



**PROGRAMMABLE AC POWER SOURCE**

**DP020AS**

---

Instruction Manual (Remote Control)

**NF Corporation**



DA00102294-001

**PROGRAMMABLE AC POWER SOURCE**

**DP020AS**

Instruction Manual (Remote Control)



## Preface

This instruction manual explains the remote interfaces of the Programmable AC Power Source DP020AS (hereinafter abbreviated as “DP020AS”). For operations from the panel, see the instruction manual of the main unit.

The remote interface allows you to use not only the basic DP020AS features (e.g., power source control and measurement function), but also its extensive features (e.g., sequence output and simulation execution control) that cover various details of user needs.

- This manual consists of the following chapters.

When you use this product for the first time, start with "**1. Preparation before Use.**"

- 1. Preparation before Use**

Explains the setting of the remote interface.

- 2. Status System**

Explains the content of service requests (SRQ) and status bytes.

- 3. Remote Command Overview**

Explains the remote command formats accepted by the DP020AS.

- 4. Remote Command Reference**

Includes the list of DP020AS remote commands and descriptions for each command.

- 5. Programming Sample**

Explains the programming samples for remote-controlling the DP020AS.

- 6. Error Message List**

Includes the list of error messages replied by the DP020AS in response to remote commands.

- 7. Specifications**

Includes the specifications of the remote interface.

---

# Table of Contents

---

|   | Page |
|---|------|
| Preface .....                                       | i    |
| Table of Contents .....                             | ii   |
| List of Figures .....                               | iv   |
| List of Tables .....                                | v    |
| 1. Preparation before Use.....                      | 1    |
| 1.1 Overview.....                                   | 2    |
| 1.2 USB Preparation.....                            | 4    |
| 1.3 GPIB Preparation.....                           | 7    |
| 1.4 RS232 Preparation .....                         | 10   |
| 1.5 LAN Preparation .....                           | 13   |
| 1.6 About Remote and Local .....                    | 17   |
| 1.6.1 Remote State.....                             | 17   |
| 1.6.2 Local State.....                              | 17   |
| 1.7 Considerations.....                             | 18   |
| 1.7.1 Considerations Common to the Interfaces ..... | 18   |
| 1.7.2 USB Interface .....                           | 19   |
| 1.7.3 GPIB Interface .....                          | 19   |
| 1.7.4 RS232 Interface.....                          | 20   |
| 1.7.5 LAN Interface.....                            | 20   |
| 2. Status System.....                               | 21   |
| 2.1 Service Request .....                           | 22   |
| 2.2 Status Byte .....                               | 23   |
| 2.2.1 Status Byte Register and Service Request..... | 23   |
| 2.2.2 Status Byte Register .....                    | 24   |
| 2.2.3 Service Request Enable Register.....          | 25   |
| 2.2.4 Standard Event Status Register Group .....    | 25   |
| 2.2.5 Operation Status Register Group .....         | 26   |
| 2.2.6 Warning Condition Register Group .....        | 28   |
| 2.2.7 System Lock Condition Register Group.....     | 30   |
| 3. Remote Command Overview.....                     | 33   |
| 3.1 Remote Command.....                             | 34   |
| 3.1.1 Command Type and Format Concept.....          | 34   |
| 3.1.2 About SCPI.....                               | 34   |
| 3.1.3 Supported Common Command .....                | 35   |
| 3.1.4 SCPI Command and Query Format.....            | 36   |
| 3.1.5 SCPI Subsystem Command Tree.....              | 36   |
| 3.1.6 Elements of SCPI Commands .....               | 39   |

|       |  |     |
|-------|--|-----|
| 3.1.7 | Overview of Response Message Formats ..... | 46  |
| 4.    | Remote Command Reference .....             | 51  |
| 4.1   | Program Command List .....                 | 52  |
| 4.1.1 | List of Commands Common to Functions ..... | 53  |
| 4.1.2 | List of Continuous Function Commands ..... | 58  |
| 4.1.3 | List of Sequence Function Commands .....   | 62  |
| 4.1.4 | List of Simulation Function Commands ..... | 64  |
| 4.2   | Program Command Description .....          | 67  |
| 4.2.1 | Commands Common to Functions .....         | 67  |
| 4.3   | Continuous Function Commands .....         | 125 |
| 4.3.2 | Sequence Function Commands .....           | 182 |
| 4.3.3 | Simulation Function Commands .....         | 203 |
| 5.    | Programming Sample .....                   | 251 |
| 5.1   | DP020AS Control Programming .....          | 252 |
| 5.2   | Continuous Function Setting .....          | 253 |
| 5.3   | Sequence Function Control .....            | 256 |
| 5.4   | Simulation Function Control .....          | 258 |
| 5.5   | Obtain Status Register .....               | 260 |
| 5.6   | Cautions in Programming .....              | 262 |
| 6.    | Error Message List .....                   | 263 |
| 6.1   | Error Message List .....                   | 264 |
| 7.    | Specifications .....                       | 269 |
| 7.1   | Interface Functions .....                  | 270 |
| 7.2   | GPIB Bus Driver .....                      | 270 |
| 7.3   | Code Used .....                            | 271 |
| 7.4   | Response to Interface Message .....        | 271 |
| 7.5   | Multi-Line Interface Message .....         | 272 |

---

# Figures and Tables

---

## ■List of Figures

|   | Page |
|---|------|
| Figure 2.1 Status Byte Register and Service Request Enable Register .....         | 23   |
| Figure 2.2 Standard Event Status Register Group .....                             | 25   |
| Figure 2.3 Operation Status Register Group .....                                  | 26   |
| Figure 2.4 Warning Condition Register Group .....                                 | 28   |
| Figure 2.5 System Lock Condition Register Group .....                             | 30   |
| Figure 3.1 Example of Command Tree .....  | 36   |
| Figure 3.2 Common Command Syntax .....  | 39   |
| Figure 3.3 SP Syntax .....  | 39   |
| Figure 3.4 Subsystem Command Syntax .....   | 39   |
| Figure 3.5 Numeric Parameter (<REAL>, <INT>) Syntax .....                         | 41   |
| Figure 3.6 Mantissa Syntax .....  | 42   |
| Figure 3.7 Index Syntax .....   | 42   |
| Figure 3.8 Discrete Parameter (<DISC>) Syntax .....                               | 42   |
| Figure 3.9 Boolean Value Parameter (<BOL>) Syntax .....                           | 43   |
| Figure 3.10 String Parameter (<STR>) Syntax .....                                 | 43   |
| Figure 3.11 Block Program Data (<BLK>) Syntax .....                               | 44   |
| Figure 3.12 Suffix Syntax .....   | 44   |
| Figure 3.13 Program Message Syntax .....  | 45   |
| Figure 3.14 Response Message Syntax .....   | 46   |
| Figure 3.15 Real Number Response Data (<REAL>) Syntax .....                       | 46   |
| Figure 3.16 NR2 Number Response Data (<NR2>) Syntax .....                         | 47   |
| Figure 3.17 NR3 Number Response Data (<NR3>) Syntax .....                         | 47   |
| Figure 3.18 Integer Response Data (<INT>) Syntax .....                            | 47   |
| Figure 3.19 Discrete Response Data (<DISC>) Syntax .....                          | 48   |
| Figure 3.20 Numeric Boolean Value Response Data (<NBOL>) Syntax .....             | 48   |
| Figure 3.21 String Response Data (<STR>) Syntax .....                             | 48   |
| Figure 3.22 Definite Length Arbitrary Block Response Data Syntax (<DBLK>) .....   | 49   |
| Figure 3.23 Indefinite Length Arbitrary Block Response Data Syntax (<IBLK>) ..... | 49   |
| Figure 7.1 Multi-Line Interface Message .....                                     | 272  |



■List of Tables

|  | Page |
|--|------|
| Table 2.1 Status Byte Register .....                               | 24   |
| Table 2.2 Standard Event Status Register .....                     | 25   |
| Table 2.3 Operation Status Register .....                          | 26   |
| Table 2.4 Transition Filter and Event Register Transitions .....   | 27   |
| Table 2.5 Warning Condition Register .....                         | 29   |
| Table 2.6 System Lock Condition Register .....                     | 30   |
| Table 3.1 IEEE488.2 Common Commands Supported by DP020AS .....     | 35   |
| Table 3.2 Acceptable and Unacceptable Keywords (for "OUTPut")..... | 40   |
| Table 6.1 Error Message List .....                                 | 264  |
| Table 7.1 Interface Functions .....                                | 270  |
| Table 7.2 Bus Driver Specifications .....                          | 270  |
| Table 7.3 Response to Interface Message .....                      | 271  |



# 1.Preparation before Use

|                            |    |
|----------------------------|----|
| 1.1 Overview               | 2  |
| 1.2 USB Preparation        | 4  |
| 1.3 GPIB Preparation       | 7  |
| 1.4 RS232 Preparation      | 10 |
| 1.5 LAN Preparation        | 13 |
| 1.6 About Remote and Local | 17 |
| 1.7 Considerations         | 18 |

## 1.1 Overview

The DP020AS supports following four types of remote interfaces:

- USB (USB Test and Measurement Class USB488 Subclass, abbreviated to USBTMC hereinafter)
- GPIB
- RS232
- LAN

The DP020AS is provided with the USB, RS232, GPIB and LAN remote interfaces and is remote-controllable by a computer or other instruments. By sending remote commands for each function and receiving the data, you can control the function execution and read the internal state such as setting values and error statuses.

Regardless of which communication interface is used, you can create and use a program that uses the Virtual Instrument Software Architecture (VISA) library. If you do not have a license to use VISA library, you need to obtain one (generally, it requires some cost).

Irrespective of differences in interface, each function can be used with the same command from the remote interface. However, some functions are not available in the RS232 and LAN due to interface-specific restrictions.

### [Restrictions in RS232 and LAN]

- Status and register acquisition method  
Serial poll-based status register acquisition is not available.  
For details, see 2.2.
- Handling of termination symbol at command send/receive  
The RS232, LAN always requires a termination symbol.
- Binary data handling  
Arbitrary waveform data in binary data cannot be sent.
- Remote/Local state transition condition  
The RS232, LAN cannot perform the state transition from Remote to Local using the remote command. Only the panel operation can change the state to Local.

Command control specification: IEEE Std. 488.2-1992 compatible

(Some commands are omitted or extended depending on the interface.)

Command format specification: compliant with the SCPI Specification 1990.0 command system

## 1.2 USB Preparation

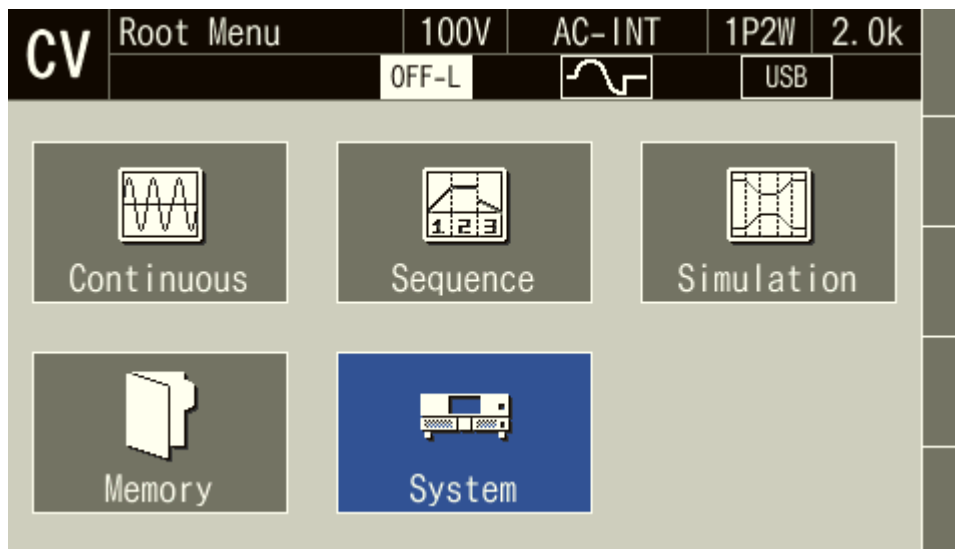
### ■ USB interface (USB2.0 Full Speed, USBTMC)

| Item       | Description                      |
|------------|----------------------------------|
| ID         | Already assigned for each device |
| Terminator | “LF”                             |

The computer used for control needs a USBTMC class driver installed on it. The USBTMC class driver is included in the hardware and software product of each company that provides the VISA library.

### ■ Operation procedure

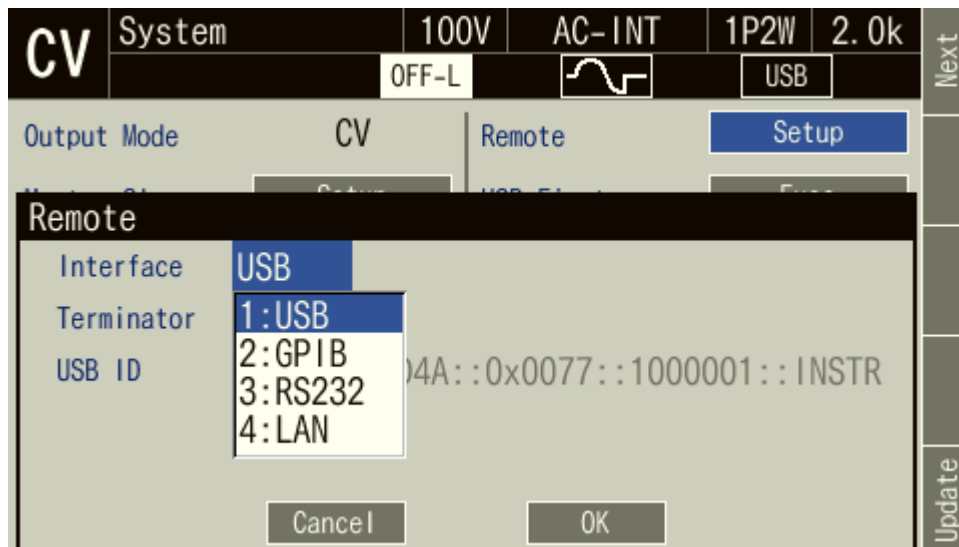
1. Press the MENU key to move to the root menu, then select [System]. The system setting window opens.



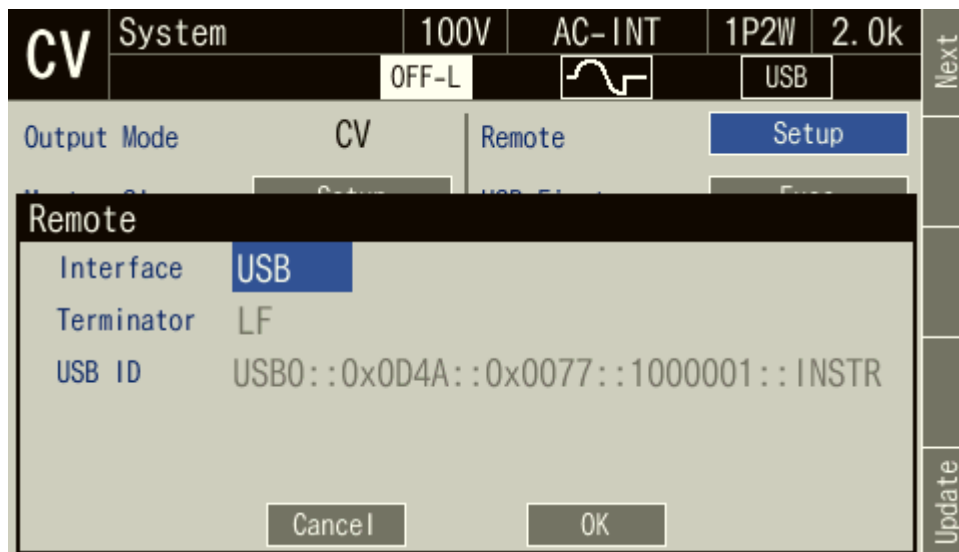
- Put the cursor on [Setup] next to the [Remote] item, then press the ENTER key. The remote setting window opens.



- In the [Interface] item, select [USB].



4. The window shows [Terminator] and [USB ID]. For the USB ID, see the explanation in the next section. Select [Close] to close the window.



5. Use a commercially available USB cable (Type A to B) to connect this product to the computer. USB connector is available on the rear of this product.

----- Notes -----

- Avoid use in a place that has under electrically noisy environment.
  - It is recommended to use a fully-shielded, short cable.
  - The use of USB hub may cause the product not to communicate properly.
- 

■ About USB ID

Used for a system connected with multiple DP020AS through USB, to enable the application identify each source. The USB ID is represented in the following format.

USB0::[Vendor number]::[Product number]::[Serial number]::INSTR

Vendor number: Fixed to 3402 (0x0D4A)

Product number: Fixed to 119 (0x0077)

Serial number: A unique number (serial number) has been set for each product.



## 1.3 GPIB Preparation

- GPIB interface (specified at the order, IEEE488.1 std 1987 compliant)

| Item       | Description or Selection | Factory default |
|------------|--------------------------|-----------------|
| Address    | 0 to 30                  | 2               |
| Terminator | “LF”                     |                 |

Note: Binary transmission is not supported.

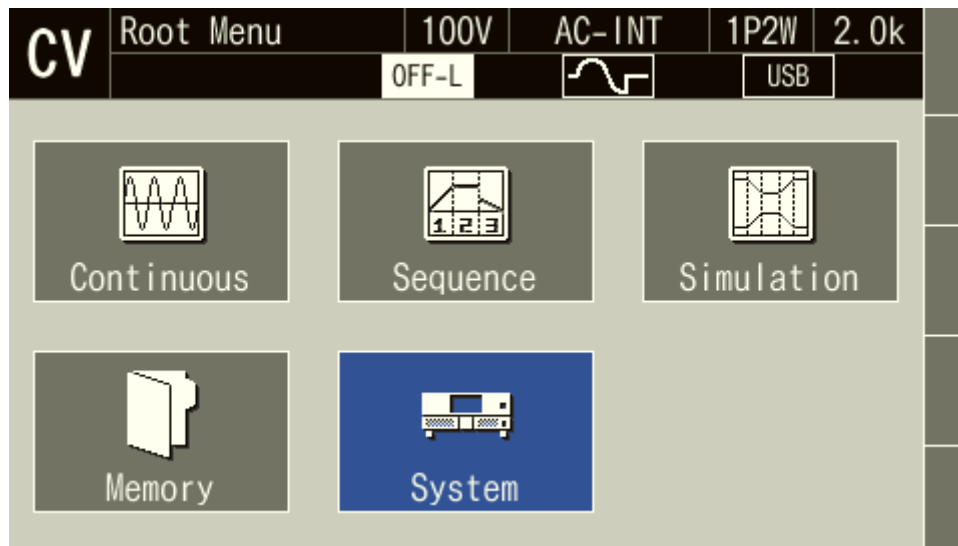
Remote command based arbitrary waveform transmission is not available.

Note: Query for the main unit status byte using a serial poll is not supported.

Mount a GPIB controller board (card) to the controlling computer, then connect between them using a commercially-available GPIB cable. For details, see the instruction manual of your GPIB controller board (card).

