specified by the following:
  SENS1:PATT:FLEX:PROG [:LENG]

In a query, the length of a PROG pattern is read using <length> if the selected pattern is a PROG pattern.

<patt>	Pattern to be selected
<prog_no>	PROG pattern Specifies the PROG pattern number. Setting range: 1 to 127 Setting resolution: 1
PRBS	PRBS pattern

<trig_signal>	Output level
HIGH	HIGH level A HIGH level signal is generated in FLEX Trigger
LOW	LOW level A LOW level signal is generated in FLEX Trigger

16. `SENSe[1]:PATTern:FLEXible:STABle:COUNt?` (Pattern option)

- Function: Sends a query for the number of patterns in a FLEX pattern sequence table.
- Availability of command and query: Query
- Query: SENSe[1]:PATTern:FLEXible:STABle:COUNt?
- Query response format: NR1 (integer type)
- Description: A query for the number of patterns in a FLEX pattern sequence table is sent.

17. `SENSe[1]:PATTern:FLEXible:STABle:NEW` (Pattern option)

- Function: Adds a pattern to a FLEX pattern sequence table
- Availability of command and query: Command
- Command: SENSe[1]:PATTern:FLEXible:STABle:NEW <index>,<patt>, <length>, <trig_signal>
- Parameter: <index> = The index to specify a pattern in a FLEX pattern sequence table
  Setting range: 1 to (the number of the patterns + 1)
  Setting resolution: 1

  <patt> = Pattern to be selected {<prog_no>| PRBS}
  <length> = The pattern length when <patt> is PRBS.
  Setting range: 128 bits to 2,097,152 bits

Setting resolution: 64 bits
<trig_signal> = Output level of FLEX Trigger: {HIGH| LOW}

- Description

Patterns are added to a FLEX pattern sequence table. A pattern is added to the position specified with <index>.
The maximum number of patterns in a pattern sequence table is 1024. No more patterns can be added if 1024 patterns already exist in the table.
The specification of <length> is effective only if the selected pattern is a PRBS pattern.
Specify a value if a PROG pattern is selected.
The PROG pattern length used in FLEX patterns is specified by the following:
SENS1:PATT:FLEX:PROG [:LENG]

<patt>	Pattern to be selected
<prog_no>	PROG pattern Specifies the PROG pattern number Setting range: 1 to 127 Setting resolution: 1
PRBS	PRBS pattern

<trig_signal>	Output level
HIGH	HIGH level A HIGH level signal is generated in FLEX Trigger
LOW	LOW level A LOW level signal is generated in FLEX Trigger

18. `SENSe[1]:PATTern:FLEXible:STABle:REMove` (Pattern option)

- Function: Delets patterns in a FLEX pattern sequence table
- Availability of command and query: Command
- Command: SENSe[1]:PATTern:FLEXible:STABle:REMove <index>
- Parameter: <index> = The index to specify a pattern in a FLEX pattern sequence table
  Setting range: 1 to the number of the patterns
  Setting resolution: 1
- Description: A pattern is deleted from a FLEX pattern sequence table.
  All of the patterns cannot be deleted. An index-1 pattern cannot be deleted if the index-2 pattern is a PBRS pattern.

19. `SENSe[1]:PATTern:FLEXible:STABle:DEFault` (Pattern option)

- Function: Specifies a default setting for the FLEX pattern sequence table
- Availability of command and query: Command
- Command: SENSe[1]:PATTern:FLEXible:STABle:DEFault
- Description: The default of a FLEX pattern sequence table is restored.

20. `SENSe[1]:PATTern:FLEXible:PROGram[:LENGth]` (Pattern option)

- Function: Specifies the PROG pattern length used in a FLEX pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:FLEXible:PROGram[:LENGth] <prog_no>, <length>
- Query: SENSe[1]:PATTern:FLEXible:PROGram[:LENGth]? <prog_no>
- Parameter: <prog_no> = The order number of a PROG pattern used in a FLEX pattern
  - Setting range: 1 to 127
  - Setting resolution: 1

  <length> = A PROG pattern length
  - Setting range: 128 bits to 65,536 bits
  - Setting resolution: 64 bits
- Query response format: <length>= NR1 (integer type)
- Description: The PROG pattern length used in a FLEX pattern is specified.

21. `SENSe[1]:PATTern:FLEXible:PROGram:DATA` (Pattern option)

- Function: Sets the PROG pattern used in a FLEX pattern using character string
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:FLEXible:PROGram:DATA <prog_no>, <start_bit>,<length>,<str>
- Query: SENSe[1]:PATTern:FLEXible:PROGram:DATA? <prog_no>, <start_bit>,<length>

- Parameter

<prog_no> = The PROG pattern number used in a FLEX pattern.
Setting range: 1 to 127
Setting resolution: 1

<start_bit> = The address where the specification of a pattern starts.
Setting range: 0 bit to (the pattern length -1) bit
Setting resolution: 1 bit

<length> = The number of bits to be specified
<str> = Pattern data to be set
"B--------": Specifying pattern data using biary numbers
--------: 0, 1
(A maximum of 512 characters)
"H--------": Defining pattern data using hexadecimal numbers
--------: 0 to 9, a to f, A to F
(A maximum of 512 characters)

- Query response format

<str> = A response pattern
"H--------": Defining pattern data using hexadecimal numbers
--------: 0 to 9, a to f, A to F
(A maximum of 512 characters)

- Description

A portion of PROG pattern specified by <prog_no> is overwritten by the content of <str> from <start_bit> up to the bit length specified by <length>.
Data exceeding the PROG pattern range specified with <prog_no> is ignored.
If the number of bits of data in <str> is smaller than <length>, the data up to the number of bits specified in <str> is set.
If <length> is smaller than the number of bits of data in <str>, the data up to the number of bits spedified in<length> is set.
In a query, data that exceeds the PROG pattern range specified in <prog_no> or 512 characters (512 × 4 = 2,048 bits) cannot be read.

- Example

To set 4142 (16-bit hexadecimal data) in the area starting at address 0 of PROG pattern 1, use the following:

SENS1:PATT:FLEX:PROG:DATA 1,0 16, "H4142"

22. `SENSe[1]:PATTern:FLEXible:PROGram:BDATa` (Pattern option)

- Function: Specifies the PROG pattern used in a FLEX pattern using block data
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:FLEXible:PROGram:BDATa <prog_no>, <start_bit>,<length>,<block>
- Query: SENSe[1]:PATTern:FLEXible:PROGram:BDATa? <prog_no>, <start_bit>,<length>
- Parameter:
  <prog_no> = The PROG pattern number used in a FLEX pattern
  Setting range: 1 to 127
  Setting resolution: 1
  <start_bit> = The address where the pattern setting starts.
  Setting range: 0 bit to (the pattern length -1) bit
  Setting resolution: 1 bit
  <length> = The number of bits to be specified
  <block> Pattern data to be set
- Query response format: <block> = A response pattern
- Description:
  A portion of a PROG pattern specified by <prog_no> is overwritten by the content of <block> from <start_bit> up to the bit length specified by <length> .
  Data exceeding the PROG pattern range specified with <prog_no> is ignored.
  If the number of bits of data in <block> is smaller than <length>, the data up to the number of bits specified in <block> is set.
  If <length> is smaller than the number of bits of data in <block>, the data up to the number of bits indicated by <length> is set.
  In a query, data that exceeds the PROG pattern range specified in <prog_no> cannot be read.
- Example:
  To set 4142 (16-bit hexadecimal data) in the area starting at address 0 of PROG pattern 1, use the following:
  SENS1: PATT: FLEX: PROG: BDAT 1, 0, 16, #12AB

23. `SENSe[1]:PATTern:FLEXible:PRBS:SELect` (Pattern option)

- Function: Sets the PRBS pattern length used in a FLEX pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:FLEXible:PRBS:SELect <type>
- Parameter: <type> = PRBS<n>
  <n> = 7,9,10,11,15,23,31
- Query response format: PRBS<n>
  <n> = 7,9,10,11,15,23,31
- Description: The PRBS pattern length used in a FLEX pattern is specified.

24. `SENSe[1]:PATTern:FLEXible:PRBS:MRATio` (Pattern option)

- Function: The specification of the mark ratio of a PRBS pattern used in a FLEX pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:PATTern:FLEXible:PRBS:MRATio <ratio>
- Parameter: <ratio> = {MRATio<n> | MINVerted4}
- Query response format: MRAT<n> | MINV4
- Description: The mark ratio of a PRBS pattern used in a FLEX pattern is specified.

Command parameter	Mark ratio
MRAT<n>	<n> = 0 Mark ratio 0/8 <n> = 1 Mark ratio 1/8 <n> = 2 Mark ratio 1/4 <n> = 4 Mark ratio 1/2 <n> = 6 Mark ratio 3/4 <n> = 7 Mark ratio 7/8 <n> = 8 Mark ratio 8/8
MINV4	Mark ratio 1/2B

25. `SENSe[1]:PATTern:POLarity`

- Function — Sets a pattern polarity
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:POLarity <type>
- Parameter — <type> = {NORMal | INVerted}
- Query response format — NORM | INV
- Description — Sets a pattern polarity

Command parameters	Pattern polarities
NORM	Positive logic
INV	Negative logic

26. `SENSe[1]:PATTern:TRACk`

- Function — Sets the same pattern as the PPG module pattern to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:PATTern:TRACk <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Specifies whether the same pattern as the PPG module pattern is used (ON) or not (OFF)

27. `SENSe[1]:SYNChronisat[:AUTO]`

- Function — Sets the automatic pattern synchronization
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat[:AUTO] ONCE | <bool>
- Parameter — ONCE: Pattern synchronization is performed only once.
  <bool>
- Query response format — 0 | 1
- Description — Specifies whether pattern synchronization is performed only once (ONCE), automatically (ON), or not at all (OFF)

28. `SENSe[1]:SYNChronisat:PATTern:PROGram:ADDRess`

- Function — Sets the synchronous pattern address when a PROG pattern is generated
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat:PATTern:PROGram:ADDRess <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the synchronous pattern address when aPROG pattern is generated
  Setting range: 0 bit to (pattern length - 1) bit
  Setting resolution: 1 bit

29. `SENSe[1]:SYNChronisat:PATTern:FLEXible:INDex` (Pattern option)

- Function — Specifies a synchronization pattern index of the FLEX patterns
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat:PATTern:FLEXible:INDex <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — A synchronization pattern index for the FLEX patterns is specified. The index setting is disabled if the selected pattern is a PRBS pattern.
  Setting range: 1 to the number of patterns
  Setting resolution: 1

30. `SENSe[1]:SYNChronisat:PATTern:FLEXible:ADDRess` (Pattern option)

- Function: Specifies a synchronization pattern address for the FLEX patterns
  Availability of command and query
  Command/Query
- Command: SENSe[1]:SYNChronisat:PATTern:FLEXible:ADDRess <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: A synchronization pattern address for the FLEX patterns is specified.

Index Patterns subsequent to a synchronization pattern index	Setting range
PROG	0 bit to (Pattern length of a synchronization pattern index - 1) bit
PRBS	0 bit to (Pattern length of a synchronization pattern index - 32) bit

Setting resolution:1 bit

31. `SENSe[1]:SYNChronisat:PATTern:LENGth`

- Function: Sets the synchronous pattern length
- Availability of command and query
  Command/Query
- Command: SENSe[1]:SYNChronisat:PATTern:LENGth <int>
- Parameter: <int> = 8, 16, 24, 32
- Query response format: NR1 (integer type)
- Description: Specifies the length (number of bits) of a synchronization pattern used to perform synchronization pull-in processing for the PROG pattern

32. `SENSe[1]:SYNChronisat:THReshold:AUTO`

- Function — Specifies whether or not the threshold error rate used for synchronization judgment is automatically set
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat:THReshold:AUTO <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Specifies whether the threshold error rate for in-sync or out-of-sync judgments is automatically set (ON) or not (OFF)

33. `SENSe[1]:SYNChronisat:THReshold:PRBS:GAIN`

- Function — Specifies the threshold for error rates to judge the establishment of synchronization of the PRBS patterns (including PRBS patterns in STM or FLEX patterns)
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat:THReshold:PRBS:GAIN <real>
- Parameter — <real> = 1E-2, 1E-3, 1E-4, 1E-5, 1E-6, 1E-7
- Query response format — NR3 (floating-point type)
- Description — Specifies the threshold for error rates to judge the establishment of synchronization of the PRBS patterns (including PRBS patterns in STM or FLEX patterns).
  The relationship between the threshold for a synchronization pull-in judgment (Sync Gain Threshold) and the out-of-sync threshold (Sync Loss Threshold) must meet the condition below:
  Sync Gain Threshold ≤ Sync Loss Threshold.

34. `SENSe[1]:SYNChronisat:THReshold:PRBS:LOSS`

- Function — Specifies the threshold for error rates to judge the out-of-synchronization state of the PRBS patterns (including PRBS patterns in STM or FLEX patterns)
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat:THReshold:PRBS:LOSS <real>
- Parameter — <real> = 1E-2, 1E-3, 1E-4, 1E-5, 1E-6, 1E-7
- Query response format — NR3 (floating-point type)
- Description — Specifies the threshold for error rates to judge the out-of-synchronization state of the PRBS patterns (including PRBS patterns in STM or FLEX patterns)
  The relationship between the threshold for a synchronization pull-in judgment (Sync Gain Threshold) and the out-of-sync threshold (Sync Loss Threshold) must meet the condition below:
  Sync Gain Threshold ≤ Sync Loss Threshold.

35. `SENSe[1]:SYNChronisat:THReshold:MEMory:GAIN`

- Function — Specifies the threshold for error rates to judge the establishment of synchronization of the ZSUB, PROG, STM, or FLEX patterns
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat:THReshold:MEMory:GAIN <real>
- Parameter — <real> = 1E-2, 1E-3, 1E-4, 1E-5, 1E-6, 1E-7, 1E-8, 1E-9, 1E-10
- Query response format — NR3 (floating-point type)
- Description — Specifies the threshold for error rates to judge the establishment of synchronization of the ZSUB, PROG, STM, or FLEX patterns. For PRBS patterns used in STM or FLEX patterns, a separate threshold is specified with the SENS[1]:SYNC:THR:PRBS:GAIN command.
  The relationship between the threshold for a synchronization pull-in judgment (Sync Gain Threshold) and the out-of-sync threshold (Sync Loss Threshold) must meet the condition below:
  Sync Gain Threshold ≤ Sync Loss Threshold.

36. `SENSe[1]:SYNChronisat:THReshold:MEMory:LOSS`

- Function — Sets the threshold error rate for out-of-sync judgments of the ZSUB, PROG, STM, or FLEX patterns
- Availability of command and query — Command/Query
- Command — SENSe[1]:SYNChronisat:THReshold:MEMory:LOSS <real>
- Parameter — <real> = 1E-2, 1E-3, 1E-4, 1E-5, 1E-6, 1E-7, 1E-8, 1E-9, 1E-10
- Query response format — NR3 (floating-point type)
- Description — The threshold for error rates to judge the out-of-synchronization state of the ZSUB, PROG, STM, or FLEX patterns is specified. For PRBS patterns used in STM or FLEX patterns, a separate threshold is specified with the SENS[1]:SYNC:THR:PRBS:LOSS command. The relationship between the threshold for a synchronization pull-in judgment (Sync Gain Threshold) and the out-of-sync threshold (Sync Loss Threshold) must meet the condition below:
  Sync Gain Threshold ≤ Sync Loss Threshold.

37. `SENSe[1]:MASK:ROUTe<n>`

- Function — Sets the measurement route masking to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:MASK:ROUTe<n> <bool>
  <n> = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Enables (ON) or disables (OFF) the measurement route masking specified in <n>

38. `SENSe[1]:ASEarch[:STATe]`

- Function — Sets auto-search on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:ASEarch[:STATe] <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether (ON) or not (OFF) to start auto search

39. `SENSe[1]:ASEarch:FAIL?`

- Function — Sends a Query for the automatic search execution result
- Availability of command and query — Query
- Query — SENSe[1]:ASEarch:FAIL?
- Query response format — 0 | 1
- Description — Queries about the result of an automatic search execution
  If auto search is unsuccessful or it is not executed at all, 1 is returned. If auto search is successful, 0 is returned.
  Before this command execution, first confirm that Auto-search has ended.

40. `SENSe[1]:ASEarch:CLOCk`

- Function — Enabling or disabling Auto-search on the delay and input polarity of clock input
- Availability of command and query — Command/Query
- Command — SENSe[1]:ASEarch:CLOCk <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Enables (ON) or disables (OFF) Auto-search on the delay and input polarity of clock input

41. `SENSe[1]:ASEarch:TVOLtage`

- Function — Enabling or disabling Auto-search on the threshold voltage of data input
- Availability of command and query — Command/Query
- Command — SENSe[1]:ASEarch:TVOLtage <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether (ON) or not (OFF) to search an automatically for the data input threshold voltage

42. `SENSe[1]:ASEarch:PATTern`

- Function — Enabling or disabling Auto-search on the settings related to patterns
- Availability of command and query — Command/Query
- Command — SENSe[1]:ASEarch:PATTern <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Enables (ON) or disables (OFF) Auto-search on the pattern length, mark ratio, and pattern polarity of the PRBS pattern, or the pattern polarity of a non-PRBS pattern

43. `SENSe[1]:MONitor:HISTory:CLEar`

- Function — Clears the monitor tool bar history information
- Availability of command and query — Command
- Command — SENSe[1]:MONitor:HISTory:CLEar
- Description — Clears monitor tool bar history information

44. `SENSe[1]:LOG:BMEasurement`

- Function — Sets the function that records a basic measurement result in the log to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:LOG:BMEasurement <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether a basic measurement result is recorded in the log (ON) or not (OFF)
  Only the last measurement result is recorded in the log.

45. `SENSe[1]:LOG:HISTory:SCLoss`

- Function — Sets the function that records the sync/clock loss history in the log to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:LOG:HISTory:SCLoss <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether sync/clock loss generation and recovery times are recorded as history in the log (ON) or not (OFF)

46. `SENSe[1]:LOG:HISTory:MDATa:IERate`

- Function — Sets the function that records the immediate error rate value history to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:LOG:HISTory:MDATa:IERate <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether measurement data for immediate error rate values are recorded as history in the log (ON) or not (OFF)

47. `SENSe[1]:LOG:HISTory:MDATa:IECount`

- Function — Sets the function that records the immediate error count value history to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:LOG:HISTory:MDATa:IECount <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether measurement data for immediate error count values are recorded as history in the log (ON) or not (OFF)

48. `SENSe[1]:LOG:HISTory:MDATa:EINTervals`

- Function — Sets the function that records the error interval value history in the log to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:LOG:HISTory:MDATa:EINTervals <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether measurement data for error interval values are recorded as history in the log (ON) or not (OFF)

49. `SENSe[1]:LOG:HISTory:MDATa:EFINtervals`

- Function — Sets the function that records the error free interval value history to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:LOG:HISTory:MDATa:EFINtervals <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether measurement data for error free interval values are recorded as history in the log (ON) or not (OFF)

50. `SENSe[1]:LOG:HISTory:MDATa:FREQuency`

- Function — Sets the function that records the frequency value history to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:LOG:HISTory:MDATa:FREQuency <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Specifies whether the measurement data of frequency values is recorded in a log (ON) or not (OFF)

51. `SENSe[1]:BMEasurement[:STATe]`

- Function — Starting and stopping basic measurement
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement[:STATe] <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether to start (ON) or stop (OFF) the basic measurement function
  If the measurement mode is in the single mode, the basic measurement function is automatically set to OFF when measurement terminates.

52. `SENSe[1]:BMEasurement:MTIMe:MODE`

- Function — Sets the measurement timer mode
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:MTIMe:MODE <type>
- Parameter — <type> = {SINGle | REPeat | UNTimed}
- Query response format — SING | REP | UNT
- Description — Sets a measurement mode

Command parameter	Measurement mode
SING	Single mode. Measures only once for the measurement period.
REP	Repeat mode. Measures repeatedly for the measurement period.
UNT	Untimed mode. Measurement is required regardless of the measurement period specified.

53. `SENSe[1]:BMEasurement:MTIMe:INTerval`

- Function — Sets the measurement interval
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:MTIMe:INTerval <real>
- Parameter — <real> = 0.1, 1.0
- Query response format — NR2 (fixed-point type)
- Description — Sets a measurement interval
  Set 0.1 or 1 second as the measurement interval.

54. `SENSe[1]:BMEasurement:MTIMe:PERiod`

- Function — Sets the measurement period
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:MTIMe:PERiod <day>, <hour>, <minute>, <second>
- Parameter
  <day>: Numeric data from 0 to 99 (days)
  <hour>: Numeric data from 0 to 23 (hours)
  <minute>: Numeric data from 0 to 59 (minutes)
  <second>: Numeric data from 0 to 59 (seconds)
- Query response format — <day>, <hour>, <minute>, <second>
  <day> = <hour> = <minute> = <second> = NR1 (integer type)
- Description — Sets a measurement period
  Setting range: 1 second to 99 days 23 hours 59 minutes 59 seconds

55. `SENSe[1]:BMEasurement:DMODe`

- Function — Sets an error detection mode
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:DMODe <type>
- Parameter — <type> = {OIT | OPT | SOT}
- Query response format — OIT | OPT | SOT
- Description — An error detection mode is set.

Command parameter	Description
OIT	Omitting errors, Inserting errors and Total errors are detected.
OPT	Errors in Overhead areas, Payload areas and Total areas are detected.
SOT	Errors in Specific areas, Other areas (excluding the specific areas) and Total areas are detected.

56. `SENSe[1]:BMEasurement:BURSt`

- Function — Sets the burst mode to on or off
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:BURSt <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Specifies whether the burst mode is enabled (ON) or disabled (OFF)

57. `SENSe[1]:BMEasurement:EVALuate:CLOSs`

- Function — Sets whether or not to include a clock loss interval in the measurement
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:EVALuate:CLOSs <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether (ON) or not (OFF) to include a clock loss interval in the measurement

58. `SENSe[1]:BMEasurement:EVALuate:SLOSs`

- Function — Sets whether or not to include a sync loss interval in the measurement
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:EVALuate:SLOSs <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Sets whether (ON) or not (OFF) to include a sync loss interval in the measurement

59. `SENSe[1]:BMEasurement:EPERformance:THReshold`

- Function — Sets the error rate threshold of the period to be included in US computation, SES computation, and DM computation
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:EPERformance:THReshold <us_ses>, <dm>
- Parameter — <us_ses> = 1E-3, 1E-4
<dm> = 1E-6, 1E-8
- Query response format — <us_ses>, <dm>
<us_ses> = <dm> = NR3 (floating-point type)
- Description — Sets the error rate threshold of the period to be included in Unavailable Seconds (US) computation, Severely Errored Seconds (SES) computation, and Degraded Minutes (DM) computation in error performance measurement

Command parameter	Description
<us_ses>	Error rate threshold of the period to be included in US computation and SES computation
<dm>	Error rate threshold of the period to be included in DM computation

<us_ses> and <dm> are paired as follows:

<us_ses>	<dm>
1E-3	1E-6
1E-4	1E-8

60. `SENSe[1]:BMEasurement:CDATa`

- Function — Sets the partial results of basic measurements to ON or OFF
- Availability of command and query — Command/Query
- Command — SENSe[1]:BMEasurement:CDATa <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Specifies whether the partial results of basic measurement are to be displayed (ON) or not (OFF)

61. `SENSe[1]:SPECific:PROGram:STARt:ADDRess`

- Function — Sets the start address in a specific area of the PROG pattern
- Availability of command and query — Command/Query
- Command — SENSe[1]:SPECific:PROGram:STARt:ADDRess <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — The start address in a specific area of the PROG pattern is specified.
  Setting range: 0 bit to (the pattern length-1) bit
  Setting resolution: 1 bit

62. `SENSe[1]:SPECific:PROGram:END:ADDRess`

- Function — Sets the end address in a specific area of the PROG pattern
- Availability of command and query — Command/Query
- Command — SENSe[1]:SPECific:PROGram:END:ADDRess <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — The end address in a specific area of the PROG pattern is specified.
  Setting range: 0 bit to (the pattern length-1) bit
  Setting resolution: 1 bit

63. `SENSe[1]:SPECific:ZSUBstitut:STARt:ADDRess`

- Function — Sets the start address of a specific area in the ZSUB pattern
- Availability of command and query — Command/Query
- Command — SENSe[1]:SPECific:ZSUBstitut:STARt:ADDRess <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — The start address in a specific area of the ZSUB pattern is specified.
  Setting range: 0 bit to (the pattern length-1) bit
  Setting resolution: 1 bit

64. `SENSe[1]:SPECific:ZSUBstitut:END:ADDRess`

- Function — Sets the end address in a specific area of the ZSUB pattern
- Availability of command and query — Command/Query
- Command — SENSe[1]:SPECific:ZSUBstitut:END:ADDRess <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — The end address in a specific area of the ZSUB patterns is specified.
  Setting range: 0 bit to (the pattern length-1) bit
  Setting resolution: 1 bit

65. `SENSe[1]:SPECific:STM:STARt:FNUMber` (Pattern option)

- Function — Sets the number of the start frame in a specific area of the STM patterns
- Availability of command and query — Command/Query
- Command — SENSe[1]:SPECific:STM:STARt:FNUMber <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — The number of the start frame of a specific area in the STM pattern is specified.
  Setting range: 1 to the number of the frames
  Setting resolution: 1

66. `SENSe[1]:SPECific:STM:STARt:ROW` (Pattern option)

- Function: Sets the start row in a specific area of the STM pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:STM:STARt:ROW <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The start row in a specific area of the STM pattern is specified.
  Setting range: 1 to 9
  Setting resolution: 1

67. `SENSe[1]:SPECific:STM:STARt:COLumn` (Pattern option)

- Function: Sets the start column in a specific area of the STM pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:STM:STARt:COLumn <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The start column in a specific area of the STM pattern is specified.