- Operation procedure

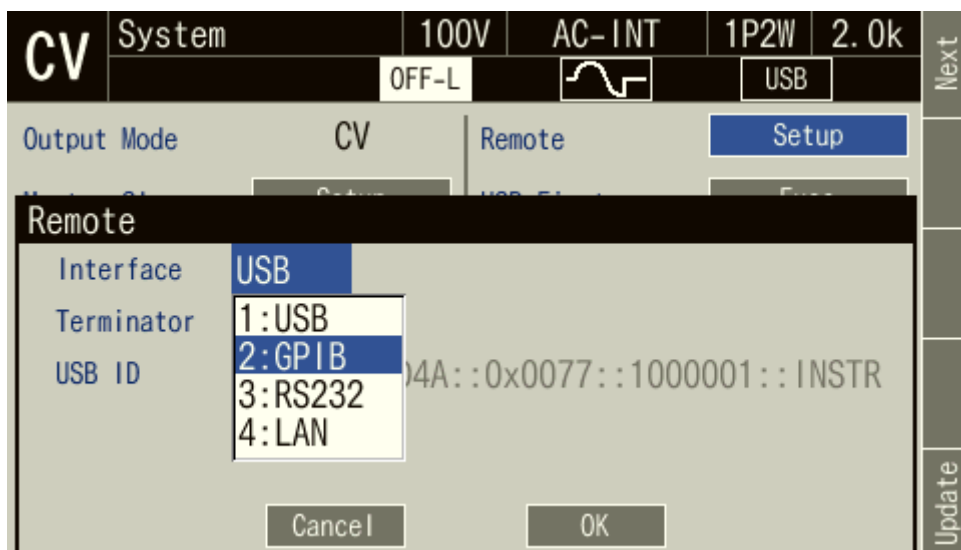
1. Press the MENU key to move to the root menu, then select [System]. The system setting window opens.



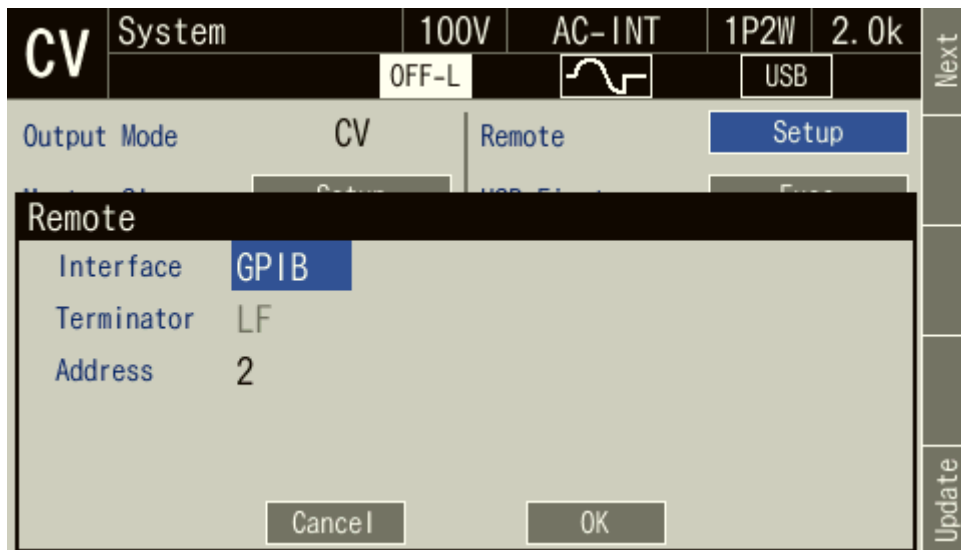
- Put the cursor on [Setup] next to the [Remote] item, then press the ENTER key. The remote setting window opens.



- In the [Interface] item, select [GPIB].



4. In the [Address] item, set the address. [Terminator] is fixed to "LF".



5. Use a GPIB cable to connect this product to the computer. GPIB connector is available on the rear of this product. Turn off this product and the computer before connecting or disconnecting the cable.

----- Notes -----

- Avoid use in a place that has under electrically noisy environment.
  - Before connecting or disconnecting the connector, turn off the power to all devices on the bus.
  - When using the GPIB, turn on the power to all devices on the bus.
  - Ensure that the total cable length is within 20 meters.
  - Ensure that the length of one cable is within 4 meters.
  - Do not set the same address as the one used for any other device connected to the same bus. Otherwise, the product cannot communicate properly.
-

## 1.4 RS232 Preparation

### ■ RS232 interface

| Item         | Description or Selection | Factory default |
|--------------|--------------------------|-----------------|
| Terminal     | D-sub 9-pin (male)       |                 |
| Baud rate    | 9600 / 19200 / 38400     | 9600bps         |
| Terminator   | “CR”“LF” / “CR” / “LF”   | “CR” “LF”       |
| Parity       | None/Odd/Even            | None            |
| Stop bit     | 1 / 2                    | 1 bit           |
| Data bit     | 7 / 8                    | 8 bit           |
| Flow control | None/Hardware/Software   | None            |

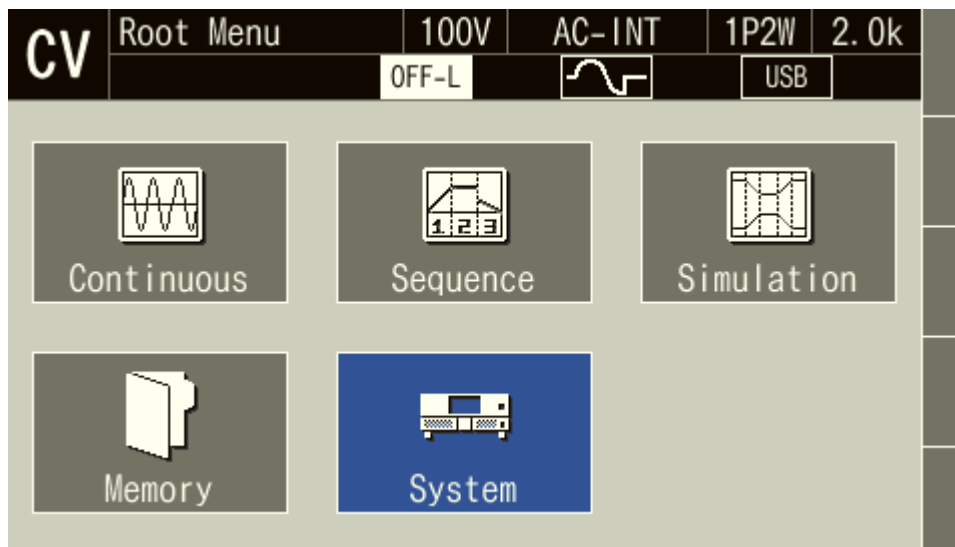
Note: Binary transmission is not supported.

Remote command based arbitrary waveform transmission is not available.

Note: Use a cross cable.

### ■ Operation procedure

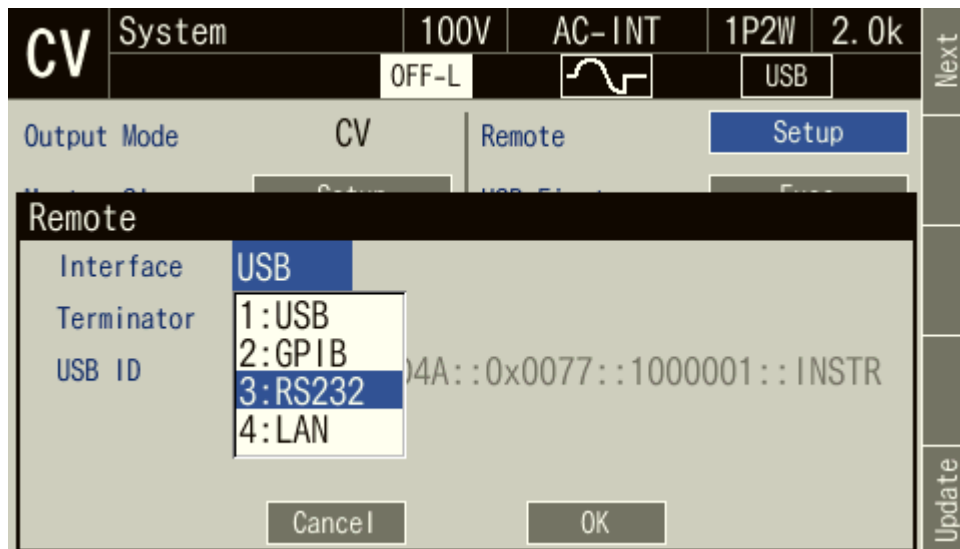
1. Press the MENU key to move to the root menu, then select [System]. The system setting window opens.



- Put the cursor on [Setup] next to the [Remote] item, then press the ENTER key. The remote setting window opens.



- In the [Interface] item, select [RS232].



4. Set each item.



5. Use a D-sub 9-pin cross cable to connect this product to the computer. RS232 connector is available on the rear of this product.

--- Notes -----

- Avoid use in a place that has under electrically noisy environment.
  - It is recommended to use a fully-shielded, short cable.
-

## 1.5 LAN Preparation

### ■ LAN interface

( specified at the order, 10BASE-T / 100BASE-TX, AutoMDI/MDI-X, SCPI-RAW : port TCP 5025)

| Item            | Description or Selection   | Note 1                              | Factory default  |
|-----------------|--|-------------------------------------|--|
| DHCP            | Enable / Disable<br>Enable : DHCP and Auto-IP function can be used.<br>Disable : The fixed IP address can be used. |                                     | Enable   |
| MAC Address     | Already assigned for each device   |                                     |  |
| IP Address      | A  | 0.0.0.0 - 255.255.255.255           | 169.254.***.***<br>(“***”shows automatically assigned by the LAN interface.) |
|                 | B  | 169.254.1.0 - 169.254.254.255       |  |
|                 | C  | Automatically assigned by the DHCP. | Automatically assigned by the DHCP.  |
| Subnet Mask     | A  | 0.0.0.0 - 255.255.255.255           | 255.255.0.0  |
|                 | B  | 255.255.0.0                         |  |
|                 | C  | Automatically assigned by the DHCP. | Automatically assigned by the DHCP.  |
| Default Gateway | A  | 0.0.0.0 - 255.255.255.255           | 169.254.***.***<br>(The same value with the ip address.)                     |
|                 | B  | 169.254.1.0 - 169.254.254.255       |  |
|                 | C  | Automatically assigned by the DHCP. | Automatically assigned by the DHCP.  |
| Terminater      | “LF”   |                                     |  |

**Note 1** : A – C in the table above are the condition of the network connection.

These shows following:

A : When “DHCP” is “Disable”.

B : When “DHCP” is “Enable”, and no DHCP server.

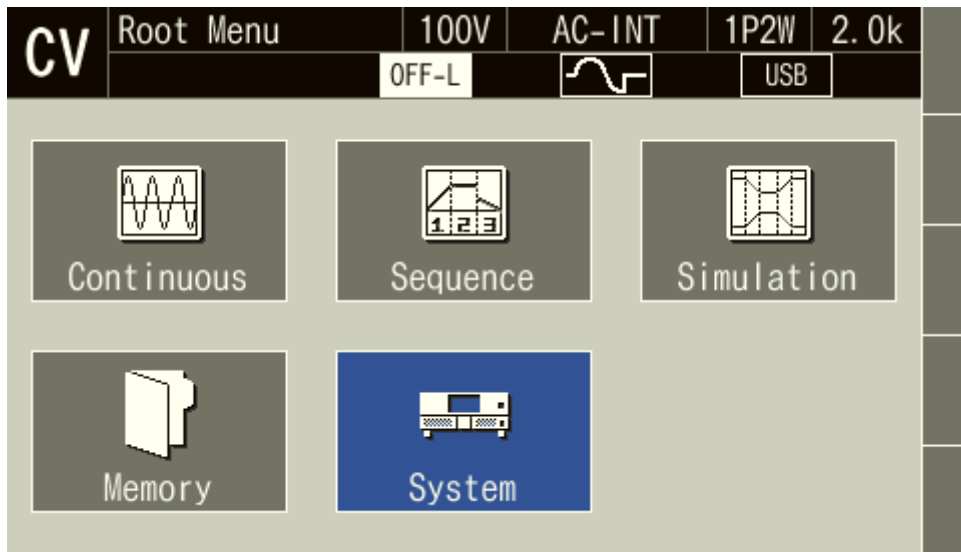
C : When “DHCP” is “Enable”, and a DHCP server is connected.

Note: Binary transmission is not supported.

Remote command based arbitrary waveform transmission is not available.

■ Operation procedure

1. Press the MENU key to move to the root menu, then select [System]. The system setting window opens.

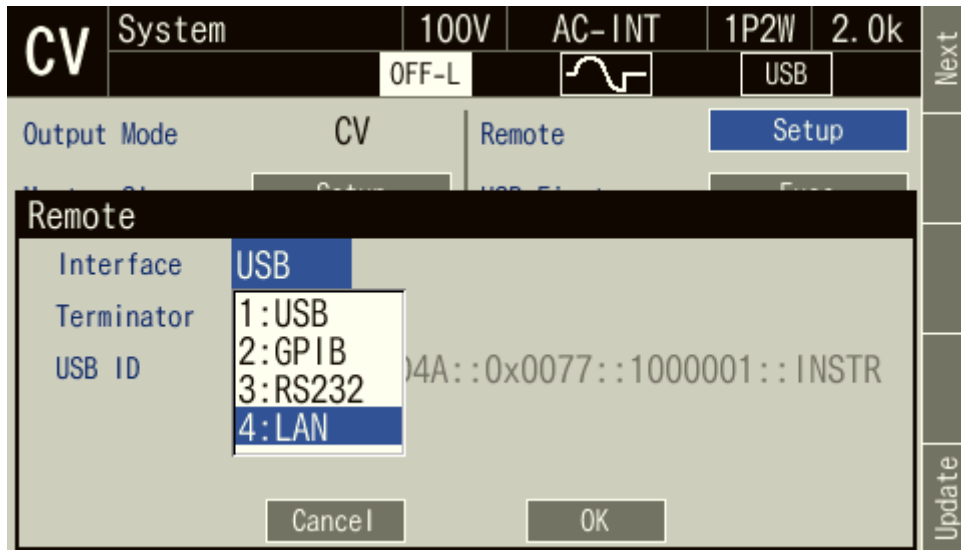


2. Put the cursor on [Setup] next to the [Remote] item, then press the ENTER key. The remote setting window opens.

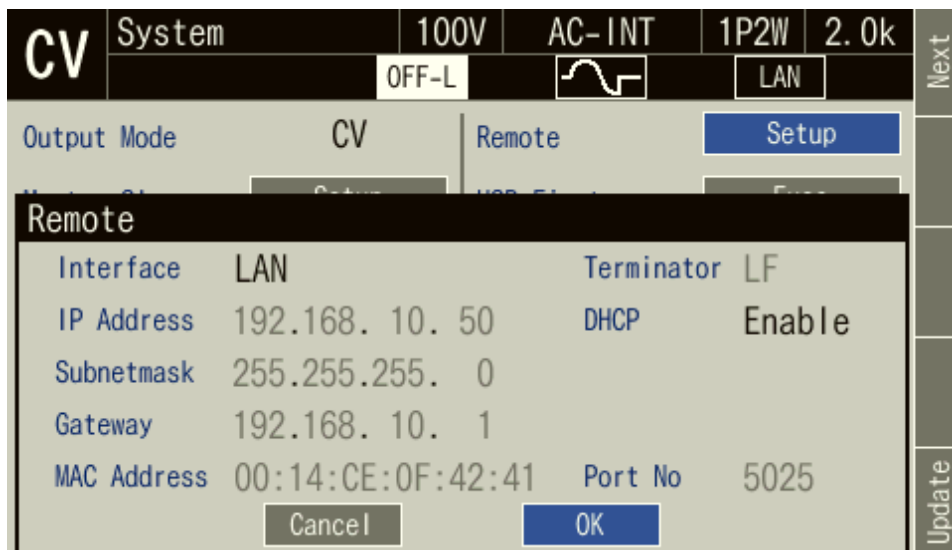
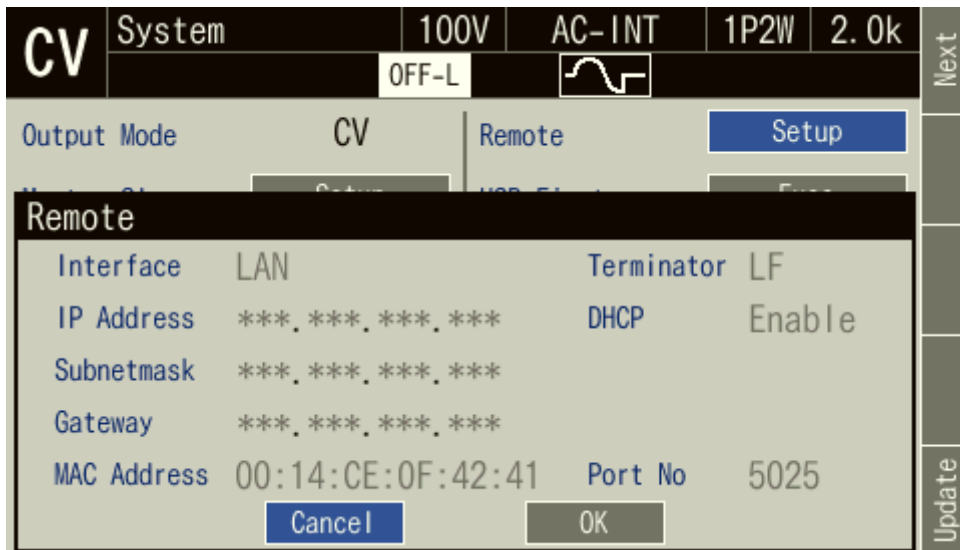




3. In the [Interface] item, select [LAN].



4. The current [MAC Address], [IP Address], [Subnetmask] and [Gateway] are shown. When not using fixed IP, undetermined parameters are shown by “\*\*\*”. After assignment, these will be updated.





5. Connect this product to a hub or computer with a commercially available LAN cable. LAN connector is available on the rear of this product.

----- Notes -----

- Avoid use in a place that has under electrically noisy environment.

## 1.6 About Remote and Local

### 1.6.1 Remote State

In the Remote state, the control panel key operations are not accepted. However, the output off (OUTPUT key) and the switching to the Local state ( + ) are available.

- Switching to the Remote state

When communication is established after the command is sent from the computer to the product, the DP020AS switches to the Remote state.

### 1.6.2 Local State

Control panel key operations are accepted.

- Switching to the Local state

The Shortcut operation  +  switches the state to Local.

The Remote state is cleared when the LOCAL key is pressed in the Remote state, when REN is set to False from the computer, or when the GTL command is received. However, the LOCAL key is disabled when the DP020AS is in the local lockout state. To clear the local lockout status, you need to set the REN to False on the computer.

---

## 1.7 Considerations

### 1.7.1 Considerations Common to the Interfaces

- The input buffer size is 36864 bytes.

The DP020AS interprets and executes the commands sequentially as it stores received commands to the input buffer. It can also handle a data of more than the size of the input buffer, where the data includes one statement which consists of multiple commands concatenated together and is terminated with the message terminator (delimiter symbol or EOI). However, the input buffer becomes full if the communication data volume is too large compared to the speed of command interpretation and execution. In that case, the computer cannot send any command, and therefore results in transmission timeout.

Note that the transmission timeout does not occur if the communication is taking place with the RS232 flow control set to None. However, other error occurs such as command execution error because the command sent from the computer is not received correctly by the DP020AS.

If there is an error during interpretation or execution of any one of the multiple commands that are concatenated and sent to the DP020AS, any commands after that command up to the message terminator are discarded; not interpreted or executed.

- The output buffer size is 4096 bytes.

The output buffer is cleared if the size of the response data in reply to the query command from the computer exceeds the output buffer size. At this time, the DP020AS sets the query error bit of the standard event status register to 1.

During the processing of data that includes multiple commands concatenated up to the message terminator (delimiter symbol or EOI) in one statement, if the size of the response data exceeds the output buffer size, the command parsing and execution is continued while the excess response data does not remain in the output buffer.

- The maximum number of errors that can be queued is 16.

Every time a command parsing and execution error occurs, the error message is queued in the error queue. If command errors occur beyond the maximum queuing number of errors, the error message stored in the 16th queue is replaced with "-350,"Queue overflow."

- The USB, RS232, GPIB and LAN interfaces are expected to be used under a relatively stable environment. Avoid using in a place that has power fluctuation or environment with much electrical noise.

## ----- Restrictions -----

- Do not insert or remove a USB memory while the main unit is performing the remote communication. This may damage the USB memory or cause the main unit to malfunction.

Insert or remove a USB memory after terminating the remote command based auto-execution program and similar activities.

## 1.7.2 USB Interface

- To use a USB interface, USBTMC device driver software needs to be installed on your computer.
- Not that not all USBTMC device driver software pieces provided by every company have been tested for operation verification.

Operation verification was performed on our USBTMC compatible device driver software (dedicated to the attached application) and the USBTMC device driver software that comes with NI-VISA provided by National Instruments.

## 1.7.3 GPIB Interface

- Power off all devices on the bus when you connect or disconnect the GPIB cable.
- When using the GPIB, turn on the power to all devices connected to the bus.
- Up to 15 devices (including the controller) can be connected to the GPIB in one system.  
The cable length has the following limitations.
  - Total cable length is 2 meters x (number of devices) or 20 meters, whichever the shorter
  - Length of a cable is 4 meters or less
- Check carefully for the GPIB address before setting it. Using a duplicated address in one system may damage devices.
- Inconsistency of the delimiter settings in the system might cause unexpected trouble. Use the same delimiter setting for every device that exists in the system.

#### **1.7.4 RS232 Interface**

- One RS232 port cannot have multiple devices connected at one time.
- When communication is taking place with no flow control, the command may not be executed properly. Use the flow control whenever possible.
- For communication that has the hardware flow control enabled, use a cable in which the RTS signal and CTS signal lines are interlink-connected. Use the one that RTS on the computer side is connected to CTS on the DP020AS, CTS to RTS.
- To avoid malfunctioning due to radiated electromagnetic noise, ensure to use a shielded cable.

#### **1.7.5 LAN Interface**

- If you set “DHCP” to “Disable” (manually determining IP address), be careful not to set the same IP address to the other device.
- Use an appropriate cable for connection to your network.

## 2. Status System

|                           |    |
|---------------------------|----|
| 2.1 Service Request ..... | 22 |
| 2.2 Status Byte .....     | 23 |

---

## 2.1 Service Request

The USB and GPIB interface supports the service request functions.

You can obtain the status of the DP020AS product by performing a serial poll (reading status byte) from your computer, under the conditions shown below. The serial poll allows you to obtain the status of the DP020AS product regardless of the Remote/Local state.

- When the output data for query is prepared
- When some error or warning occurs



## 2.2 Status Byte

The status byte can be obtained by a serial poll only when using USB or GPIB interface. When using a RS232 or LAN interface, the status byte cannot be obtained by a serial poll.

### 2.2.1 Status Byte Register and Service Request

The configuration of the status byte register and the service request enable register is shown in Figure 2.1.

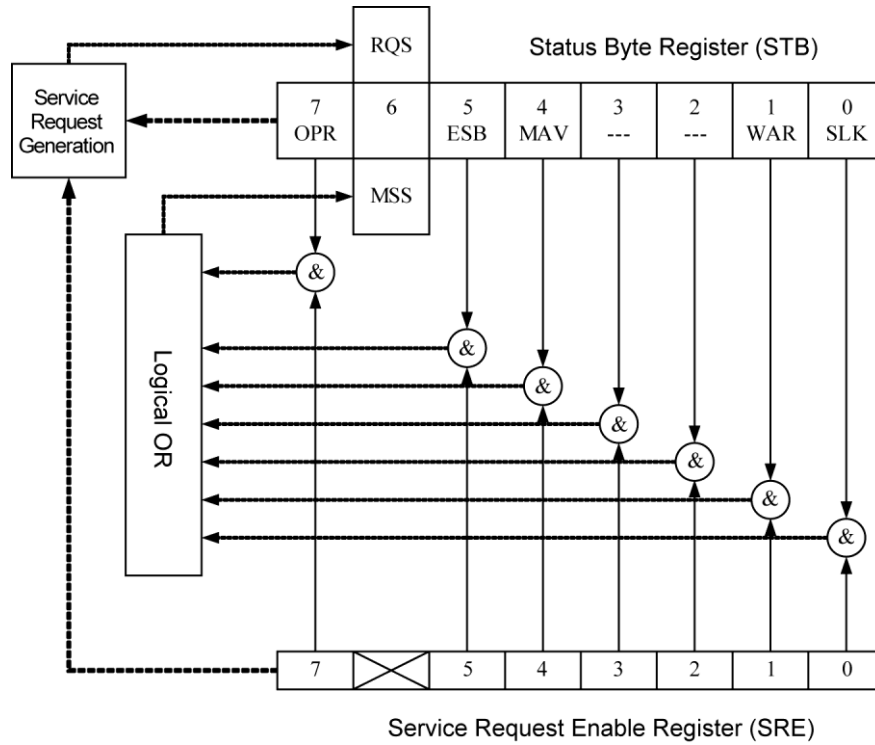


Figure 2.1 Status Byte Register and Service Request Enable Register

## 2.2.2 Status Byte Register

Each bit of the status byte register is described in Table 2.1.

**Table 2.1 Status Byte Register**

| Bit        | Weight | Symbol      | Description   | Set Condition<br>(set to 1)   | Clear Condition<br>(set to 0)   |
|------------|--------|-------------|---|---|---|
| (MSB)<br>7 | 128    | OPR         | Operation condition register summary                              | This bit is set when the logical OR of bits of the operation event register is 1.       | This bit is cleared when the logical OR of bits of the operation event register is 0.       |
| 6          | 64     | RQS/<br>MSS | Request service/master summary status                             | <b>Note 3</b>   | <b>Note 1</b>   |
| 5          | 32     | ESB         | Standard Event Status Register Summary                            | This bit is set when the logical OR of bits of the standard event status register is 1. | This bit is cleared when the logical OR of bits of the standard event status register is 0. |
| 4          | 16     | MAV         | This bit notifies that the output data for the query is prepared. | This bit is set when the output data for the query is prepared.                         | This bit is cleared when there is no output data for the query. <b>Note 4</b>               |
| 3          | 8      | Unused      | Unused  | <b>Note 2</b>   | <b>Note 2</b>   |
| 2          | 4      | Unused      | Unused  | <b>Note 2</b>   | <b>Note 2</b>   |
| 1          | 2      | WAR         | Warning condition register summary                                | This bit is set when the logical OR of bits of the warning event register is 1.         | This bit is cleared when the logical OR of bits of the warning event register is 0.         |
| (LSB)<br>0 | 1      | SLK         | System Lock condition register summary                            | This bit is set when the logical OR of bits of the System Lock event register is 1.     | This bit is cleared when the logical OR of bits of the System Lock event register is 0.     |

**Note 1** : • When a device clear (DCL or SDC) is received.  
• After reading the status byte (reading the status byte by a serial poll during the SRQ output or a query message).

**Note 2** : • Always 0 because this bit is not used.

**Note 3** : • In case of reading by a serial poll, this bit is set when the SRQ is sent.  
• In case of reading by a query message, this bit is set if the logical OR of bits of the status byte is 1.

**Note 4** : • When a device clear (DCL or SDC) is received.  
• USB : After reading the status byte by using a query message.  
When using USB, this bit is not cleared by a serial poll.

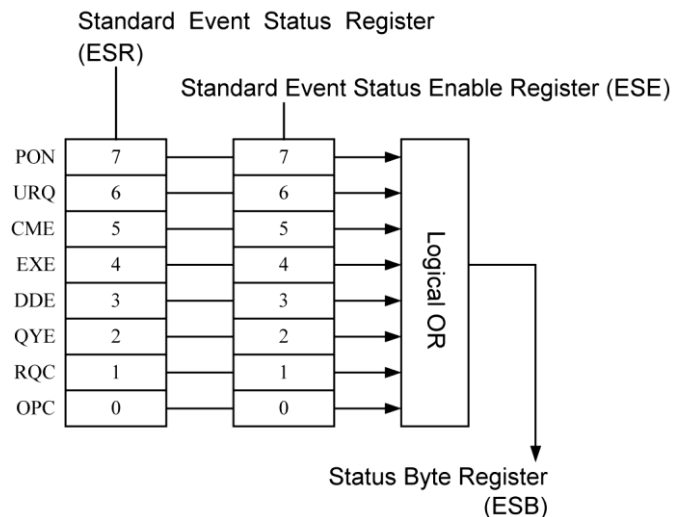
The status byte register is cleared when a \*CLS command is received.

### 2.2.3 Service Request Enable Register

The service request enable register is used to select the summary bit within the status byte register that generates service requests shown in Figure 2.1.

### 2.2.4 Standard Event Status Register Group

The configuration of the standard event status register group is shown in Figure 2.2.



**Figure 2.2 Standard Event Status Register Group**

#### 2.2.4.1 Standard Event Status Register

Each bit of the standard event status register is described in Table 2.2.

**Table 2.2 Standard Event Status Register**

| Bit | Weight | Symbol | Description              |
|-----|--------|--------|--------------------------|
| 7   | 128    | PON    | Power on                 |
| 6   | 64     | URQ    | User request             |
| 5   | 32     | CME    | Command error            |
| 4   | 16     | EXE    | Execution error          |
| 3   | 8      | DDE    | Equipment-specific error |
| 2   | 4      | QYE    | Query error              |
| 1   | 2      | RQC    | Request control          |
| 0   | 1      | OPC    | Operation completion     |

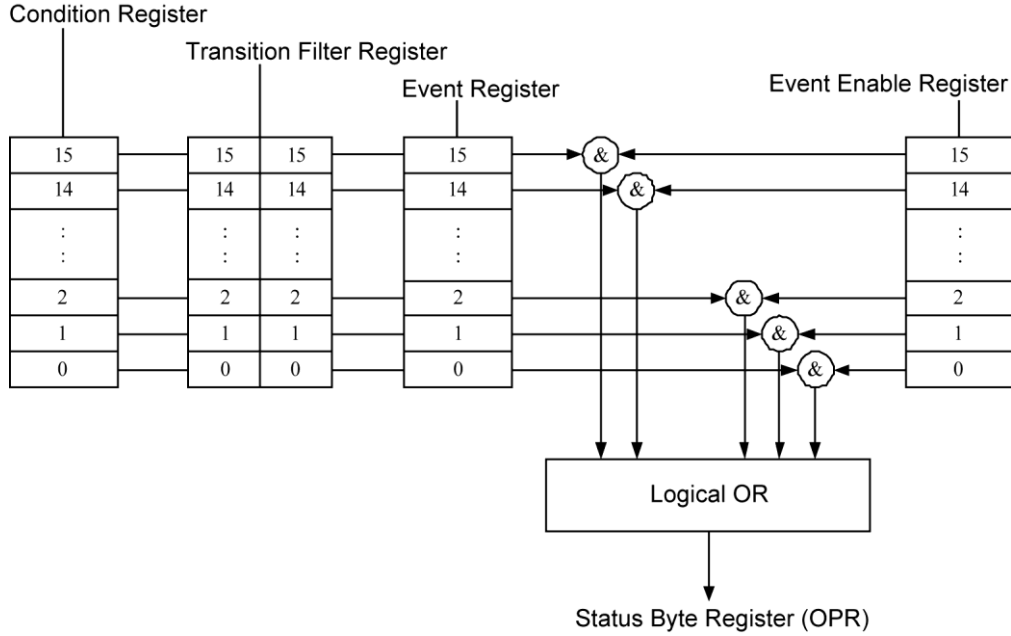
The standard event status register is cleared when the \*ESR? query or \*CLS command is received.

#### 2.2.4.2 Standard Event Status Enable Register

The standard event status enable register is used to select the bit of the standard event status register as shown in Figure 2.2, and reflect the status of the selected bit to the ESB of the status byte register.

### 2.2.5 Operation Status Register Group

The configuration of operation status register group is shown in Figure 2.3.



**Figure 2.3 Operation Status Register Group**

#### 2.2.5.1 Operation Status Condition Register

The operation status condition register is a register that represents the current status of the device. This register is not cleared even when a query addressed to the condition register is received. Each bit of operation status register group is described in Table 2.3.

**Table 2.3 Operation Status Register**

| Bit | Weight | Description                                  |
|-----|--------|--|
| 15  | -      | Always 0                                     |
| 14  | 16384  | Sequence or Simulation is in the Run status  |
| 13  | -      | (Unused)                                     |
| 12  | 4096   | Sequence is in the Hold status               |
| 11  | -      | (Unused)                                     |
| 10  | -      | (Unused)                                     |
| 9   | 512    | Measured value overrange                     |
| 8   | 256    | LOCK status (SYNC status)                    |
| 7   | -      | (Unused)                                     |
| 6   | -      | (Unused)                                     |
| 5   | -      | (Unused)                                     |
| 4   | -      | (Unused)                                     |
| 3   | 8      | Soft start or soft stop status(Sweep status) |
| 2   | -      | (Unused)                                     |
| 1   | 2      | Busy status                                  |
| 0   | -      | (Unused)                                     |

\* The state that 40 °C of ambient temperature exceeds the reverse power flow specification value at the environment bottom.

### 2.2.5.2 Operation Status Transition Filter

The operation status transition filter is a filter for determining the event bit transition.

The operation status transition filter is not cleared even when the \*CLS command is received.

The relationship between the transition filter setting and the event register transitions is shown in Table 2.4.

**Table 2.4 Transition Filter and Event Register Transitions**

| Each Bit Setting of Positive Transition Filter | Each Bit Setting of Negative Transition Filter | Transition of Condition Register to Set the Bit of Event Register to 1 |
|--|--|--|
| 0  | 0  | Not changed  |
| 0  | 1  | 1 → 0 (detects the falling)  |
| 1  | 0  | 0 → 1 (detects the rising)   |
| 1  | 1  | 0 → 1 or 1 → 0   |

### 2.2.5.3 Operation Status Event Register

The operation status event register is a register that reflects the changes of the condition register according to the transition filter setting.

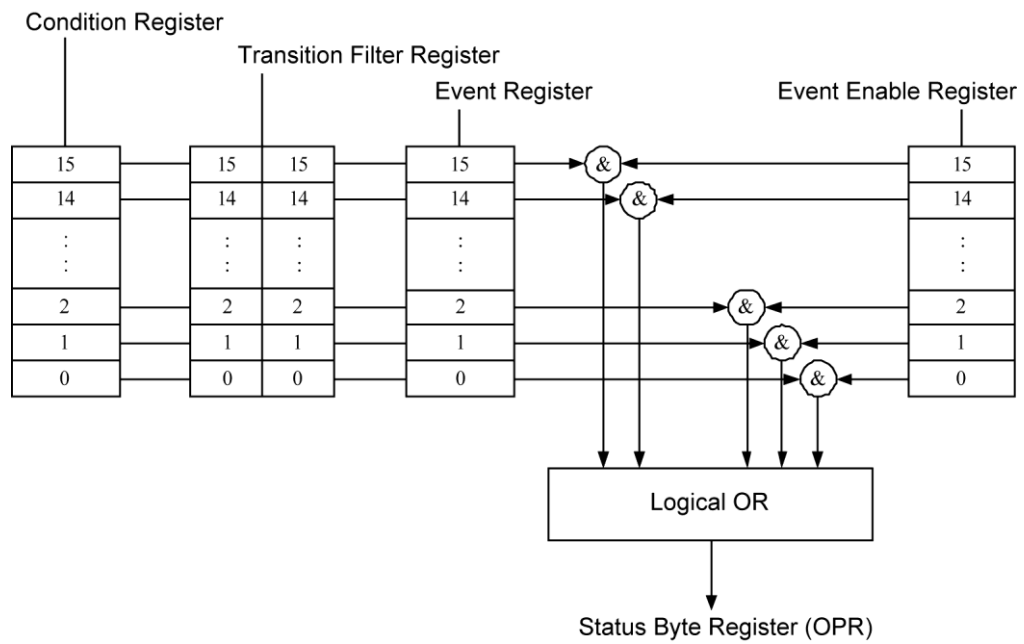
The event register is cleared upon reception of a query addressed to the event register or the \*CLS command.

### 2.2.5.4 Operation Status Event Enable Register

The operation status event enable register is used to select bits in the event register to be summarized.

## 2.2.6 Warning Condition Register Group

The configuration of the warning condition register group is shown in Figure 2.4.



**Figure 2.4 Warning Condition Register Group**

**2.2.6.1 Warning Condition Register**

This register is not cleared even when a query addressed to the condition register is received.  
Each bit of the warning condition register is described in Table 2.5.

**Table 2.5 Warning Condition Register**

| Bit | Weight | Description  |
|-----|--------|--|
| 15  | -      | Always 0   |
| 14  | 16384  | Peak current limiter is operating<br>Peak voltage limiter is operating                                   |
| 13  | 8192   | RMS current limiter is operating<br>RMS voltage limiter is operation                                     |
| 12  | 4096   | Active wattage limiter is operating  |
| 11  | 2048   | Output OFF after peak current limiter is activated<br>Output OFF after peak voltage limiter is activated |
| 10  | 1024   | Output OFF after RMS current limiter is activated<br>Output OFF after RMS voltage limiter is activated   |
| 9   | 512    | Sensing voltage anomaly  |
| 8   | 256    | Power unit DCPS anomaly  |
| 7   | 128    | Synchronization frequency anomaly  |
| 6   | 64     | Overheat   |
| 5   | 32     | Power unit DCPS overvoltage  |
| 4   | 16     | Power unit DCPS undervoltage   |
| 3   | 8      | Output overcurrent (Peak)  |
| 2   | 4      | Power unit anomaly   |
| 1   | 2      | Output overcurrent (RMS)   |
| 0   | 1      | Output overvoltage   |

**2.2.6.2 Warning Status Transition Filter**

The warning status transition filter is a filter for determining the event bit transition.

The warning status transition filter is not cleared even when the \*CLS command is received.

For the relationship between the transition filter setting and the event register transitions, refer to Table 2.4.

**2.2.6.3 Warning Event Register**

The warning event register is a register that reflects the changes of the condition register according to the transition filter setting.

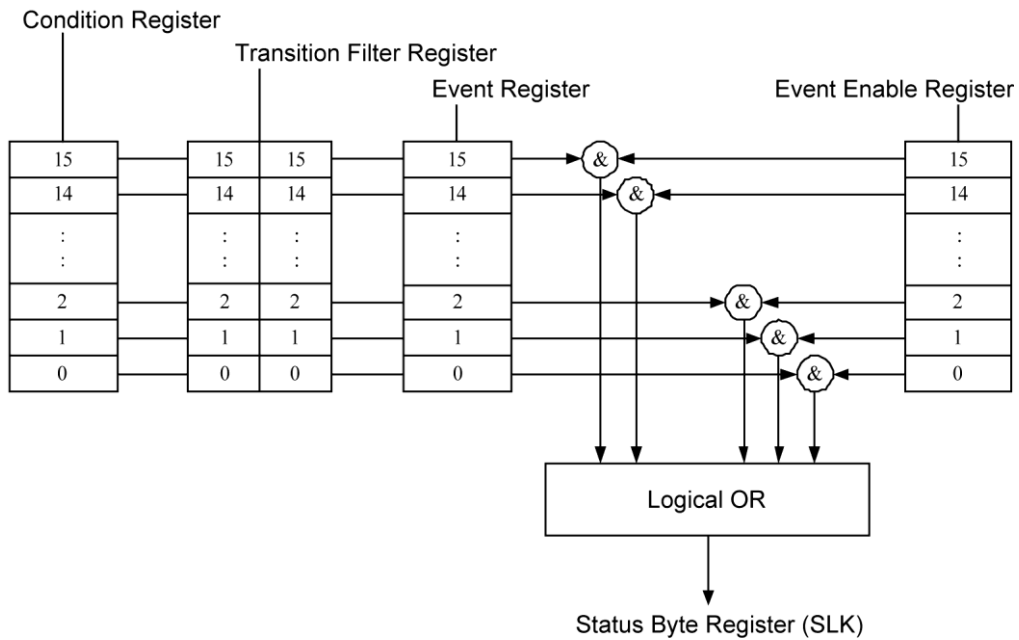
The event register is cleared upon reception of a query addressed to the event register or the \*CLS command.