STM pattern	Setting range
STM4	1 byte to 1080 byte
STM16	1 byte to 4320 byte

Setting resolution: 1 byte

68. `SENSe[1]:SPECific:STM:END:FNUMber` (Pattern option)

- Function: Sets the end frame number in a specific area of the STM pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:STM:END:FNUMber <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The end frame in a specific area of the STM pattern is specified.
  Setting range: 1 to the number of the frames
  Setting resolution: 1

69. `SENSe[1]:SPECific:STM:END:ROW` (Pattern option)

- Function: Sets the end row in a specific area of the STM pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:STM:END:ROW <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The end row in a specific area of the STM pattern is specified.
  Setting range: 1 to 9
  Setting resolution: 1

70. `SENSe[1]:SPECific:STM:END:COLumn` (Pattern option)

- Function: Sets the end column in a specific area of the STM patterns
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:STM:END:COLumn <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The end column in a specific area of the STM pattern is specified.

STM pattern	Setting range
STM4	1 byte to 1080 byte
STM16	1 byte to 4320 byte

Setting resolution: 1 byte

71. `SENSe[1]:SPECific:STM:MODE` (Pattern option)

- Function: Sets the measurement mode for a specific area of the STM pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:STM:MODE <type>
- Parameter: <type> = {PATTern | FRAMe}
- Query response format: PATT | FRAM
- Description: The measurement mode for a specific area of the STM pattern is specified.

Command parameters	Specific area measurement modes
PATT	Pattern mode Measurement is conducted in a specific area for every pattern.
FRAM	Frame mode Measurement is conducted in a specific area for every frame.

72. `SENSe[1]:SPECific:FLEXible:STARt:INDex` (Pattern option)

- Function: Sets the start index in a specific area of the FLEX pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:FLEXible:STARt:INDex <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The start index in a specific area of the FLEX pattern is specified.
  Setting range: 1 to the number of patterns
  Setting resolution: 1

73. `SENSe[1]:SPECific:FLEXible:STARt:ADDRess` (Pattern option)

- Function: Sets the start address in a specific area of the FLEX pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:FLEXible:STARt:ADDRess <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The start address in a specific area of the FLEX pattern is specified.
  Setting range: 0 bits to (pattern length of the start index of a specific area-1) bits
  Setting resolution: 1 bit

74. `SENSe[1]:SPECific:FLEXible:END:INDex` (Pattern option)

- Function: Sets the end index in a specific area of the FLEX pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:FLEXible:END:INDex <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The end index in a specific area of the FLEX pattern is specified.
  Setting range: 1 to the number of patterns
  Setting resolution: 1

75. `SENSe[1]:SPECific:FLEXible:END:ADDRess` (Pattern option)

- Function: Sets the end address in a specific area of the FLEX pattern
- Availability of command and query: Command/Query
- Command: SENSe[1]:SPECific:FLEXible:END:ADDRess <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The end address in a specific area of the FLEX pattern is specified.
  Setting range: 0 bits to (the pattern length of the end index of a specific area - 1) bits
  Setting resolution: 1 bit

76. `SENSe[1]:EPANalysis:ERECord[:STATe]` (Error phase analysis option)

- Function: Sets the error recording function to on or off
- Availability of command and query: Command/Query
- Command: SENSe[1]:EPANalysis:ERECord[:STATe] <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Specifies whether error recording, together with basic measurement, is enabled (ON) or disabled (OFF). In order for error recording to actually be executed, the following conditions must be met: (1) the specified pattern can not be a PRBS pattern, and (2) the pattern length must be 256 bits or longer and an integral multiple of 32 bits.

Pattern type	Description
PRBS	Error recording is impossible
PROG	Errors can be recorded if the pattern length is 256 bits or longer and an integral multiple of 32 bits.
ZSUB	Errors can not be recorded for ZSUB7.
STM	Errors can be recorded.
FLEX	Errors can be recorded if the total pattern length is 256 bits or longer.

77. `SENSe[1]:EPANalysis:ERECord:AREa` (Error phase analysis option)

- Function: Sets the error recording area
- Availability of command and query: Command/Query
- Command: SENSe[1]:EPANalysis:ERECord:AREa <type>
- Parameter: <type> = { SPECific | TOTal }
- Query response format: SPEC | TOT
- Description: An areas is specified to record errors.

Command parameter	An area to record errors
SPEC	A specific area To enable the setting, it is necessary to set the error detection mode to Specific/ Other/Total. Commands in the SENS[1]:SPEC layer are used to specify a specific area.
TOT	Every pattern

5.5.2 INPut[1] Subsystem

1. `INPut[1]:POLarity`

- Function — Sets the DATA input polarity
- Availability of command and query — Command/Query
- Command — INPut[1]:POLarity <type>
- Parameter — <type> = {NORMal | INVerted}
- Query response format — NORM | INV
- Description — Sets the DATA input polarity

Command parameter	DATA input polarity
NORM	Normal
INV	Inverted

2. `INPut[1]:TVOLtage`

- Function — Sets the DATA input threshold voltage
- Availability of command and query — Command/Query
- Command — INPut[1]:TVOLtage <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point type)
- Description — Sets the DATA input threshold voltage

Termination type	Setting range
PECL (+3 V)	+3.150 to +4.250 V
LVPECL (+1.3 V)	+1.450 to +2.550 V
to GND (0 V)	-2.040 to +2.040 V
ECL (-2 V)	-1.850 to -0.750 V
CML	(Vcc-1.100) V to (Vcc+0.100) V *1

*1 Vcc indicates DATA input termination voltage.

Setting resolution: 0.001 V
Setting unit: V

3. `INPut[1]:TERMination`

- Function — Sets the DATA input termination type
- Availability of command and query — Command/Query
- Command — INPut[1]:TERMination <type>
- Parameter — <type> = {P3V | P1R3V | GND | M2V | CML}
- Query response format — P3V | P1R3V | GND | M2V | CML
- Description — Sets the DATA input termination type

Command parameter	Termination type
P3V	PECL (+3 V)
P1R3V	LVPECL (+1.3 V)
GND	to GND (0 V)
M2V	ECL (-2 V)
CML	CML

4. `INPut[1]:TERMination:VARiable`

- Function — Sets the DATA input termination voltage variable function to on or off
- Availability of command and query — Command/Query
- Command — INPut[1]:TERMination:VARiable <bool>s enabled (ON) or disabled (OFF) is specified.
  The input is terminated with a specified termination voltage if the setting is set to ON, and terminated with the initial voltage if it is set to OFF.
  If the termination type is set to GND, no setting is possible.

5. `INPut[1]:TERMination:VOLTage`

- Function — Sets the DATA input termination voltage
- Availability of command and query — Command/Query
- Command — INPut[1]:TERMination:VOLTage <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point-type)
- Description — The DATA input termination voltage is specified.
  To make the setting effective, the DATA input termination voltage variable function must be set to ON.
  If the termination type is set to GND, no setting is possible.

Termination type	Setting range
PECL (+3 V)	+2.70 V to +3.30 V
LVPECL (+1.3 V)	+1.00 V to +1.60 V
ECL (-2 V)	-2.30 V to -1.70 V
CML	0.00 V to +3.50 V

Setting resolution: 0.05 V
Setting unit: V

5.5.3 INPut2 Subsystem

1. `INPut2:POLarity`

- Function — Setting of the CLOCK input polarity
- Availability of command and query — Command/Query
- Command — INPut2:POLarity <type>
- Parameter — <type> = {NORMal | INVerted}
- Query response format — NORM | INV
- Description — Sets a CLOCK input polarity

Command parameter	CLOCK input polarity
NORM	Normal
INV	Inverted

2. `INPut2:DELay`

- Function — Sets the CLOCK delay time
- Availability of command and query — Command/Query
- Command — INPut2:DELay <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the CLOCK delay time
  Setting range: -1000 to +1000 ps
  Setting resolution: 1 ps
  Setting unit: ps

3. `INPut2:TERMination`

- Function — Sets the CLOCK input termination type
- Availability of command and query — Command/Query
- Command — INPut2:TERMination <type>
- Parameter — <type> = {P3V | P1R3V | GND | M2V | CML}
- Query response format — P3V | P1R3V | GND | M2V | CML
- Description — Sets the CLOCK input termination type.

Command parameter	Termination type
P3V	PECL (+3 V)
P1R3V	LVPECL (+1.3 V)
GND	to GND (0 V)
M2V	ECL (-2 V)
CML	CML

4. `INPut2:TERMination:VARiable`

- Function — The enabling (ON) and disabling (OFF) of the CLOCK input termination voltage variable function
- Availability of command and query — Command/Query
- Command — INPut2:TERMination:VARiable <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Whether the CLOCK input termination voltage variable function is enabled (ON) or disabled (OFF) is specified. If the setting is set to ON, a specified termination voltage is used for the termination, and the initial voltage is used if it is set to OFF.
  If the termination type is set to GND, no setting is possible.

5. `INPut2:TERMination:VOLTage`

- Function — Sets the CLOCK input termination voltage
- Availability of command and query — Command/Query
- Command — INPut2:TERMination:VOLTage <real>
- Parameter — <real>
- Query response format — NR2 (fixed-point-type)
- Description — The CLOCK input termination type is specified.
  To make the setting effective, the CLOCK input termination voltage variable function must be set to ON.
  If the termination type is set to GND, no setting is possible.

Termination type	Setting range
PECL (+3 V)	+2.70 V to +3.30 V
LVPECL (+1.3 V)	+1.00 V to +1.60 V
ECL (-2 V)	-2.30 V to -1.70 V
CML	0.00 V to +3.50 V

Setting resolution: 0.05 V
Setting unit: V

5.5.4 SOURce8 Subsystem

1. `SOURce8:TRIGger[:MODE]`

- Function — Specifies a trigger output source
- Availability of command and query — Command/Query
- Command — SOURce8:TRIGger[:MODE] <type>
- Parameter — <type> = {CLOCk16 | PATTern | FLEXible}
- Query response format — CLOC16 | PATT | FLEX
- Description — Specifies a trigger output source.

Command parameter	Trigger output source
CLOC16	1/16 frequency clock of the output source clock
PATT	Trigger synchronized with the pattern
FLEX	The level specified for FLEX Trigger in the pattern sequence table Setting is possible if the pattern option is installed and the pattern type is FLEX.

2. `SOURce8:TRIGger:STM:MODE` (Pattern option)

- Function — Sets a trigger output mode in the STM pattern
- Availability of command and query — Command/Query
- Command — SOURce8:TRIGger:STM:MODE <type>
- Parameter — <type> = {PATTern | FRAMe}
- Query response format — PATT | FRAM
- Description — A trigger output mode in the STM pattern is specified.

Command parameter	STM pattern trigger output mode
PATT	Pattern mode A trigger is generated for every pattern.
FRAM	Frame mode A trigger is generated for every frame.

5.5.5 SOURce9 Subsystem

1. `SOURce9:AUX[:MODE]`

- Function — Sets an AUX output source
- Availability of command and query — Command/Query
- Command — SOURce9:AUX[:MODE] <type>
- Parameter — <type> = {DTYPe | SYNChronisat}
- Query response format — DTYP | SYNC
- Description — An AUX output source is specified.

Command parameter	Specification of auxiliary output source
DTYP	Pattern data type
SYNC	Synchronized state

5.6 Results and System Commands

5.6.1 FETCh and CFETch Subsystems

The FETCh or CFETch query is used to obtain measurement data from the ED module.

The FETCh query inquires about the result of a measurement, and the CFETch query inquires about the value of a partial measurement. For example, if a measurement is being repeated, the FETC:BME:ECO? query inquires about the error count obtained during the previous measurement period, and the CFET:BME:ECO? query inquires about a partial error count which is obtained during the current measurement period.

1.
```
FETCh[:SENSe[1]]:BMEasurement:BCOunt[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:BCOunt:OMITting?
FETCh[:SENSe[1]]:BMEasurement:BCOunt:INSerting?
FETCh[:SENSe[1]]:BMEasurement:BCOunt:OVERhead?   (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:BCOunt:PAYLoad?    (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:BCOunt:SPECific?
FETCh[:SENSe[1]]:BMEasurement:BCOunt:OTHer?
CFETch[:SENSe[1]]:BMEasurement:BCOunt[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:BCOunt:OMITting?
CFETch[:SENSe[1]]:BMEasurement:BCOunt:INSerting?
CFETch[:SENSe[1]]:BMEasurement:BCOunt:OVERhead?  (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:BCOunt:PAYLoad?   (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:BCOunt:SPECific?
CFETch[:SENSe[1]]:BMEasurement:BCOunt:OTHer?
```

- Function — Sends a query for a bit count value
- Availability of command and query — Query
- Query
  FETCh[:SENSe[1]]:BMEasurement:BCOunt[:TOTal]?
  FETCh[:SENSe[1]]:BMEasurement:BCOunt:OMITting?
  FETCh[:SENSe[1]]:BMEasurement:BCOunt:INSerting?
  FETCh[:SENSe[1]]:BMEasurement:BCOunt:OVERhead?
  FETCh[:SENSe[1]]:BMEasurement:BCOunt:PAYLoad?
  FETCh[:SENSe[1]]:BMEasurement:BCOunt:SPECific?
  FETCh[:SENSe[1]]:BMEasurement:BCOunt:OTHer?
  CFETch[:SENSe[1]]:BMEasurement:BCOunt[:TOTal]?
  CFETch[:SENSe[1]]:BMEasurement:BCOunt:OMITting?
  CFETch[:SENSe[1]]:BMEasurement:BCOunt:INSerting?
  CFETch[:SENSe[1]]:BMEasurement:BCOunt:OVERhead?
  CFETch[:SENSe[1]]:BMEasurement:BCOunt:PAYLoad?
  CFETch[:SENSe[1]]:BMEasurement:BCOunt:SPECific?
  CFETch[:SENSe[1]]:BMEasurement:BCOunt:OTHer?
- Query response format — When the display format is integer format: NR1 (integer type)
  When the display format is exponent format: NR3 (floating-point type)

- Description

Inquires about a bit count that has been measured
Depending on the mnemonic at the end of a header, this command inquires about one of the following bit counts.
Bit counts that can be acquired are dependent on the setting of an error detection mode.

Mnemonic at the header end	Type of bit count value
[TOTal]	Received bit count value
OMITting	Bit count value measured when an omission error occurs
INSerting	Bit count value measured when an insertion error occurs
OVERhead	Bit count value in Overhead areas
PAYLoad	Bit count value in Payload areas
SPECific	Bit count value in Specific areas
OTHer	Bit count value in other than Specific areas

If a data overflow occurs or data is undefined or invalid, one of the values listed in the table below is returned.

Display formats	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Exponent format	9.9998E+98	9.9999E+99	9.9997E+97

2.

```
FETCh[:SENSe[1]]:BMEasurement:ERATe[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:ERATe:OMITting?
FETCh[:SENSe[1]]:BMEasurement:ERATe:INSerting?
FETCh[:SENSe[1]]:BMEasurement:ERATe:OVERhead?     (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:ERATe:PAYLoad?      (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:ERATe:SPECific?
FETCh[:SENSe[1]]:BMEasurement:ERATe:OTHer?
CFETch[:SENSe[1]]:BMEasurement:ERATe[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:ERATe:OMITting?
CFETch[:SENSe[1]]:BMEasurement:ERATe:INSerting?
CFETch[:SENSe[1]]:BMEasurement:ERATe:OVERhead?    (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:ERATe:PAYLoad?     (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:ERATe:SPECific?
CFETch[:SENSe[1]]:BMEasurement:ERATe:OTHer?
```

- Function — Sends a query for an error rate value
- Availability of command and query — Query
- Query

```
FETCh[:SENSe[1]]:BMEasurement:ERATe[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:ERATe:OMITting?
FETCh[:SENSe[1]]:BMEasurement:ERATe:INSerting?
FETCh[:SENSe[1]]:BMEasurement:ERATe:OVERhead?
FETCh[:SENSe[1]]:BMEasurement:ERATe:PAYLoad?
FETCh[:SENSe[1]]:BMEasurement:ERATe:SPECific?
FETCh[:SENSe[1]]:BMEasurement:ERATe:OTHer?
CFETch[:SENSe[1]]:BMEasurement:ERATe[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:ERATe:OMITting?
CFETch[:SENSe[1]]:BMEasurement:ERATe:INSerting?
CFETch[:SENSe[1]]:BMEasurement:ERATe:OVERhead?
CFETch[:SENSe[1]]:BMEasurement:ERATe:PAYLoad?
CFETch[:SENSe[1]]:BMEasurement:ERATe:SPECific?
CFETch[:SENSe[1]]:BMEasurement:ERATe:OTHer?
```

- Query response format — NR3 (floating-point type)

- Description

Inquires about an error rate value that has been measured.
Depending on the mnemonic at the end of a header, this command inquires about one of the following error rate values.
Error rates that can be acquired are dependent on the specification of an error detection mode.

Mnemonic at the header end	Type of error rate value
[TOTal]	Total errors
OMITting	Omitting errors
INSerting	Inserting errors
OVERhead	Errors in Overhead areas
PAYLoad	Errors in Payload areas
SPECific	Errors in Specific areas
OTHer	Errors in other than Specific areas

9.9998E+98 is returned if data overflows, 9.9999E+99 is returned if data is undefined, and 9.9997E+97 is returned if data is invalid.

3.
```
FETCh[:SENSe[1]]:BMEasurement:ECOunt[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:ECOunt:OMITting?
FETCh[:SENSe[1]]:BMEasurement:ECOunt:INSerting?
FETCh[:SENSe[1]]:BMEasurement:ECOunt:OVERhead?   (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:ECOunt:PAYLoad?    (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:ECOunt:SPECific?
FETCh[:SENSe[1]]:BMEasurement:ECOunt:OTHer?
CFETch[:SENSe[1]]:BMEasurement:ECOunt[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:ECOunt:OMITting?
CFETch[:SENSe[1]]:BMEasurement:ECOunt:INSerting?
CFETch[:SENSe[1]]:BMEasurement:ECOunt:OVERhead?  (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:ECOunt:PAYLoad?   (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:ECOunt:SPECific?
CFETch[:SENSe[1]]:BMEasurement:ECOunt:OTHer?
```

- Function — Sends a query for an error count value
- Availability of command and query — Query
- Query
  FETCh[:SENSe[1]]:BMEasurement:ECOunt[:TOTal]?
  FETCh[:SENSe[1]]:BMEasurement:ECOunt:OMITting?
  FETCh[:SENSe[1]]:BMEasurement:ECOunt:INSerting?
  FETCh[:SENSe[1]]:BMEasurement:ECOunt:OVERhead?
  FETCh[:SENSe[1]]:BMEasurement:ECOunt:PAYLoad?
  FETCh[:SENSe[1]]:BMEasurement:ECOunt:SPECific?
  FETCh[:SENSe[1]]:BMEasurement:ECOunt:OTHer?
  CFETch[:SENSe[1]]:BMEasurement:ECOunt[:TOTal]?
  CFETch[:SENSe[1]]:BMEasurement:ECOunt:OMITting?
  CFETch[:SENSe[1]]:BMEasurement:ECOunt:INSerting?
  CFETch[:SENSe[1]]:BMEasurement:ECOunt:OVERhead?
  CFETch[:SENSe[1]]:BMEasurement:ECOunt:PAYLoad?
  CFETch[:SENSe[1]]:BMEasurement:ECOunt:SPECific?
  CFETch[:SENSe[1]]:BMEasurement:ECOunt:OTHer?
- Query response format
  When the display format is integer format:
  NR1 (integer type)
  When the display format is exponent format:
  NR3 (floating-point type)

- Description

Inquires about an error count value that has been measured
Depending on the mnemonic at the end of a header, this command inquires about one of the following error count values.
Error counts that can be acquired are dependent on the setting of an error detection mode.

Mnemonic at the header end	Types of error count value
[TOTal]	Total errors
OMITting	Omitting errors
INSerting	Inserting errors
OVERhead	Errors in Overhead areas
PAYLoad	Errors in Payload areas
SPECific	Errors in Specific areas
OTHer	Errors in other than Specific areas

If a data overflow occurs or data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Exponent format	9.9998E+98	9.9999E+99	9.9997E+97

4.
```
CFETch[:SENSe[1]]:BMEasurement:IERate[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:IERate:OMITting?
CFETch[:SENSe[1]]:BMEasurement:IERate:INSerting?
CFETch[:SENSe[1]]:BMEasurement:IERate:OVERhead?  (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:IERate:PAYLoad?   (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:IERate:SPECific?
CFETch[:SENSe[1]]:BMEasurement:IERate:OTHer?
```

- Function — Sends a query for an immediate error rate value
- Availability of command and query — Query
- Query —
  CFETch[:SENSe[1]]:BMEasurement:IERate[:TOTal]?
  CFETch[:SENSe[1]]:BMEasurement:IERate:OMITting?
  CFETch[:SENSe[1]]:BMEasurement:IERate:INSerting?
  CFETch[:SENSe[1]]:BMEasurement:IERate:OVERhead?
  CFETch[:SENSe[1]]:BMEasurement:IERate:PAYLoad?
  CFETch[:SENSe[1]]:BMEasurement:IERate:SPECific?
  CFETch[:SENSe[1]]:BMEasurement:IERate:OTHer?
- Query response format — NR3 (floating-point type)
- Description — Inquires about an immediate error rate value that has been measured
  Depending on the mnemonic at the end of a header, this command inquires about one of the following immediate error rate values.
  Error rates that can be acquired are dependent on the error detection mode setting.

Mnemonic at the header end	Type of immediate error rate value
[TOTal]	Total errors
OMITting	Omitting errors
INSerting	Inserting errors
OVERhead	Errors in Overhead areas
PAYLoad	Errors in Payload areas
SPECific	Errors in Specific areas
OTHer	Errors in other than Specific areas

9.9998E+98 is returned if data overflows, 9.9999E+99 is returned if data is undefined, and 9.9997E+97 is returned if data is invalid.

5.
```
CFETch[:SENSe[1]]:BMEasurement:IECount[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:IECount:OMITting?
CFETch[:SENSe[1]]:BMEasurement:IECount:INSerting?
CFETch[:SENSe[1]]:BMEasurement:IECount:OVERhead? (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:IECount:PAYLoad?  (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:IECount:SPECific?
CFETch[:SENSe[1]]:BMEasurement:IECount:OTHer?
```

- Function — Sends a query for an immediate error count value
- Availability of command and query — Query
- Query

```
CFETch[:SENSe[1]]:BMEasurement:IECount[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:IECount:OMITting?
CFETch[:SENSe[1]]:BMEasurement:IECount:INSerting?
CFETch[:SENSe[1]]:BMEasurement:IECount:OVERhead?
CFETch[:SENSe[1]]:BMEasurement:IECount:PAYLoad?
CFETch[:SENSe[1]]:BMEasurement:IECount:SPECific?
CFETch[:SENSe[1]]:BMEasurement:IECount:OTHer?
```

- Query response format — When the display format is integer format: NR1 (integer type)
  When the display format is exponent format: NR3 (floating-point type)
- Description — Inquires about an immediate error count value that has been measured
  Depending on the mnemonic at the end of a header, this command inquires about one of the following immediate error count values.
  Error counts that can be acquired are dependent on the error detection mode setting.

Mnemonic at the header end	Type of immediate error count value
[TOTal]	Total errors
OMITting	Omitting errors
INSerting	Inserting errors
OVERhead	Errors in Overhead areas
PAYLoad	Errors in Payload areas
SPECific	Errors in Specific areas
OTHer	Errors in other than Specific areas

If a data overflow occurs or data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Exponent format	9.9998E+98	9.9999E+99	9.9997E+97

6.
```
FETCh[:SENSe[1]]:BMEasurement:EINTervals[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EINTervals:OMITting?
FETCh[:SENSe[1]]:BMEasurement:EINTervals:INSerting?
FETCh[:SENSe[1]]:BMEasurement:EINTervals:OVERhead? (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:EINTervals:PAYLoad?  (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:EINTervals:SPECific?
FETCh[:SENSe[1]]:BMEasurement:EINTervals:OTHer?
CFETch[:SENSe[1]]:BMEasurement:EINTervals[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EINTervals:OMITting?
CFETch[:SENSe[1]]:BMEasurement:EINTervals:INSerting?
CFETch[:SENSe[1]]:BMEasurement:EINTervals:OVERhead?(Pattern option)
CFETch[:SENSe[1]]:BMEasurement:EINTervals:PAYLoad? (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:EINTervals:SPECific?
CFETch[:SENSe[1]]:BMEasurement:EINTervals:OTHer?
```

- Function — Sends a query for an error interval value
- Availability of command and query — Query
- Query
  FETCh[:SENSe[1]]:BMEasurement:EINTervals[:TOTal]?
  FETCh[:SENSe[1]]:BMEasurement:EINTervals:OMITting?
  FETCh[:SENSe[1]]:BMEasurement:EINTervals:INSerting?
  FETCh[:SENSe[1]]:BMEasurement:EINTervals:OVERhead?
  FETCh[:SENSe[1]]:BMEasurement:EINTervals:PAYLoad?
  FETCh[:SENSe[1]]:BMEasurement:EINTervals:SPECific?
  FETCh[:SENSe[1]]:BMEasurement:EINTervals:OTHer?
  CFETch[:SENSe[1]]:BMEasurement:EINTervals[:TOTal]?
  CFETch[:SENSe[1]]:BMEasurement:EINTervals:OMITting?
  CFETch[:SENSe[1]]:BMEasurement:EINTervals:INSerting?
  CFETch[:SENSe[1]]:BMEasurement:EINTervals:OVERhead?
  CFETch[:SENSe[1]]:BMEasurement:EINTervals:PAYLoad?
  CFETch[:SENSe[1]]:BMEasurement:EINTervals:SPECific?
  CFETch[:SENSe[1]]:BMEasurement:EINTervals:OTHer?
- Query response format — When the display format is integer format: NR1 (integer type)
  When the display format is percentage format: NR2 (fixed-point type)

- Description

Inquires about an error interval value that has been measured
Depending on the mnemonic at the end of a header, this command inquires about one of the following error interval values.
Error interval values that can be acquired are dependent on the specification of an error detection mode.

Mnemonic at the header end	Type of error interval values
[TOTal]	Total errors
OMITting	Omitting errors
INSerting	Inserting errors
OVERhead	Errors in Overhead areas
PAYLoad	Errors in Payload areas
SPECific	Errors in Specific areas
OTHer	Errors in other than Specific areas

If a data overflow occurs or if data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Percentage format	999.9998	999.9999	999.9997

7.
```
FETCh[:SENSe[1]]:BMEasurement:EFINtervals[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EFINtervals:OMITting?
FETCh[:SENSe[1]]:BMEasurement:EFINtervals:INSerting?
FETCh[:SENSe[1]]:BMEasurement:EFINtervals:OVERhead?  (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:EFINtervals:PAYLoad?   (Pattern option)
FETCh[:SENSe[1]]:BMEasurement:EFINtervals:SPECific?
FETCh[:SENSe[1]]:BMEasurement:EFINtervals:OTHer?
CFETch[:SENSe[1]]:BMEasurement:EFINtervals[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EFINtervals:OMITting?
CFETch[:SENSe[1]]:BMEasurement:EFINtervals:INSerting?
CFETch[:SENSe[1]]:BMEasurement:EFINtervals:OVERhead? (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:EFINtervals:PAYLoad?  (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:EFINtervals:SPECific?
CFETch[:SENSe[1]]:BMEasurement:EFINtervals:OTHer?
```

- Function — Sends a query for an error free interval value
- Availability of command and query
  Query
- Query
  FETCh[:SENSe[1]]:BMEasurement:EFINtervals[:TOTal]?
  FETCh[:SENSe[1]]:BMEasurement:EFINtervals:OMITting?
  FETCh[:SENSe[1]]:BMEasurement:EFINtervals:INSerting?
  FETCh[:SENSe[1]]:BMEasurement:EFINtervals:OVERhead?
  FETCh[:SENSe[1]]:BMEasurement:EFINtervals:PAYLoad?
  FETCh[:SENSe[1]]:BMEasurement:EFINtervals:SPECific?
  FETCh[:SENSe[1]]:BMEasurement:EFINtervals:OTHer?
  CFETch[:SENSe[1]]:BMEasurement:EFINtervals[:TOTal]?
  CFETch[:SENSe[1]]:BMEasurement:EFINtervals:OMITting?
  CFETch[:SENSe[1]]:BMEasurement:EFINtervals:INSerting?
  CFETch[:SENSe[1]]:BMEasurement:EFINtervals:OVERhead?
  CFETch[:SENSe[1]]:BMEasurement:EFINtervals:PAYLoad?
  CFETch[:SENSe[1]]:BMEasurement:EFINtervals:SPECific?
  CFETch[:SENSe[1]]:BMEasurement:EFINtervals:OTHer?
- Query response format
  When the display format is integer format:
  NR1 (integer type)
  When the display format is percentage format:
  NR2 (fixed-point type)

- Description — Inquires about an error free interval value that has been measured
Depending on the mnemonic at the end of a header, this command inquires about one of the following error free interval values.
Error-free interval values that can be acquired are dependent on the error detection mode setting.

Mnemonic at the header end	Type of error free interval value
[TOTal]	Total errors
OMITting	Omitting errors
INSerting	Inserting errors
OVERhead	Errors in Overhead areas
PAYLoad	Errors in Payload areas
SPECific	Errors in Specific areas
OTHer	Errors in other than Specific areas

If a data overflow occurs or if data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Percentage format	999.9998	999.9999	999.9997

8.
```
FETCh[:SENSe[1]]:BMEasurement:FREQuency?
CFETch[:SENSe[1]]:BMEasurement:FREQuency?
```

- Function — Sends a query for a frequency value
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:BMEasurement:FREQuency?
CFETch[:SENSe[1]]:BMEasurement:FREQuency?
- Query response format — NR1 (integer type)
- Description — Returns the measured frequency value in Hz
Returns 9999999999 if data is undefined.

9.
```
FETCh[:SENSe[1]]:BMEasurement:CLINtervals?
CFETch[:SENSe[1]]:BMEasurement:CLINtervals?
```

- Function — Sends a query for a clock loss interval value
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:BMEasurement:CLINtervals?
CFETch[:SENSe[1]]:BMEasurement:CLINtervals?
- Query response format — NR1 (integer type)
- Description — Inquires about a clock loss interval value that has been measured 9999999998 is returned if data overflows, 9999999999 is returned if data is undefined, and 9999999997 is returned if data is invalid.

10.
```
FETCh[:SENSe[1]]:BMEasurement:SLINtervals?
CFETch[:SENSe[1]]:BMEasurement:SLINtervals?
```

- Function — Sends a query for a sync loss interval value
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:BMEasurement:SLINtervals?
CFETch[:SENSe[1]]:BMEasurement:SLINtervals?
- Query response format — NR1 (integer type)
- Description — Inquires about a sync loss interval value that has been measured 9999999998 is returned if data overflows, 9999999999 is returned if data is undefined, and 9999999997 if data is invalid.

11.
```
FETCh[:SENSe[1]]:BMEasurement:TEINtervals:EM<n>[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:TEINtervals:LEM8[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:TEINtervals:EM<n>[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:TEINtervals:LEM8[:TOTal]?
```

- Function — Sends a query for a threshold EI value
- Availability of command and query — Query
- Query

FETCh[:SENSe[1]]:BMEasurement:TEINtervals:EM<n>[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:TEINtervals:LEM8[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:TEINtervals:EM<n>[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:TEINtervals:LEM8[:TOTal]?
<n>: 3, 4, 5, 6, 7, 8

- Query response format — When the display format is integer format: NR1 (integer type)
When the display format is percentage format: NR2 (fixed-point type)
- Description — Inquires about a threshold EI value that has been measured
The threshold EI value is the number of intervals for which the error rate of one interval satisfies the threshold condition.
Depending on the mnemonic at the end of a header, this command inquires about one of the following threshold conditions:

Mnemonic at the header end	Threshold condition
EM3[:TOTal]	> 1E-3
EM4[:TOTal]	> 1E-4
EM5[:TOTal]	> 1E-5
EM6[:TOTal]	> 1E-6
EM7[:TOTal]	> 1E-7
EM8[:TOTal]	> 1E-8
LEM8[:TOTal]	≤ 1E-8

If a data overflow occurs or if data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Percentage format	999.9998	999.9999	999.9997

12.
```
FETCh[:SENSe[1]]:BMEasurement:TEFintervals:EM<n>[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:TEFintervals:LEM8[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:TEFintervals:EM<n>[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:TEFintervals:LEM8[:TOTal]?
```

- Function — Sends a query for a threshold EFI value
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:BMEasurement:TEFintervals:EM<n>[:TOTal]?
  FETCh[:SENSe[1]]:BMEasurement:TEFintervals:LEM8[:TOTal]?
  CFETch[:SENSe[1]]:BMEasurement:TEFinter-vals:EM<n>[:TOTal]?
  CFETch[:SENSe[1]]:BMEasurement:TEFintervals:LEM8[:TOTal]?
- Parameter — <n>: 3, 4, 5, 6, 7, 8
- Query response format — When the display format is integer format:
  NR1 (integer type)
  When the display format is percentage format:
  NR2 (fixed-point type)
- Description — Inquires about a threshold EFI value that has been measured
  The threshold EFI value is the number of intervals for which the error rate of one interval does not satisfy the threshold condition.
  Depending on the mnemonic at the end of a header, this command inquires about one of the following threshold conditions:

Mnemonic at the header end	Threshold condition
EM3[:TOTal]	> 1E-3
EM4[:TOTal]	> 1E-4
EM5[:TOTal]	> 1E-5
EM6[:TOTal]	> 1E-6
EM7[:TOTal]	> 1E-7
EM8[:TOTal]	> 1E-8
LEM8[:TOTal]	≤ 1E-8

If a data overflow occurs or if data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Percentage format	999.9998	999.9999	999.9997

13.
```
FETCh[:SENSe[1]]:BMEasurement:EPERformance:ESEConds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:EFSeconds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:SESeconds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:USEConds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:DMINutes[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:ESEConds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:EFSeconds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:SESeconds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:USEConds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:DMINutes[:TOTal]?
```

- Function — Sends a query for an error performance value
- Availability of command and query — Query
- Query

FETCh[:SENSe[1]]:BMEasurement:EPERformance:ESEConds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:EFSeconds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:SESeconds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:USEConds[:TOTal]?
FETCh[:SENSe[1]]:BMEasurement:EPERformance:DMINutes[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:ESEConds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:EFSeconds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:SESeconds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:USEConds[:TOTal]?
CFETch[:SENSe[1]]:BMEasurement:EPERformance:DMINutes[:TOTal]?

- Query response format — When the display format is integer format: NR1 (integer type)
  When the display format is percentage format: NR2 (fixed-point type)
- Description — Inquires about an error performance value that has been measured Depending on the mnemonic at the end of a header, this command inquires about one of the following error performance values:

Mnemonic at the header end	Error performance item
ESEConds[:TOTal]	Error Seconds
EFSeconds[:TOTal]	Error Free Seconds
SESeconds[:TOTal]	Severe Error Seconds
USEConds[:TOTal]	Unavailable Seconds
DMINutes[:TOTal]	Degraded Minutes

If a data overflow occurs or if data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Percentage format	999.9998	999.9999	999.9997

14. `FETCh[:SENSe[1]]:BMEasurement:B1ERror:BCOunt?` (Pattern option)
`CFETch[:SENSe[1]]:BMEasurement:B1ERror:BCOunt?` (Pattern option)

- Function: Sends a query for the number of bits included in all frames
- Availability of command and query: Query
- Query: FETCh[:SENSe[1]]:BMEasurement:B1ERror:BCOunt?
  CFETch[:SENSe[1]]:BMEasurement:B1ERror:BCOunt?
- Query response format: If an integer expression is used for display:NR1 (integer type)
  If an exponential expression is used for display: NR3 (floating-point type)
- Description: A query is sent for the number of bits included in all frames.
  The number of bits included in all frames is calculated with the following equation:
  The number of bits included in all frames = The number of frames for which B1-byte measurement is performed multiplied by the number of bits per frame.
  If a data overflow occurs or if data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Exponential expression	9.9998E+98	9.9999E+99	9.9997E+97

15. `FETCh[:SENSe[1]]:BMEasurement:B1ERror:ERATe?` (Pattern option)
`CFETch[:SENSe[1]]:BMEasurement:B1ERror:ERATe?` (Pattern option)

- Function: Sends a query for the error rate value in the B1-byte measurement
- Availability of command and query: Command/Query
- Query: FETCh[:SENSe[1]]:BMEasurement:B1ERror:ERATe?
  CFETch[:SENSe[1]]:BMEasurement:B1ERror:ERATe?
- Query response format: NR3 (floating-point type)
- Description: A query is sent for the error rate value in the B1-byte measurement. 9.9998E+98 is returned if data overflows, 9.9999E+99 is returned if data is undefined, and 9.9997E+97 is returned if data is invalid.

16.
```
FETCh[:SENSe[1]]:BMEasurement:B1ERror:ECOunt?     (Pattern option)
CFETch[:SENSe[1]]:BMEasurement:B1ERror:ECOunt?    (Pattern option)
```

- Function — Sends a query for the error count value in the B1-byte measurement
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:BMEasurement:B1ERror:ECOunt?
CFETch[:SENSe[1]]:BMEasurement:B1ERror:ECOunt?
- Query response format — If an integer expression is used for display:NR1 (integer type)
If an exponential expression is used for display:
NR3 (floating-point type)
- Description — A query is sent for the error count in the B1-byte measurement.
If a data overflow occurs or if data is undefined or invalid, one of the values listed in the table below is returned.

Display format	Overflow	Undefined	Invalid
Integer format	9999999998	9999999999	9999999997
Exponential expression	9.9998E+98	9.9999E+99	9.9997E+97

17.
```
FETCh[:SENSe[1]]:MTIMe:ETIMe?
```

- Function — Sends a query for the elapsed measurement time
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:MTIMe:ETIMe?
- Query response format — <day>, <hour>, <minute>, <second>
<day> = <hour> = <minute> = <second> = NR1 (integer type)
- Description — A query is sent for the elapsed measurement time.
If the elapsed measurement time overflows, 98,98,98,98 is returned.

18.
```
FETCh[:SENSe[1]]:MTIMe:TIMed?
```

- Function — Sends a query for the remaining measurement time
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:MTIMe:TIMed?
- Query response format — <day>, <hour>, <minute>, <second>
<day> = <hour> = <minute> = <second> = NR1 (integer type)
- Description — Inquires about the remaining measurement time
If the remaining measurement time is undefined, 99,99,99,99 is returned.

19. `FETCh[:SENSe[1]]:MTIMe:STIMe?`

- Function — Sends a query for the measurement start time
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:MTIMe:STIMe?
- Query response format — <year>, <month>, <day>, <hour>, <minute>, <second>
  <year> = <month> = <day> = <hour> = <minute> = <second> = NR1 (integer type)
- Description — Inquires about the measurement start time
  If the measurement start time is undefined, 9999,99,99,99,99,99 is returned.

20. `FETCh[:SENSe[1]]:MTIMe:RTIMe?`

- Function — Sends a query for the current date and time
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:MTIMe:RTIMe?
- Query response format — <year> = <month> = <day> = <hour> = <minute> = <second> = NR1 (integer type)
- Description — Inquires about the current date and time

21.
```
FETCh[:SENSe[1]]:MONitor:RTIMe:DERRor?
FETCh[:SENSe[1]]:MONitor:RTIMe:CLOSs?
FETCh[:SENSe[1]]:MONitor:RTIMe:SLOSs?
FETCh[:SENSe[1]]:MONitor:HISTory:DERRor?
FETCh[:SENSe[1]]:MONitor:HISTory:CLOSs?
FETCh[:SENSe[1]]:MONitor:HISTory:SLOSs?
```

- Function — Sends a query for the monitoring status
- Availability of command and query — Query
- Query —
  FETCh[:SENSe[1]]:MONitor:RTIMe:DERRor?
  FETCh[:SENSe[1]]:MONitor:RTIMe:CLOSs?
  FETCh[:SENSe[1]]:MONitor:RTIMe:SLOSs?
  FETCh[:SENSe[1]]:MONitor:HISTory:DERRor?
  FETCh[:SENSe[1]]:MONitor:HISTory:CLOSs?
  FETCh[:SENSe[1]]:MONitor:HISTory:SLOSs?
- Query response format — 0 | 1
- Description — Inquires about the monitoring status
  There are two types of monitoring statuses: ireal-time and history. This command inquires about the current status in the Real-time mode and about whether a specified event has occurred during measurement in the History mode.
  Depending on the mnemonic at the end of a header, this command inquires about one of the following monitoring statuses:

Mnemonic at the header end	Monitoring status
RTIMe:DERRor	A real-time data error occurred.
RTIMe:CLOSs	A real-time clock loss occurred.
RTIMe:SLOSs	A real-time sync loss occurred.
HISTory:DERRor	A history data error occurred.
HISTory:CLOSs	A history clock loss occurred.
HISTory:SLOSs	A history sync loss occurred.

In the Real-time mode, this command returns 1 if an event is occurring. In the History mode, this command returns 1 if an event has occurred.

22. `FETCh[:SENSe[1]]:EPANalysis:MDATa:TSERies:ERRor:COUNt?` (Error phase analysis option)

- Function — Sends a query for the error row number in the time-series recording format
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:EPANalysis:MDATa:TSERies:ERRor:COUNt?
- Query response format — NR1 (integer type)
- Description — Inquires about the error row number in the time-series recording format

23. `FETCh[:SENSe[1]]:EPANalysis:MDATa:TSERies:CYCLe:COUNt?` (Error phase analysis option)

- Function — Sends a query for the pattern cycles number in the time-series recording format
- Availability of command and query — Query
- Query — FETCh[:SENSe[1]]:EPANalysis:MDATa:TSERies:CYCLe:COUNt?
- Query response format — NR1 (integer type)
- Description — Inquires about the pattern cycle number in the time-series recording format

24. `FETCh:JTOLerance:MDATa:JFRequency:COUNt?` (Jitter tolerance option)

- Function — Sends a query for the number of the jitter frequency targeted for jitter tolerance measurement
- Availability of command and query — Query
- Query — FETCh:JTOLerance:MDATa:JFRequency:COUNt?
- Query response format — NR1 (integer type)
- Description — Inquires about the number of the jitter frequenciy targeted for jitter tolerance measurement.

25. `FETCh:JTOLerance:MDATa:JAMPlitude:COUNt?` (Jitter tolerance option)

- Function: Sends a query for the number of the jitter amplitud targeted for jitter tolerance measurement
- Availability of command and query: Query
- Query: FETCh:JTOLerance:MDATa:JAMPlitude:COUNt? <index_freq>
- Parameter: <index_freq> = An index value to specify measured jitter frequency.
  Setting range: 1 to (the number of the jitter frequency targeted for jitter tolerance measurement)
  Setting resolution: 1
- Query response format: NR1 (integer type)
- Description: For a jitter frequency, a query is sent for the number of the jitter amplitud targeted for jitter tolerance measurement.

26. `FETCh:JTOLerance:MDATa?` (Jitter tolerance option)

- Function: Sends a query for jitter tolerance measurement data
- Availability of command and query: Query
- Query: FETCh:JTOLerance:MDATa? <index_freq>,<index_ampl>
- Parameter: <index_freq> = An index value to specify a measured jitter frequency
  Setting range: 1 to (the number of the jitter frequency targeted for jitter tolerance measurement)
  Setting resolution: 1
  <index_ampl> =An index value to specify a measured jitter amplitude in <index_freq>
  Setting range: 1 to (the number of the jitter amplitude targeted in <index_freq>)
  However, "1" is always specified in the search mode.
  Setting resolution: 1

- Query response format <str> = 1-point measurement results
  "FREQ, M, AMPL, RES"
  FREQ Jitter frequency [Hz] 10 to 20000000
  M Modifier (1 space if unnecessary)
  >Upper measurement limit
  <Lower measurement limit or measurement was impossible
  1 space is always specified in the sweep mode.
  AMPL Jitter amplitude [UI_{P-P}] 0.01 to 800.00
  RES Results

Measurement mode	RES (Results)	Meaning
Search mode	OK	Jitter amplitude at the jitter measurement point is larger than the template.
	NG	Jitter amplitude at the jitter measurement point is equal to the template or smaller.
	FAIL	Measurement was impossible.
	-----	No template or out of the range
Sweep mode	O Alphabetical characters (capital letters)	Less than the specified threshold
	X Alphabetical characters (capital letters)	Equal to or larger than the specified threshold
	FAIL	Measurement was impossible

- Description A query is sent for jitter tolerance measurement data for one measurement point.

27. `FETCh:JTOLerance:MTIMe:ETIMe?` (Jitter tolerance option)

- Function Sends a query for the time elapsed for jitter tolerance measurement
- Availability of command and query
  Query
- Query FETCh:JTOLerance:MTIMe:ETIMe?
- Query response format <day>,<hour>,<minute>,<second>
  <day> = <hour> = <minute> = <second> = NR1 (integer type)
- Description A query is sent for the time for jitter tolerance measurement.

28. `FETCh:JTOLerance:MTIMe:STIMe?` (Jitter tolerance option)

- Function: Sends a query for the jitter tolerance measurement start time
- Availability of command and query: Query
- Query: FETCh:JTOLerance:MTIMe:STIMe?
- Query response format: <year>,<month>,<day>,<hour>,<minute>,<second>
  <year>=<month>=<day>=<hour>=<minute>=<second>= NR1 (integer type)
- Description: A query is sent for the jitter tolerance measurement start time. If the measurement start time is undefined, (unspecified?), "9999,99,99,99,99,99" is returned.

5.6.2 DISPlay Subsystem

1. `DISPlay:WINDow:BCOunt:DFORmat`

- Function — Sets the bit count value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:BCOunt:DFORmat <type>
- Parameter — <type> = {INTegral | EXPonential}
- Query response format — INT | EXP
- Description — Sets the display format for the measured bit count value
  Changing the display format changes both the display format of the results screen and the response format of the bit count value returned to a query in the FETCh or CFETch subsystem. The format to display bit counts in the B1 measurement is also changed at the same time.

Command parameter	Display format
INT	Integer format
EXP	Exponent format

2. `DISPlay:WINDow:ECOunt:DFORmat`

- Function — Sets the error count value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:ECOunt:DFORmat <type>
- Parameter — <type> = {INTegral | EXPonential}
- Query response format — INT | EXP
- Description — Sets the display format for the measured error count value
  Changing the display format changes both the display format of the results screen and the response format of the error count value returned to a query in the FETCh or CFETch subsystem. The format to display error counts in the B1 measurement is altered at the same time.

Command parameter	Display format
INT	Integer format
EXP	Exponent format

3. `DISPlay:WINDow:IECount:DFORmat`

- Function — Sets the immediate error count value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:IECount:DFORmat <type>
- Parameter — <type> = {INTegral | EXPonential}
- Query response format — INT | EXP
- Description — Sets the display format for the measured immediate error count value
  Changing the display format changes both the display format of the results screen and the response format of the immediate error count value returned to a query in the CFETch subsystem.

Command parameter	Display format
INT	Integer format
EXP	Exponent format

4. `DISPlay:WINDow:EINTervals:DFORmat`

- Function — Sets the error interval value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:EINTervals:DFORmat <type>
- Parameter — <type> = {INTegral | PERCent}
- Query response format — INT | PERC
- Description — Sets the display format for the measured error interval value
  Changing the display format changes both the display format of the results screen and the response format of the error interval value returned to a query in the CFETch subsystem.

Command parameter	Display format
INT	Integer format
PERC	Percentage format

5. `DISPlay:WINDow:EFINtervals:DFORmat`

- Function — Sets the error free interval value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:EFINtervals:DFORmat <type>
- Parameter — <type> = {INTegral | PERCent}
- Query response format — INT | PERC
- Description — Sets the display format for the measured error free interval value Changing the display format changes both the display format of the results screen and the response format of the error free interval value returned to a query in the FETCh or CFETch subsystem.

Command parameter	Display format
INT	Integer format
PERC	Percentage format

6. `DISPlay:WINDow:TEINtervals:DFORmat`

- Function — Sets the threshold EI value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:TEINtervals:DFORmat <type>
- Parameter — <type> = {INTegral | PERCent}
- Query response format — INT | PERC
- Description — Sets the display format for the measured threshold EI value Changing the display format changes both the display format of the results screen and the response format of the threshold EI value returned to a query in the FETCh or CFETch subsystem.

Command parameter	Display format
INT	Integer format
PERC	Percentage format

7. `DISPlay:WINDow:TEFintervals:DFORmat`

- Function — Sets the threshold EFI value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:TEFintervals:DFORmat <type>
- Parameter — <type> = {INTegral | PERCent}
- Query response format — INT | PERC
- Description — Sets the display format for the measured threshold EFI value Changing the display format changes both the display format of the results screen and the response format of the threshold EFI value returned to a query in the FETCh or CFETch subsystem.

Command parameter	Display format
INT	Integer format
PERC	Percentage format

8. `DISPlay:WINDow:EPERformance:DFORmat`

- Function — Sets the error performance value display format
- Availability of command and query — Command/Query
- Command — DISPlay:WINDow:EPERformance:DFORmat <type>
- Parameter — <type> = {INTegral | PERCent}
- Query response format — INT | PERC
- Description — Sets the display format for the measured error performance value Changing the display format changes both the display format of the results screen and the response format of the error performance value returned to a query in the FETCh or CFETch subsystem.

Command parameter	Display format
INT	Integer format
PERC	Percentage format

5.6.3 HCOPy Subsystem

1. `HCOPy[:ITEM]:ED:BMEasurement`

   - Function — Prints the basic measurement result
   - Availability of command and query — Command
   - Command — HCOPy[:ITEM]:ED:BMEasurement
   - Description — Prints the basic measurement result

2. `HCOPy[:ITEM]:JTOLerance` (Jitter tolerance option)

   - Function — Prints jitter tolerance measurement results
   - Availability of command and query — Command
   - Command — HCOPy[:ITEM]:JTOLerance
   - Description — jitter tolerance measurement results are printed.

5.6.4 MMEMory Subsystem

1. `MMEMory:LOAD:SETup`

- Function — Reads the D3371 setup file
- Availability of command and query — Command
- Command — MMEMory:LOAD:SETup <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Reads the D3371 setup file

2. `MMEMory:LOAD:SSG:SETup`

- Function — Reads the settings file of the SSG module
- Availability of command and query — Command
- Command — MMEMory:LOAD:SSG:SETup <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Reads the settings file of the SSG module

3. `MMEMory:LOAD:PPG:SETup`

- Function — Reads the settings file of the PPG module
- Availability of command and query — Command
- Command — MMEMory:LOAD:PPG:SETup <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Reads the settings file of the PPG module

4. `MMEMory:LOAD:PPG:PROGram`

- Function — Reads the PROG pattern file into the PPG module
- Availability of command and query — Command
- Command — MMEMory:LOAD:PPG:PROGram <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Reads the PROG pattern file into the PPG module

5. `MMEMory:LOAD:PPG:STM` (Pattern option)

- Function: Reads an STM pattern file into the PPG module
- Availability of command and query: Command
- Command: MMEMory:LOAD:PPG:STM <str>
- Parameter: <str> = A file name (specified as an absolute path)
- Description: An STM pattern file is read into the PPG module.

6. `MMEMory:LOAD:PPG:FLEXible` (Pattern option)

- Function: Reads a FLEX pattern file into the PPG module
- Availability of command and query: Command
- Command: MMEMory:LOAD:PPG:FLEXible <str>
- Parameter: <str> = A file name (specified as an absolute path)
- Description: A FLEX pattern file is read into the PPG module.

7. `MMEMory:LOAD:PPG:FLEXible:PROGram` (Pattern option)

- Function: A PROG pattern used in a FLEX pattern is read into the PPG module
- Availability of command and query: Command
- Command: MMEMory:LOAD:PPG:FLEXible:PROGram <prog_no>,<str>
- Parameter: <prog_no> =The PROG pattern number used in a FLEX pattern
  Setting range: 1 to 127
  Setting resolution: 1
  <str> = A file name (specified as an absolute path)
- Description: A PROG pattern used in a FLEX pattern is read into the PPG module.