**2.2.6.4 Warning Event Enable Register**

The warning event enable register is used to select bits in the event register to be summarized.

### 2.2.7 System Lock Condition Register Group

The configuration of the System Lock condition register group is shown in Figure 2.5.



**Figure 2.5 System Lock Condition Register Group**

#### 2.2.7.1 System Lock Condition Register

This register is not cleared even when a query addressed to the condition register is received. Each bit of the System Lock condition register is described in Table 2.6.

**Table 2.6 System Lock Condition Register**

| Bit | Weight | Description                                |
|-----|--------|--|
| 15  | -      | Always 0                                   |
| 14  | 16384  | (Unused)                                   |
| 13  | 8192   | (Unused)                                   |
| 12  | 4096   | (Unused)                                   |
| 11  | 2048   | (Unused)                                   |
| 10  | 1024   | (Unused)                                   |
| 9   | 512    | Shut down                                  |
| 8   | 256    | Cabinet connect mismatched                 |
| 7   | 128    | System settings mismatched between cabinet |
| 6   | 64     | Internal auxiliary power anomaly           |
| 5   | 32     | Polyphase system communication anomaly     |
| 4   | 16     | Internal communication anomaly 2           |
| 3   | 8      | Internal communication anomaly 1           |
| 2   | 4      | (Unused)                                   |
| 1   | 2      | Power input undervoltage                   |
| 0   | 1      | Power input overvoltage                    |



### 2.2.7.2 System Lock Status Transition Filter

The System Lock status transition filter is a filter for determining the event bit transition.

The System Lock status transition filter is not cleared even when the \*CLS command is received.

For the relationship between the transition filter setting and the event register transitions, refer to Table 2.4.

### 2.2.7.3 System Lock Event Register

The System Lock event register is a register that reflects the changes of the condition register according to the transition filter setting.

The event register is cleared upon reception of a query addressed to the event register or the \*CLS command.

### 2.2.7.4 System Lock Event Enable Register

The System Lock event enable register is used to select bits in the event register to be summarized.

## 2.2 Status Byte



## **3.Remote Command Overview**

|                          |    |
|--------------------------|----|
| 3.1 Remote Command ..... | 34 |
|--------------------------|----|

## 3.1 Remote Command

### 3.1.1 Command Type and Format Concept

There are two types of remote commands of DP020AS; the common commands defined in IEEE488.2, and the commands defined according to SCPI (Standard Commands for Programmable Instruments) command specifications.

### 3.1.2 About SCPI

SCPI is a standard that defines the communication method between external controllers and measuring instruments.

For general information about SCPI, refer to the following document:

Standard Commands for Programmable Instruments (SCPI) VERSION 1999.0

<http://www.ivifoundation.org/scpi/>

#### Conventions

The commands mentioned in the description are written following conventions:

|                      |   |
|----------------------|---|
| < >                  | < > indicates something other than itself. In the case of parameter or response data, the abbreviation of the type is enclosed in < >. <NL> indicates an ASCII character that has the value of 10 decimal, and similarly, <^END> indicates EOI. |
| [ ]                  | The items enclosed in [ ] are optional. Note that "[" and "]" are not part of the command.  |
| { abc xyz }          | Means that either "abc" or "xyz" should be used.  |
| [ abc xyz ]          | Indicates that either "abc" or "xyz" can be used, but both are optional.  |
| Uppercase, lowercase | A whole word consisting of the combination of upper and lowercase letters makes a long form of the keyword, while an uppercase only part makes a short form.  |

### 3.1.3 Supported Common Command

The DP020AS supports the instrument common commands which comply with the IEEE488.2 standard.

All common commands begin with an asterisk (\*), and some of them can use parameters.

The header of common command and the first parameter must be separated by a space code. The common commands supported by the DP020AS are shown in Table 3.1.

**Table 3.1 IEEE488.2 Common Commands Supported by DP020AS**

| Command Query | Name                                 |
|---------------|--------------------------------------|
| *IDN?         | Identification Query                 |
| *RST          | Reset Command                        |
| *TST?         | Self-Test Query                      |
| *OPC          | Operation Complete Command           |
| *OPC?         | Operation Complete Query             |
| *WAI          | Wait-to-Continue Command             |
| *CLS          | Clear Status Command                 |
| *ESE          | Standard Event Status Enable Command |
| *ESE?         | Standard Event Status Enable Query   |
| *ESR?         | Standard Event Status Register Query |
| *SRE          | Service Request Enable Command       |
| *SRE?         | Service Request Enable Query         |
| *STB?         | Read Status Byte Query               |

### 3.1.4 SCPI Command and Query Format

A SCPI command has a hierarchical structure that consists of a root keyword, one or more lower-level keywords, parameters, and suffixes.

The followings show examples of command and query.

```
:OUTPut:STATe ON<NL><^END>
```

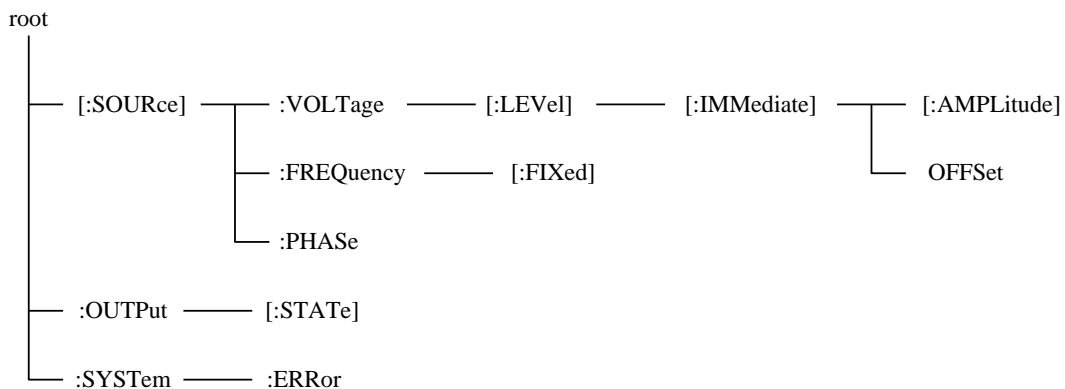
```
:OUTPut:STATe?<NL><^END>
```

OUTPut is the root-level keyword that links second-level keywords. ON is a command parameter.

### 3.1.5 SCPI Subsystem Command Tree

#### 3.1.5.1 Structure of Command Tree

SCPI applies a hierarchical structure similar to the one used in a file system to subsystem commands. In SCPI, this command structure is called a command tree, and Figure 3.1 shows an example of a command tree.



**Figure 3.1 Example of Command Tree**

In the command tree shown in Figure 3.1, the keywords closest to the top ("[:SOURce]", ":OUTPut", and ":SYSTem") are root-level keywords. To reach a lower-level keyword, you must follow the path of keywords defined above it.

For example, to access ":OFFSet", the command should be specified by following the path [:SOURce] - :VOLTage - [:LEVel] - [:IMMediate] - :OFFSet.

#### 3.1.5.2 Current Path Movement

The current path is a level within the command tree, and becomes the first command path from which the command parser starts the search when the next command is sent. The command parser determines the current path according to the following rules.

(1) At power-on and reset

The current path is set to the root.

(2) Message terminator

Upon reception of <NL> (message terminator), the current path is set to the root.

(3) Colon (command separator)

When a colon is placed between two keywords, the colon moves the current path down one level in the command tree.

(4) Colon (root specifier)

When a colon is placed at the beginning of a command, the colon sets the current path to the root.

(5) Semicolon

Semicolons do not affect to the current path.

(6) Space

Spaces do not affect to the current path.

(7) Comma

Commas do not affect to the current path.

(8) IEEE488.2 common command

Common commands do not affect to the current path.

By using semicolons properly, multiple commands can be sent efficiently.

For example,

```
:SOURce:VOLTage:LEVel:IMMEDIATE:AMPLitude 1.0; OFFSet 1.0<NL><^END>
```

is the same as sending the following two commands:

```
:SOURce:VOLTage:LEVel:IMMEDIATE:AMPLitude 1.0<NL><^END>  
:SOURce:VOLTage:LEVel:IMMEDIATE:OFFSet 1.0<NL><^END>
```

When an optional keyword is omitted, you should pay careful attention to the current path movement.

For example,

```
:VOLTage 1.0<NL><^END>
```

sets the current path to ":SOURce."

If you want to send the following two program messages as one program message,

```
:SOURce:VOLTage:LEVel:IMMediate:AMPLitude 1.0<NL><^END>  
:SOURce:FREQuency:FIXed 100.0<NL><^END>
```

<Program message which is executed properly>

```
:SOURce:VOLTage 1.0; FREQuency:FIXed 100.0<NL><^END>
```

<Program message which causes parsing error>

```
:SOURce:VOLTage:LEVel:IMMediate:AMPLitude1.0; FREQuency:FIXed 100.0<NL><^END>
```

Here, the current path of the second command is ":SOURce:VOLTage:LEVel:IMMediate" (by right, it must be ":SOURce"). Therefore, ":FREQuency:FIXed 100.0" causes parsing error.



### 3.1.6 Elements of SCPI Commands

#### 3.1.6.1 Common Command Syntax

The syntax of common commands is shown in Figure 3.2. In this figure, a keyword consists of three alphanumeric characters, beginning with an alphabetic character.

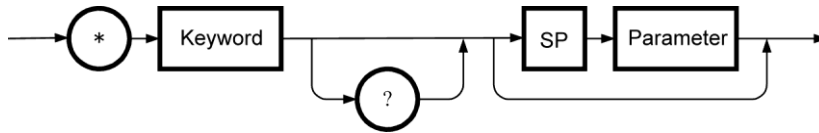


Figure 3.2 Common Command Syntax

"SP" in Figure 3.2 is technically as shown in Figure 3.3 (hereinafter, all SP means the same).

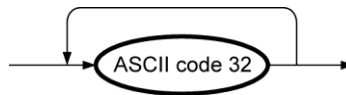


Figure 3.3 SP Syntax

#### 3.1.6.2 Subsystem Command Syntax

The syntax of the subsystem commands is shown in Figure 3.4.

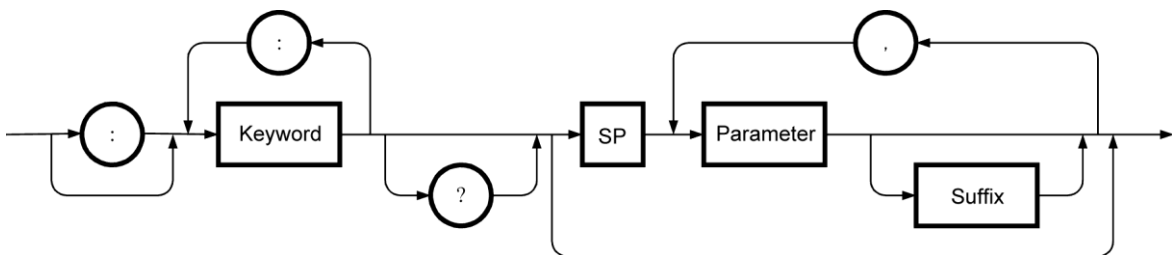


Figure 3.4 Subsystem Command Syntax

**3.1.6.3 Keyword**

The keyword in the command syntax is a string of up to 12 characters consisting of uppercase and lowercase alphabetic characters, underscore (\_), and numeric characters, beginning with an alphabetic character.

Most keywords are shown as a mixture of uppercase and lowercase characters.

The uppercase characters represent the short form for commands. Either long or short form format can be accepted; commands are not case sensitive actually.

Examples for the keyword "OUTPut" are listed in Table 3.2.

**Table 3.2 Acceptable and Unacceptable Keywords (for "OUTPut")**

| Keyword | Description   |
|---------|---|
| OUTPUT  | Accepted as long form.  |
| OUTP    | Accepted as short form.   |
| OuTpUt  | Accepted as long form without distinguishing uppercase and lowercase characters.  |
| oUtP    | Accepted as short form without distinguishing uppercase and lowercase characters. |
| OUTPU   | Not accepted because it corresponds to neither long form nor short form.          |
| OUT     | Not accepted because it corresponds to neither long form nor short form.          |

**3.1.6.4 Keyword Separator**

Keyword separators serve to separate the current keyword and the next lower-level keyword as shown in the example below. Colons (:) are used as keyword separators.

:OUTPut:STATe?

**3.1.6.5 Root Specifier**

A colon (:) at the beginning of the subsystem command serves as a root specifier.

The root specifier sets the current path to the root.

As colons (:) are also used as command separators, you need to carefully use them depending on your purpose.

SOURce:FREQuency:FIXed 100.0

→ All colons are header separators

:SOURce:FREQuency:FIXed 1.0

→ The first colon is the root specifier

SOURce:FREQuency:FIXed 100.0; :OUTPut ON

→ The third colon is the root specifier

**3.1.6.6 Optional Keywords**

The keywords enclosed in square brackets ([]) are optional, and can be omitted. However, the square brackets ([]) are used for description purpose only, and are not included in the actual command to be sent.

If the keyword is omitted, the command parser interprets the command as if that optional keyword is received.

For example,

```
:OUTPut[:STATe]
```

means that you can use either of the following commands:

```
:OUTPut:STATe
```

```
:OUTPut
```

### 3.1.6.7 Implied Channels (Optional Numeric Keyword Suffix)

Many commands allow you to specify channels by using an optional numeric keyword suffix.

For example,

```
:OUTPut[1|2]:STATe {ON|OFF}
```

means commands for channel 1 and 2 as follows:

```
:OUTPut[1]:STATe {ON|OFF}
```

```
:OUTPut2:STATe {ON|OFF}
```

Here, you need to note that if you do not specify the channel number, the channel 1 will be selected as an implied channel.

For example, to control the channel 1 output to be turned ON, you can use either of the following commands.

```
:OUTPut1:STATe ON
```

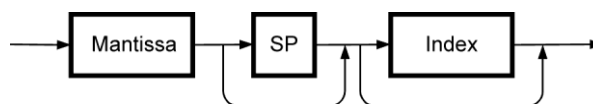
```
:OUTPut:STATe ON
```

### 3.1.6.8 Parameter Type

In the command list, the angle brackets (<>) indicate parameters. The parameter type is one of the following types.

#### Numeric parameter (<REAL>, <INT>)

The syntax of numeric parameters is shown in Figure 3.5.



**Figure 3.5 Numeric Parameter (<REAL>, <INT>) Syntax**

The syntaxes of the mantissa and index in Figure 3.5 are shown in Figure 3.6 and Figure 3.7.

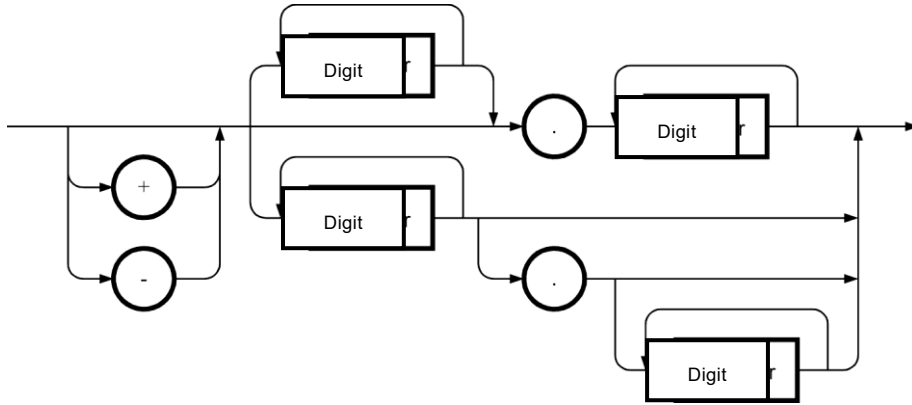


Figure 3.6 Mantissa Syntax

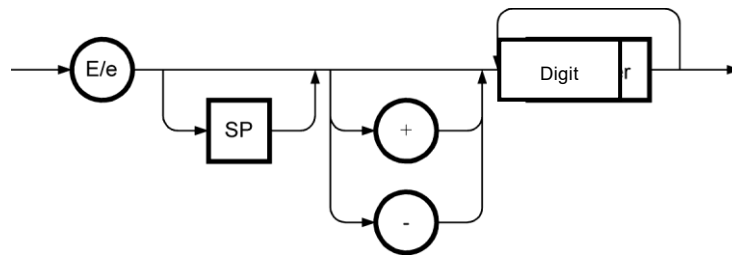


Figure 3.7 Index Syntax

**3.1.6.9 Discrete Parameter (<DISC>)**

The syntax of the discrete parameter is shown in Figure 3.8.

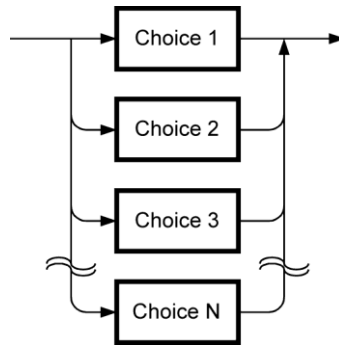
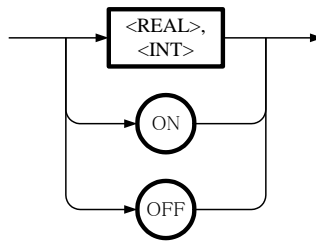


Figure 3.8 Discrete Parameter (<DISC>) Syntax

**3.1.6.10 Boolean Value Parameter (<BOL>)**

The syntax of the Boolean value parameter is shown in Figure 3.9.



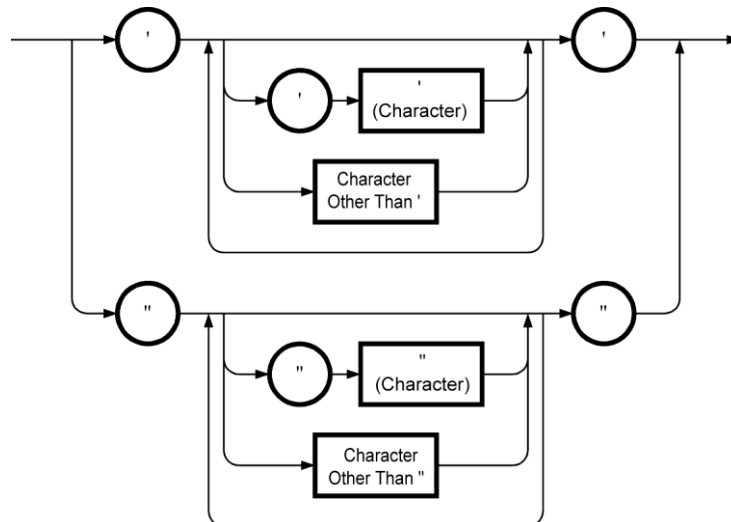
**Figure 3.9 Boolean Value Parameter (<BOL>) Syntax**

The boolean parameter is interpreted the values other than 0 as true, and 0 as false.

If the specified value includes a fractional part, that value is interpreted as the whole number obtained by rounding the fractional part. Therefore, "0.4" is false (rounded off to 0), and "0.5" is true (rounded off to 1).

#### 3.1.6.11 String Program Data (<STR>)

The syntax of the string program data is shown in Figure 3.10.

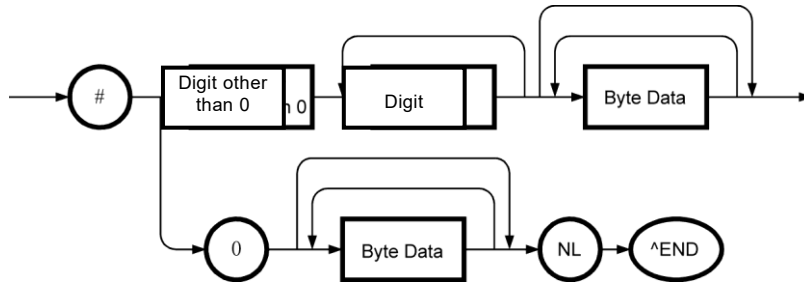


**Figure 3.10 String Parameter (<STR>) Syntax**

In the DP020AS, you cannot use the following nine characters as a part of a string: \, /, :, \*, ?, ", <, >, |. If the string includes one of these characters, an error process will be raised. (The error message returns an error code :-150 "String data error.")