8. `MMEMory:LOAD:ED:SETup`

- Function — Reads the settings file of the ED module
- Availability of command and query — Command
- Command — MMEMory:LOAD:ED:SETup <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Reads the settings file of the ED module

9. `MMEMory:LOAD:ED:PROGram`

- Function — Reads the PROG pattern file into the ED module
- Availability of command and query — Command
- Command — MMEMory:LOAD:ED:PROGram <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Reads the PROG pattern file into the ED module

10. `MMEMory:LOAD:ED:STM` (Pattern option)

- Function — Reads an STM pattern file into the ED module
- Availability of command and query — Command
- Command — MMEMory:LOAD:ED:STM <str>
- Parameter — <str> = a file name (specified as an absolute path)
- Description — An STM pattern file is read into the ED module.

11. `MMEMory:LOAD:ED:FLEXible` (Pattern option)

- Function — Reads a FLEX pattern file into the ED module
- Availability of command and query — Command
- Command — MMEMory:LOAD:ED:FLEXible <str>
- Parameter — <str> = A file name (specified as an absolute path)
- Description — A FLEX pattern file is read into the ED module.

12. `MMEMory:LOAD:ED:FLEXible:PROGram` (Pattern option)

- Function: Reads a PROG pattern utilized in a FLEX pattern into the ED module
- Availability of command and query: Command
- Command: MMEMory:LOAD:ED:FLEXible:PROGram <prog_no>,<str>
- Parameter: <prog_no> =The order number of a PROG pattern used in a FLEX pattern
  Setting range: 1 to 127
  Setting resolution: 1
  <str> = A file name (specified as an absolute path)
- Description: A PROG pattern used in a FLEX pattern is read into the ED module.

13. `MMEMory:LOAD:JTOLerance` (Jitter tolerance option)

- Function: Reads jitter tolerance measurement result files with a binary format
- Availability of command and query: Command
- Command: MMEMory:LOAD:JTOLerance <str>
- Parameter: <str> = A file name (specified as an absolute path)
- Description: A jitter tolerance measurement result file saved with a binary format is read (JID files are permissible).

14. `MMEMory:STORe:SETup`

- Function: Saves the D3371 settings to a file
- Availability of command and query: Command
- Command: MMEMory:STORe:SETup <str>
- Parameter: <str>: File name (specified as an absolute path)
- Description: Saves the D3371 settings to a file

15. `MMEMory:STORe:SSG:SETup`

- Function — Saves the SSG module settings to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:SSG:SETup <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Saves the SSG module settings to a file

16. `MMEMory:STORe:PPG:SETup`

- Function — Saves the PPG module settings to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:PPG:SETup <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Saves the PPG module settings to a file

17. `MMEMory:STORe:PPG:PROGram`

- Function — Saves the PROG pattern of the PPG module to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:PPG:PROGram <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Saves the PROG pattern of the PPG module to a file

18. `MMEMory:STORe:PPG:STM` (Pattern option)

- Function — Saves an STM pattern in the PPG module to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:PPG:STM <str>
- Parameter — <str> = A file name (specified as an absolute path)
- Description — An STM pattern in the PPG module is saved to a file.

19. `MMEMory:STORe:PPG:FLEXible` (Pattern option)

- Function: Saves a FLEX pattern in the PPG module to a file
- Availability of command and query: Command
- Command: MMEMory:STORe:PPG:FLEXible <str>
- Parameter: <str> = A file name (specified as an absolute path)
- Description: A FLEX pattern in the PPG module is saved to a file.

20. `MMEMory:STORe:PPG:FLEXible:PROGram` (Pattern option)

- Function: Saves a PROG pattern used in a FLEX pattern in the PPG module to a file
- Availability of command and query: Command
- Command: MMEMory:STORe:PPG:FLEXible:PROGram <prog_no>,<str>
- Parameter: <prog_no> =The order number of a PROG pattern used in a FLEX pattern
  Setting range: 1 to 127
  Setting resolution: 1
  <str> = A file name (specified as an absolute path)
- Description: A PROG pattern used in FLEX pattern in the PPG module is saved to a file.

21. `MMEMory:STORe:ED:SETup`

- Function: Saves the ED module setting to a file
- Availability of command and query: Command
- Command: MMEMory:STORe:ED:SETup <str>
- Parameter: <str>: File name (specified as an absolute path)
- Description: Saves the ED module setting to a file

22. `MMEMory:STORe:ED:PROGram`

- Function — Saves the PROG pattern of the ED module to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:ED:PROGram <str>
- Parameter — <str>: File name (specified as an absolute path)
- Description — Saves the PROG pattern of the ED module to a file

23. `MMEMory:STORe:ED:STM` (Pattern option)

- Function — Saves an STM pattern in the ED module to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:ED:STM <str>
- Parameter — <str> = a file name (specified as an absolute path)
- Description — An STM pattern in the ED module is saved to a file.

24. `MMEMory:STORe:ED:FLEXible` (Pattern option)

- Function — Saves an STM pattern in the ED module to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:ED:FLEXible <str>
- Parameter — <str> = a file name (specified as an absolute path)
- Description — An STM pattern in the ED module is saved to a file.

25. `MMEMory:STORe:ED:FLEXible:PROGram` (Pattern option)

- Function — Saves a PROG pattern used in FLEX pattern in the ED module to a file
- Availability of command and query — Command
- Command — MMEMory:STORe:ED:FLEXible:PROGram <prog_no>,<str>
- Parameter — <prog_no> =The order number of a PROG pattern used in a FLEX pattern
  Setting range: 1 to 127
  Setting resolution: 1
  <str> = A file name (specified as an absolute path)
- Description — A PROG pattern used in a FLEX patterns in the ED module is saved to a file.

26. `MMEMory:STORe:ED:BMEasurement:LOG`

- Function: Saves measurement log information to a file
- Availability of command and query: Command
- Command: MMEMory:STORe:ED:BMEasurement:LOG <str>
- Parameter: <str>: File name (specified as an absolute path)
- Description: Saves measurement log information to a file

27. `MMEMory:STORe:ED:EPANalysis:TSERies:TEXT` (Error phase analysis option)

- Function: Saves error-recording data in a file with a time-series format
- Availability of command and query: Command
- Command: MMEMory:STORe:ED:EPANalysis:TSERies:TEXT <cn>,<sp1>,<sp2>,<sp3>,<nr>,<str>
- Parameter:

<cn> the cycle number when the saving pattern starts
Setting range: 1 to the number of the pattern cycles
Setting resolution: 1

<sp1> = <sp2> = <sp3> = the location where the saving starts

Pattern type	Start location		Setting range	Resolution
PROG, ZSUB	<sp1>	Address	0 bits to (pattern length -1) bits	32 bits
	<sp2>	---	Any value	---
	<sp3>	---	Any value	---
STM	<sp1>	Frame number	1 to number of frames	1
	<sp2>	Row	1 to 9	1
	<sp3>	Column	1 byte to 1080 bytes (STM4) 1 byte to 4320 bytes (STM16)	4 bytes
FLEX	<sp1>	Index	1 to number of patterns	1
	<sp2>	Address	0 bit to (pattern length -1) bit	32 bits
	<sp3>	---	Any value	---

<nr> = The number of error rows to be saved
Setting range: 1 to the number of the error rows
Setting resolution: 1

<str> = a file name (specified as an absolute path)

- Description: Error-recording data is saved in a text file with a time-series format. In a screen displaying time series data, a row indicates error data. Error data whose row numbers are specified by <cn>,<sp1>,<sp2>, and <sp3> to <nr> is saved in the file.

28. `MMEMory:STORe:ED:EPANalysis:STATistics:TEXT` (Error phase analysis option)

- Function: Saves error-recording data in a file with a statistics format
- Availability of command and query: Command
- Command: MMEMory:STORe:ED:EPANalysis:STATistics:TEXT <sp1>, <sp2>,<sp3>,<range>,<str>
  Parameter<spl> = <sp2> = <sp3> = Save-starting location

Pattern type	Start location		Setting range	Resolution
PROG, ZSUB	<sp1>	Address	0 bits to (pattern length -1) bits	1 bit
	<sp2>	---	Any value	---
	<sp3>	---	Any value	---
STM	<sp1>	Frame number	1 to number of frames	1
	<sp2>	Row	1 to 9	1
	<sp3>	Column	1 byte to 1080 bytes (STM4) 1 byte to 4320 bytes (STM16)	1 byte
FLEX	<sp1>	Index	1 to number of patterns	1
	<sp2>	Address	0 bits to (pattern length -1) bits	1 bit
	<sp3>	---	Any value	---

  <range> = The range used for error saving (the number of bits)
  <str> = File name (specified as an absolute path)
- Description: Error-recording data is saved in a text file with a time-series format. Error bits specified from <sp1>,<sp2>,and <sp3> to <range> are saved in the file.

29. `MMEMory:STORe:ED:EPANalysis:CANCel` (Error phase analysis option)

- Function: Cancels the function which saves the error-recording data in a file
- Availability of command and query: Command
- Command: MMEMory:STORe:ED:EPANalysis:CANCel
- Description: Operations to save error-recording data in a file are cancelled.

30. `MMEMory:STORe:ED:EPANalysis?` (Error phase analysis option)

- Function: Sends a query for how the operations to save error-recording data in a file are being handled
- Availability of command and query: Query
- Command: MMEMory:STORe:ED:EPANalysis?
- Query response format: 0 | 1
- Description: A query is sent for how the operations to save error-recording data in a file are being handled.
  If error-recording data is being saved in a file, "1" will be returned.
  "0" will be returned if saving operations are not being performed.

31. `MMEMory:STORe:ED:EPANalysis:FAIL?` (Error phase analysis option)

- Function: Sends a query for the present status of the error-recording data saved in a file
- Availability of command and query: Query
- Command: MMEMory:STORe:ED:EPANalysis:FAIL?
- Query response format: 0 | 1
- Description: A query is sent for the present status of the error-recording data saved in a file.
  If saving operations have failed, there is no data, or saving operations have not been completed yet, "1" will be returned, "0" will be returned if a saving operation has been successful.
  Execute this command after confirming that the saving of error data in a file has been completed.

32. `MMEMory:STORe:JTOLerance` (Jitter tolerance option)

- Function: Saves jitter tolerance measurement results in a file in a binary format
- Availability of command and query: Command
- Command: MMEMory:STORe:JTOLerance <str>
- Parameter: <str> = file name (specified as an absolute path)
- Description: jitter tolerance measurement results are saved in a file in a binary format (JID files are permissible).

33. `MMEMory:STORe:JTOLerance:TEXT` (Jitter tolerance option)

- Function: Saves jitter tolerance measurement results in a file in a text format.
- Availability of command and query: Command
- Command: MMEMory:STORe:JTOLerance:TEXT <str>
- Parameter: <str> = file name (specified as an absolute path)
- Description: jitter tolerance measurement results are saved in a file in a text format (only TXT files are permissible).

5.6.5 SYSTem Subsystem

1. `SYSTem:BEEPer:DERRor[:STATe]`

- Function: Sets the buzzer that sounds when a data error occurs to on or off
- Availability of command and query: Command/Query
- Command: SYSTem:BEEPer:DERRor[:STATe] <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Enables (ON) or disables (OFF) the buzzer that sounds when a data error occurs

2. `SYSTem:BEEPer:ALARm[:STATe]`

- Function — Sets the buzzer that sounds when a data error occurs to on or off
- Availability of command and query — Command/Query
- Command — SYSTem:BEEPer:ALARm[:STATe] <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — Enables (ON) or disables (OFF) the buzzer that sounds when a data error occurs.

3. `SYSTem:ERRor?`

- Function — Sends a query for errors
- Availability of command and query — Query
- Query — SYSTem:ERRor?
- Query response format — <errno>, <errmsg>
  <errno>: NR1 (integer type)
  <errmsg>: Error message (string)
- Description — The analyzer can queue up to 10 errors in an error queue. If more than 10 errors occur, the 10th error is replaced as follows.

  -350, "Queue overflow"

  The 10th and subsequent errors are not retained.
  SYSTem:ERRor? Removes errors from the error queue.
  Errors are queued in the error queue on a first in, first out (FIFO) basis. For this reason, errors are removed from the error queue in the order in which they were queued.
  When an error is removed from the error queue, it disappears from the error queue, enabling the next error to be queued.
  If no error occurs, the following message is returned:

  0, "No error"

  The error queue has been cleared by *CLS.

4. `SYSTem:DATE`

- Function — Sets the date
- Availability of command and query — Command/Query
- Command — SYSTem:DATE <year>, <month>, <day>
- Parameter — <year>: Numeric data from 1980 to 2099
  <month>: Numeric data from 1 to 12
  <day>: Numeric data from 1 to 31
- Query response format — <year>, <month>, <day>
  <year> = <month> = <day> = NR1 (integer type)
- Description — Sets the date of the built-in clock in the analyzer.
  Set <year> with four digits (e.g., 2000, 2001).

5. `SYSTem:TIME`

- Function — Sets the time
- Availability of command and query — Command/Query
- Command — SYSTem:TIME <hour>, <minute>, <second>
- Parameter — <hour>: Numeric data from 0 to 23
  <minute>: Numeric data from 0 to 59
  <second>: Numeric data from 0 to 59
- Query response format — <hour>, <minute>, <second>
  <hour> = <minute> = <second> = NR1 (integer type)
- Description — Sets the time of the built-in clock in the analyzer.
  The military time system is used for the time.

5.6.6 STATus Subsystem

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

- Function — Reads the operation status event register
- Availability of command and query — Query
- Query — STATus:OPERation[:EVENt]?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the event register of the operation status register
  When the register is read, it is cleared and bit 7 of the corresponding status byte register is also cleared.
  For details, refer to Chapter 4, "STATUS BYTES."

2. `STATus:OPERation:CONDition?`

- Function — Reads the operation status condition register
- Availability of command and query — Query
- Query — STATus:OPERation:CONDition?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the operation status condition register. This register is not cleared when it is read.
  For details, refer to Chapter 4, "STATUS BYTES."

3. `STATus:OPERation:ENABle`

- Function — Sets the operation status enable register
- Availability of command and query — Command/Query
- Command — STATus:OPERation:ENABle <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the contents of the operation status enable register
  The event register corresponding to the bit set to 1 in this register is reflected in bit 7 of the status byte register as the effective bit.
  For details, refer to Chapter 4, "STATUS BYTES."
- Example — If the SET bit (bit 1) and the END bit (bit 9) are set to enable, STAT:OPER:ENAB 514 is set because 514 is the result of the expression "$2^1 + 2^9 = 2 + 512 = 514$."

4.
```
STATus:OPERation:PTRansition
STATus:OPERation:NTRansition
```

- Function — Sets the operation status transition filter
- Availability of command and query — Command/Query
- Command — STATus:OPERation:PTRansition <int>
  STATus:OPERation:NTRansition <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the contents of the transition filter of the operation status register
  When the condition register changes, specifies whether or not to set the contents of the event register. Specifies the transition direction for filtering according to the mnemonic coded at the header end.

Mnemonic at the header end	Transition directions
PTRansition	Positive direction
NTRansition	Negative direction

  For details, refer to Chapter 4, "STATUS BYTES."

5.
```
STATus:OPERation:SETTling[:EVENt]?
```

- Function — Reads the settling status event register
- Availability of command and query — Query
- Query — STATus:OPERation:SETTling[:EVENt]?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the settling status event register
  After this register is read, it is cleared.

  For details, refer to Chapter 4, "STATUS BYTES."

6.
```
STATus:OPERation:SETTling:CONDition?
```

- Function — Reads the settling status condition register
- Availability of command and query — Query
- Query — STATus:OPERation:SETTling:CONDition?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the settling status condition register
  This register is not cleared when it is read.

For details, refer to Chapter 4, "STATUS BYTES."

7. `STATus:OPERation:SETTling:ENABle`

- Function: Sets the settling status enable register
- Availability of command and query: Command/Query
- Command: STATus:OPERation:SETTling:ENABle <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: Sets the contents of the settling status enable register
  The event register corresponding to the bit set to 1 in this register is reflected in bit 1 of the operation status condition register as the effective bit.

  For details, refer to Chapter 4, "STATUS BYTES."
- Example: When the PDLY bit (bit 1) and EDLY bit (bit 2)are set to enable, STAT:OPER:SETT:ENAB 6 is set because 6 is the result of the expression "$2^1 + 2^2 = 2 + 4 = 6$."

8. `STATus:OPERation:SETTling:PTRansition`
   `STATus:OPERation:SETTling:NTRansition`

- Function: Sets the settling status transition filter
- Availability of command and query: Command/Query
- Command: STATus:OPERation:SETTling:PTRansition <int>
  STATus:OPERation:SETTling:NTRansition <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: Sets the contents of the transition filter of the settling status register When the condition register changes, specifies whether or not to set the contents of the event register. Specifies the transition for filtering according to the mnemonic coded at the header end.

Mnemonic at the header end	Transition direction
PTRansition	Positive direction
NTRansition	Negative direction

For details, refer to Chapter 4, "STATUS BYTES."

9. `STATus:QUEStionable[:EVENt]?`

- Function — Reads the questionable status event register
- Availability of command and query — Query
- Query — STATus:QUEStionable[:EVENt]?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the questionable status event register
  When the register is read, it is cleared and bit 3 of the corresponding status byte register is also cleared.

  For details, refer to Chapter 4, "STATUS BYTES."

10. `STATus:QUEStionable:CONDition?`

- Function — Reads the questionable status condition register
- Availability of command and query — Query
- Query — STATus:QUEStionable:CONDition?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the questionable status condition register
  This register is not cleared when it is read.

  For details, refer to Chapter 4, "STATUS BYTES."

11. `STATus:QUEStionable:ENABle`

- Function — Sets the questionable status enable register
- Availability of command and query — Command/Query
- Command — STATus:QUEStionable:ENABle <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the contents of the questionable status enable register
  The event register corresponding to the bit set to 1 in this register is reflected in bit 3 of the status byte register as the effective bit.

  For details, refer to Chapter 4, "STATUS BYTES."
- Example — When the CLOSS bit (bit 9) and SLOSS bit (bit 10) are set to enable, STAT:QUES:ENAB 1536 is set because 1536 is the result of the expression "$2^9 + 2^{10} = 512 + 1024 = 1536$."

12.
```
STATus:QUEStionable:PTRansition
STATus:QUEStionable:NTRansition
```

- Function — Sets the questionable status transition filter
- Availability of command and query — Command/Query
- Command — STATus:QUEStionable:PTRansition <int>
  STATus:QUEStionable:NTRansition <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the contents of the transition filter of the questionable status register
  When the condition register changes, specifies whether or not to set the contents of the event register. Specifies the transition for filtering according to the mnemonic coded at the header end.

Mnemonic at the header end	Transition direction
PTRansition	Positive direction
NTRansition	Negative direction

  For details, refer to Chapter 4, "STATUS BYTES."

13.
```
STATus:QUEStionable:CLOSs[:EVENt]?
```

- Function — Reads the clock loss status event register
- Availability of command and query — Query
- Query — STATus:QUEStionable:CLOSs[:EVENt]?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the clock loss status event register
  After this register is read, it is cleared.
  For details, refer to Chapter 4, "STATUS BYTES."

14.
```
STATus:QUEStionable:CLOSs:CONDition?
```

- Function — Reads the clock loss status condition register
- Availability of command and query — Query
- Query — STATus:QUEStionable:CLOSs:CONDition?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the clock loss status condition register
  This register is not cleared when it is read.
  For details, refer to Chapter 4, "STATUS BYTES."

15. `STATus:QUEStionable:CLOSs:ENABle`

- Function — Sets the clock loss status enable register
- Availability of command and query — Command/Query
- Command — STATus:QUEStionable:CLOSs:ENABle <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the contents of the clock loss status enable register
  The event register corresponding to the bit set to 1 in this register is reflected in bit 9 of the questionable status condition register as the effective bit.

  For details, refer to Chapter 4, "STATUS BYTES."
- Example — When the PPG bit (bit 1) and ED bit (bit 2) are set to enable, STAT:QUES:CLOS:ENAB 6 is set because 6 is the result of the expression "$2^1 + 2^2 = 2 + 4 = 6$."

16. `STATus:QUEStionable:CLOSs:PTRansition`
`STATus:QUEStionable:CLOSs:NTRansition`

- Function — Sets the clock loss status transition filter
- Availability of command and query — Command/Query
- Command — STATus:QUEStionable:CLOSs:PTRansition <int>
  STATus:QUEStionable:CLOSs:NTRansition <int>
- Parameter — <int>
- Query response format — NR1 (integer type)
- Description — Sets the contents of the transition filter of the clock loss status register
  When the condition register changes, specifies whether or not to set the contents of the event register.
  Specifies the transition for filtering according to the mnemonic coded at the header end.

Mnemonic at the header end	Transition directions
PTRansition	Positive direction
NTRansition	Negative direction

For details, refer to Chapter 4, "STATUS BYTES."

17. `STATus:DEVice[:EVENt]?`

- Function — Reads the device status event register
- Availability of command and query — Query
- Query — STATus:DEVice[:EVENt]?
- Query response format — NR1 (integer type)
- Description — Reads the contents of the device status event register
  When the register is read, it is cleared and bit 1 of the corresponding status byte register is also cleared.
  For details, refer to Chapter 4, "STATUS BYTES."

18. `STATus:PRESet`

- Function — Initializes the status register
- Availability of command and query — Command
- Command — STATus:PRESet
- Description — Initializes the status register
  This command initializes the enable register and transition filter of the operation status register, questionable status register, clock loss status register, and settling status register only.
  For details, refer to Chapter 4, "STATUS BYTES."

5.7 SSG Module

5.7.1 SOURce10 Subsystem

1. `SOURce10:FREQuency[:CW]`

- Function: Sets the frequency
- Availability of command and query: Command/Query
- Command: SOURce10:FREQuency[:CW] <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: Specifies the frequency of the clock signal generated from the CLOCK OUTPUT on the SSG module rear panel
  Clock frequency may not be set lower than 800MHz when the jitter modulation function is on, depending on the jitter frequency or jitter amplitude setting.
  Setting range: 10,000 kHz to 3,600,000 kHz (when jitter modulation function is off)
  10,000 kHz to 3,200,000 kHz (when jitter modulation function is on)
  Setting resolution: 1 kHz
  Setting unit: kHz

2. `SOURce10:OUTPut[:STATe]`

- Function: Sets the clock output to on or off
- Availability of command and query: Command/Query
- Command: SOURce10:OUTPut[:STATe] <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Specifies whether the clock signal is output (ON) or not (OFF) to the CLOCK OUTPUT on the SSG module rear panel

3. `SOURce10:REFerence`

- Function — Sets the input source of the clock reference signal
- Availability of command and query — Command/Query
- Command — SOURce10:REFerence <type>
- Parameter — <type> = {INTernal | EXTernal}
- Query response format — INT | EXT
- Description — Sets the input source of the clock reference signal used to generate the SSG module signal

Command parameter	Reference cycle signal
INT	Internal signal
EXT	External signal

4. `SOURce10:JMODulation[:STATe]` (Jitter tolerance option)

- Function — Sets the jitter modulation function to on or off
- Availability of command and query — Command/Query
- Command — SOURce10:JMODulation[:STATe] <bool>
- Parameter — <bool>
- Query response format — 0 | 1
- Description — The jitter modulation function is enabled (ON) or disabled (OFF).

5. `SOURce10:JMODulation:FREQuency` (Jitter tolerance option)

- Function: Sets the jitter frequency
- Availability of command and query: Command/Query
- Command: SOURce10:JMODulation:FREQuency <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: Jitter frequency is specified.

Clock frequency (f)	Jitter frequency-setting range
10 MHz ≤ f < 175 MHz	10 Hz to 2,000,000 Hz
175 MHz ≤ f < 800 MHz	10 Hz to 5,000,000 Hz
800 MHz ≤ f ≤ 3.2 GHz	10 Hz to 20,000,000 Hz

Setting resolution: 10 Hz
Setting unit: Hz

6. `SOURce10:JMODulation:AMPLitude` (Jitter tolerance option)

- Function: Sets the jitter amplitude
- Availability of command and query: Command/Query
- Command: SOURce10:JMODulation:AMPLitude <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point-type)

- Description

Jitter amplitude is specified.

[When the clock frequency is 10 MHz or higher and is lower than 175 MHz]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0 UI_{P-P} to 200 UI_{P-P}
20 Hz	0 UI_{P-P} to 120 UI_{P-P}
20 Hz < fm < 200 Hz	0 UI_{P-P} to (2,400/fm)UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0 UI_{P-P} to 12 UI_{P-P}
5,000 Hz ≤ fm < 200,000 Hz	0 UI_{P-P} to 5 UI_{P-P}
200,000 Hz < fm < 2,000,000 Hz	0 UI_{P-P} to (1,000,000/fm)UI_{P-P}
2,000,000 Hz	0 UI_{P-P} to 0.5 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0 UI_{P-P} to 1 UI_{P-P}	0.01 UI_{P-P}
1 UI_{P-P} to 10 UI_{P-P}	0.1 UI_{P-P}
10 UI_{P-P} to 100 UI_{P-P}	1 UI_{P-P}
100 UI_{P-P} to 200 UI_{P-P}	2 UI_{P-P}

[When the clock frequency is 175 MHz or higher and is lower than 800 MHz]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0 UI_{P-P} to 800 UI_{P-P}
20 Hz	0 UI_{P-P} to 500 UI_{P-P}
20 Hz < fm < 200 Hz	0 UI_{P-P} to (10,000/fm)UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0 UI_{P-P} to 50 UI_{P-P}
5,000 Hz ≤ fm < 125,000 Hz	0 UI_{P-P} to 20 UI_{P-P}
125,000 Hz ≤ fm < 5,000,000 Hz	0 UI_{P-P} to (2,500,000/fm)UI_{P-P}
5,000,000 Hz	0 UI_{P-P} to 0.5 UI_{P-P}

(continued)

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0 UI_{P-P} to 5 UI_{P-P}	0.01 UI_{P-P}
5 UI_{P-P} to 50 UI_{P-P}	0.1 UI_{P-P}
50 UI_{P-P} to 500 UI_{P-P}	1 UI_{P-P}
500 UI_{P-P} to 800 UI_{P-P}	2 UI_{P-P}

[When the clock frequency is 800 MHz or higher and is 3.2 GHz or lower]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0 UI_{P-P} to 800 UI_{P-P}
20 Hz	0 UI_{P-P} to 500 UI_{P-P}
20 Hz < fm < 200 Hz	0 UI_{P-P} to (10,000/fm)UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0 UI_{P-P} to 50 UI_{P-P}
5,000 Hz ≤ fm < 300,000 Hz	0 UI_{P-P} to 20 UI_{P-P}
300,000 Hz ≤ fm < 20,000,000 Hz	0 UI_{P-P} to (6,000,000/fm)UI_{P-P}
20,000,000 Hz	0 UI_{P-P} to 0.3 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0 UI_{P-P} to 5 UI_{P-P}	0.01 UI_{P-P}
5 UI_{P-P} to 50 UI_{P-P}	0.1 UI_{P-P}
50 UI_{P-P} to 500 UI_{P-P}	1 UI_{P-P}
500 UI_{P-P} to 800 UI_{P-P}	2 UI_{P-P}

Setting unit: UI_{P-P}

5.8 Jitter Tolerance Measurement

5.8.1 JTOLerance subsystem

The commands that control the jitter tolerance measurement setting include ones for a jitter tolerance parameter table, clock frequency, measurement mode, and measurement conditions. (These commands can be used only when the jitter tolerance option is installed.)

The command that edits the template does not exist. Therefore, edit the template before executing the remote control.

1. `JTOLerance[:STATe]` (Jitter tolerance option)

   - Function: Starts or stops the jitter tolerance measurement
   - Availability of command and query: Command/Query
   - Command: JTOLerance[:STATe] <bool>
   - Parameter: <bool>
   - Query response format: 0 | 1
   - Description: The start (ON)/stop (OFF) of jitter tolerance measurement is specified. The screen is set for measuring jitter tolerance measurement.

2. `JTOLerance:TABLe` (Jitter tolerance option)

   - Function: Lists in the parameter table for jitter tolerance are edited
   - Availability of command and query: Command/Query
   - Command: JTOLerance:TABLe <index>,<freq>,<min_ampl>,<max_ampl>, <points>
   - Query: JTOLerance:TABLe? <index>

- Parameter

<index> = The index to specify a list in the parameter table for jitter tolerance

Setting range: 1 to the number of the lists
Setting resolution: 1

<freq> = Jitter frequency

Clock frequency (f)	Jitter frequency-setting range
10 MHz ≤ f < 175 MHz	10 Hz to 2,000,000 Hz
175 MHz ≤ f < 800 MHz	10 Hz to 5,000,000 Hz
800 MHz ≤ f ≤ 3.2 GHz	10 Hz to 20,000,000 Hz

Setting resolution: 10 Hz
Setting unit: Hz

<min_ampl> = The minimum jitter amplitude
<max_ampl> = The maximum jitter amplitude

[In the condition that the clock frequency is 10 MHz or higher and is lower than 175 MHz]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0.01 UI_{P-P} to 200 UI_{P-P}
20 Hz	0.01 UI_{P-P} to 120 UI_{P-P}
20 Hz < fm < 200 Hz	0.01 UI_{P-P} to (2,400/fm) UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0.01 UI_{P-P} to 12 UI_{P-P}
5,000 Hz ≤ fm < 200,000 Hz	0.01 UI_{P-P} to 5 UI_{P-P}
200,000 Hz ≤ fm < 2,000,000 Hz	0.01 UI_{P-P} to (1,000,000/fm) UI_{P-P}
2,000,000 Hz	0.01 UI_{P-P} to 0.5 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0.01 UI_{P-P} to 1 UI_{P-P}	0.01 UI_{P-P}
1 UI_{P-P} to 10 UI_{P-P}	0.1 UI_{P-P}
10 UI_{P-P} to 100 UI_{P-P}	1 UI_{P-P}
100 UI_{P-P} to 200 UI_{P-P}	2 UI_{P-P}

[In the condition that the clock frequency is 175 MHz or higher and is lower than 800 MHz]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0.01 UI_{P-P} to 800 UI_{P-P}
20 Hz	0.01 UI_{P-P} to 500 UI_{P-P}
20 Hz < fm < 200 Hz	0.01 UI_{P-P} to (10,000/fm) UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0.01 UI_{P-P} to 50 UI_{P-P}
5,000 Hz ≤ fm < 125,000 Hz	0.01 UI_{P-P} to 20 UI_{P-P}
125,000 Hz ≤ fm < 5,000,000 Hz	0.01 UI_{P-P} to (2,500,000/fm) UI_{P-P}
5,000,000 Hz	0.01 UI_{P-P} to 0.5 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0.01 UI_{P-P} to 5 UI_{P-P}	0.01 UI_{P-P}
5 UI_{P-P} to 50 UI_{P-P}	0.1 UI_{P-P}
50 UI_{P-P} to 500 UI_{P-P}	1 UI_{P-P}
500 UI_{P-P} to 800 UI_{P-P}	2 UI_{P-P}

[In the condition that the clock frequency is 800 MHz or higher and 3.2 GHz or lower]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0.01 UI_{P-P} to 800 UI_{P-P}
20 Hz	0.01 UI_{P-P} to 500 UI_{P-P}
20 Hz < fm < 200 Hz	0.01 UI_{P-P} to (10,000/fm) UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0.01 UI_{P-P} to 50 UI_{P-P}
5,000 Hz ≤ fm < 300,000 Hz	0.01 UI_{P-P} to 20 UI_{P-P}
300,000 Hz ≤ fm < 20,000,000 Hz	0.01 UI_{P-P} to (6,000,000/fm) UI_{P-P}
20,000,000 Hz	0.01 UI_{P-P} to 0.3 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0.01 UI_{P-P} to 5 UI_{P-P}	0.01 UI_{P-P}
5 UI_{P-P} to 50 UI_{P-P}	0.1 UI_{P-P}
50 UI_{P-P} to 500 UI_{P-P}	1 UI_{P-P}
500 UI_{P-P} to 800 UI_{P-P}	2 UI_{P-P}

Setting unit: UI_{P-P}

<points> = The number of jitter amplitude measurement points
Setting range: 1 to 40
Setting resolution: 1

- Query response format: <freq>,<min_ampl>,<max_ampl>,<points>
  <freq> = NR1 (integer type)
  <min_ampl> = NR2 (fixed-point-type)
  <max_ampl> = NR2 (fixed-point-type)
  <points> = NR1 (integer type)
- Description: A list, which is registered in the parameter table for jitter tolerance, is edited. Make the setting so that the condition, <min_ampl> ≤ <max_ampl>, is satisfied.

3. `JTOLerance:TABLe:COUNt?` (Jitter tolerance option)

- Function: Sends a query for the number of parameter table lists for jitter tolerance measurement
- Availability of command and query
  Query
- Query: JTOLerance:TABLe:COUNt?
- Query response format: NR1 (integer type)
- Description: A query is sent for the number of lists in the parameter table for jitter tolerance measurement.

4. `JTOLerance:TABLe:NEW` (Jitter tolerance option)

- Function: Adds a list to the parameter table for jitter tolerance
- Availability of command and query
  Command
- Command: JTOLerance:TABLe:NEW <index>,<freq>,<min_ampl>, <max_ampl>,<points>
- Parameter: <index> = The index to specify a list in the parameter table for jitter tolerance.
  Setting range: 1 to (the number of the lists + 1)
  Setting resolution: 1
  <freq> = jitter frequency

Clock frequency (f)	Jitter frequency-setting range
10 MHz ≤ f < 175 MHz	10 Hz to 2,000,000 Hz
175 MHz ≤ f < 800 MHz	10 Hz to 5,000,000 Hz
800 MHz ≤ f ≤ 3.2 GHz	10 Hz to 20,000,000 Hz

Setting resolution: 10 Hz
Setting unit: Hz

<min_ampl> = Minimum jitter amplitude
<max_ampl> = Maximum jitter amplitude

[In the condition that the clock frequency is 10 MHz or higher and is lower than 175 MHz]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0.01 UI_{P-P} to 200 UI_{P-P}
20 Hz	0.01 UI_{P-P} to 120 UI_{P-P}
20 Hz < fm < 200 Hz	0.01 UI_{P-P} to (2,400/fm) UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0.01 UI_{P-P} to 12 UI_{P-P}
5,000 Hz ≤ fm < 200,000 Hz	0.01 UI_{P-P} to 5 UI_{P-P}
200,000 Hz ≤ fm < 2,000,000 Hz	0.01 UI_{P-P} to (1,000,000/fm) UI_{P-P}
2,000,000 Hz	0.01 UI_{P-P} to 0.5 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0.01 UI_{P-P} to 1 UI_{P-P}	0.01 UI_{P-P}
1 UI_{P-P} to 10 UI_{P-P}	0.1 UI_{P-P}
10 UI_{P-P} to 100 UI_{P-P}	1 UI_{P-P}
100 UI_{P-P} to 200 UI_{P-P}	2 UI_{P-P}

[In the condition that the clock frequency is 175 MHz or higher and is lower than 800 MHz]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0.01 UI_{P-P} to 800 UI_{P-P}
20 Hz	0.01 UI_{P-P} to 500 UI_{P-P}
20 Hz < fm < 200 Hz	0.01 UI_{P-P} to (10,000/fm) UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0.01 UI_{P-P} to 50 UI_{P-P}
5,000 Hz ≤ fm < 125,000 Hz	0.01 UI_{P-P} to 20 UI_{P-P}
125,000 Hz ≤ fm < 5,000,000 Hz	0.01 UI_{P-P} to (2,500,000/fm) UI_{P-P}
5,000,000 Hz	0.01 UI_{P-P} to 0.5 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0.01 UI_{P-P} to 5 UI_{P-P}	0.01 UI_{P-P}
5 UI_{P-P} to 50 UI_{P-P}	0.1 UI_{P-P}
50 UI_{P-P} to 500 UI_{P-P}	1 UI_{P-P}
500 UI_{P-P} to 800 UI_{P-P}	2 UI_{P-P}

[In the condition that the clock frequency is 800 MHz or higher and is lower than 3.2 GHz]

Jitter frequency (fm)	Jitter amplitude-setting range
10 Hz	0.01 UI_{P-P} to 800 UI_{P-P}
20 Hz	0.01 UI_{P-P} to 500 UI_{P-P}
20 Hz < fm < 200 Hz	0.01 UI_{P-P} to (10,000/fm) UI_{P-P}
200 Hz ≤ fm < 5,000 Hz	0.01 UI_{P-P} to 50 UI_{P-P}
5,000 Hz ≤ fm < 300,000 Hz	0.01 UI_{P-P} to 20 UI_{P-P}
300,000 Hz ≤ fm < 20,000,000 Hz	0.01 UI_{P-P} to (6,000,000/fm) UI_{P-P}
20,000,000 Hz	0.01 UI_{P-P} to 0.3 UI_{P-P}

Jitter amplitude-setting range	Jitter amplitude-setting resolution
0.01 UI_{P-P} to 5 UI_{P-P}	0.01 UI_{P-P}
5 UI_{P-P} to 50 UI_{P-P}	0.1 UI_{P-P}
50 UI_{P-P} to 500 UI_{P-P}	1 UI_{P-P}
500 UI_{P-P} to 800 UI_{P-P}	2 UI_{P-P}

Setting unit: UI_{P-P}

<points> = Point of measured jitter amplitude

Setting range: 1 to 40

Setting resolution: 1

- Description: A list is added to the parameter table for jitter tolerance at the location specified with <index>. Make the setting so that the condition, <min_ampl> ≤ <max_ampl> is satisfied.
  The maximum number of lists in the parameter table for jitter tolerance is 20. If there are already 20, list can not be added.

5. `JTOLerance:TABLe:REMove` (Jitter tolerance option)

- Function: Deletes a list from the parameter table for jitter tolerance
- Availability of command and query: Command
- Command: JTOLerance:TABLe:REMove <index>
- Parameter: <index> = The index to specify a list in the parameter table for jitter tolerance

  Setting range: 1 to the number of lists

  Setting resolution: 1
- Description: A list is deleted from the parameter table for jitter tolerance. All of the lists cannot be deleted.

6. `JTOLerance:TEMPlate:SELect` (Jitter tolerance option)

- Function: Sets a template
- Availability of command and query: Command/Query
- Command: JTOLerance:TEMPlate:SELect <str>
- Parameter: <str> = A template name registered in the template table
- Query response format: <str> = A specified template name
- Description: Sets a template to be used in the search mode. If "NONE" is set, no template can be used.

7. `JTOLerance:MODE` (Jitter tolerance option)

- Function: Sets the jitter tolerance measurement mode
- Availability of command and query: Command/Query
- Command: JTOLerance:MODE <type>
- Parameter: <type> = {SEARch | SWEep}
- Query response format: SEAR | SWE
- Description: The jitter tolerance measurement mode is set.

Command parameter	Measurement mode
SEAR	Search mode
SWE	Sweep mode

8. `JTOLerance:CFRequency` (Jitter tolerance option)

- Function: Sets the clock frequency in jitter tolerance measurement
- Availability of command and query: Command/Query
- Command: JTOLerance:CFRequency <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: Clock frequency in jitter tolerance measurement is specified.
  Clock frequency may not be set under 800 MHz depending on how jitter frequency and jitter amplitude have already been specified.
  Setting range: 10,000 kHz to 3,200,000 kHz
  Setting resolution: 1 kHz
  Setting unit: kHz

9. `JTOLerance:STIMe` (Jitter tolerance option)

- Function: Specifies settling time in jitter tolerance measurement
- Availability of command and query: Command/Query
- Command: JTOLerance:STIMe <real>
- Parameter: <real>
- Query response format: NR2 (fixed-point-type)
- Description: Settling time in jitter tolerance measurement is specified.
  - Setting range: 0.0 s to 100.0 s
  - Setting resolution: 0.1 s
  - Setting unit: s

10. `JTOLerance:PERiod` (Jitter tolerance option)

- Function: Sets the time to conduct a point measurement in jitter tolerance measurement
- Availability of command and query: Command/Query
- Command: JTOLerance:PERiod <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The time to perform jitter tolerance measurement for one point is specified.
  - Setting range: 1 s to 1000 s
  - Setting resolution: 1 s
  - Setting unit: s

11. `JTOLerance:ETHReshold` (Jitter tolerance option)

- Function: Sets the error counts providing basic data to evaluate jitter tolerance measurement results
- Availability of command and query: Command/Query
- Command: JTOLerance:ETHReshold <int>
- Parameter: <int>
- Query response format: NR1 (integer type)
- Description: The error counts providing basic data to evaluate jitter tolerance measurement results is set.
  - Setting range: 1 bit to 1,000,000 bits
  - Setting resolution: 1 bit

12. `JTOLerance:ASEarch` (Jitter tolerance option)

- Function: Sets the auto-search function for jitter tolerance measurement to on or off
- Availability of command and query: Command/Query
- Command: JTOLerance:ASEarch <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: The auto-search function for jitter tolerance measurement is enabled (ON) or disabled (OFF).

13. `JTOLerance:MDATa` (Jitter tolerance option)

- Function: Sets the displaying of jitter tolerance measurement results to on or off
- Availability of command and query: Command/Query
- Command: JTOLerance:MDATa <bool>
- Parameter: <bool>
- Query response format: 0 | 1
- Description: Enabling (ON)/disabling (OFF) of the displaying of jitter tolerance measurement results is specified.

14. `JTOLerance:MDATa:ALL:PASS?` (Jitter tolerance option)

- Function: Sends a query for total evaluations of jitter tolerance measurement data.
- Availability of command and query: Query
- Query: JTOLerance:MDATa:ALL:PASS?
- Query response format: 0 | 1
- Description: A query is sent for total evaluations of jitter tolerance measurement data. If evaluations at every point are satisfactory, "1" will be returned. Otherwise, "0" will be returned. Put a query after jitter tolerance measurement has been completed.

Measurement modes	Condition for satisfactory evaluations
Search mode	Jitter amplitude at the jitter measurement point is larger than the template.
Sweep mode	The error counter is lower than the one providing basic data to evaluate jitter tolerance measurement results.

6. SAMPLE PROGRAMS

This chapter provides examples for remote control using the GPIB port of the D3371.

CAUTION: ***The sample programs below use Visual Basic 6.0 (VB hereafter) as the programming language. The GPIB control board used is manufactured by National Instrument (NI hereinafter). The control driver is a driver provided by NI.***

6.1 Sample Programs for Setting and Reading Measurement Conditions

Example 1 Initialize this device.

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "*RST")                          ' Initialization
```

Example 2 Setting the SSG frequency to 3.6 GHz

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "SOURCE:FREQ 3600000")           ' Sets frequency to 3.6 GHz.
```

Example 3 Executing an automatic search and displaying the result

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "SENSE:ASE ON")                  ' Starts an automatic search.
Call ibwrt(ud, "*WAI")                          ' Waits for end of auto search.
Call ibwrt(ud, "SENSE:ASE:FAIL?")               ' Query command for an automatic search result.
ReadBuff = Space(30)                            ' Acquires 30-byte-space character string.
Call ibrd(ud, ReadBuff)                         ' Reads an automatic search result.
Text1.Text = Left(ReadBuff, InStr(ReadBuff, Chr(10))   1)
                                                ' Displays an automatic search result.
```

Example 4 Save the D3371 settings to a file.

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "MMEM:STOR:SET ""D:\D3371\SETUP1""")
                                                ' SETUP1.EDS, SETUP1.PGS, SETUP1.SGS, and
                                                  SETUP1.TAS are created.
```

Example 5 Setting the PROG pattern as the pattern type

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "SOURCE:PATT PROG")              ' Sets PPG pattern type to PROG.
Call ibwrt(ud, "SENSE:PATT PROG")               ' Sets ED pattern type to PROG.
```

Example 6 Setting the PROG pattern length.

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "SOUR1:PATT:PROG 128")           ' Sets PPG PROG pattern length to 128 bits.
Call ibwrt(ud, "SENS1:PATT:PROG 128")           ' Sets ED PROG pattern length to 128 bits.
```

Example 7 Transferring a PROG pattern as character-string data (setting the first 16 bits to 0x1234)

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "SOUR1:PATT:PROG:DATA 0,16,""#1234""")
                                                ' PROG pattern transfer to PPG (HEX format)
Call ibwrt(ud, "SOUR1:PATT:PROG:DATA 0,16,""#B0001001000110100""")'
                                                ' PROG pattern transfer to PPG (BIN format)
```

Example 8 Transferring a PROG pattern as block data (setting the first 16 bits to 0x1234)

```
Dim PattBuff(50) As Byte                        ' Defines PattBuff area.
Call ibclr(ud)                                  ' Device clear
PattBuff(0) = &H12                              ' Sets 0x12 in PattBuff(0).
PattBuff(1) = &H34                              ' Sets 0x34 in PattBuff(1).
Call ibeot(ud, 0)                               ' Disables an automatic assertion of EOI line.
Call ibwrt(ud, "SOUR1:PATT:PROG:BDAT 0,16,#12")
                                                ' Sends up to block data header.
Call ibeot(ud, 1)                               ' Enables an automatic assertion of EOI line.
Call ibwrt32(ud, PattBuff(0), 2)                ' Sends 16-bit binary data.
Call copy_ibvars                                ' Sets global variable for NI driver.
```

Example 9 Reading a PROG pattern as character-string data (first 16 bits)

```
Call ibclr(ud)                                  ' Device clear
Call ibwrt(ud, "SOUR1:PATT:PROG:DATA? 0,16")
                                                ' Reads PROG pattern from PPG.
ReadBuff = Space(30)                            ' Acquires 30-byte-space character string.
Call ibrd(ud, ReadBuff)                         ' Receives pattern as character-string data.
Text1.Text = "PROG Pattern=" & Left(ReadBuff, InStr(ReadBuff, Chr(0)) - 1)
                                                ' Displays read pattern.
```

Example of results

```
PROG Pattern="H1234"
```

Example 10 Reading a PROG pattern as block data (first 16 bits)

```
Dim PattBuff(50) As Byte                          ' Defines PattBuff area.
Dim DigitNum As Integer                           ' Number of block data digits
Dim BlockLen As String                            ' Block data length
Dim DataByte As String                            ' For character string conversion of binary data

Call ibclr(ud)                                    ' Device clear
Call ibwrt(ud, "SOUR1:PATT:PROG:BDAT? 0,16")
                                                  ' Reads PROG pattern from PPG.
Call ibrd32(ud, PattBuff(0), 20)                  ' Reads block data.
Call copy_ibvars                                  ' Sets global variables for NI driver.
DigitNum = PattBuff(1)    &H30                    ' Acquires number of block data digits.

For i = 0 To DigitNum - 1                         ' Executes loop as many times as number of block
                                                    data digits.
    BlockLen = BlockLen & Chr(PattBuff(i + 2))
                                                  ' Creates block data length.
Next i

For i = DigitNum + 2 To Val(BlockLen) + DigitNum + 2 - 1
                                                  ' Loop for acquiring binary data
    DataByte = DataByte & Hex(PattBuff(i)) ' Converts binary data to character string.
Next i

Text1.Text = "PROG Pattern=" & DataByte           ' Displays character string as pattern in Text1.
```

Example of results

```
PROG Pattern=1234
```

Example 11 Specifying tracking

```
Call ibclr(ud)                                    ' Device clear
Call ibwrt(ud, "SENS:PATT:TRAC ON")               ' Turns on pattern tracking.
Call ibwrt(ud, "OUTP3:TRAC ON")                   ' Turns on data output tracking.
```

6.2 Sample Measurement Programs

Example 1 When SRQ is not used

```
Dim Result As Integer                          ' For numeric conversion of character-string result
Dim State As Integer                           ' For status management

Call ibdev(0, GpibAddress, 0, T300s, 1, 0, ud)
                                               ' Sets timeout value to 300 s.
Call ibclr(ud)                                 ' Device clear

Call ibwrt(ud, "SENS1:BME:MTIM:MODE SING") ' Sets measurement timer mode to single mode.
Call ibwrt(ud, "SENS1:BME:MTIM:PER 0,0,0,10")
                                               ' Sets the measurement period to 10 seconds.
Call ibwrt(ud, "SENS1:PATT:TRAC ON")           ' Turns on pattern tracking.
Call ibwrt(ud, "SOUR1:PATT PRBS23")            ' Sets pattern to PRBS23.
Call ibwrt(ud, "SOUR1:PATT:PRBS:MRAT MRAT4")
                                               ' Sets mark ratio to 1/2.

Call ibwrt(ud, "SENS1:ASE ON")                 ' Starts an automatic search.
Call ibwrt(ud, "*WAI")                         ' Waits for an automatic search to end.
Call ibwrt(ud, "SENS1:ASE:FALL?")              ' Acquires an automatic search result.
ReadBuff = Space(30)                           ' Acquires 30-byte-space character string.
Call ibrd(ud, ReadBuff)                        ' Reads an automatic search result.

Result = Val(ReadBuff)                         ' Converts result to value.
If Result = 0 Then                             ' When result is 0
    Call ibwrt(ud, "SENS1:BME ON")             ' Starts measurement.

    State = 1                                  ' Initializes state variable.
    While State = 1                            ' Loop for checking end of measurement
        Call ibwrt(ud, "SENS1:BME?")           ' Query command for acquiring measurement
                                               status
        ReadBuff = Space(30)                   ' Acquires 30-byte-space character string.
        Call ibrd(ud, ReadBuff)                ' Reads measurement state.
        State = Val(ReadBuff)                  ' Sets measurement state in state variable.
        DoEvents
    Wend

    Call ibwrt(ud, "FETC:BME:ERAT?")           ' Acquires error rate.
    ReadBuff = Space(30)                       ' Acquires 30-byte-space character string.
    Call ibrd(ud, ReadBuff)                    ' Reads error rate.

    Text1.Text = "Error Rate=" & Left(ReadBuff, InStr(ReadBuff, Chr(10)) - 1)
                                               ' Displays error rate in Text1.

End If
```

Example of results

```
Error Rate=0.0000E-09
```

Example 2 When SRQ is used

```
Dim Result As Integer                          ' For numeric conversion of character-string result
Dim Stb As Integer                             ' For storing serial polling response bytes

Call ibdev(0, GpibAddress, 0, T300s, 1, 0, ud)
                                               ' Sets timeout value to 300 s.
Call ibclr(ud)                                 ' Device clear
Call ibwrt(ud, "*CLS")                         ' Clear status byte.
Call ibwrt(ud, "*SRE 128")                     ' Sets OPER of status byte.
Call ibwrt(ud, "STAT:OPER:ENAB 512")           ' Sets END bit to enable.

Call ibwrt(ud, "SENS1:BME:MTIM:MODE SING") ' Sets measurement timer mode to single mode.
Call ibwrt(ud, "SENS1:BME:MTIM:PER 0,0,0,10")
                                               ' Sets measurement period to 10 s.
Call ibwrt(ud, "SENS1:PATT:TRAC ON")           ' Turns on pattern tracking.
Call ibwrt(ud, "SOUR1:PATT PRBS23")            ' Sets pattern to PRBS23.
Call ibwrt(ud, "SOUR1:PATT:PRBS:MRAT MRAT4")
                                               ' Sets mark ratio to 1/2.

Call ibwrt(ud, "SENS1:ASE ON")                 ' Starts an automatic search.
Call ibwrt(ud, "*WAI")                         ' Waits for an automatic search to end.
Call ibwrt(ud, "SENS1:ASE:FAIL?")              ' Acquires an automatic search result.
ReadBuff = Space(30)                           ' Acquires 30-byte-space character string.
Call ibrd(ud, ReadBuff)                        ' Reads an automatic search result.

Result = Val(ReadBuff)                         ' Converts result to value.
If Result = 0 Then                             ' When result is 0
    Call ibwrt(ud, "SENS1:BME ON")               Starts measurement.

    Call WaitSRQ(boardID, Stb)                 ' Waits for end of measurement.
    Call ibrsp(ud, Stb)                        ' Executes serial polling.
    Call ibwrt(ud, "FETC:BME:ERAT?")           ' Acquires error rate.
    ReadBuff = Space(30)                       ' Acquires 30-byte-space character string.
    Call ibrd(ud, ReadBuff)                    ' Reads error rate.

    Text1.Text = "Error Rate=" & Left(ReadBuff, InStr(ReadBuff, Chr(10)) - 1)
                                               ' Displays error rate in Text1.