#### 3.1.6.12 Block Program Data (<BLK>)

The syntax of the block program data is shown in Figure 3.11.



**Figure 3.11 Block Program Data (<BLK>) Syntax**

For the definite length block program data, if "^END" is received before receiving the specified number of bytes, it is treated as a command error.

The DP020AS only accept the definite length block program data, and the number string between # and the first byte data is fixed to 48192. The byte data must be 8192 bytes.

**3.1.6.13 Parameter Separator**

If you want to send multiple parameters in one command, you need to separate the adjacent parameters with a comma (,).

**3.1.6.14 Query Parameter**

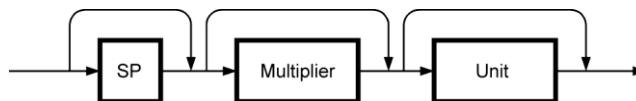
Most of the queries supported by the commands that have decimal numeric parameters can return the minimum or maximum values if "MINimum" or "MAXimum" is specified after "?".

For example, to query the minimum and maximum values of the frequency, send the following queries:

```
:SOURce:FREQuency? MINimum
:SOURce:FREQuency? MAXimum
```

**3.1.6.15 Suffix**

The syntax of suffix is shown in Figure 3.12.



**Figure 3.12 Suffix Syntax**

**3.1.6.16 Command Terminator**

A command terminator is either <NL> (ASCII character decimal 10), an EOI (End-of Identify) asserted by the last byte, or an EOI asserted by <NL>.

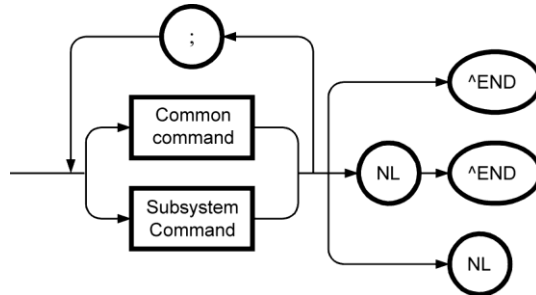
Using multiple commands

### Program message

A program message is a combination of one or more IEEE488.2 common commands and SCPI commands, and used to send multiple commands at once.

### Program Message Syntax

The syntax of the program message is shown in Figure 3.13.



**Figure 3.13 Program Message Syntax**

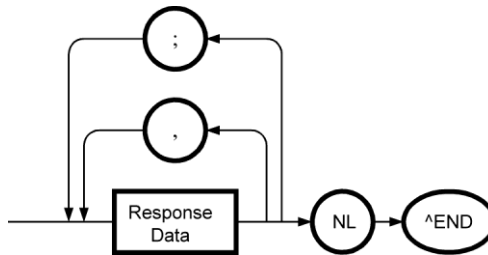
### 3.1.7 Overview of Response Message Formats

#### 3.1.7.1 Response Message

The response data that can be read from the DP020AS when the command is sent by the computer to inquire something is called "response message." The command sent by the computer is called "query", and a question mark (?) is placed at the end of the command.

#### 3.1.7.2 Response Message Syntax

The syntax of the response message is shown in Figure 3.14.



**Figure 3.14 Response Message Syntax**

In response messages, commas (,) and semicolons (;) are used as separators. When multiple values are returned by one command, data items are separated by commas (,).

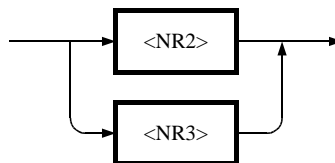
When multiple queries are sent in the same message, the data item groups for each query are separated by semicolons (;).

#### 3.1.7.3 Response Message Data Type

The data type of the response message is one of the followings:

##### **Real number response data syntax (<REAL>)**

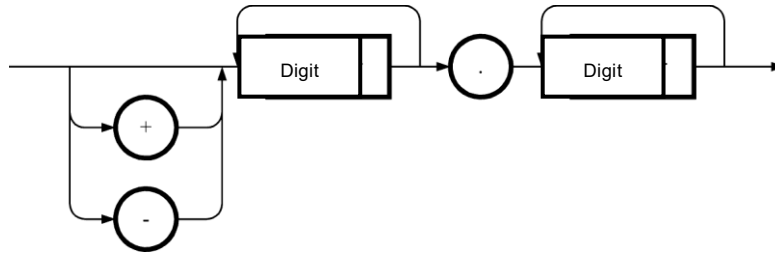
The syntax of the real number response data is shown in Figure 3.15.



**Figure 3.15 Real Number Response Data (<REAL>) Syntax**

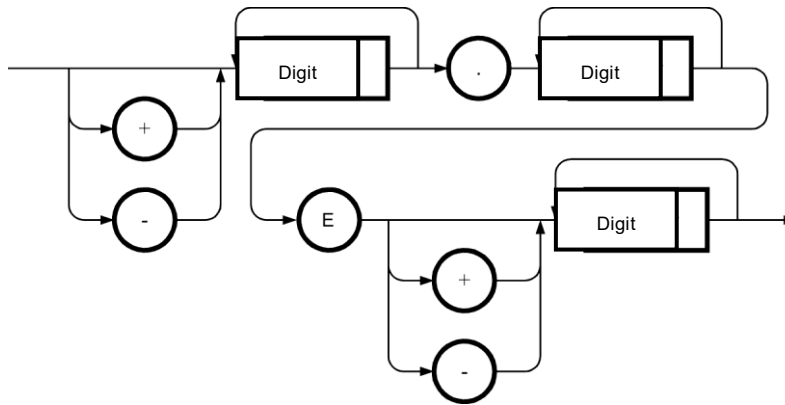
The syntax of the NR2 number response data is shown in Figure 3.16.





**Figure 3.16 NR2 Number Response Data (<NR2>) Syntax**

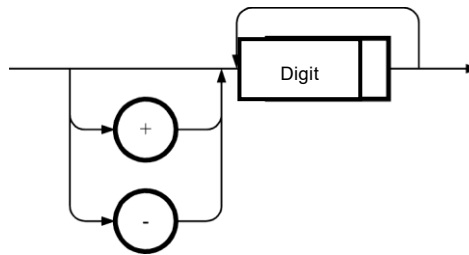
The syntax of the NR3 number response data is shown in Figure 3.17.



**Figure 3.17 NR3 Number Response Data (<NR3>) Syntax**

**Integer response data (<INT>)**

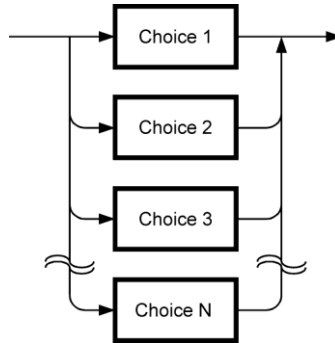
The syntax of the integer response data is shown in Figure 3.18. The integer response data syntax is same as that of the NR1 number response data defined in IEEE488.2.



**Figure 3.18 Integer Response Data (<INT>) Syntax**

**Discrete response data (<DISC>)**

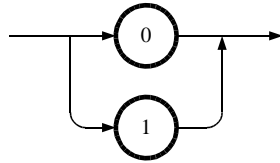
The syntax of the discrete response data is shown in Figure 3.19.  
The response data is always in short form.



**Figure 3.19 Discrete Response Data (<DISC>) Syntax**

**Numeric Boolean value response data (<NBOL>)**

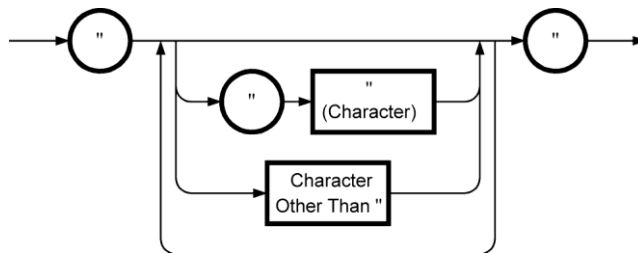
The syntax of the numeric Boolean value response data is shown in Figure 3.20.



**Figure 3.20 Numeric Boolean Value Response Data (<NBOL>) Syntax**

**String response data (<STR>)**

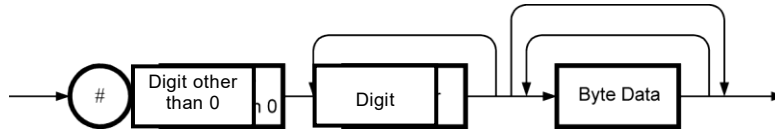
The syntax of the string response data is shown in Figure 3.21.  
Note that double quotation (") is never contained in a response string in the DP020AS.



**Figure 3.21 String Response Data (<STR>) Syntax**

**Definite length arbitrary block response data (<DBLK>)**

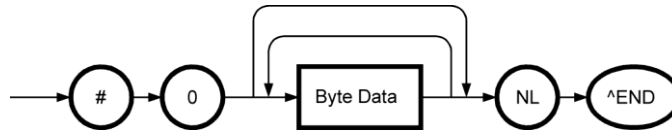
The syntax of the definite length arbitrary block response data is shown in Figure 3.22. However, the DP020AS does not return this type of response message.



**Figure 3.22 Definite Length Arbitrary Block Response Data Syntax (<DBLK>)**

**Indefinite length arbitrary block response data (<IBLK>)**

The syntax of the indefinite length arbitrary block response data is shown in Figure 3.23. However, the DP020AS does not return this type of response message.



**Figure 3.23 Indefinite Length Arbitrary Block Response Data Syntax (<IBLK>)**



## 4. Remote Command Reference

|                                       |    |
|---------------------------------------|----|
| 4.1 Program Command List.....         | 52 |
| 4.2 Program Command Description ..... | 67 |

## 4.1 Program Command List

This chapter describes the program commands for the DP020AS.

Some commands may be given permission or inhibition of acceptance for each output function. Some other commands may be handled as a command error or may not be accepted, to maintain the integrity of the internal operation state of the DP020AS. In addition, some commands may not be run depending on the output phase mode supported by the purchased product.

For these commands that require special attentions, tips are described to run them properly in "Program Command Description".

## 4.1.1 List of Commands Common to Functions

| Function   | Command                  | Set | Query |
|--|--------------------------|-----|-------|
| <b>Common Command (IEEE488.2)</b>                    |                          |     |       |
| Device ID  | *IDN?                    | —   | ○     |
| Initialize the basic setting                         | *RST                     | ○   | —     |
| Self-Check   | *TST?                    | —   | ○     |
| Clear event register and error queue                 | *CLS                     | ○   | —     |
| Set/get the standard event status enable register    | *ESE                     | ○   | ○     |
| Get the standard event status register               | *ESR?                    | —   | ○     |
| Set/get the service request enable register          | *SRE                     | ○   | ○     |
| Get the status byte register                         | *STB?                    | —   | ○     |
| Wait for synchronization of the overlap command      | *WAI                     | ○   | —     |
| Set the OPC bit at the operation completion          | *OPC                     | ○   | —     |
| Get the operation completion state                   | *OPC?                    | —   | ○     |
| <b>Operate System Lock Condition Register</b>        |                          |     |       |
| Get the System Lock condition register               | :STATus:LOCK:CONDition?  | —   | ○     |
| Set/get the System Lock event enable register        | :STATus:LOCK:ENABle      | ○   | ○     |
| Get the System Lock event register                   | :STATus:LOCK[:EVENT]?    | —   | ○     |
| Set/get the System Lock transition filter (negative) | :STATus:LOCK:NTRansition | ○   | ○     |
| Set/get the System Lock transition filter (positive) | :STATus:LOCK:PTRansition | ○   | ○     |

| <b>Operate Operation Condition Register</b>      |                               |   |   |
|--|-------------------------------|---|---|
| Operation condition register                     | :STATus:OPERation:CONDition?  | — | ○ |
| Operation event enable register                  | :STATus:OPERation:ENABle      | ○ | ○ |
| Operation event register                         | :STATus:OPERation[:EVENT]?    | — | ○ |
| Operation transition filter (negative)           | :STATus:OPERation:NTRansition | ○ | ○ |
| Operation transition filter (positive)           | :STATus:OPERation:PTRansition | ○ | ○ |
| <b>Get Error Message</b>                         |                               |   |   |
| Get error messages                               | :SYSTem:ERRor?                | — | ○ |
| <b>Get Power On Error Message</b>                |                               |   |   |
| Get Power on error messages                      | :SYSTem:MESSage?              | — | ○ |
| <b>Operate Warning Condition Register</b>        |                               |   |   |
| Get the warning condition register               | :STATus:WARning:CONDition?    | — | ○ |
| Set/get the warning event enable register        | :STATus:WARning:ENABle        | ○ | ○ |
| Get the warning event register                   | :STATus:WARning[:EVENT]?      | — | ○ |
| Set/get the warning transition filter (negative) | :STATus:WARning:NTRansition   | ○ | ○ |
| Set/get the warning transition filter (positive) | :STATus:WARning:PTRansition   | ○ | ○ |
| Release the warning                              | :SYSTem:WRELease              | ○ | — |
| <b>Output Control</b>                            |                               |   |   |
| Set/get the output on/off                        | :OUTPut[:STATe]               | ○ | ○ |
| Set/get the output at power-on                   | :OUTPut:PON                   | ○ | ○ |
| Set/get the output relay control                 | :OUTPut:RELAy                 | ○ | ○ |
| Set/get the high impedance output off            | :OUTPut:OFFImpedance          | ○ | ○ |
| Set/get the trigger output polarity              | :TRIGger:POLarity             | ○ | ○ |
| Set/get the trigger output pulse width           | :TRIGger:WIDTh                | ○ | ○ |



| <b>Device Configuration</b>  |   |                       |                       |
|--|---|-----------------------|-----------------------|
| Set/get the phase configuration  | :SYSTem:CONFIgure:PHASe                 | <input type="radio"/> | <input type="radio"/> |
| Set/get L3 phase power on  | :SYSTem:CONFIgure:PHASe:THRee:ENABLE    | <input type="radio"/> | <input type="radio"/> |
| <b>Set/Get Output Function</b>   |   |                       |                       |
| Set/Get the L3 phase invalid   | :SYSTem:CONFIgure[:MODE]                | <input type="radio"/> | <input type="radio"/> |
| <b>Set/Get CV/CC Mode</b>  |   |                       |                       |
| Set/Get the CV/CC Mode   | :SYSTem:CONFIgure:CONStant[:MODE]       | <input type="radio"/> | <input type="radio"/> |
| <b>Set Panel Display Operation</b>   |   |                       |                       |
| Set/get the LCD BRIGHTness value   | :DISPlay:CONTRast   BRIGHtness          | <input type="radio"/> | <input type="radio"/> |
| Set/get the Key lock   | :SYSTem:KLOCK                           | <input type="radio"/> | <input type="radio"/> |
| Operation Beep   | :SYSTem:BEEPer:STATe                    | <input type="radio"/> | <input type="radio"/> |
| Limiter Beep   | :SYSTem:BEEPer:LIMit:STATe              | <input type="radio"/> | <input type="radio"/> |
| <b>Get Power Unit State</b>  |   |                       |                       |
| Get the number of connected unit   | :SYSTem:CONFIgure:NPU[:STATe]?          | —                     | <input type="radio"/> |
| Get the error status   | :SYSTem:CONFIgure:NPU:ERRor?            | —                     | <input type="radio"/> |
| <b>Set Monitor Output</b>  |   |                       |                       |
| Set/get the monitor output   | :OUTPut:MONitor:MODE                    | <input type="radio"/> | <input type="radio"/> |
| <b>Current Limiter</b>   |   |                       |                       |
| Set/get the current limiter (effective value)  | [:SOURce]:CURRent:LIMit:RMS[:AMPLitude] | <input type="radio"/> | <input type="radio"/> |
| Set/get off after the current limiter (effective value) is activated                         | [:SOURce]:CURRent:LIMit:RMS:MODE        | <input type="radio"/> | <input type="radio"/> |
| Set/get the time before turning off after the current limiter (effective value) is activated | [:SOURce]:CURRent:LIMit:RMS:TIME        | <input type="radio"/> | <input type="radio"/> |
| Set/get the current limiter (peak value <maximum>)   | [:SOURce]:CURRent:LIMit:PEAK:HIGH       | <input type="radio"/> | <input type="radio"/> |
| Set/get the current limiter (peak value <minimum>)   | [:SOURce]:CURRent:LIMit:PEAK:LOW        | <input type="radio"/> | <input type="radio"/> |
| Set/get off after the current limiter (peak value) is activated                              | [:SOURce]:CURRent:LIMit:PEAK:MODE       | <input type="radio"/> | <input type="radio"/> |

|  |  |                       |                       |
|--|--|-----------------------|-----------------------|
| Set/get the time before turning off after the current limit (peak value) is activated        | [:SOURce]:CURRent:LIMit:PEAK:TIME          | <input type="radio"/> | <input type="radio"/> |
| <b>Voltage Limiter</b>   |  |                       |                       |
| Set/get the voltage limiter (effective value)  | [:SOURce]:CC:VOLTage:LIMit:RMS[:AMPLitude] | <input type="radio"/> | <input type="radio"/> |
| Set/get off after the voltage limiter (effective value) is activated                         | [:SOURce]:CC:VOLTage:LIMit:RMS:MODE        | <input type="radio"/> | <input type="radio"/> |
| Set/get the time before turning off after the voltage limiter (effective value) is activated | [:SOURce]:CC:VOLTage:LIMit:RMS:TIME        | <input type="radio"/> | <input type="radio"/> |
| Set/get the voltage limiter (peak value <maximum>)   | [:SOURce]:CC:VOLTage:LIMit:PEAK:HIGH       | <input type="radio"/> | <input type="radio"/> |
| Set/get the voltage limiter (peak value <minimum>)   | [:SOURce]:CC:VOLTage:LIMit:PEAK:LOW        | <input type="radio"/> | <input type="radio"/> |
| Set/get off after the voltage limiter (peak value) is activated                              | [:SOURce]:CC:VOLTage:LIMit:PEAK:MODE       | <input type="radio"/> | <input type="radio"/> |
| Set/get the time before turning off after the voltage limit (peak value) is activated        | [:SOURce]:CC:VOLTage:LIMit:PEAK:TIME       | <input type="radio"/> | <input type="radio"/> |
| <b>Output Setting (for All Phases)</b>   |  |                       |                       |
| Set/get the output range(CV)*1   | [:SOURce]:VOLTage:RANGe                    | <input type="radio"/> | <input type="radio"/> |
| Set/get the output range(CC)*1   | [:SOURce]:CURRent:RANGe                    | <input type="radio"/> | <input type="radio"/> |
| Set/get the AC/DC mode*1   | [:SOURce]:MODE                             | <input type="radio"/> | <input type="radio"/> |
| <b>CLIP Waveform Setting</b>   |  |                       |                       |
| Set/get the clipped sine wave type specification   | [:SOURce]:FUNCTion:CSINE:TYPE              | <input type="radio"/> | <input type="radio"/> |
| Set/get the clipped sine wave crest factor   | [:SOURce]:FUNCTion:CSINE:CFACTOR           | <input type="radio"/> | <input type="radio"/> |

## 4.1 Program Command List

|  |                               |   |   |
|--|-------------------------------|---|---|
| Set/get the clipped sine wave clip ratio | [:SOURce]:FUNCtion:CSINE:CLIP | ○ | ○ |
| <b>ARB Waveform Setting(Binary)</b>      |                               |   |   |
| Initialize the arbitrary wave memory     | :TRACe DATA:WAVE:CLEAr        | ○ | — |
| Transfer the arbitrary wave data(Binary) | :TRACe DATA:WAVE[:DATA]       | ○ | — |
| Set/get the arbitrary wave name          | :TRACe DATA:WAVE:NAME         | ○ | ○ |
| <b>ARB Waveform Setting(ASCII)</b>       |                               |   |   |
| Transfer the arbitrary wave data (ASCII) | :TRACe DATA:WAVE:ASCii        | ○ | — |
| Set arbitrary wave data to wave memory   | :TRACe DATA:WAVE:BUFFer:SET   | ○ | — |
| <b>Factory Setting</b>                   |                               |   |   |
| Execute factory set                      | :SYSTem:INIT                  | ○ | — |
| <b>Get CC Mode Enable/Disable</b>        |                               |   |   |
| Get CC Mode Enable/Disable               | :SYSTem:CONFIgure:OPTion?     | — | ○ |
| <b>Date and Time Settings</b>            |                               |   |   |
| Set/get Date and Time                    | :SYSTem:DATE                  | ○ | ○ |

\*1 The setting is held by each output function.