End If
```

Example of results

```
Error Rate=0.0000E-09
```

APPENDIX

This chapter lists commands and the commands used on setting menus.

A.1 LIST OF COMMANDS

The commands of the D3371 for each subsystem are shown.

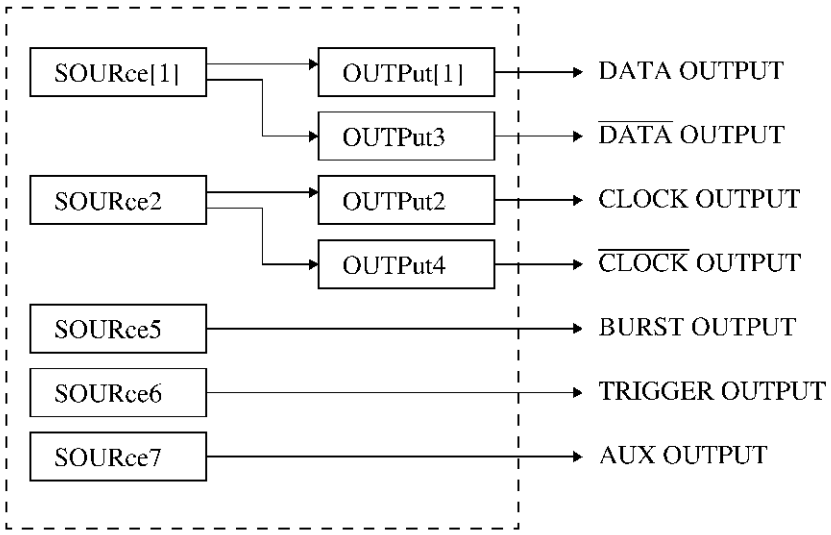
For detailed information on commands, refer to Chapter 5, "COMMAND REFERENCE."

A.1.1 Subsystems and I/O Interfaces

The relationships between subsystems and the I/O interfaces of measurement modules are shown below.

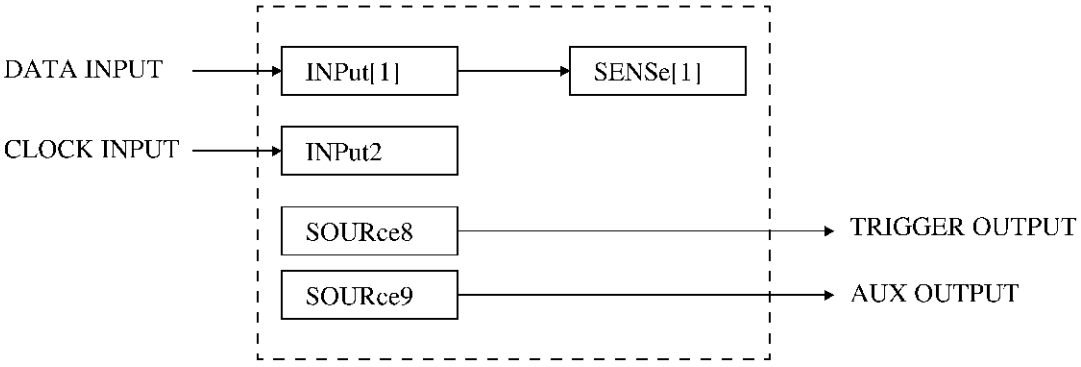
A.1.1.1 PPG Module I/O and Subsystems

The following diagram shows the correspondence between the I/O interfaces of the PPG module of the D3371 and subsystems:



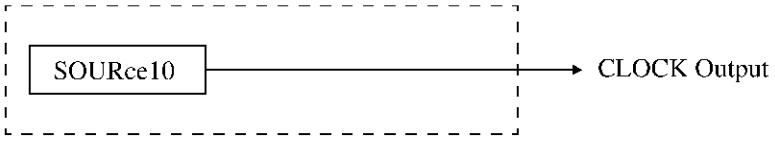
A.1.1.2 ED Module I/O and Subsystems

The following diagram shows the correspondence between the I/O interfaces of the ED module of the D3371 and subsystems:



A.1.1.3 SSG Module I/O and Subsystems

The following diagram shows the correspondence between the I/O interfaces of the SSG module of the D3371 and subsystems:



A.1.2 Common Commands

KEYWORD	PARAMETER FORM	COMMENTS
*CLS	---	no query
*ESE	<int>	
*ESE?	<NR1>	
*ESR?	<NR1>	query only
*IDN?	<manufacturer>,<model>,	
	<serial number>,<firmware level>	query only
*OPC	---	
*OPC?	1	
*RST	---	no query
*SRE	<int>	
*SRE?	<NR1>	
*STB?	<NR1>	query only
*TST?	0	query only
*WAI	---	
*OPT?	<opt>,<opt>,...	query only

A.1.3 PPG Module

A.1.3.1 SOURce[1] Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SOURce[1]		
:PATTern		
[:SELect]	PRBS<n>|ZSUBstitut<n>|PROGram|STM<n>|FLEXible	
[:SELect]?	PRBS<n>|ZSUB<n>|PROG|STM<n>|FLEX	
:PRBS		
:MRATio	MRATio<n>|MINVerted4	
:MRATio?	MRAT<n>|MINV4	
:ZSUBstitut		
[:ZLENgth]	<int>	
[:ZLENgth]?	<NR1>	
:PROGram		
[:LENGth]	<int>	
[:LENGth]?	<NR1>	
:DATA	<start-bit>,<length>,<str>	
:DATA?	<start-bit>,<length>/<str>	
:BDATa	<start-bit>,<length>,<block>	
:BDATa?	<start-bit>,<length>/<block>	
:STM		
:NFRame	<int>	
:NFRame?	<NR1>	
:DATA	<start_frm>,<start_row>,<start_col>,<length>,<str>	
:DATA?	<start_frm>,<start_row>,<start_col>,<length>/<str>	
:BDATa	<start_frm>,<start_row>,<start_col>,<length>,<block>	
:BDATa?	<start_frm>,<start_row>,<start_col>,<length>/<block>	
:PAYLoad		
:PRBS	<bool>	
:PRBS?	0|1	
:SELect	PRBS<n>	
:SELect?	PRBS<n>	
:MRATio	MRATio<n>|MINVerted4	
:MRATio?	MRAT<n>|MINV4	
:SCRamble	<bool>	
:SCRamble?	0|1	
:B1	<bool>	
:B1?	0|1	
:FLEXible		
:STABle	<index>,<prog_no>|PRBS,<length>,HIGH|LOW	
:STABle?	<index>/<prog_no>|PRBS,<length>,HIGH|LOW	

```
            :COUNt?          <NR1>                                  query only
            :NEW             <index>,<prog_no>|PRBS,<length>,HIGH|LOW
                                                                    no query
            :REMove          <index>
            :DEFault
         :PROGram
            [:LENGth]        <prog_no>,<length>
            [:LENGth]?       <prog_no>/<NR1>
            :DATA            <prog_no>,<start_bit>,<length>,<str>
            :DATA?           <prog_no>,<start_bit>,<length>/<str>
            :BDATa           <prog_no>,<start_bit>,<length>,<block>
            :BDATa?          <prog_no>,<start_bit>,<length>/<block>
         :PRBS
            :SELect          PRBS<n>
            :SELect?         PRBS<n>
            :MRATio          MRATio<n>|MINVerted4
            :MRATio?         MRAT<n>|MINV4
      :POLarity              NORMal|INVerted
      :POLarity?             NORM|INV
   :EADDition                <bool>
   :EADDition?               0|1
      :SINGle                                                       no query
      :MODE                  SINGle|REPeat|EXTernal
      :MODE?                 SING|REP|EXT
      :RATE                  <real>
      :RATE?                 <NR3>
      :ROUTe                 <int>
      :ROUTe?                <NR1>
```

A.1.3.2 OUTPut[1] Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
OUTPut[1]		
[:STATe]	<bool>	
[:STATe]?	0|1	
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<real>	
[:AMPLitude]?	<NR2>	
:HIGH	<real>	
:HIGH?	<NR2>	
:MIDDle	<real>	
:MIDDle?	<NR2>	
:LOW	<real>	
:LOW?	<NR2>	
:CROSs	<int>	
:CROSs?	<NR1>	
:TERMination	P1R3V|GND|M2V|CML	
:TERMination?	P1R3V|GND|M2V|CML	
:VOLTage	<real>	
:VOLTage?	<NR2>	

A.1.3.3 OUTPut3 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
OUTPut3		
[:STATe]	<bool>	
[:STATe]?	0|1	
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<real>	
[:AMPLitude]?	<NR2>	
:HIGH	<real>	
:HIGH?	<NR2>	
:MIDDle	<real>	
:MIDDle?	<NR2>	
:LOW	<real>	
:LOW?	<NR2>	
:CROSs	<int>	
:CROSs?	<NR1>	
:TERMination	P1R3V|GND|M2V|CML	
:TERMination?	P1R3V|GND|M2V|CML	
:VOLTage	<real>	
:VOLTage?	<NR2>	
:TRACk	<bool>	
:TRACk?	0|1	

A.1.3.4 SOURce2 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SOURce2		
:DELay	<int>	
:DELay?	<NR1>	

A.1.3.5 OUTPut2 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
OUTPut2		
[:STATe]	<bool>	
[:STATe]?	0|1	
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<real>	
[:AMPLitude]	<NR2>	
:HIGH	<real>	
:HIGH?	<NR2>	
:MIDDle	<real>	
:MIDDle?	<NR2>	
:LOW	<real>	
:LOW?	<NR2>	
:TERMination	P1R3V|GND|M2V|CML	
:TERMination?	P1R3V|GND|M2V|CML	
:VOLTage	<real>	
:VOLTage?	<NR2>	

A.1.3.6 OUTPut4 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
OUTPut4		
[:STATe]	<bool>	
[:STATe]?	0|1	
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<real>	
[:AMPLitude]	<NR2>	
:HIGH	<real>	
:HIGH?	<NR2>	
:MIDDle	<real>	
:MIDDle?	<NR2>	
:LOW	<real>	
:LOW?	<NR2>	
:TERMination	P1R3V|GND|M2V|CML	
:TERMination?	P1R3V|GND|M2V|CML	
:VOLTage	<real>	
:VOLTage?	<NR2>	
:TRACk	<bool>	
:TRACk?	0|1	

A.1.3.7 SOURce5 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SOURce5		
:BOUTput		
[:STATe]	<bool>	
[:STATe]?	0|1	
:SOURce	INTernal|EXTernal	
:SOURce?	INT|EXT	
:CYCLe	<int>	
:CYCLe?	<NR1>	
:OTIMe	<int>	
:OTIMe?	<NR1>	

A.1.3.8 SOURce6 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SOURce6		
:TRIGger		
[:MODE]	CLOCk8|CLOCk32|PATTern|FLEXible	
[:MODE]?	CLOC8|CLOC32|PATT|FLEX	
:PRBS		
:POSition	<int>	
:POSition?	<NR1>	
:ZSUBstitut		
:POSition	<int>	
:POSition?	<NR1>	
:PROGram		
:POSition	<int>	
:POSition?	<NR1>	
:STM		
:FNUMber	<int>	
:FNUMber?	<NR1>	
:ROW	<int>	
:ROW?	<NR1>	
:COLumn	<int>	
:COLumn?	<NR1>	
:MODE	PATTern|FRAMe	
:MODE?	PATT|FRAM	
:FLEXible		
:INDex	<int>	
:INDex?	<NR1>	
:ADDRess	<int>	
:ADDRess?	<NR1>	

A.1.3.9 SOURce7 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SOURce7		

A.1.4 ED Module

A.1.4.1 SENSe[1] Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SENSe[1]		
:PATTern		
[:SELect]	PRBS<n>|ZSUBstitut<n>|PROGram|STM<n>|FLEXible	
[:SELect]?	PRBS<n>|ZSUB<n>|PROG|STM<n>|FLEX	
:PRBS		
:MRATio	MRATio<n>|MINVerted4	
:MRATio?	MRAT<n>|MINV4	
:ZSUBstitut		
[:ZLENgth]	<int>	
[:ZLENgth]?	<NR1>	
:PROGram		
[:LENGth]	<int>	
[:LENGth]?	<NR1>	
:DATA	<start-bit>,<length>,<str>	
:DATA?	<start-bit>,<length>/<str>	
:BDATa	<start-bit>,<length>,<block>	
:BDATa?	<start-bit>,<length>/<block>	
:STM		
:NFRame	<int>	
:NFRame?	<NR1>	
:DATA	<start_frm>,<start_row>,<start_col>,<length>,<str>	
:DATA?	<start_frm>,<start_row>,<start_col>,<length>/<str>	
:BDATa	<start_frm>,<start_row>,<start_col>,<length>,<block>	
:BDATa?	<start_frm>,<start_row>,<start_col>,<length>/<block>	
:PAYLoad		
:PRBS	<bool>	
:PRBS?	0|1	
:SELect	PRBS<n>	
:SELect?	PRBS<n>	
:MRATio	MRATio<n>|MINVerted4	
:MRATio?	MRAT<n>|MINV4	
:SCRamble	<bool>	
:SCRamble?	0|1	
:B1	<bool>	
:B1?	0|1	
:FLEXible		
:STABle	<index>,<prog_no>|PRBS,<length>,HIGH|LOW	
:STABle?	<index>,<prog_no>|PRBS,<length>,HIGH|LOW	

A.1.4 ED Module

```
            :COUNt?          <NR1>                                  query only
           :NEW              <index>,<prog_no>|PRBS,<length>,HIGH|LOW
                                                                    no query
            :REMove          <index>                                no query
            :DEFault                                                no query
         :PROGram
            [:LENGth]        <prog_no>,<length>
            [:LENGth]?       <prog_no>/<NR1>
            :DATA            <prog_no>,<start_bit>,<length>,<str>
            :DATA?           <prog_no>,<start_bit>,<length>/<str>
            :BDATa           <prog_no>,<start_bit>,<length>,<block>
            :BDATa?          <prog_no>,<start_bit>,<length>/<block>
         :PRBS
            :SELect          PRBS<n>
            :SELect?         PRBS<n>
            :MRATio          MRATio<n>|MINVerted4
            :MRATio?         MRAT<n>|MINV4
      :POLarity              NORMal|INVerted
      :POLarity?             NORM|INV
      :TRACk                 <bool>
      :TRACk?                0|1
   :SYNChronisat
      [:AUTO]                ONCE|<bool>
      [:AUTO]?               0|1
      :PATTern
         :PROGram
            :ADDRess         <int>
            :ADDRess?        <NR1>
         :FLEXible
            :INDex           <int>
            :INDex?          <NR1>
            :ADDRess         <int>
            :ADDRess?        <NR1>
         :LENGth             <int>
         :LENGth?            <NR1>
      :THReshold
         :AUTO               <bool>
         :AUTO?              0|1
         :PRBS
            :GAIN            <real>
            :GAIN?           <NR3>
            :LOSS            <real>
            :LOSS?           <NR3>
```

```
      :MEMory
         :GAIN          <real>
         :GAIN?         <NR3>
         :LOSS          <real>
         :LOSS?         <NR3>
:MASK
   :ROUTe<n>            <bool>
   :ROUTe<n>?           0|1
:ASEarch
   [:STATe]             <bool>
   [:STAEe]?            0|1
   :FAIL?               0|1                                  query only
   :CLOCk               <bool>
   :CLOCk?              0|1
   :TVOLtage            <bool>
   :TVOLtage?           0|1
   :PATTern             <bool>
   :PATTern?            0|1
:MONitor
   :HISTory
      :CLEar                                                no query
:LOG
   :BMEasurement        <bool>
   :BMEasurement?       0|1
   :HISTory
      :SCLoss           <bool>
      :SCLoss?          0|1
      :MDATa
         :IERate        <bool>
         :IERate?       0|1
         :IECount       <bool>
         :IECount?      0|1
         :EINTervals    <bool>
         :EINTervals?   0|1
         :EFINtervals   <bool>
         :EFINtervals?  0|1
         :FREQuency     <bool>
         :FREQuency?    0|1
:BMEasurement
   [:STATe]             <bool>
   [:STATe]?            0|1
   :MTIMe
      :MODE             SINGle|REPeat|UNTimed
      :MODE?            SING|REP|UNT
```

A.1.4 ED Module

```
      :INTerval          <real>
      :INTerval?         <NR2>
      :PERiod            <day>,<hour>,<minute>,<second>
      :PERiod?           <day>,<hour>,<minute>,<second>
   :DMODe                OIT|OPT|SOT
   :DMODe?               OIT|OPT|SOT
   :BURSt                <bool>
   :BURSt?               0|1
   :EVALuate
      :CLOSs             <bool>
      :CLOSs?            0|1
      :SLOSs             <bool>
      :SLOSs?            0|1
   :EPERformance
      :THReshold         <us_ses>,<dm>
      :THReshold?        <us_ses>,<dm>
   :CDATa                <bool>
   :CDATa?               0|1
:SPECific
   :PROGram
      :STARt
         :ADDRess        <int>
         :ADDRess?       <NR1>
      :END
         :ADDRess        <int>
         :ADDRess?       <NR1>
   :ZSUBstitut
      :STARt
         :ADDRess        <int>
         :ADDRess?       <NR1>
      :END
         :ADDRess        <int>
         :ADDRess?       <NR1>
   :STM
      :STARt
         :FNUMber        <int>
         :FNUMber?       <NR1>
         :ROW            <int>
         :ROW?           <NR1>
         :COLumn         <int>
         :COLumn?        <NR1>
      :END
         :FNUMber        <int>
         :FNUMber?       <NR1>
```

```
            :ROW            <int>
            :ROW?           <NR1>
            :COLumn         <int>
            :COLumn?        <NR1>
         :MODE              PATTern|FRAMe
         :MODE?             PATT|FRAM
      :FLEXible
         :STARt
            :INDex          <int>
            :INDex?         <NR1>
            :ADDRess        <int>
            :ADDRess?       <NR1>
         :END
            :INDex          <int>
            :INDex?         <NR1>
            :ADDRess        <int>
            :ADDRess?       <NR1>
   :EPANalysis
      :ERECord
         [:STATe]           <bool>
         [:STATe]?          0|1
         :AREa              SPECific|TOTal
         :AREa?             SPEC|TOT
```

A.1.4.2 INPut[1] Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
INPut[1]		
:POLarity	NORMal|INVerted	
:POLarity?	NORM|INV	
:TVOLtage	<real>	
:TVOLtage?	<NR2>	
:TERMination	P3V|P1R3V|GND|M2V|CML	
:TERMination?	P3V|P1R3V|GND|M2V|CML	
:VARiable	<bool>	
:VARiable?	0|1	
:VOLTage	<real>	
:VOLTage?	<NR2>	

A.1.4.3 INPut2 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
INPut2		
:POLarity	NORMal|INVerted	
:POLarity?	NORM|INV	
:DELay	<int>	
:DELay?	<NR1>	
:TERMination	P3V|P1R3V|GND|M2V|CML	
:TERMination?	P3V|P1R3V|GND|M2V|CML	
:VARiable	<bool>	
:VARiable?	0|1	
:VOLTage	<real>	
:VOLTage?	<NR2>	

A.1.4.4 SOURce8 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SOURce8		
:TRIGger		
[:MODE]	CLOCk16|PATTern|FLEXible	
[:MODE]?	CLOC16|PATT|FLEX	
STM		
:MODE	PATTern|FRAMe	
:MODE?	PATT|FRAM	

A.1.4.5 SOURce9 Subsystem

KEYWORD	PARAMETER FORM	COMMENTS
SOURce9		
:AUX		
[:MODE]	DTYPe|SYNChronisat	
[:MODE]?	DTYP|SYNC	

A.1.5 Results and System Commands

A.1.5.1 FETCh and CFETch Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
FETCh|CFETch		
[:SENSe[1]]		
:BMEasurement		
:BCOunt		
[:TOTal]?	<NR1>|<NR3>	query only
:OMITting?	<NR1>|<NR3>	query only
:INSerting?	<NR1>|<NR3>	query only
:OVERhead?	<NR1>|<NR3>	query only
:PAYLoad?	<NR1>|<NR3>	query only
:SPECific?	<NR1>|<NR3>	query only
:OTHer?	<NR1>|<NR3>	query only
:ERATe		
[:TOTal]?	<NR3>	query only
:OMITting?	<NR3>	query only
:INSerting?	<NR3>	query only
:OVERhead?	<NR3>	query only
:PAYLoad?	<NR3>	query only
:SPECific?	<NR3>	query only
:OTHer?	<NR3>	query only
:ECOunt		
[:TOTal]?	<NR1>|<NR3>	query only
:OMITting?	<NR1>|<NR3>	query only
:INSerting?	<NR1>|<NR3>	query only
:OVERhead?	<NR1>|<NR3>	query only
:PAYLoad?	<NR1>|<NR3>	query only
:SPECific?	<NR1>|<NR3>	query only
:OTHer?	<NR1>|<NR3>	query only
:IERate		
[:TOTal]?	<NR3>	query, CFETch only
:OMITting?	<NR3>	query, CFETch only
:INSerting?	<NR3>	query, CFETch only
:OVERhead?	<NR3>	query, CFETch only
:PAYLoad?	<NR3>	query, CFETch only
:SPECific?	<NR3>	query, CFETch only
:OTHer?	<NR3>	query, CFETch only
:IECount		
[:TOTal]?	<NR1>|<NR3>	query, CFETch only
:OMITting?	<NR1>|<NR3>	query, CFETch only

```
   :INSerting?     <NR1>|<NR3>                    query, CFETch only
   :OVERhead?      <NR1>|<NR3>                    query, CFETch only
   :PAYLoad?       <NR1>|<NR3>                    query, CFETch only
   :SPECific?      <NR1>|<NR3>                    query, CFETch only
   :OTHer?         <NR1>|<NR3>                    query, CFETch only
:EINTervals
   [:TOTal]?       <NR1>|<NR2>                    query only
   :OMITting?      <NR1>|<NR2>                    query only
   :INSerting?     <NR1>|<NR2>                    query only
   :OVERhead?      <NR1>|<NR2>                    query only
   :PAYLoad?       <NR1>|<NR2>                    query only
   :SPECific?      <NR1>|<NR2>                    query only
   :OTHer?         <NR1>|<NR2>                    query only
:EFINtervals
   [:TOTal]?       <NR1>|<NR2>                    query only
   :OMITting?      <NR1>|<NR2>                    query only
   :INSerting?     <NR1>|<NR2>                    query only
   :OVERhead?      <NR1>|<NR2>                    query only
   :PAYLoad?       <NR1>|<NR2>                    query only
   :SPECific?      <NR1>|<NR2>                    query only
   :OTHer?         <NR1>|<NR2>                    query only
:FREQuency?        <NR1>                          query only
:CLINtervals?      <NR1>                          query only
:SLINtervals?      <NR1>                          query only
:TEINtervals
   :EM<n>
      [:TOTal]?    <NR1>|<NR2>                   query only
   :LEM8
      [:TOTal]?    <NR1>|<NR2>                   query only
:TEFintervals
   :EM<n>
      [:TOTal]?    <NR1>|<NR2>                   query only
   :LEM8
      [:TOTal]?    <NR1>|<NR2>                   query only
:EPERformance
   :ESEConds
      [:TOTal]?    <NR1>|<NR2>                   query only
   :EFSeconds
      [:TOTal]?    <NR1>|<NR2>                   query only
   :SESeconds
      [:TOTal]?    <NR1>|<NR2>                   query only
   :USEConds
      [:TOTal]?    <NR1>|<NR2>                   query only
   :DMINutes
```

[:TOTal]?	<NR1>|<NR2>	query only
:B1ERror		
:BCOunt?	<NR1>|<NR3>	query only
:ERATe?	<NR3>	query only
:ECOunt?	<NR1>|<NR3>	query only
:MTIMe		
:ETIMe?	<day>,<hour>,<minute>,<second>	query, FETCh only
:TIMed?	<day>,<hour>,<minute>,<second>	query, FETCh only
:STIMe?	<year>,<month>,<day>,<hour>, <minute>,<second>	query, FETCh only
:RTIMe?	<year>,<month>,<day>,<hour>, <minute>,<second>	query, FETCh only
:MONitor		
:RTIMe		
:DERRor?	0|1	query, FETCh only
:CLOSs?	0|1	query, FETCh only
:SLOSs?	0|1	query, FETCh only
:HISTory		
:DERRor?	0|1	query, FETCh only
:CLOSs?	0|1	query, FETCh only
:SLOSs?	0|1	query, FETCh only
:EPANalysis		
:MDATa		
:TSERies		
:ERRor		
:COUNt?	<NR1>	query, FETCh only
:CYCLe		
:COUNt?	<NR1>	query, FETCh only
:JTOLerance:		
:MDATa		
:JFRequency		
:COUNt?	<NR1>	query, FETCh only
:JAMPlitude		
:COUNt?	<NR1>	query, FETCh only
:MDATa?	<index_freq>,<index_ampl>/<str>	query, FETCh only
:MTIMe		
:ETIMe?	<day>,<hour>,<minute>,<second>	query, FETCh only
:STIMe?	<year>,<month>,<day>,<hour>, <minute>,<second>	query, FETCh only

A.1.5.2 DISPlay Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
DISPlay		
:WINDow		
:BCOunt		
:DFORmat	INTegral|EXPonential	
:DFORmat?	INT|EXP	
:ECOunt		
:DFORmat	INTegral|EXPonential	
:DFORmat?	INT|EXP	
:IECount		
:DFORmat	INTegral|EXPonential	
:DFORmat?	INT|EXP	
:EINTervals		
:DFORmat	INTegral|PERCent	
:DFORmat?	INT|PERC	
:EFINtervals		
:DFORmat	INTegral|PERCent	
:DFORmat?	INT|PERC	
:TEINtervals		
:DFORmat	INTegral|PERCent	
:DFORmat?	INT|PERC	
:TEFintervals		
:DFORmat	INTegral|PERCent	
:DFORmat?	INT|PERC	
:EPERformance		
:DFORmat	INTegral|PERCent	
:DFORmat?	INT|PERC	

A.1.5.3 HCOPy Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
HCOPy		
[:ITEM]		
:ED		
:BMEasurement		no query
:JTOLerance		no query

A.1.5.4 MMEMory Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
MMEMory		
:LOAD		
:SETup	<str>	no query
:SSG		
:SETup	<str>	no query
:PPG		
:SETup	<str>	no query
:PROGram	<str>	no query
:STM	<str>	no query
:FLEXible	<str>	no query
:PROGram	<prog_no>,<str>	no query
:ED		
:SETup	<str>	no query
:PROGram	<str>	no query
:STM	<str>	no query
:FLEXible	<str>	no query
:PROGram	<prog_no>,<str>	no query
:JTOLerance	<str>	no query
:STORe		
:SETup	<str>	no query
:SSG		
:SETup	<str>	no query
:PPG		
:SETup	<str>	no query
:PROGram	<str>	no query
:STM	<str>	no query
:FLEXible	<str>	no query
:PROGram	<prog_no>,<str>	no query
:ED		
:SETup	<str>	no query
:PROGram	<str>	no query
:STM	<str>	no query
:FLEXible	<str>	no query
:PROGram	<prog_no>,<str>	no query
:BMEasurement		
:LOG	<str>	no query
:EPANalysis		
:EPANalysis?	0|1	query only
:TSERies		
:TEXT	<cn>,<sp1>,<sp2>,<sp3>,<nr>,<str>	no query
:STATistics		

:TEXT	<sp1>,<sp2>,<sp3>,<range>,<str>	no query
:CANCel		no query
:FAIL?	0|1	query only
:JTOLerance	<str>	no query
:TEXT	<str>	no query

A.1.5.5 SYSTem Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
SYSTem		
:BEEPer		
:DERRor		
[:STATe]	<bool>	
[:STATe]?	0|1	
:ALARm		
[:STATe]	<bool>	
[:STATe]?	<0|1>	
:ERRor?	<NR1>,<str>	query only
:DATE	<year>,<month>,<day>	
:DATE?	<year>,<month>,<day>	
:TIME	<hour>,<minute>,<second>	
:TIME?	<hour>,<minute>,<second>	

A.1.5.6 STATus Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
STATus		
:OPERation		
[:EVENt]?	<NR1>	query only
:CONDition?	<NR1>	query only
:ENABle	<int>	
:ENABle?	<NR1>	
:PTRansition	<int>	
:PTRansition?	<NR1>	
:NTRansition	<int>	
:NTRansition?	<NR1>	
:SETTling		
[:EVENt]?	<NR1>	query only
:CONDition?	<NR1>	query only
:ENABle	<int>	
:ENABle?	<NR1>	
:PTRansition	<int>	
:PTRansition?	<NR1>	
:NTRansition	<int>	
:NTRansition?	<NR1>	
:QUEStionable		
[:EVENt]?	<NR1>	query only
:CONDition?	<NR1>	query only
:ENABle	<int>	
:ENABle?	<NR1>	
:PTRansition	<int>	
:PTRansition?	<NR1>	
:NTRansition	<int>	
:NTRansition?	<NR1>	
:CLOSs		
[:EVENt]?	<NR1>	query only
:CONDition?	<NR1>	query only
:ENABle	<int>	
:ENABle?	<NR1>	
:PTRansition	<int>	
:PTRansition?	<NR1>	
:NTRansition	<int>	
:NTRansition?	<NR1>	
:DEVice		
[:EVENt]?	<NR1>	query only
:PRESet	---	no query

A.1.6 SSG Module

A.1.6.1 SOURce10 Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
SOURce10		
:FREQuency		
[:CW]	<int>	
[:CW]?	<NR1>	
:OUTPut		
[:STATe]	<bool>	
[:STATe]?	0|1	
:REFerence	INTernal|EXTernal	
:REFerence?	INT|EXT	
:JMODulation		
[:STATe]	<bool>	
[:STATe]?	0|1	
:FREQuency	<int>	
:FREQuency?	<NR1>	
:AMPLitude	<real>	
:AMPLitude?	<NR2>	

A.1.7 Jitter Tolerance Measurement

A.1.7.1 JTOLerance Subsystems

KEYWORD	PARAMETER FORM	COMMENTS
JTOLerance		
[:STATe]	<bool>	
[:STATe]?	0|1	
:TABLe	<index>,<freq>,<min_ampl>,<max_ampl>,<points>	
:TABLe?	<index>/<freq>,<min_ampl>,<max_ampl>,<points>	
:COUNt?	<NR1>	query only
:NEW	<index>,<freq>,<min_ampl>,<max_ampl>,	
	<points>	no query
:REMove	<index>	no query
:TEMPlate		
:SELect	<str>	
:SELect?	<str>	
:MODE	SEARch|SWEep	
:MODE?	SEAR|SWE	
:CFRequency	<int>	
:CFRequency?	<NR1>	
:STIMe	<real>	
:STIMe?	<NR2>	
:PERiod	<int>	
:PERiod?	<NR1>	
:ETHReshold	<int>	
:ETHReshold?	<NR1>	
:ASEarch	<bool>	
:ASEarch?	0|1	
:MDATa	<bool>	
:MDATa?	0|1	
:ALL		
:PASS?	0|1	query only

A.2 COMMANDS USED ON SETTING MENUS

The following table lists the GPIB commands used on the setting menus of the D3371.

A.2.1 Menus

Menu	Sub menu	Command
File	Open	MMEMory:LOAD:SETup <str> MMEMory:LOAD:SSG:SETup <str> MMEMory:LOAD:PPG:SETup <str> MMEMory:LOAD:PPG:PROGram <str> MMEMory:LOAD:PPG:STM <str> MMEMory:LOAD:PPG:FLEXible <str> MMEMory:LOAD:PPG:FLEXible:PROGram <prog_no>,<str> MMEMory:LOAD:ED:SETup <str> MMEMory:LOAD:ED:PROGram <str> MMEMory:LOAD:ED:STM <str> MMEMory:LOAD:ED:FLEXible <str> MMEMory:LOAD:ED:FLEXible:PROGram <prog_no>,<str> MMEMory:LOAD:JTOLerance <str>
	Save As	MMEMory:STORe:SETup <str> MMEMory:STORe:SSG:SETup <str> MMEMory:STORe:PPG:SETup <str> MMEMory:STORe:PPG:PROGram <str> MMEMory:STORe:PPG:STM <str> MMEMory:STORe:PPG:FLEXible <str> MMEMory:STORe:PPG:FLEXible:PROGram <prog_no>,<str> MMEMory:STORe:ED:SETup <str> MMEMory:STORe:ED:PROGram <str> MMEMory:STORe:ED:STM <str> MMEMory:STORe:ED:FLEXible <str> MMEMory:STORe:ED:FLEXible:PROGram <prog_no>,<str> MMEMory:STORe:ED:BMEasurement:LOG <str> MMEMory:STORe:ED:EPANalysis:TSERies:TEXT <cr>,<sp1>,<sp2>,<sp3>,<nr>,<str> MMEMory:STORe:ED:EPANalysis:STATistics:TEXT <sp1>,<sp2>,<sp3>,<range>,<str> MMEMory:STORe:JTOLerance <str> MMEMory:STORe:JTOLerance:TEXT <str>
	Print	HCOPy[:ITEM]:ED:BMEasurement HCOPy[:ITEM]:JTOLerance

Menu	Sub menu	Command
Measurement	Start	SENSe[1]:BMEasurement[:STATe] ON CTOLerance[:STATe] ON
	Stop	SENSe[1]:BMEasurement[:STATe] OFF CTOLerance[:STATe] OFF
	Auto Search...	SENSe[1]:ASEarch[:STATe] ON
	Output Clock & Data	OUTPut[1][:STATe] <bool> OUTPut2[:STATe] <bool> OUTPut3[:STATe] <bool> OUTPut4[:STATe] <bool>
	Single Error Addition	SOURce[1]:EADDition:SINGle
	ReSync	SENSe[1]:SYNChronisat[:AUTO] ONCE
	Buzzer	SYSTem:BEEPer:ERRor[:STATe] <bool> SYSTem:BEEPer:ALARm[:STATe] <bool>
	Clear History	SENSe[1]:MONitor:HISTory:CLEar
	Set Installation Defaults	*RST

A.2.2 [Settings] Dialog

A.2.2.1 SSG Module

- Frequency

Item		Command
Output		SOURce10:OUTPut[:STATe] <bool>
Frequency		SOURce10:FREQuency[:CW] <int>
Reference		SOURce10:REFerence INTernal EXTernal
Jitter Modulation		SOURce10:JMODulation[:STATe] <bool>
	Frequency	SOURce10:JMODulation:FREQuency <int>
	Amplitude	SOURce10:JMODulation:AMPLitude <real>

A.2.2.2 PPG Module

- Data

Item			Command
Track Data			OUTPut3:TRACk <bool>
Data	Enable DATA Output		OUTPut[1][:STATe] <bool>
	Cross Point		OUTPut[1]:CROSs <int>
	Termination		OUTPut[1]:TERMination P1R3V|GND M2V CML
	Termination Voltage		OUTPut[1]:TERMination:VOLTage <real>
	Amplitude		OUTPut[1]:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
	Offset	High	OUTPut[1]:VOLTage[:LEVel][:IMMediate]:HIGH <real>
		Middle	OUTPut[1]:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
		Low	OUTPut[1]:VOLTage[:LEVel][:IMMediate]:LOW <real>
XData	Enable XDATA Output		OUTPut3[:STATe] <bool>
	Cross Point		OUTPut3:CROSs <int>
	Termination		OUTPut3:TERMination P1R3V GND|M2V|CML
	Termination Voltage		OUTPut3:TERMination:VOLTage <real>
	Amplitude		OUTPut3:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
	Offset	High	OUTPut3:VOLTage[:LEVel][:IMMediate]:HIGH <real>
		Middle	OUTPut3:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
		Low	OUTPut3:VOLTage[:LEVel][:IMMediate]:LOW <real>

- Clock

Item			Command
Clock Delay			SOURce2:DELay <int>
Track Clock			OUTPut4:TRACk <bool>
Clock	Enable CLOCK Output		OUTPut2[:STATe] <bool>
	Termination		OUTPut2:TERMination P1R3V GND|M2V|CML
	Termination Voltage		OUTPut2:TERMination:VOLTage <real>
	Amplitude		OUTPut2:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
	Offset	High	OUTPut2:VOLTage[:LEVel][:IMMediate]:HIGH <real>
		Middle	OUTPut2:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
		Low	OUTPut2:VOLTage[:LEVel][:IMMediate]:LOW <real>
XClock	Enable XCLOCK Output		OUTPut4[:STATe] <bool>
	Termination		OUTPut4:TERMination P1R3V GND|M2V|CML
	Termination Voltage		OUTPut4:TERMination:VOLTage <real>
	Amplitude		OUTPut4:VOLTage[:LEVel][:IMMediate][:AMPLitude] <real>
	Offset	High	OUTPut4:VOLTage[:LEVel][:IMMediate]:HIGH <real>
		Middle	OUTPut4:VOLTage[:LEVel][:IMMediate]:MIDDle <real>
		Low	OUTPut4:VOLTage[:LEVel][:IMMediate]:LOW <real>
Track Clock			OUTPut4[:TRACk] <bool>

- Pattern

Item		Command
Pattern Polarity		SOURce[1]:PATTern:POLarity NORMal|INVerted
Pattern Type		SOURce[1]:PATTern[:SELect] PRBS<n>|ZSUBstitut<n>|PROGram STM<n>|FLEXible
PRBS	Pattern Length	SOURce[1]:PATTern[:SELect] PRBS<n>
	Mark Ratio	SOURce[1]:PATTern:PRBS:MRATio MRATio<n>|MINVerted4
ZSUB	Pattern Length	SOURce[1]:PATTern[:SELect] ZSUBstitut<n>
	Zero Length	SOURce[1]:PATTern:ZSUBstitut[:ZLENgth] <int>
PROG	Pattern Length	SOURce[1]:PATTern:PROGram[:LENGth] <int>
	Pattern	SOURce[1]:PATTern:PROGram:DATA <start_bit>,<length>,<str> SOURce[1]:PATTern:PROGram:BDATa <start_bit>,<length>,<block>

A.2.2 [Settings] Dialog

Item			Command
STM	Insert PRBS into Payload		SOURce[1]:PATTern:STM:PAYLoad:PRBS <bool>
		Pattern Length	SOURce[1]:PATTern:STM:PAYLoad:PRBS:SELect PRBS<n>
		Mark Ratio	SOURce[1]:PATTern:STM:PAYLoad:PRBS:MRATio MRATio<n>|MINVerted4
	Scramble		SOURce[1]:PATTern:STM:SCRamble <bool>
	Insert B1		SOURce[1]:PATTern:STM:B1 <bool>
	Number of Frame		SOURce[1]:PATTern:STM:NFRame <int>
	Pattern		SOURce[1]:PATTern:STM:DATA <start_frm>,<start_row>,<start_col>,<length>,<str> SOURce[1]:PATTern:STM:BDATa <start_frm>,<start_row>,<start_col>,<length>,<block>
FLEX	Pattern Sequence Table	New	SOURce[1]:PATTern:FLEXible:STABle:NEW <index>,<prog_no>|PRBS,<length>,HIGH|LOW
		Edit	SOURce[1]:PATTern:FLEXible:STABle <index>,<prog_no>|PRBS,<length>,HIGH|LOW
		Remove	SOURce[1]:PATTern:FLEXible:STABle:REMove <index>
		Default	SOURce[1]:PATTern:FLEXible:STABle:DEFault
	PROG	Pattern Length	SOURce[1]:PATTern:FLEXible:PROGram[:LENGth] <prog_no>,<length>
		Pattern	SOURce[1]:PATTern:FLEXible:PROGram:DATA <prog_no>,<start_bit>,<length>,<str> SOURce[1]:PATTern:FLEXible:PROGram:BDATa <prog_no>,<start_bit>,<length>,<block>
	PRBS	Pattern Length	SOURce[1]:PATTern:FLEXible:PRBS:SELect PRBS<n>
		Mark Ratio	SOURce[1]:PATTern:FLEXible:PRBS:MRATio MRATio<n>|MINVerted4

- Burst

Item		Command
Burst Mode		SOURce5:BOUTput[:STATe] <bool>
Source		SOURce5:BOUTput:SOURce INTernal EXTernal
Internal	Cycle	SOURce5:BOUTput:CYCLe <int>
	OFF Time	SOURce5:BOUTput:OTIMe <int>

- Error Addition

Item			Command
Addition Route			SOURce[1]:EADDition:ROUTe <int>
Add Errors			SOURce[1]:EADDition <bool>
	Single		SOURce[1]:EADDition:MODE SINGle
	Repeat		SOURce[1]:EADDition:MODE REPeat
		Rate	SOURce[1]:EADDition:RATE <real>
	External		SOURce[1]:EADDition:MODE EXTernal

- Trigger/AUX

Item			Command
Trigger Output			SOURce6:TRIGger[1]:MODE[1] CLOCk8|CLOCk32|PATTern FLEXible
Position	PRBS		SOURce6:TRIGger:PRBS:POSition <int>
	PROG		SOURce6:TRIGger:PROGram:POSition <int>
	ZSUB		SOURce6:TRIGger:ZSUBstitut:POSition <int>
	STM	Pattern/Frame	SOURce6:TRIGger:STM:MODE PATTern|FRAMe
		Frame No.	SOURce6:TRIGger:STM:FNUMber <int>
		Row	SOURce6:TRIGger:STM:ROW <int>
		Column	SOURce6:TRIGger:STM:COLumn <int>
	FLEX	Index	SOURce6:TRIGger:FLEXible:INDex <int>
		Address	SOURce6:TRIGger:FLEXible:ADDRess <int>

A.2.2.3 ED Module

- Data

Item	Command
Data Polarity	INPut[1]:POLarity NORMal INVerted
Termination	INPut[1]:TERMination P3V P1R3V|GND M2V CML
Variable	INPut[1]:TERMination:VARiable <bool>
Termination Voltage	INPut[1]:TERMination:VOLTage <real>
Threshold Voltage	INPut[1]:TVOLtage <real>

- Clock

Item	Command
Clock Delay	INPut2:DELay <int>
Clock Polarity	INPut2:POLarity NORMal|INVerted
Termination	INPut2:TERMination P3V|P1R3V GND|M2V|CML
Variable	INPut2:TERMination:VARiable <bool>
Termination Voltage	INPut2:TERMination:VOLTage <real>

- Pattern

Item			Command
Use the same Pattern as PPG			SENSe[1]:PATTern:TRACk <bool>
Pattern Polarity			SENSe[1]:PATTern:POLarity NORMal INVerted
Pattern Type			SENSe[1]:PATTern[:SELect] PRBS<n>|ZSUBstitut<n> PROGram|STM<n> FLEXible
PRBS	Pattern Length		SENSe[1]:PATTern[:SELect] PRBS<n>
	Mark Ratio		SENSe[1]:PATTern:PRBS:MRATio MRATio<n>|MINVerted4
ZSUB	Pattern Length		SENSe[1]:PATTern[:SELect] ZSUBstitut<n>
	Zero Length		SENSe[1]:PATTern:ZSUBstitut[:ZLENgth] <int>
PROG	Pattern Length		SENSe[1]:PATTern:PROGram[:LENGth] <int>
	Pattern		SENSe[1]:PATTern:PROGram:DATA <start_bit>,<length>,<str> SENSe[1]:PATTern:PROGram:BDATa <start_bit>,<length>,<block>
STM	Insert PRBS into Payload		SENSe[1]:PATTern:STM:PAYLoad:PRBS <bool>
		Pattern Length	SENSe[1]:PATTern:STM:PAYLoad:PRBS:SELect PRBS<n>
		Mark Ratio	SENSe[1]:PATTern:STM:PAYLoad:PRBS:MRATio MRATio<n> MINVerted4
	Scramble		SENSe[1]:PATTern:STM:SCRamble <bool>
	Insert B1		SENSe[1]:PATTern:STM:B1 <bool>
	Number of Frame		SENSe[1]:PATTern:STM:NFRame <int>
	Pattern		SENSe[1]:PATTern:STM:DATA <start_frm>,<start_row>,<start_col>,<length>,<str> SENSe[1]:PATTern:STM:BDATa <start_frm>,<start_row>,<start_col>,<length>,<block>
FLEX	Pattern Sequence Table	New	SENSe[1]:PATTern:FLEXible:STABle:NEW <index>,<prog_no>|PRBS,<length>,HIGH|LOW
		Edit	SENSe[1]:PATTern:FLEXible:STABle <index>,<prog_no>|PRBS,<length>,HIGH|LOW
		Remove	SENSe[1]:PATTern:FLEXible:STABle:REMove <index>
		Default	SENSe[1]:PATTern:FLEXible:STABle:DEFault
	PROG	Pattern Length	SENSe[1]:PATTern:FLEXible:PROGram[:LENGth] <prog_no>,<length>
		Pattern	SENSe[1]:PATTern:FLEXible:PROGram:DATA <prog_no>,<start_bit>,<length>,<str> SENSe[1]:PATTern:FLEXible:PROGram:BDATa <prog_no>,<start_bit>,<length>,<block>
	PRBS	Pattern Length	SENSe[1]:PATTern:FLEXible:PRBS:SELect PRBS<n>
		Mark Ratio	SENSe[1]:PATTern:FLEXible:PRBS:MRATio MRATio<n> MINVerted4

- Sync

Item		Command
Auto Sync		SENSe[1]:SYNChronisat[:AUTO] <bool>
Sync Pattern Length		SENSe[1]:SYNChronisat:PATTern:LENGth <int>
Sync PROG Address		SENSe[1]:SYNChronisat:PATTern:PROGram:ADDRess <int>
Sync FLEX Index		SENSe[1]:SYNChronisat:PATTern:FLEXible:INDex <int>
Sync FLEX Address		SENSe[1]:SYNChronisat:PATTern:FLEXible:ADDRess <int>
Auto Threshold		SENSe[1]:SYNChronisat:THReshold:AUTO <bool>
Sync Gain Threshold	PRBS	SENSe[1]:SYNChronisat:THReshold:PRBS:GAIN <real>
	Memory	SENSe[1]:SYNChronisat:THReshold:MEMory:GAIN <real>
Sync Loss Threshold	PRBS	SENSe[1]:SYNChronisat:THReshold:PRBS:LOSS <real>
	Memory	SENSe[1]:SYNChronisat:THReshold:MEMory:LOSS <real>

- Auto Search

Item	Command
Clock Polarity & Delay	SENSe[1]:ASEarch:CLOCk <bool>
Threshold Voltage	SENSe[1]:ASEarch:TVOLtage <bool>
Pattern	SENSe[1]:ASEarch:PATTern <bool>

- Mask

Item	Command
Mask Route	SENSe[1]:MASK:ROUTe<n> <bool>

- Trigger/AUX

Item		Command
Trigger Output		SOURce8:TRIGger[:MODE] CLOCk16|PATTern FLEXible
	STM	SOURce8:TRIGger:STM:MODE PATTern|FRAMe
AUX		SOURce9:AUX[:MODE] DTYPe SYNChronisat

- Log

Item		Command
Basic Measurement Results		SENSe[1]:LOG:BMEasurement <bool>
Sync & Clock Loss History		SENSe[1]:LOG:HISTory:SCLoss <bool>
Measurement Data History		
	Frequency	SENSe[1]:LOG:HISTory:MDATa:FREQuency <bool>
	Immediate Error Rate	SENSe[1]:LOG:HISTory:MDATa:IERate <bool>
	Immediate Error Count	SENSe[1]:LOG:HISTory:MDATa:IECount <bool>
	Error Intervals	SENSe[1]:LOG:HISTory:MDATa:EINTervals <bool>
	Error Free Intervals	SENSe[1]:LOG:HISTory:MDATa:EFINtervals <bool>

- Condition

Item		Command
Timer Mode		SENSe[1]:BMEasurement:MTIMe:MODE SINGle|REPeat|UNTimed
Interval		SENSe[1]:BMEasurement:MTIMe:INTerval <real>
Error Performance Threshold		SENSe[1]:BMEasurement:EPERformance:THReshold <us_ses>,<dm>
Period		SENSe[1]:BMEasurement:MTIMe:PERiod <day>,<hour>,<minute>,<second>
Detection Mode		SENSe[1]:BMEasurement:DMODe OIT OPT SOT
Burst Mode		SENSe[1]:BMEasurement:BURSt <bool>
Current Data		SENSe[1]:BMEasurement:CDATa <bool>
Evaluate Clock Loss Intervals		SENSe[1]:BMEasurement:EVALuate:CLOSs <bool>
Evaluate Sync Loss Intervals		SENSe[1]:BMEasurement:EVALuate:SLOSs <bool>
Error Record		SENSe[1]:EPANalysis:ERECord[:STATe] <bool>
	Specific/Total	SENSe[1]:EPANalysis:ERECord:AREa SPECific TOTal

- Display Format

Item	Command
Bit Count	DISPlay:WINDow:BCOunt:DFORmat INTegral EXPonential
Error Count	DISPlay:WINDow:ECOunt:DFORmat INTegral EXPonential
Immediate Error Count	DISPlay:WINDow:IECount:DFORmat INTegral|EXPonential
Error Intervals	DISPlay:WINDow:EINTervals:DFORmat INTegral|PERCent
Error Free Intervals	DISPlay:WINDow:EFINtervals:DFORmat INTegral|PERCent
Threshold EI	DISPlay:WINDow:TEINtervals:DFORmat INTegral|PERCent
Threshold EFI	DISPlay:WINDow:TEFIntervals:DFORmat INTegral PERCent
Error Performance	DISPlay:WINDow:EPERformance:DFORmat INTegral PERCent

- Buzzer

Item	Command
Error	SYSTem:BEEPer:ERRor[:STATe] <bool>
Alarm	SYSTem:BEEPer:ALARm[:STATe] <bool>

- Jitter Tolerance

Item	Command
New	JTOLerance:TABLe:NEW <index>,<freq>,<min_ampl>,<max_ampl>,<points>
Edit	JTOLerance:TABLe <index>,<freq>,<min_ampl>,<max_ampl>,<points>
Remove	JTOLerance:TABLe:REMove <index>
Measurement Mode	JTOLerance:MODE SEARch|SWEep
Display Measurement Data	JTOLerance:MDATa <bool>
Template Type	JTOLerance:TEMPlate:SELect <str>
Auto Search	JTOLerance:ASEarch <bool>
Clock Frequency	JTOLerance:CFRequency <int>
Settling Time	JTOLerance:STIMe <real>
Period	JTOLerance:PERiod <int>
Error Threshold	JTOLerance:ETHReshold <int>

- Specific Field

Item		Command
PROG	Start Address	SENSe[1]:SPECific:PROGram:STARt:ADDRess <int>
	End Address	SENSe[1]:SPECific:PROGram:END:ADDRess <int>
ZSUB	Start Address	SENSe[1]:SPECific:ZSUBstitut:STARt:ADDRess <int>
	End Address	SENSe[1]:SPECific:ZSUBstitut:END:ADDRess <int>
STM	Pattern/Frame	SENSe[1]:SPECific:STM:MODE PATTern FRAMe
	Start Frame No.	SENSe[1]:SPECific:STM:STARt:FNUMber <int>
	Start Row	SENSe[1]:SPECific:STM:STARt:ROW <int>
	Start Column	SENSe[1]:SPECific:STM:STARt:COLumn <int>
	End Frame No.	SENSe[1]:SPECific:STM:END:FNUMber <int>
	End Row	SENSe[1]:SPECific:STM:END:ROW <int>
	End Column	SENSe[1]:SPECific:STM:END:COLumn <int>
FLEX	Start Index	SENSe[1]:SPECific:FLEXible:STARt:INDex <int>
	Start Address	SENSe[1]:SPECific:FLEXible:STARt:ADDRess <int>
	End Index	SENSe[1]:SPECific:FLEXible:END:INDex <int>
	End Address	SENSe[1]:SPECific:FLEXible:END:ADDRess <int>

A.2.2.4 System Module

- Utility

Item	Command
Date	SYSTem:DATE <year>,<month>,<day>
Time	SYSTem:TIME <hour>,<minute>,<second>

A.3 DEFAULTS

A.3.1 PPG Module

Item	Default value
Pattern type	PRBS15
PRBS pattern mark ratio	1/2
ZSUB pattern length	2^7
Consecutive zero bit length	7
PROG pattern length	16
PROG pattern	1010 1010 1010 1010 (Binary)
Setting the function to insert a PRBS pattern into STM-pattern payload to on or off	OFF
The PRBS pattern length to be inserted into STM-pattern payload	PRBS15
The mark ratio of a PRBS pattern to be inserted into STM-pattern payload	1/2
Setting the function to scramble STM pattern to on or off	OFF
Setting the function to insert B1 bytes in STM pattern to on or off	OFF
STM pattern length	STM-16, 1 frame
STM pattern	Refer to "A.3.5 STM Pattern."
The number of patterns in the FLEX pattern sequence table	1
FLEX pattern sequence table	Refer to "A.3.6 FLEX Pattern."
PROG pattern length used in a FLEX pattern	128 bits for all patterns
PROG pattern used in a FLEX pattern	Refer to "A.3.6 FLEX Pattern."
PRBS pattern length used in a FLEX pattern	PRBS15
Mark ratio of PRBS pattern used in a FLEX pattern	1/2
Pattern polarity	Positive logic (NORMal)
Setting error addition to on or off	ON
Error addition mode	Single mode (SINGle)
Error addition rate	1E-8
Error addition route	1
Setting dATA output to on or off	OFF
DATA output amplitude (LVPECL termination)	0.80 V_{P-P}
DATA output amplitude (GND termination)	1.00 V_{P-P}
DATA output amplitude (ECL termination)	0.80 V_{P-P}

Item	Default value
DATA output amplitude (CML termination)	0.40 V_{P-P}
DATA output high-level offset value (LVPECL termination)	2.40 V
DATA output intermediate level offset value (LVPECL termination)	2.00 V
DATA output low-level offset value (LVPECL termination)	1.60 V
DATA output high-level offset value (GND termination)	0.00 V