After the output function is switched, the setting held by the new output function is forcibly used.

## 4.1.2 List of Continuous Function Commands

| Function  | Command   | Set | Query |
|---|---|-----|-------|
| <b>Common Command (IEEE488.2)</b>                                 |   |     |       |
| Read the basic setting data                                       | *RCL  | ○   | —     |
| Save the basic setting data                                       | *SAV  | ○   | —     |
| <b>Set Panel Display</b>  |   |     |       |
| Set/get the measurement display phase (phase number)              | :DISPlay[:WINDow]:INSTrument:NSElect              | ○   | ○     |
| Set/get the measurement display phase (phase)                     | :DISPlay[:WINDow]:INSTrument:SElect               | ○   | ○     |
| Set/get the measurement display selection                         | :DISPlay[:WINDow]:MEASure:MODE                    | ○   | ○     |
| <b>Output Setting (for All Phases)</b>                            |   |     |       |
| Set/get the output frequency                                      | [:SOURce]:FREQuency[:IMMediate]                   | ○   | ○     |
| Set the output on phase   | [:SOURce]:PHASe:STARt[:IMMediate]                 | ○   | ○     |
| Enable/disable the output off phase specification                 | [:SOURce]:PHASe:STOP:ENABLE                       | ○   | ○     |
| Set the output off phase  | [:SOURce]:PHASe:STOP[:IMMediate]                  | ○   | ○     |
| Set/get the balanced/unbalanced mode                              | [:SOURce]:PHASe:MODE                              | ○   | ○     |
| Set/get the remote sensing function                               | :MEASure:CONFigure:SENSing                        | ○   | ○     |
| <b>Output Setting (for Specific Phase)</b>                        |   |     |       |
| Set/get the phase to set (phase number)                           | :INSTrument:NSElect                               | ○   | ○     |
| Set/get the phase to set (phase)                                  | :INSTrument:SElect                                | ○   | ○     |
| Set/get the AC output voltage                                     | [:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude] | ○   | ○     |
| Set/get the DC output voltage                                     | [:SOURce]:VOLTage[:LEVel][:IMMediate]:OFFSet      | ○   | ○     |
| Set/get the AC output current                                     | [:SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude] | ○   | ○     |
| Set/get the DC output current                                     | [:SOURce]:CURRent[:LEVel][:IMMediate]:OFFSet      | ○   | ○     |
| Set/get the output waveform                                       | [:SOURce]:FUNctioN[:SHAPE][:IMMediate]            | ○   | ○     |
| Set/get the phase angle at polyphase system                       | [:SOURce]:PHASe:PHASe                             | ○   | ○     |
| Set/get the DC offset voltage adjustment value (AC and ACHF mode) | [:SOURce]:VOLTage:ADJust:OFFSet:AC                | ○   | ○     |

|  |                                       |   |   |
|--|---------------------------------------|---|---|
| Set/get the DC offset voltage adjustment value (ACDC and DC modes) | [:SOURce]:VOLTage:ADJust:OFFSet:DC    | ○ | ○ |
| Set/get the DC offset current adjustment value (AC and ACHF mode)  | [:SOURce]:CURRent:ADJust:OFFSet:AC    | ○ | ○ |
| Set/get the DC offset current adjustment value (ACDC and DC modes) | [:SOURce]:CURRent:ADJust:OFFSet:DC    | ○ | ○ |
| <b>Voltage Setting Limit</b>                                       |                                       |   |   |
| Set/get the voltage setting limit (effective value)                | [:SOURce]:VOLTage:LIMit:RMS           | ○ | ○ |
| Set/get the voltage setting limit (peak value <maximum>)           | [:SOURce]:VOLTage:LIMit:HIGH          | ○ | ○ |
| Set/get the voltage setting limit (peak value <minimum>)           | [:SOURce]:VOLTage:LIMit:LOW           | ○ | ○ |
| <b>Current Setting Limit</b>                                       |                                       |   |   |
| Set/get the current setting limit (effective value)                | [:SOURce]:CC:CURRent:RANGe:Limit:RMS  | ○ | ○ |
| Set/get the current setting limit (peak value <maximum>)           | [:SOURce]:CC:CURRent:RANGe:Limit:HIGH | ○ | ○ |
| Set/get the current setting limit (peak value <minimum>)           | [:SOURce]:CC:CURRent:RANGe:Limit:LOW  | ○ | ○ |
| <b>Frequency Setting Range Limit</b>                               |                                       |   |   |
| Set/get the upper limit value of the frequency setting range limit | [:SOURce]:FREQUency:LIMit:HIGH        | ○ | ○ |
| Set/get the lower limit value of the frequency setting range limit | [:SOURce]:FREQUency:LIMit:LOW         | ○ | ○ |
| <b>Get Output Measured Value</b>                                   |                                       |   |   |
| Set/get the measurement query phase (phase number)                 | :MEASure:INSTrument:NSElect           | ○ | ○ |
| Set/get the measurement query phase (phase)                        | :MEASure:INSTrument:SElect            | ○ | ○ |
| Clear the output current peak-hold value                           | :MEASure[:SCALar]:CURRent:PEAK:CLEar  | ○ | — |
| Output current peak-hold value                                     | :MEASure[:SCALar]:CURRent:PEAK:HOLD?  | — | ○ |

#### 4.1 Program Command List

|   |   |   |   |
|---|---|---|---|
| Clear the output voltage peak-hold value                  | :MEASure[:SCALar]:VOLTage:PEAK:CLEAr      | ○ | — |
| Output voltage peak-hold value                            | :MEASure[:SCALar]:VOLTage:PEAK:HOLD?      | — | ○ |
| Synchronization signal source frequency                   | :MEASure[:SCALar]:FREQuency?              | — | ○ |
| Output voltage (effective value)                          | :MEASure[:SCALar]:VOLTage[:RMS]?          | — | ○ |
| Output voltage (average value)                            | :MEASure[:SCALar]:VOLTage:AVErage?        | — | ○ |
| Output voltage (peak value <maximum>)                     | :MEASure[:SCALar]:VOLTage:HIGH?           | — | ○ |
| Output voltage (peak value <minimum>)                     | :MEASure[:SCALar]:VOLTage:LOW?            | — | ○ |
| Output voltage crest factor                               | :MEASure[:SCALar]:VOLTage:CFACTOR?        | — | ○ |
| Output current (effective value)                          | :MEASure[:SCALar]:CURRent[:RMS]?          | — | ○ |
| Output current (average value)                            | :MEASure[:SCALar]:CURRent:AVErage?        | — | ○ |
| Output current (peak value <maximum>)                     | :MEASure[:SCALar]:CURRent:HIGH?           | — | ○ |
| Output current (peak value <minimum>)                     | :MEASure[:SCALar]:CURRent:LOW?            | — | ○ |
| Set/get harmonic measurement type                         | :MEASure[:SCALar]:HARMonic:TYPE           | ○ | ○ |
| Harmonic voltage value                                    | :MEASure[:SCALar]:VOLTage:HARMonic[:RMS]? | — | ○ |
| Harmonic voltage percentage                               | :MEASure[:SCALar]:VOLTage:HARMonic:RATio? | — | ○ |
| Set/get Control the harmonic current measurement function | :MEASure[:SCALar]:CURRent:HARMonic:ENABle | ○ | ○ |
| Harmonic current value                                    | :MEASure[:SCALar]:CURRent:HARMonic[:RMS]? | — | ○ |
| Harmonic current percentage                               | :MEASure[:SCALar]:CURRent:HARMonic:RATio? | — | ○ |
| Output current crest factor                               | :MEASure[:SCALar]:CURRent:CFACTOR?        | — | ○ |
| Apparent power  | :MEASure[:SCALar]:POWEr[:AC]:APPArEnt?    | — | ○ |
| Power factor  | :MEASure[:SCALar]:POWEr[:AC]:PFACTOR?     | — | ○ |
| Active power  | :MEASure[:SCALar]:POWEr[:AC][:REAL]?      | — | ○ |
| <b>AGC/Autocal Function Setting</b>                       |   |   |   |
| Enable/disable the AGC function                           | :OUTPut:AGC                               | ○ | ○ |
| Enable/disable the Autocal function                       | :OUTPut:ACALibration                      | ○ | ○ |
| <b>Set External Input Signal</b>                          |   |   |   |
| Set/get the external input gain                           | :INPut:GAIN                               | ○ | ○ |
| Set/get the external synchronization signal source        | :INPut:SYNC:SOURce                        | ○ | ○ |
| <b>Set External Control Signal</b>                        |   |   |   |
| Permit/inhibit the  | :SYSTem:CONFIgure:EXTio[:STATe]           | ○ | ○ |

## 4.1 Program Command List

|  |                                  |   |   |
|--|----------------------------------|---|---|
| external control input                             |                                  |   |   |
| Set/get the external control state output polarity | :SYSTem:CONFigure:EXTio:POLarity | ○ | ○ |
| <b>Set Soft Start / Soft Stop</b>                  |                                  |   |   |
| Enable/disable soft start function                 | :OUTPut:SStart[:STATe][:RISE]    | ○ | ○ |
| Set/get soft start time                            | :OUTPut:SStart:TIME[:RISE]       | ○ | ○ |
| Enable/disable soft stop function                  | :OUTPut:SStart[:STATe]:FALL      | ○ | ○ |
| Set/get soft stop time                             | :OUTPut:SStart:TIME:FALL         | ○ | ○ |

## 4.1.3 List of Sequence Function Commands

| Function   | Command   | Set | Query |
|--|---|-----|-------|
| <b>Set Panel Display</b>   |   |     |       |
| Set/get the Step Time display unit                               | :DISPlay[:WINDow]:TIME:UNIT                                 | ○   | ○     |
| <b>Sequence Function Control</b>                                 |   |     |       |
| Get the state  | [:SOURce]:SEQuence:CONTRol[:STATe]?                         | —   | ○     |
| Control the execution  | :TRIGger:SEQuence:SELEcted:EXECute                          | ○   | —     |
| Switch to the Edit mode  | [:SOURce]:SEQuence:EDIT                                     | ○   | —     |
| Compile the setting parameter and switch to the Control mode     | :TRIGger:SEQuence:COMPile                                   | ○   | —     |
| Get the running step number                                      | [:SOURce]:SEQuence:CSTep?                                   | —   | ○     |
| <b>Set Parameter Common to Steps (Enabled at Edit Mode)</b>      |   |     |       |
| Set/get the Step-control parameter                               | [:SOURce]:SEQuence:CPARameter                               | ○   | ○     |
| <b>Set Parameter for Specified Step (Enabled at Edit Mode)</b>   |   |     |       |
| Set/get the step number to set                                   | [:SOURce]:SEQuence:STEP                                     | ○   | ○     |
| Set/get the phase to set (phase number)                          | [:SOURce]:SEQuence:INSTrument:NSELEct                       | ○   | ○     |
| Set/get the phase to set (phase)                                 | [:SOURce]:SEQuence:INSTrument:SELEct                        | ○   | ○     |
| Set/get the intra-Step parameter                                 | [:SOURce]:SEQuence:SPARameter                               | ○   | ○     |
| <b>Initialize/Read/Save Sequence Data (Enabled at Edit Mode)</b> |   |     |       |
| Initialize the Sequence data                                     | :TRACe DATA:SEQuence:CLEAr                                  | ○   | —     |
| Set/get the sequence data name                                   | :TRACe DATA:SEQuence:NAME                                   | ○   | ○     |
| Read the Sequence data   | :TRACe DATA:SEQuence:RECall                                 | ○   | —     |
| Save the Sequence data   | :TRACe DATA:SEQuence:STORE                                  | ○   | —     |
| <b>Set Parameter for Step 0</b>                                  |   |     |       |
| Set/get the output frequency                                     | [:SOURce]:SEQuence:FREQuency[:IMM ediate]                   | ○   | ○     |
| Set/get the AC output voltage                                    | [:SOURce]:SEQuence:VOLTage[:LEVel][:IMM ediate][:AMPLitude] | ○   | ○     |
| Set/get the DC output voltage                                    | [:SOURce]:SEQuence:VOLTage[:LEVel][:IMM ediate]:OFFSet      | ○   | ○     |
| Set/get the AC output current                                    | [:SOURce]:SEQuence:CURRent[:LEVel][:IMM ediate][:AMPLitude] | ○   | ○     |
| Set/get the DC output current                                    | [:SOURce]:SEQuence:CURRent[:LEVel][:IMM ediate]:OFFSet      | ○   | ○     |
| <b>Remote Sensing</b>  |   |     |       |
| Set/get the remote sensing function                              | :MEASure:CONFIgure:SENSing                                  | ○   | ○     |



## 4.1 Program Command List

| <b>Autocal Function Setting</b>     |                      |                       |                       |
|-------------------------------------|----------------------|-----------------------|-----------------------|
| Enable/disable the Autocal function | :OUTPut:ACALibration | <input type="radio"/> | <input type="radio"/> |

## 4.1.4 List of Simulation Function Commands

| Function  | Command  | Set | Query |
|---|--|-----|-------|
| <b>Set Panel Display</b>                                      |  |     |       |
| Set/get the Step Time display unit                            | :DISPlay[:WINDow]:TIME:UNIT                          | ○   | ○     |
| <b>Simulation Function Control</b>                            |  |     |       |
| Get the state   | [:SOURce]:SIMulation:CONTRol[:STATe]?                | —   | ○     |
| Control the execution   | :TRIGger:SIMulation:SELEcted:EXECute                 | ○   | —     |
| Switch to the Edit mode                                       | [:SOURce]:SIMulation:EDIT                            | ○   | —     |
| Compile the setting parameter and switch to the Control mode  | :TRIGger:SIMulation:COMPIle                          | ○   | —     |
| Get the running step number                                   | [:SOURce]:SIMulation:CSTep?                          | —   | ○     |
| <b>Set Repeat Execution (Enabled at Edit Mode)</b>            |  |     |       |
| Enable/disable the repeat execution                           | [:SOURce]:SIMulation:REPEat:ENABLE                   | ○   | ○     |
| Set/get the repeat execution count                            | [:SOURce]:SIMulation:REPEat:COUNt                    | ○   | ○     |
| <b>Set Parameter for Initial Step (Enabled at Edit Mode)</b>  |  |     |       |
| Set/get the output voltage                                    | [:SOURce]:SIMulation:INITial:VOLTage                 | ○   | ○     |
| Set/get the frequency   | [:SOURce]:SIMulation:INITial:FREQuency               | ○   | ○     |
| Enable/disable the start phase specification                  | [:SOURce]:SIMulation:INITial:PHASe:STARt:ENABLE      | ○   | ○     |
| Set/get the start phase                                       | [:SOURce]:SIMulation:INITial:PHASe:STARt[:IMMEDIATE] | ○   | ○     |
| Enable/disable the stop phase specification                   | [:SOURce]:SIMulation:INITial:PHASe:STOP:ENABLE       | ○   | ○     |
| Set/get the stop phase  | [:SOURce]:SIMulation:INITial:PHASe:STOP[:IMMEDIATE]  | ○   | ○     |
| Set/get the synchronization output                            | [:SOURce]:SIMulation:INITial:CODE                    | ○   | ○     |
| <b>Set Parameter for Normal 1 Step (Enabled at Edit Mode)</b> |  |     |       |
| Set/get the execution time                                    | [:SOURce]:SIMulation:NORMal1:TIME                    | ○   | ○     |
| Set/get the output voltage                                    | [:SOURce]:SIMulation:NORMal1:VOLTage                 | ○   | ○     |
| Set/get the frequency   | [:SOURce]:SIMulation:NORMal1:FREQuency               | ○   | ○     |
| Enable/disable the start phase specification                  | [:SOURce]:SIMulation:NORMal1:PHASe:STARt:ENABLE      | ○   | ○     |
| Set/get the start phase                                       | [:SOURce]:SIMulation:NORMal1:PHASe:STARt[:IMMEDIATE] | ○   | ○     |

## 4.1 Program Command List

|   |   |   |   |
|---|---|---|---|
| Enable/disable the stop phase specification                             | [:SOURce]:SIMulation:NORMal1:PHASe:STOP:ENABLE        | ○ | ○ |
| Set/get the stop phase  | [:SOURce]:SIMulation:NORMal1:PHASe:STOP[:IMMEDIATE]   | ○ | ○ |
| Set/get the synchronization output                                      | [:SOURce]:SIMulation:NORMal1:CODE                     | ○ | ○ |
| Set/get the trigger output (G1)   | [:SOURce]:SIMulation:NORMal1:TRIGger[:STATe]          | ○ | ○ |
| <b>Set Parameter for Trans 1 Step (Enabled at Edit Mode)</b>            |   |   |   |
| Set/get the execution time  | [:SOURce]:SIMulation:TRANSition1:TIME                 | ○ | ○ |
| Set/get the synchronization output                                      | [:SOURce]:SIMulation:TRANSition1:CODE                 | ○ | ○ |
| Set/get the trigger output (G2)   | [:SOURce]:SIMulation:TRANSition1:TRIGger[:STATe]      | ○ | ○ |
| <b>Set Parameter for Abnormal Step Parameter (Enabled at Edit Mode)</b> |   |   |   |
| Set/get the execution time  | [:SOURce]:SIMulation:ABNormal:TIME                    | ○ | ○ |
| Set/get the output voltage  | [:SOURce]:SIMulation:ABNormal:VOLTage                 | ○ | ○ |
| Set/get the frequency   | [:SOURce]:SIMulation:ABNormal:FREQuency               | ○ | ○ |
| Enable/disable the start phase specification                            | [:SOURce]:SIMulation:ABNormal:PHASe:START:ENABLE      | ○ | ○ |
| Set/get the start phase   | [:SOURce]:SIMulation:ABNormal:PHASe:START[:IMMEDIATE] | ○ | ○ |
| Enable/disable the stop phase specification                             | [:SOURce]:SIMulation:ABNormal:PHASe:STOP:ENABLE       | ○ | ○ |
| Set/get the stop phase  | [:SOURce]:SIMulation:ABNormal:PHASe:STOP[:IMMEDIATE]  | ○ | ○ |
| Set/get the synchronization output                                      | [:SOURce]:SIMulation:ABNormal:CODE                    | ○ | ○ |
| Set/get the trigger output (G3)   | [:SOURce]:SIMulation:ABNormal:TRIGger[:STATe]         | ○ | ○ |
| <b>Set Parameter for Trans 2 Step (Enabled at Edit Mode)</b>            |   |   |   |
| Set/get the execution time  | [:SOURce]:SIMulation:TRANSition2:TIME                 | ○ | ○ |
| Set/get the synchronization output                                      | [:SOURce]:SIMulation:TRANSition2:CODE                 | ○ | ○ |
| Set/get the trigger output (G4)   | [:SOURce]:SIMulation:TRANSition2:TRIGger[:STATe]      | ○ | ○ |
| <b>Set Parameter for Normal 2 Step (Enabled at Edit Mode)</b>           |   |   |   |
| Set/get the execution time  | [:SOURce]:SIMulation:NORMal2:TIME                     | ○ | ○ |
| Enable/disable the start phase specification                            | [:SOURce]:SIMulation:NORMal2:PHASe:START:ENABLE       | ○ | ○ |

#### 4.1 Program Command List

|   |  |   |   |
|---|--|---|---|
| Set/get the start phase                                       | [:SOURce]:SIMulation:NORMal2:PHASe:START[:IMMediate] | ○ | ○ |
| Enable/disable the stop phase specification                   | [:SOURce]:SIMulation:NORMal2:PHASe:STOP:ENABLE       | ○ | ○ |
| Set/get the stop phase  | [:SOURce]:SIMulation:NORMal2:PHASe:STOP[:IMMediate]  | ○ | ○ |
| Set/get the synchronization output                            | [:SOURce]:SIMulation:NORMal2:CODE                    | ○ | ○ |
| Set/get the trigger output (G5)                               | [:SOURce]:SIMulation:NORMal2:TRIGger[:STATe]         | ○ | ○ |
| <b>Initialize/Read/Save Simulation (Enabled at Edit Mode)</b> |  |   |   |
| Initialize the Simulation data                                | :TRACe DATA:SIMulation:CLEar                         | ○ | — |
| Set/get the simulation data name                              | :TRACe DATA:SIMulation:NAME                          | ○ | ○ |
| Read the Simulation data                                      | :TRACe DATA:SIMulation:RECall                        | ○ | — |
| Save the Simulation data                                      | :TRACe DATA:SIMulation:STORE                         | ○ | — |

## 4.2 Program Command Description

### 4.2.1 Commands Common to Functions

#### 4.2.1.1 Common Command (IEEE488.2): Device ID

\*IDN?