DATA output intermediate level offset value (GND termination)	-0.50
DATA output low-level offset value (GND termination)	-1.00
DATA output high-level offset value (ECL termination)	-0.90 V
DATA output intermediate level offset value (ECL termination)	-1.30 V
DATA output low-level offset value (ECL termination)	-1.70 V
DATA output high-level offset value (CML termination)	0.00 V
DATA output intermediate level offset value (CML termination)	-0.20 V
DATA output low-level offset value (CML termination)	-0.40 V
DATA output cross point	50%
DATA output termination type	GND (0 V)
DATA output termination voltage (CML termination)	0.00 V
Setting $\overline{\text{DATA}}$ output to on or off	OFF
$\overline{\text{DATA}}$ output amplitude (LVPECL termination)	0.80 V_{P-P}
$\overline{\text{DATA}}$ output amplitude (GND termination)	1.00 V_{P-P}
$\overline{\text{DATA}}$ output amplitude (ECL termination)	0.80 V_{P-P}
$\overline{\text{DATA}}$ output amplitude (CML termination)	0.40 V_{P-P}
$\overline{\text{DATA}}$ output high-level offset (LVPECL termination)	2.40 V
$\overline{\text{DATA}}$ output intermediate level offset value (LVPECL termination)	2.00 V
$\overline{\text{DATA}}$ output low-level offset value (LVPECL termination)	1.60 V
$\overline{\text{DATA}}$ output high-level offset value (GND termination)	0.00 V
$\overline{\text{DATA}}$ output intermediate level offset value (GND termination)	-0.50 V
$\overline{\text{DATA}}$ output low-level offset value (GND termination)	-1.00 V
$\overline{\text{DATA}}$ output high-level offset value (ECL termination)	-0.90 V
$\overline{\text{DATA}}$ output intermediate level offset value (ECL termination)	-1.30 V
$\overline{\text{DATA}}$ output low-level offset value (ECL termination)	-1.70 V
$\overline{\text{DATA}}$ output high-level offset value (CML termination)	0.00 V
$\overline{\text{DATA}}$ output intermediate level offset value (CML termination)	-0.20 V
$\overline{\text{DATA}}$ output low-level offset value (CML termination)	-0.40 V

Item	Default value
$\overline{\text{DATA}}$ output cross point	50%
$\overline{\text{DATA}}$ output termination type	GND (0 V)
$\overline{\text{DATA}}$ output termination voltage (CML termination)	0.00 V
Setting data output tracking function to on or off	OFF
Clock delay	Not initialized
Setting CLOCK output to on or off	OFF
CLOCK output amplitude (LVPECL termination)	0.80 V_{P-P}
CLOCK output amplitude (GND termination)	1.00 V_{P-P}
CLOCK output amplitude (ECL termination)	0.80 V_{P-P}
CLOCK output amplitude (CML termination)	0.40 V_{P-P}
CLOCK output high-level offset value (LVPECL termination)	2.40 V
CLOCK output intermediate level offset value (LVPECL termination)	2.00 V
CLOCK output low-level offset value (LVPECL termination)	1.60 V
CLOCK output high-level offset value (GND termination)	0.00 V
CLOCK output intermediate level offset value (GND termination)	-0.50
CLOCK output low-level offset value (GND termination)	-1.00
CLOCK output high-level offset value (ECL termination)	-0.90 V
CLOCK output intermediate level offset value (ECL termination)	-1.30 V
CLOCK output low-level offset value (ECL termination)	-1.70 V
CLOCK output low-level offset value(CML termination)	0.00 V
CLOCK output intermediate level offset value (CML termination)	-0.20 V
CLOCK output low-level offset value (CML termination)	-0.40 V
CLOCK output termination type	GND (0 V)
CLOCK output termination voltage (CML termination)	0.00 V
Setting $\overline{\text{CLOCK}}$ output to on or off	OFF
$\overline{\text{CLOCK}}$ output amplitude (LVPECL termination)	0.80 V_{P-P}
$\overline{\text{CLOCK}}$ output amplitude (GND termination)	1.00 V_{P-P}
$\overline{\text{CLOCK}}$ output amplitude (ECL termination)	0.80 V_{P-P}
$\overline{\text{CLOCK}}$ output amplitude (CML termination)	0.40 V_{P-P}
$\overline{\text{CLOCK}}$ output high-level offset value (LVPECL termination)	2.40 V
$\overline{\text{CLOCK}}$ output intermediate level offset value (LVPECL termination)	2.00 V
$\overline{\text{CLOCK}}$ output low-level offset value (LVPECL termination)	1.60 V

Item	Default value
$\overline{\text{CLOCK}}$ output high-level offset value (GND termination)	0.00 V
$\overline{\text{CLOCK}}$ output intermediate level offset value (GND termination)	-0.50 V
$\overline{\text{CLOCK}}$ output low-level offset value (GND termination)	-1.00 V
$\overline{\text{CLOCK}}$ output high-level offset value (ECL termination)	-0.90 V
$\overline{\text{CLOCK}}$ output intermediate level offset value (ECL termination)	-1.30 V
$\overline{\text{CLOCK}}$ output low-level offset value (ECL termination)	-1.70 V
$\overline{\text{CLOCK}}$ output high-level offset value (CML termination)	0.00 V
$\overline{\text{CLOCK}}$ output intermediate level offset value (CML termination)	-0.20 V
$\overline{\text{CLOCK}}$ output low-level offset value (CML termination)	-0.40 V
$\overline{\text{CLOCK}}$ output termination type	GND (0 V)
$\overline{\text{CLOCK}}$ output termination voltage (CML termination)	0.00 V
Setting clock output tracking function to on or off	OFF
Setting Burst Mode output to on or off	OFF
BURST OUTPUT source	Internal signal (INTernal)
Burst cycle	1000 µs
OFF Time in burst cycle	500 µs
Trigger output source	1/8 frequency clock of the output source clock (CLOCk8)
Trigger signal output position in the PRBS pattern	0 bit
Trigger signal output position in the ZSUB pattern	0 bit
Trigger signal output position in the PROG pattern	0 bit
Trigger output mode in the STM patterns	Pattern mode
Trigger output frame number in the STM patterns	1
Trigger output row in the STM patterns	1
Trigger output column in the STM patterns	1 byte
Trigger output index in the FLEX patterns	1
Trigger output address in the FLEX patterns	0 bit

A.3.2 ED Module

Item	Default value
Pattern type	PRBS15
PRBS pattern mark ratio	1/2
ZSUB pattern length	2^7 bits
Consecutive zero bit length	7 bits
PROG pattern length	16 bits
PROG pattern	1010 1010 1010 1010
Setting the function to insert a PRBS pattern into STM-pattern payloads to on or off	OFF
The PRBS pattern length to be inserted into the STM-pattern payload	PRBS15
The mark ratio of a PRBS pattern to be inserted into the STM-pattern payload	1/2
Setting the function to scramble STM patterns to on or off	OFF
Setting the function to insert B1 byte in STM patterns to on or off	OFF
STM pattern length	STM-16, 1 frame
STM patterns	Refer to "A.3.5 STM Pattern."
The number of patterns in the FLEX pattern sequence table	1
FLEX pattern sequence table	Refer to "A.3.6 FLEX Pattern."
PROG pattern length used in a FLEX pattern	128 bits for all patterns
PROG pattern used in a FLEX pattern	Refer to "A.3.6 FLEX Pattern."
PRBS pattern length used in a FLEX pattern	PRBS15
Mark ratio of PRBS pattern used in a FLEX pattern	1/2
Pattern polarity	Positive logic (NORMal)
Setting the same pattern set for the PPG module to on or off	OFF
Automatic pattern synchronization	ON
Synchronization pattern address for PROG pattern	0 bit
Synchronization pattern length	32 bits
Synchronization pattern index in the FLEX pattern	1 bit
Synchronization pattern address in the FLEX pattern	0 bit
Setting the automatic error rate threshold setting function used for synchronization judgments to on or off	ON
Sync Gain Threshold in the PRBS pattern	1E-2
Sync Loss Threshold in the PRBS pattern	1E-2
Sync Gain Threshold in the ZSUB or PROG pattern	1E-4

Item	Default value
Sync Loss Threshold in the ZSUB or PROG pattern	1E-4
Setting measurement route mask to on or off	OFF for all routes
Setting the function that automatically searches for clock input delay and input polarity to on or off	ON
Setting the data input threshold voltage auto-search function to on or off	ON
Setting the pattern setting auto-search function to on or off	ON
Setting the log function for basic measurement results to on or off	ON
Setting the history record function for sync loss and clock loss to on or off	ON
Setting the history record function for immediate error rate value to on or off	ON
Setting the history record function for immediate error count value to on or off	ON
Setting the history record function for error interval value to on or off	ON
Setting the history record function for error-free interval value to on or off	ON
Setting the history record function for frequency value to on or off	ON
Measurement mode	Untimed mode (UNTimed)
Measurement interval	0.1 s
Measurement period	10 s
Error detection mode	Omitting errors / Inserting errors / Total errors
Setting the BURST mode to on or off	OFF
Selecting whether or not to include clock loss interval in measurement	ON
Selecting whether or not to include sync loss interval in measurement	ON
Error rate threshold for period included in US/SES, DM calculation	US/SES:1E-3, DM:1E-6
Setting the function to display results in progress to on or off	ON
Setting the error-recording function to on or off	ON
Error-recording area	The whole pattern
The number of lists in the parameter table for jitter tolerance	15
Parameter table for jitter tolerance	Refer to "A.3.7 Parameter table for jitter tolerance."
Jitter tolerance-measuring mode	Search mode
Setting the displaying of jitter tolerance measurement to on or off	ON

Item	Default value
Templates to measure jitter tolerance	STM-16 (A)
Setting the auto-search function to measure jitter tolerance to on or off	ON
Clock frequency to measure jitter tolerance	2,488,320 kHz
Settling time to measure jitter tolerance	0.0 s
One-point measurement time in jitter tolerance measurement	1 s
Error counts providing basis for evaluation of jitter tolerance measurements	1 bit
Start address of a specific area for PROG pattern	0 bit
Stop address of a specific area for PROG pattern	0 bit
Start address of a specific area for ZSUB pattern	0 bit
Stop address of a specific area for ZSUB pattern	0 bit
Specific area measurement mode for STM pattern	Pattern mode
Start frame number of a specific area for STM patterns	1
Start row of a specific area for STM patterns	1
Start column of a specific area for STM patterns	1 byte
Stop frame number of a specific area for STM patterns	1
Stop row of a specific area for STM patterns	1
Stop column of a specific area for STM patterns	1 byte
Start index of a specific area for FLEX patterns	1
Start address of a specific area for FLEX patterns	0 bit
Stop index of a specific area for FLEX patterns	1
Stop address of a specific area for FLEX patterns	0 bit
DATA input polarity	Normal (NORMal)
DATA threshold voltage (PECL termination)	3.700 V
DATA threshold voltage (LVPECL termination)	2.000 V
DATA threshold voltage (GND termination)	-0.500 V
DATA threshold voltage (ECL termination)	-1.300 V
DATA threshold voltage (CML termination)	-0.200 V
DATA input termination type	GND (0 V)
Setting the DATA input termination voltage-variable function to on or off (PECL termination)	OFF
Setting the DATA input termination voltage-variable function to on or off (LVPECL termination)	OFF
Setting the DATA input termination voltage-variable function to on or off (ECL termination)	OFF

Item	Default value
Setting the DATA input termination voltage-variable function to on or off (CML termination)	OFF
DATA input termination voltage (PECL termination)	+3.00 V
DATA input termination voltage (LVPECL termination)	+1.30 V
DATA input termination voltage (ECL termination)	-2.00 V
DATA input termination voltage (CML termination)	0.00 V
CLOCK input polarity	Normal (NORMal)
CLOCK delay	Not initialized
CLOCK termination voltage	GND (0 V)
Setting the CLOCK input termination voltage-variable function to on or off (PECL termination)	OFF
Setting the CLOCK input termination voltage-variable function to on or off (LVPECL termination)	OFF
Setting the CLOCK input termination voltage-variable function to on or off (ECL termination)	OFF
Setting the CLOCK input termination voltage-variable function to on or off (CML termination)	OFF
CLOCK input termination voltage (PECL termination)	+3.00 V
CLOCK input termination voltage (LVPECL termination)	+1.30 V
CLOCK input termination voltage (ECL termination)	-2.00 V
CLOCK input termination voltage (CML termination)	0.00 V
Trigger output source	1/16 frequency clock of the input source clock (CLOCk16)
Trigger output mode in handling STM patterns	Pattern mode
AUX output mode	Data type

A.3.3 Results and System

Item	Default value
Bit count display format	Exponential format (EXPonential)
Error count display format	Exponential format (EXPonential)
Immediate error count value display format	Exponential format (EXPonential)
Error interval value display format	Percentage format (PERCent)
Error-free interval value display format	Percentage format (PERCent)
Threshold EI value display format	Percentage format (PERCent)
Threshold EFI value display format	Percentage format (PERCent)
Error performance value display format	Percentage format (PERCent)
Turningaudible alarm on or off in the event of a data error	OFF
Turning audible alarm on or offin the event of an alarm	OFF
Remote interface	Not initialized
GPIB address	Not initialized

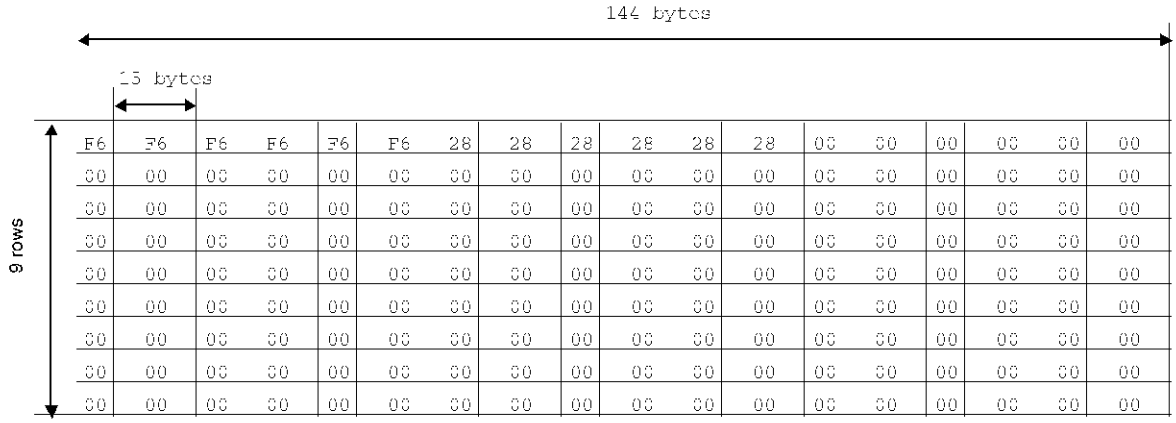
A.3.4 SSG Module

Item	Default value
Turning the clock output on or off	ON
Frequency	2,500,000 kHz
Clock reference signal input source	Internal signal (INTernal)
Setting the jitter modulation function to on or off	OFF
Jitter frequency	10 Hz
Jitter amplitude	0.00 UI_{P-P}

A.3.5 STM Patterns

An STM pattern is initialized with one of the STM-16 frame patterns shown below:

A.3.5.1 Overhead



A.3.5.2 Payload

The number of rows	9
The number of columns	4176 bytes
Pattern	A pattern where the binary data pattern 01010101 repeatedly appears

A.3.6 FLEX Patterns

A.3.6.1 Pattern Sequence Table

Index	Pattern	Length	FLEX Trigger
1	PROG001	128 bits	Low

A.3.6.2 PROG Patterns

All PROG patterns between PROG No.1 and No.127 used in FLEX are initialized with the following pattern:

Pattern No.	Pattern Length	Pattern
1to 127	128 bits	The pattern where the binary data pattern 10101010 repeatedly appears

A.3.7 Parameter Table for Jitter Tolerance

Index	Jitter frequency	Jitter amplitude		
		Minimum	Maximum	The number of measurement points
1	10 Hz	1.50 UI_{P-P}	800.00 UI_{P-P}	5
2	100 Hz	1.50 UI_{P-P}	100.00 UI_{P-P}	5
3	1,000 Hz	1.50 UI_{P-P}	50.00 UI_{P-P}	5
4	5,000 Hz	1.50 UI_{P-P}	50.00 UI_{P-P}	5
5	10,000 Hz	1.50 UI_{P-P}	20.00 UI_{P-P}	5
6	20,000 Hz	1.50 UI_{P-P}	20.00 UI_{P-P}	5
7	50,000 Hz	1.50 UI_{P-P}	20.00 UI_{P-P}	5
8	100,000 Hz	1.50 UI_{P-P}	20.00 UI_{P-P}	10
9	200,000 Hz	0.75 UI_{P-P}	20.00 UI_{P-P}	10
10	500,000 Hz	0.30 UI_{P-P}	12.00 UI_{P-P}	10
11	1,000,000 Hz	0.15 UI_{P-P}	6.00 UI_{P-P}	10
12	2,000,000 Hz	0.15 UI_{P-P}	3.00 UI_{P-P}	10
13	5,000,000 Hz	0.15 UI_{P-P}	1.20 UI_{P-P}	10
14	10,000,000 Hz	0.15 UI_{P-P}	0.60 UI_{P-P}	5
15	20,000,000 Hz	0.15 UI_{P-P}	0.30 UI_{P-P}	5

ALPHABETICAL INDEX

[T]

[W]

IMPORTANT INFORMATION FOR ADVANTEST SOFTWARE

PLEASE READ CAREFULLY: This is an important notice for the software defined herein. Computer programs including any additions, modifications and updates thereof, operation manuals, and related materials provided by Advantest (hereafter referred to as "SOFTWARE"), included in or used with hardware produced by Advantest (hereafter referred to as "PRODUCTS").

SOFTWARE License

All rights in and to the SOFTWARE (including, but not limited to, copyright) shall be and remain vested in Advantest. Advantest hereby grants you a license to use the SOFTWARE only on or with Advantest PRODUCTS.

Restrictions

(1) You may not use the SOFTWARE for any purpose other than for the use of the PRODUCTS.
(2) You may not copy, modify, or change, all or any part of, the SOFTWARE without permission from Advantest.
(3) You may not reverse engineer, de-compile, or disassemble, all or any part of, the SOFTWARE.

Liability

Advantest shall have no liability (1) for any PRODUCT failures, which may arise out of any misuse (misuse is deemed to be use of the SOFTWARE for purposes other than it's intended use) of the SOFTWARE. (2) For any dispute between you and any third party for any reason whatsoever including, but not limited to, infringement of intellectual property rights.

LIMITED WARRANTY

1. Unless otherwise specifically agreed by Seller and Purchaser in writing, Advantest will warrant to the Purchaser that during the Warranty Period this Product (other than consumables included in the Product) will be free from defects in material and workmanship and shall conform to the specifications set forth in this Operation Manual.

2. The warranty period for the Product (the "Warranty Period") will be a period of one year commencing on the delivery date of the Product.

3. If the Product is found to be defective during the Warranty Period, Advantest will, at its option and in its sole and absolute discretion, either (a) repair the defective Product or part or component thereof or (b) replace the defective Product or part or component thereof, in either case at Advantest's sole cost and expense.

4. This limited warranty will not apply to defects or damage to the Product or any part or component thereof resulting from any of the following:

   (a) any modifications, maintenance or repairs other than modifications, maintenance or repairs (i) performed by Advantest or (ii) specifically recommended or authorized by Advantest and performed in accordance with Advantest 's instructions;

   (b) any improper or inadequate handling, carriage or storage of the Product by the Purchaser or any third party (other than Advantest or its agents);

   (c) use of the Product under operating conditions or environments different than those specified in the Operation Manual or recommended by Advantest, including, without limitation, (i) instances where the Product has been subjected to physical stress or electrical voltage exceeding the permissible range and (ii) instances where the corrosion of electrical circuits or other deterioration was accelerated by exposure to corrosive gases or dusty environments;

   (d) use of the Product in connection with software, interfaces, products or parts other than software, interfaces, products or parts supplied or recommended by Advantest;

   (e) incorporation in the Product of any parts or components (i) provided by Purchaser or (ii) provided by a third party at the request or direction of Purchaser or due to specifications or designs supplied by Purchaser (including, without limitation, any degradation in performance of such parts or components);

   (f) Advantest's incorporation or use of any specifications or designs supplied by Purchaser;

   (g) the occurrence of an event of force majeure, including, without limitation, fire, explosion, geological change, storm, flood, earthquake, tidal wave, lightning or act of war; or

   (h) any negligent act or omission of the Purchaser or any third party other than Advantest.

5. **EXCEPT TO THE EXTENT EXPRESSLY PROVIDED HEREIN, ADVANTEST HEREBY EXPRESSLY DISCLAIMS, AND THE PURCHASER HEREBY WAIVES, ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, (A) ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND (B) ANY WARRANTY OR REPRESENTATION AS TO THE VALIDITY, SCOPE, EFFECTIVENESS OR USEFULNESS OF ANY TECHNOLOGY OR ANY INVENTION.**

6. **THE REMEDY SET FORTH HEREIN SHALL BE THE SOLE AND EXCLUSIVE REMEDY OF THE PURCHASER FOR BREACH OF WARRANTY WITH RESPECT TO THE PRODUCT.**

7. **ADVANTEST WILL NOT HAVE ANY LIABILITY TO THE PURCHASER FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, INCLUDING, WITHOUT LIMITATION, LOSS OF ANTICIPATED PROFITS OR REVENUES, IN ANY AND ALL CIRCUMSTANCES, EVEN IF ADVANTEST HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND WHETHER ARISING OUT OF BREACH OF CONTRACT, WARRANTY, TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE. TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE.**

8. **OTHER THAN THE REMEDY FOR THE BREACH OF WARRANTY SET FORTH HEREIN, ADVANTEST SHALL NOT BE LIABLE FOR, AND HEREBY DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ANY LIABILITY FOR, DAMAGES FOR PRODUCT FAILURE OR DEFECT, WHETHER ARISING OUT OF BREACH OF CONTRACT, TORT (INCLUDING, WITHOUT LIMITATION, NEGLEGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE.**

CUSTOMER SERVICE DESCRIPTION

In order to maintain safe and trouble-free operation of the Product and to prevent the incurrence of unnecessary costs and expenses, Advantest recommends a regular preventive maintenance program under its maintenance agreement.

Advantest's maintenance agreement provides the Purchaser on-site and off-site maintenance, parts, maintenance machinery, regular inspections, and telephone support and will last a maximum of ten years from the date the delivery of the Product. For specific details of the services provided under the maintenance agreement, please contact the nearest Advantest office listed at the end of this Operation Manual or Advantest 's sales representatives.

Some of the components and parts of this Product have a limited operating life (such as, electrical and mechanical parts, fan motors, unit power supply, etc.). Accordingly, these components and parts will have to be replaced on a periodic basis. If the operating life of a component or part has expired and such component or part has not been replaced, there is a possibility that the Product will not perform properly. Additionally, if the operating life of a component or part has expired and continued use of such component or part damages the Product, the Product may not be repairable. Please contact the nearest Advantest office listed at the end of this Operation Manual or Advantest's sales representatives to determine the operating life of a specific component or part, as the operating life may vary depending on various factors such as operating condition and usage environment.

SALES & SUPPORT OFFICES

Advantest Korea Co., Ltd.
22BF, Kyobo KangNam Tower,
1303-22, Seocho-Dong, Seocho-Ku, Seoul #137-070, Korea
Phone: +82-2-532-7071
Fax: +82-2-532-7132

Advantest (Suzhou) Co., Ltd.
Shanghai Branch Office:
Bldg. 6D, NO.1188 Gumei Road, Shanghai, China 201102 P.R.C.
Phone: +86-21-6485-2725
Fax: +86-21-6485-2726

Shanghai Branch Office:
406/F, Ying Building, Quantum Plaza, No. 23 Zhi Chun Road,
Hai Dian District, Beijing,
China 100083
Phone: +86-10-8235-3377
Fax: +86-10-8235-6717

Advantest (Singapore) Pte. Ltd.
438A Alexandra Road, #08-03/06
Alexandra Technopark Singapore 119967
Phone: +65-6274-3100
Fax: +65-6274-4055

Advantest America, Inc.
3201 Scott Boulevard, Suite, Santa Clara, CA 95054, U.S.A
Phone: +1-408-988-7700
Fax: +1-408-987-0691

ROHDE & SCHWARZ Europe GmbH
Mühldorfstraße 15 D-81671 München, Germany
(P.O.B. 80 14 60 D-81614 München, Germany)
Phone: +49-89-4129-13711
Fax: +49-89-4129-13723

ADVANTEST

http://www.advantest.co.jp

ADVANTEST CORPORATION
Shin-Marunouchi Center Building, 1-6-2 Marunouchi, Chiyoda-ku, Tokyo 100-0005, Japan
Phone: +81-3-3214-7500