Description

Gets the device ID

Query parameter

None

Response format

<STR>

Query example

\*IDN?

Response example

NF Corporation,DP020AS,1234567,1.00

#### 4.2.1.2 Common Command (IEEE488.2): Initialize Basic Setting

\*RST

Description

Initializes the setting

Setting parameter

None

Setting example

\*RST

Remarks

The operation cannot be done during the output on.

**4.2.1.3** Common Command (IEEE488.2): Self-Check

\*TST?

Description

Gets the self-check result

For the DP020AS, it always returns 0 without performing the self-check

Query parameter

None

Response format

<INT>

Query example

\*TST?

Response example

0

**4.2.1.4** Common Command (IEEE488.2): Clear Event Register and Error Queue

\*CLS

Description

Clears event register and error queue

For the event register, see **2.2.2**, **2.2.4.1**, **2.2.5.3**, **2.2.6.3**, and **2.2.7.3**

Setting parameter

None

Setting example

\*CLS

**4.2.1.5** Common Command (IEEE488.2): Set/Get Standard Event Status Enable Register

\*ESE

Description

Sets/Gets the standard event status enable register

For the standard event status enable register, see **2.2.4.2**

Setting parameter

<value> ::= <INT>

<INT> → Register setting value: 0 to 255

Query parameter

None

Response format

<INT>

Setting example

\*ESE 8

Query example

\*ESE?

Response example

8

**4.2.1.6** Common Command (IEEE488.2): Get Standard Event Status Register

\*ESR?

Description

Gets the standard event status register

For the standard event status register, see 2.2.4.1

Query parameter

None

Response format

<INT>

Query example

\*ESR?

Response example

8

**4.2.1.7** Common Command (IEEE488.2): Set/Get Service Request Enable Register

\*SRE

**Description**

Sets/gets the service request enable register

For the service request enable register, see **2.2.3**

**Setting parameter**

<value> ::= <INT>

<INT> → Register setting value: 0 to 255

**Query parameter**

None

**Response format**

<INT>

**Setting example**

\*SRE 8

**Query example**

\*SRE?

**Response example**

8

**4.2.1.8** Common Command (IEEE488.2): Get Status Byte Register

\*STB?

**Description**

Gets the status byte register

For the status byte register, see 2.2.2

**Query parameter**

None

**Response format**

<INT>

**Query example**

\*STB?

**Response example**

8



**4.2.1.9** Common Command (IEEE488.2): Wait for Synchronization of Overlap Command

\*WAI

**Description**

Waits for completion of all the previous commands.

DP020AS does not have overlap command.

Setting parameter

None

**Setting example**

\*WAI

**4.2.1.10** Common Command (IEEE488.2): Set OPC Bit at Operation Completion

\*OPC

**Description**

Sets the OPC bit to 1 when all the previous commands are completed

DP020AS does not have overlap command

For the OPC bit, see 2.2.4.1

**Setting parameter**

None

**Setting example**

\*OPC

**4.2.1.11** Common Command (IEEE488.2): Get Operation Completion State

\*OPC?

**Description**

Sets the output buffer to 1 when all the previous commands are completed

DP020AS does not have overlap command

**Query parameter**

None

**Response format**

<INT>

**Query example**

\*OPC?

**Response example**

1

**4.2.1.12 Operate System Lock Condition Register: System Lock Condition Register**  
:STATus:LOCK:CONDition?**Description**

Gets the System Lock condition register

For the System Lock condition register, see **2.2.7.1**

**Query parameter**

None

**Response format**

<INT>

**Query example**

STAT:LOCK:COND?

**Response example**

1

**4.2.1.13 Operate System Lock Condition Register: System Lock Event Enable Register**  
:STATus:LOCK:ENABLe**Description**

Sets/gets the System Lock event enable register

For the System Lock event enable register, see **2.2.7.4**

**Setting parameter**

<value> ::= <INT>

<INT> → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**

<INT>

**Setting example**

STAT:LOCK:ENAB 1

**Query example**

STAT:LOCK:ENAB?

**Response example**

1

**4.2.1.14 Operate System Lock Condition Register: System Lock Event Register**`:STATus:LOCK[:EVENT]?`**Description**

Gets the System Lock event register

For the System Lock event register, see **2.2.7.3**

**Query parameter**

None

**Response format**

<INT>

**Query example**

STAT:LOCK?

**Response example**

16384

**4.2.1.15 Operate System Lock Condition Register: System Lock Transition Filter (Negative)**`:STATus:LOCK:NTRansition`**Description**

Sets/gets the System Lock transition filter (negative)

For the System Lock transition filter, see 2.2.7.2

**Setting parameter**

<value> ::= <INT>

<INT> → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**

<INT>

**Setting example**

STAT:LOCK:NTR 16384

**Query example**

STAT:LOCK:NTR?

**Response example**

16384

**4.2.1.16 Operate System Lock Condition Register: System Lock Transition Filter (Positive)**

:STATus:LOCK:PTRansition

**Description**

Sets/gets the System Lock transition filter (positive)

For the System Lock transition filter, see 2.2.7.2

**Setting parameter**

<value> ::= <INT>

<INT> → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**

<INT>

**Setting example**

STAT:LOCK:PTR 16384

**Query example**

STAT:LOCK:PTR?

**Response example**

16384

**4.2.1.17 Operate Operation Condition Register: Operation Condition Register**

:STATus:OPERation:CONDition?

**Description**

Gets the operation condition register

For the operation condition register, see **2.2.5.1**

**Query parameter**

None

**Response format**

<INT>

**Query example**

STAT:OPER:COND?

**Response example**

16384

**4.2.1.18 Operate Operation Condition Register: Operation Event Enable Register**

:STATus:OPERation:ENABle

**Description**

Sets/gets the operation event enable register

For the operation event enable register, see **2.2.5.4****Setting parameter**

&lt;value&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**

&lt;INT&gt;

**Setting example**

STAT:OPER:ENAB 16384

**Query example**

STAT:OPER:ENAB?

**Response example**

16384

**4.2.1.19 Operate Operation Condition Register: Operation Event Register**

:STATus:OPERation[:EVENT]?

**Description**

Gets the operation event register

For the operation event register, see **2.2.5.3****Query parameter**

None

**Response format**

&lt;INT&gt;

**Query example**

STAT:OPER?

**Response example**

16384

**4.2.1.20 Operate Operation Condition Register: Operation Transition Filter (Negative)**

:STATus:OPERation:NTRansition

**Description**

Sets/gets the operation transition filter (negative)

For the operation transition filter, see **2.2.5.2**

**Setting parameter**

<value> ::= <INT>

<INT> → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**

<INT>

**Setting example**

STAT:OPER:NTR 16384

**Query example**

STAT:OPER:NTR?

**Response example**

16384

**4.2.1.21 Operate Operation Condition Register: Operation Transition Filter (Positive)**

:STATus:OPERation:PTRansition

**Description**

Sets/gets the operation transition filter (positive)

For the operation transition filter, see 2.2.5.2

**Setting parameter**

<value> ::= <INT>

<INT> → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**

<INT>

**Setting example**

STAT:OPER:PTR 16384

**Query example**

STAT:OPER:PTR?

**Response example**

16384

**4.2.1.22** Get Error Message: Get Error Message

:SYSTem:ERRor?

**Description**

Gets the error queue output

It returns the error ID and the error message in the order

**Query parameter**

None

**Response format**

<INT>,<STR>

**Query example**

SYST:ERR?

**Response example**

0,"No error"

**4.2.1.23** Get Error Message: Get Error Message

:SYSTem:MESSage?

**Description**

Get error message at startup

When "Press enter key" appears on the screen, this command replaces the Enter key.  
for example, after setting the phase configuration.

**Query parameter**

None

**Response format**

<INT>,<STR>

**Query example**

SYST:MESS?

**Response example**

0,"No error"

**4.2.1.24 Operate Warning Condition Register: Get Warning Condition Register**  
:STATus:WARNing:CONDition?

Description

Gets the warning condition register

For the warning condition register, see **2.2.6.1**

Query parameter

None

Response format

<INT>

Query example

STAT:WARN:COND?

Response example

16384

**4.2.1.25 Operate Warning Condition Register: Set/Get Warning Event Enable Register**  
:STATus:WARNing:ENABLE

Description

Sets/gets the warning event enable register

For the warning event enable register, see **2.2.6.4**

Setting parameter

<value> ::= <INT>

<INT> → Register setting value: 0 to 65535

Query parameter

None

Response format

<INT>

Setting example

STAT:WARN:ENAB 16384

Query example

STAT:WARN:ENAB?

Response example

16384



**4.2.1.26 Operate Warning Condition Register: Get Warning Event Register**`:STATus:WARNing[:EVENT]?`**Description**

Gets the warning event register

For the warning event register, see 2.2.6.3

**Query parameter**

None

**Response format**

<INT>

**Query example**

STAT:WARN?

**Response example**

16384

**4.2.1.27 Operate Warning Condition Register: Set/Get Warning Transition Filter (Negative)**`:STATus:WARNing:NTRansition`**Description**

Sets/gets the warning transition filter (negative)

For the warning transition filter, see **2.2.6.2**

**Setting parameter**

<value> ::= <INT>

<INT> → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**

<INT>

**Setting example**

STAT:WARN:NTR 16384

**Query example**

STAT:WARN:NTR?

**Response example**

16384

**4.2.1.28 Operate Warning Condition Register: Set/Get Warning Transition Filter (Positive)**`:STATus:WARNing:PTRansition`**Description**

Sets/gets the warning transition filter (positive)

For the warning transition filter, see 2.2.6.2

**Setting parameter**`<value> ::= <INT>`

<INT> → Register setting value: 0 to 65535

**Query parameter**

None

**Response format**`<INT>`**Setting example**`STAT:WARN:PTR 16384`**Query example**`STAT:WARN:PTR?`**Response example**`16384`**4.2.1.29 Operate Warning Condition Register: Release Warning**`:SYSTem:WRELease`**Description**

Sets the release of the warning

**Setting parameter**

None

**Setting example**`SYST:WREL`**Remarks**

The setting is error during the System Lock.

**4.2.1.30** Output Control: Set/Get Output On/Off

:OUTPut[:STATe]

**Description**

Sets/gets the output on/off

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

OUTP ON

**Query example**

OUTP?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.31** Output Control: Set/Get Output at Power-On

:OUTPut:PON

Description

Sets/gets the output at power-on

Setting parameter

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

Query parameter

None

Response format

<NBOL>

Setting example

OUTP:PON ON

Query example

OUTP:PON?

Response example

1

Remarks

The setting is error during the System Lock/warning.

**4.2.1.32** Output Control: Set/Get Output Relay Control

:OUTPut:RELAy

**Description**

Sets/gets the output relay control

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

OUTP:REL ON

**Query example**

OUTP:REL?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.33** Output Control: Set/Get High Impedance Output Off

:OUTPut:OFFImpedance

Description

Sets/gets high impedance output off function

Setting parameter

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

Query parameter

None

Response format

<NBOL>

Setting example

OUTP:OFFI ON

Query example

OUTP:OFFI?

Response example

1

Remarks

The setting is error during the System Lock/warning.

**4.2.1.34** Output Control: Trigger Set/Get Output Polarity

:TRIGger:POLarity

**Description**

Sets/gets the trigger output polarity

**Setting parameter**

<trig slope> ::= <DISC>

<DISC> ::= POSitive | NEGative

POSitive → Rising

NEGative → Falling

**Query parameter**

None

**Response format**

None

**Setting example**

TRIG:POL POSITIVE

**Query example**

TRIG:POL?

**Response example**

POS

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.35** Output Control: Trigger Set/Get Output Pulse Width

:TRIGger:WIDTh

## Description

Sets/gets the trigger output pulse width

The unit is millisecond (ms)

## Setting parameter

&lt;trig width&gt; | MINimum | MAXimum

&lt;trig width&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Trigger output pulse width: 0.1 to 10.0, Resolution 0.1

MINimum → 0.1

MAXimum → 10.0

## Query parameter

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

## Response format

&lt;NR2&gt;

## Setting example

TRIG:WIDTh 0.5

## Query example

TRIG:WIDTh?

## Response example

0.5

## Remarks

The setting is error during the System Lock/warning.



**4.2.1.36 Device Configuration: Set/Get Phase Configuration**

:SYSTem:CONFigure:PHASe?

**Description**

Gets the phase configuration

Sets the phase configuration (Limited to the model corresponding to change function)

**Setting parameter**

<system phase> ::= <INT>

<INT> → 0: single-phase, 1: single-phase three-wire, 2: three-phase

**Query parameter**

None

**Response format**

<INT> → 0: single-phase, 1: single-phase three-wire, 2: three-phase

**Setting example**

SYST:CONF:PHAS 0

**Query example**

SYST:CONF:PHAS?

**Response example**

0

**Remarks**

Regarding the setting, there are the following restrictions.

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence or Simulation.

The setting cannot be done during the output on.

After changing the phase configuration and turning the power back on,

use the :SYSTem:MESSage? command to receive a message that the phase configuration has changed.

**4.2.1.37** Device Configuration: Set/Get L3 Phase invalid`:SYSTem:CONFigure:PHASe:THRee:ENABle`**Description**

Set/Get L3 Phase invalid

This command switches the L3 phase power ON/OFF when using single-phase 3-wire.

**Setting parameter**

`<state> ::= <BOL>`

`<BOL>` → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

`<NBOL>`

**Setting example**

`:SYST:CONF:PHAS:THR:ENAB ON`

**Query example**

`:SYST:CONF:PHAS:THR:ENAB?`

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

This command can be used when the phase configuration switching option for single-phase 3-wire and no need to restart the power supply is enabled.

**4.2.1.38** Device Configuration: Set/Get Output Function`:SYSTem:CONFigure[:MODE]`**Description**

Sets/gets the output function

**Setting parameter**

<system mode> ::= <DISC>

<DISC> ::= CONTInuous | SEQuence | SIMulation

CONTInuous → Continuous

SEQuence → Sequence

SIMulation → Simulation

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

SYST:CONF SIMULATION

**Query example**

SYST:CONF?

**Response example**

SIM

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done during the output on.

The setting cannot be done in the Sequence control state or Simulation control state.

**4.2.1.39** Device Configuration: Set/Get the CV/CC Mode`:SYSTem:CONFigure:CONStant[:MODE]`**Description**

Sets/gets the CV/CC Mode

**Setting parameter**

<system mode> ::= <DISC>

<DISC> ::= CV | CC

CV → CV Mode

CC → CC Mode

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

SYST:CONF:CONS:MODE CV

**Query example**

SYST:CONF:CONS:MODE?

**Response example**

CV

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done during the output on.

The setting cannot be done in the Sequence or Simulation.

CC mode cannot be set unless CC mode is enabled.

**4.2.1.40** Set Panel Display Operation: Set/Get LCD brightness value

:DISPlay:CONTRast | BRIGHtness

**Description**

Sets/gets the LCD brightness

**Setting parameter**

<contrast> | MINimum | MAXimum

<contrast> ::= <INT>

<INT> → LCD brightness: 0 to 99

MINimum → 0

MAXimum → 99

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

DISP:CONT 55

**Query example**

DISP:CONT?

**Response example**

55

**4.2.1.41 Set Panel Display Operation: Set/Get Key Lock**

:SYSTem:KLOCK

**Description**

Sets/gets the key lock

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SYST:KLOC ON

**Query example**

SYST:KLOC?

**Response example**

1

**4.2.1.42** Set Panel Display Operation: Set/Get Operation Beep

:SYSTem:BEEPer:STATe

**Description**

Sets/gets the operation beep

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SYST:BEEP:STAT ON

**Query example**

SYST:BEEP:STAT?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.43** Set Panel Display Operation: Set/Get Limiter Beep

:SYSTem:BEEPer:LIMit:STATe

**Description**

Sets/gets the limiter beep

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SYST:BEEP:LIM:STAT ON

**Query example**

SYST:BEEP:LIM:STAT?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.



**4.2.1.44** Get State of Power Unit: Get Number of Connected Power Units

:SYSTem:CONFigure:NPU[:STATe]?

**Description**

Obtain the connection configuration of L1 phase, L2 phase, and L3 phase.

The status of each phase is assigned to each bit as follows.

| Bit | Weight | Description  |
|-----|--------|--------------|
| 2   | 4      | Booster 2    |
| 1   | 2      | Booster 1    |
| 0   | 1      | Phase Master |

**Query parameter**

None

**Response format**

&lt;INT&gt;,&lt;INT&gt;,&lt;INT&gt;

The response for the three phases is returned regardless of the phase configuration.

**Query example**

SYST:CONF:NPU?

**Response example**

7,7,7

(shows that phase master/booster 1/booster 2 is connected for each phase of L1, L2, and L3.)

**4.2.1.45** Get State of Power Unit: Get Error State

:SYSTem:CONFigure:NPU:ERRor?

**Description**

Gets the power unit in which an error occurs

It gets the power unit in which an error occurs from the L1, L2, and L3 phases.

The status of each phase is assigned for each bit as follows:

If an error occurs in a power unit, the corresponding bit returns 1.

| Bit | Weight | Description  |
|-----|--------|--------------|
| 2   | 4      | Booster 2    |
| 1   | 2      | Booster 1    |
| 0   | 1      | Phase Master |

**Query parameter**

None

**Response format**

<INT>,<INT>,<INT>

The response for the three phases is returned regardless of the phase configuration.

**Query example**

SYST:CONF:NPU:ERR?

**Response example**

0,6,0

shows

L1 phase: no error

L2 phase:Booster 2 and Booster 1 error

L3 phase: no error

**4.2.1.46** Set Monitor Output: Set/Get Monitor Output

:OUTPut:MONitor:MODE

**Description**

Sets/gets the monitor output

**Setting parameter**

<state> ::= <DISC>

<DISC> ::= CURRent | VOLTage

CURRent → Current monitor output mode

SCURrent → Single Current monitor output mode

VOLTage → Voltage monitor output mode

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

OUTP:MON:MODE CURR

**Query example**

OUTP:MON:MODE?

**Response example**

CURR

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.47 Set Output Current Limit: Set/Get Current Limit (Effective Value)**

[:SOURce]:CURRent:LIMit:RMS[:AMPLitude]

**Description**

Sets/gets the current effective value [Arms] limiter

**Setting parameter**

&lt;cur lim rms&gt; | MINimum | MAXimum

&lt;cur lim rms &gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Output current effective value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CURR:LIM:RMS 5

**Query example**

CURR:LIM:RMS?

**Response example**

5.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.

**4.2.1.48** Set Output Current Limit: Set/Get Output OFF at Current Limit (Effective Value) Detection  
[:SOURce]:CURRent:LIMit:RMS:MODE**Description**

Sets/gets the output off after the RMS current limiter is activated

**Setting parameter**

<cur lim mode> ::= <DISC>

<DISC> ::= CONTInuous | OFF

CONTInuous → Continuous operation, output off disabled

OFF → Output off enabled

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

CURR:LIM:RMS:MODE OFF

**Query example**

CURR:LIM:RMS:MODE?

**Response example**

OFF

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.

**4.2.1.49** Set Output Current Limit: Set/Get Time Before Turning Output OFF After Current Limit (Effective Value) is Detected

[:SOURce]:CURRent:LIMit:RMS:TIME

**Description**

Sets/gets the time [s] before turning the output off after the RMS current limiter is activated

**Setting parameter**

<cur lim time> | MINimum | MAXimum

<cur lim time> ::= <INT>

<INT> → Limiter time: 1 to 10

MINimum → 1

MAXimum → 10

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

CURR:LIM:RMS:TIME 5

**Query example**

CURR:LIM:RMS:TIME?

**Response example**

5

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.

**4.2.1.50** Set Output Current Limit: Set/Get Current Limit (Peak Value <Maximum>)

[:SOURce]:CURRent:LIMit:PEAK:HIGH

**Description**

Sets/gets the peak current limiter (positive) [A]

**Setting parameter**

&lt;cur lim peak high&gt; | MINimum | MAXimum

&lt;cur lim peak high&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Peak current limiter (positive) setting value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CURR:LIM:PEAK:HIGH 10

**Query example**

CURR:LIM:PEAK:HIGH?

**Response example**

10.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.

**4.2.1.51 Set Output Current Limit: Set/Get Current Limit (Peak Value <Minimum>)**

[:SOURce]:CURRent:LIMit:PEAK:LOW

**Description**

Sets/gets the peak current limiter (negative) [A]

**Setting parameter**

&lt;cur lim peak low&gt; | MINimum | MAXimum

&lt;cur lim peak low&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Peak current limiter (negative) setting value

MINimum → Minimum value

MAXimum → Maximum value  
For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CURR:LIM:PEAK:LOW -10

**Query example**

CURR:LIM:PEAK:LOW?

**Response example**

-10.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.



**4.2.1.52** Set Output Current Limit: Set/Get Output OFF at Current Limit (Peak Value) Detection

[:SOURce]:CURRent:LIMit:PEAK:MODE

**Description**

Sets/gets the output off after the peak current limiter is activated

**Setting parameter**

<cur lim mode> ::= <DISC>

<DISC> ::= CONTInuous | OFF

CONTInuous → Continuous operation, output off disabled

OFF → Output off enabled

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

CURR:LIM:PEAK:MODE OFF

**Query example**

CURR:LIM:PEAK:MODE?

**Response example**

OFF

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.

**4.2.1.53** Set Output Current Limit: Set/Get Time Before Turning Output OFF After Current Limit (Peak Value) is Detected

[:SOURce]:CURRent:LIMit:PEAK:TIME

**Description**

Sets/gets the time [s] before turning the output off after the peak current limiter is activated

**Setting parameter**

<cur lim time> | MINimum | MAXimum

<cur lim time> ::= <INT>

<INT> → Limiter time: 1 to 10

MINimum → 1

MAXimum → 10

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

CURR:LIM:PEAK:TIME 5

**Query example**

CURR:LIM:PEAK:TIME?

**Response example**

5

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.

**4.2.1.54 Set Output Voltage Limit: Set/Get Voltage Limit (Effective Value)**

[:SOURce]:CC:VOLTage:LIMit:RMS[:AMPLitude]

**Description**

Sets/gets the voltage effective value [Vrms] limiter

**Setting parameter**

&lt;volt lim rms&gt; | MINimum | MAXimum

&lt;volt lim rms &gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Output voltage effective value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CC:VOLT:LIM:RMS 100

**Query example**

CC:VOLT:LIM:RMS?

**Response example**

100

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.

**4.2.1.55** Set Output Voltage Limit: Set/Get Output OFF at Voltage Limit (Effective Value) Detection  
[:SOURce]:CC:VOLTage:LIMit:RMS:MODE**Description**

Sets/gets the output off after the RMS voltage limiter is activated

**Setting parameter**

<volt lim mode> ::= <DISC>

<DISC> ::= CONTInuous | OFF

CONTInuous → Continuous operation, output off disabled

OFF → Output off enabled

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

CC:VOLT:LIM:RMS:MODE OFF

**Query example**

CC:VOLT:LIM:RMS:MODE?

**Response example**

OFF

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.

**4.2.1.56 Set Output Voltage Limit: Set/Get Time Before Turning Output OFF After Voltage Limit (Effective Value) is Detected**

[:SOURce]:CC:VOLTage:LIMit:RMS:TIME

**Description**

Sets/gets the time [s] before turning the output off after the RMS voltage limiter is activated

**Setting parameter**

&lt;volt lim time&gt; | MINimum | MAXimum

&lt;volt lim time&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Limiter time: 1 to 10

MINimum → 1

MAXimum → 10

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

CC:VOLT:LIM:RMS:TIME 5

**Query example**

CC:VOLT:LIM:RMS:TIME?

**Response example**

5

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.

**4.2.1.57** Set Output Voltage Limit: Set/Get Voltage Limit (Peak Value <Maximum>)

[:SOURce]:CC:VOLTage:LIMit:PEAK:HIGH

**Description**

Sets/gets the peak voltage limiter (positive) [V]

**Setting parameter**

&lt;volt lim peak high&gt; | MINimum | MAXimum

&lt;volt lim peak high&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Peak voltage limiter (positive) setting value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CC:VOLT:LIM:PEAK:HIGH 150

**Query example**

CC:VOLT:LIM:PEAK:HIGH?

**Response example**

150

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.

**4.2.1.58** Set Output Voltage Limit: Set/Get Voltage Limit (Peak Value <Minimum>)

[:SOURce]:CC:VOLTage:LIMit:PEAK:LOW

**Description**

Sets/gets the peak voltage limiter (negative) [V]

**Setting parameter**

<volt lim peak low> | MINimum | MAXimum

<volt lim peak low> ::= <REAL>

<REAL> → Peak voltage limiter (negative) setting value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<NR2>

**Setting example**

CC:VOLT:LIM:PEAK:LOW -150

**Query example**

CC:VOLT:LIM:PEAK:LOW?

**Response example**

-150

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.

**4.2.1.59** Set Output Voltage Limit: Set/Get Output OFF at Voltage Limit (Peak Value) Detection  
[:SOURce]:CC:VOLTage:LIMit:PEAK:MODE**Description**

Sets/gets the output off after the peak voltage limiter is activated

**Setting parameter**

<volt lim mode> ::= <DISC>

<DISC> ::= CONTInuous | OFF

CONTInuous → Continuous operation, output off disabled

OFF → Output off enabled

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

CC:VOLT:LIM:PEAK:MODE OFF

**Query example**

CC:VOLT:LIM:PEAK:MODE OFF?

**Response example**

OFF

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.



**4.2.1.60** Set Output Voltage Limit: Set/Get Time Before Turning Output OFF After Voltage Limit (Peak Value) is Detected

[[:SOURce]:CC:VOLTage:LIMit:PEAK:TIME

#### Description

Sets/gets the time [s] before turning the output off after the peak voltage limiter is activated

#### Setting parameter

<volt lim time> | MINimum | MAXimum

<volt lim time> ::= <INT>

<INT> → Limiter time: 1 to 10

MINimum → 1

MAXimum → 10

#### Query parameter

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

#### Response format

<INT>

#### Setting example

CC:VOLT:LIM:PEAK:TIME 5

#### Query example

CC:VOLT:LIM:PEAK:TIME?

#### Response example

5

#### Remarks

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.

**4.2.1.61** Output Setting (for All Phases): Set/Get Output Range(CV)

[:SOURce]:VOLTage:RANGe

**Description**

Sets/gets the output range(CV)

**Setting parameter**

&lt;volt range&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= R100V | R200V

R100V → 100V range

R200V → 200V range

**Query parameter**

None

**Response format**

&lt;DISC&gt;

**Setting example**

VOLT:RANG R100V

**Query example**

VOLT:RANG?

**Response example**

R100V

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CC mode.

The setting cannot be done in the Sequence control state or Simulation control state.

The setting cannot be done during the output on.

**4.2.1.62** Output Setting (for All Phases): Set/Get Output Range(CC)

[:SOURce]:CURRent:RANGe

**Description**

Sets/gets the output range(CC)

**Setting parameter**

&lt;current range&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= R10A | R20A

R10A → 10A range

R20A → 20A range

**Query parameter**

None

**Response format**

&lt;DISC&gt;

**Setting example**

CURR:RANG R10A

**Query example**

CURR:RANG?

**Response example**

R10A

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the CV mode.

The setting cannot be done in the Sequence control state or Simulation control state.

The setting cannot be done during the output on.

**4.2.1.63** Output Setting (for All Phases): Set/Get Operation Mode

[:SOURce]:MODE

**Description**

Sets/gets the operation mode

**Setting parameter**

&lt;mode&gt; ::= &lt;DISC&gt;

<DISC> ::= AC\_INT | AC\_VCA | AC\_SYNC | AC\_EXT | AC\_ADD | ACHF\_INT  
 | ACHF\_VCA | DC\_INT | DC\_VCA | DC\_EXT | ACDC\_INT | ACDC\_SYNC |  
 ACDC\_EXT | ACDC\_ADD

AC\_INT → AC-INT mode

AC\_VCA → AC-VCA mode

AC\_SYNC → AC-SYNC mode

AC\_EXT → AC-EXT mode

AC\_ADD → AC-ADD mode

ACHF\_INT → ACHF-INT mode

ACHF\_VCA → ACHF-VCA mode

DC\_INT → DC-INT mode

DC\_VCA → DC-VCA mode

ACDC\_INT → ACDC-INT mode

ACDC\_SYNC → ACDC-SYNC mode

ACDC\_EXT → ACDC-EXT mode

ACDC\_ADD → ACDC-ADD mode

**Query parameter**

None

**Response format**

&lt;DISC&gt;

**Setting example**

MODE ACDC\_INT

**Query example**

MODE?

**Response example**

ACDC\_INT

**Remarks**

The setting is error during the System Lock/warning.

When the output function is the Sequence, only INT can be selected as the signal source.

When the output function is the Simulation, only the ACDC-INT mode can be selected as the operation mode.

**4.2.1.64 Set/Get Output Waveform: Set/Get Clipped Sine Wave Type Specification**

[:SOURce]:FUNCtion:CSINe:TYPE

**Description**

Sets/gets the clipped sine wave setting type

**Setting parameter**

&lt;func shap&gt;,&lt;type&gt;

&lt;func shap&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= CLP1 | CLP2 | CLP3

CLP1 → Clipped sine wave 1

CLP2 → Clipped sine wave 2

CLP3 → Clipped sine wave 3

&lt;type&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= CFACtor | CLIP

CFACtor → Crest factor

CLIP → Clip

**Query parameter**

&lt;func shap&gt;

&lt;func shap&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= CLP1 | CLP2 | CLP3

**Response format**

&lt;DISC&gt;

**Setting example**

FUNC:CSIN:TYPE CLP1, CFACTOR

**Query example**

FUNC:CSIN:CFAC? CLP1

**Response example**

CFAC

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.65 Set/Get Output Waveform: Set/Get Clipped Sine Wave Crest Factor**

[:SOURce]:FUNCtion:CSINe:CFACtor

**Description**

Sets/gets the clipped sine wave crest factor

**Setting parameter**

&lt;func shap&gt;,{&lt;cf&gt; | MINimum | MAXimum}

&lt;func shap&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= CLP1 | CLP2 | CLP3

CLP1 → Clipped sine wave 1

CLP2 → Clipped sine wave 2

CLP3 → Clipped sine wave 3

&lt;cf&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Crest factor: 1.10 to 1.41, Resolution 0.01

MINimum → 1.10

MAXimum → 1.41

**Query parameter**

&lt;func shap&gt;[, MINimum | MAXimum]

&lt;func shap&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= CLP1 | CLP2 | CLP3

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

FUNC:CSIN:CFAC CLP1,1.2

**Query example**

FUNC:CSIN:CFAC? CLP1

**Response example**

1.20

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.66 Set/Get Output Waveform: Set/Get Clipped Sine Wave Clip Ratio**

[:SOURce]:FUNCtion:CSINe:CLIP

**Description**

Sets/gets the clipped sine wave clip ratio

**Setting parameter**

&lt;func shap&gt;,{&lt;clip&gt; | MINimum | MAXimum}

&lt;func shap&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= CLP1 | CLP2 | CLP3

CLP1 → Clipped sine wave 1

CLP2 → Clipped sine wave 2

CLP3 → Clipped sine wave 3

&lt;clip&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Clip ratio: 40.0 to 100.0, Resolution 0.1

MINimum → 40.0

MAXimum → 100.0

**Query parameter**

&lt;func shap&gt;[, MINimum | MAXimum]

&lt;func shap&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= CLP1 | CLP2 | CLP3

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

FUNC:CSIN:CLIP CLP1, 80

**Query example**

FUNC:CSIN:CLIP? CLP1

**Response example**

80.0

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.67** Set/Get Output Waveform: Initialize Arbitrary Wave Memory

:TRACe|DATA:WAVe:CLEAr

**Description**

Initializes the arbitrary wave memory

**Setting parameter**

<memory> | MINimum | MAXimum

<memory> ::= <INT>

<INT> →Arbitrary wave memory number: 1 to 16

MINimum → 1

MAXimum → 16

**Setting example**

TRAC:WAV:CLE 1

**Remarks**

The setting is error during the System Lock/warning.



**4.2.1.68** Set/Get Output Waveform: Transfer Arbitrary Wave Data

:TRACe[DATA:WAVe[:DATA]

**Description**

Sets the arbitrary wave

**Setting parameter**

&lt;memory&gt;,&lt;data&gt;

&lt;memory&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Arbitrary wave memory number: 1 to 16

&lt;data&gt; ::= &lt;BLK&gt;

&lt;BLK&gt; → Arbitrary waveform data

#532768&lt;DAB&gt;...&lt;DAB&gt;

#: Represents that the block data is sent

5: Represents the number of the subsequent numbers

32768: Represents the number of the subsequent byte data

&lt;DAB&gt;...&lt;DAB&gt;:16-bit, 16384 words waveform data.

The waveform data format is the complement form of 2 in the little endian

**Setting example**

TRAC:WAV 1, #532768&lt;DAB&gt;...&lt;DAB&gt;

**Remarks**

Not available for RS232 and LAN.

The setting is error during the System Lock/warning.

**4.2.1.69** Set/Get Output Waveform: Set/Get Arbitrary Wave Name

:TRACe|DATA:WAVe:NAME

**Description**

Sets/gets the arbitrary wave name

**Setting parameter**

&lt;memory&gt;,&lt;name&gt;

&lt;memory&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Arbitrary wave memory number: 1 to 16

&lt;name&gt; ::= &lt;STR&gt;

&lt;STR&gt; → Arbitrary wave name (16 characters or less)

The arbitrary wave name should contain 16 one-byte characters (alphanumeric characters) or less. The characters that can be used are as follows.

| valid character       |                                    |
|-----------------------|------------------------------------|
| Number                | 0~9                                |
| Alphabet (upper case) | A-Z                                |
| Alphabet (lower case) | a-z                                |
| Symbol                | !#\$%&'()+,.-:;=@[]^_`{ }~ (space) |

**Query parameter**

&lt;memory&gt; | MINimum | MAXimum

&lt;memory&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Arbitrary wave memory number: 1 to 16

MINimum → Gets the arbitrary wave name for memory number 1

MAXimum → Gets the arbitrary wave name for memory number 16

**Response format**

&lt;STR&gt;

**Setting example**

TRAC:WAV:NAME 1,"ARB1"

**Query example**

TRAC:WAV:NAME? 1

**Response example**

"ARB1"

**Remarks**

The setting is error during the System Lock/warning.

**4.2.1.70** Set/Get Output Waveform: Transfer Arbitrary Wave Data To Buffer

:TRACe|DATA:WAVe:ASCii

**Description**

Sets the arbitrary wave buffer(ASCII)

**Setting parameter**

&lt;address&gt;,&lt;data&gt;

&lt;address&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Arbitrary wave buffer address: 0 to 16383

&lt;data&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Arbitrary waveform data: -32768 to 32768

**Setting example**

TRAC:WAV:ASC 0, 18000

**4.2.1.71** Set/Get Output Waveform: Setting From Arbitrary Waveform Buffer to Waveform Memory

:TRACe|DATA:WAVe:BUFFer:SET

**Description**

Set arbitrary waveform buffer to waveform memory

**Setting parameter**

NONE

**Setting example**

TRAC:WAV:BUFF:SET

**Remark**

The setting is error during the System Lock/warning.

Usage example

//Set wave data to buffer

:TRAC:WAV:ASC 0, 11

:TRAC:WAV:ASC 1, 12

:TRAC:WAV:ASC 2, 13

. . .

:TRAC:WAV:ASC 16383, 10000

//Set to waveform memory number 1

:TRAC:WAV:BUFF:SET 1

**4.2.1.72** Factory Setting: Execute Factory Set

:SYSTem:INIT

**Description**

Execute factory set

**Setting parameter**

NONE

**Setting example**

:SYST:INIT

**Remark**

The setting is error during the System Lock/warning.

This command can only be used when only one device is running.

The setting cannot be done during the output on.

Executing this command will switch to the USB interface.

**4.2.1.73** Get CC Mode Enable/Disable: Get CC Mode Enable/Disable  
:SYSTem:CONFiGure:OPTion?

Description

Get the CC Mode enable

Query parameter

None

Response format

<INT>

Query example

SYST:CONF:OPT?

Response example

1

Remark

This command returns 1 if the CC mode is enable.

**4.2.1.74** Date and Time Settings: Set/get Date and Time

:SYSTem:DATE

**Description**

Sets/gets Date and Time

**Setting parameter**

&lt;year&gt;,&lt;month&gt;,&lt;day&gt;,&lt;hour&gt;,&lt;minute&gt;,&lt;second&gt;

&lt;year&gt; ::= &lt;INT&gt;

&lt;INT&gt; ::= year: 2022~2099

&lt;month&gt; ::= &lt;INT&gt;

&lt;INT&gt; ::= month: 1~12

&lt;day&gt; ::= &lt;INT&gt;

&lt;INT&gt; ::= day: 1~31

&lt;hour&gt; ::= &lt;INT&gt;

&lt;INT&gt; ::= hour: 0~23

&lt;minute&gt; ::= &lt;INT&gt;

&lt;INT&gt; ::= minute: 0~59

&lt;second&gt; ::= &lt;INT&gt;

&lt;INT&gt; ::= second: 0~59

**Query parameter**

NONE

**Response format**

&lt;INT&gt;,&lt;INT&gt;,&lt;INT&gt;,&lt;INT&gt;,&lt;INT&gt;,&lt;INT&gt;

Returns year, month, day, hour, minute, second in order.

**Setting example**

SYST:DATE 2023,9,14,10,46,40

**Query example**

SYST:DATE?

**Response example**

2023,9,14,10,46,45

## 4.3 Continuous Function Commands

### 4.3.1.1 Common Command (IEEE488.2): Read Basic Setting Data

\*RCL

#### Description

Read the basic setting data

#### Setting parameter

<memory> ::= <INT>

<INT> → Basic Setting Memory number: 0 to 30

#### Setting example

\* RCL 8

#### Remarks

The setting is error during the System Lock/warning.

The operation cannot be done in the Sequence control state or Simulation control state.

The operation cannot be done during the output on.

### 4.3.1.2 Common Command (IEEE488.2): Save the Basic Setting Data

\*SAV

#### Description

Save the basic setting data

#### Setting parameter

<memory> ::= <INT>

<INT> → Basic Setting Memory number: 1 to 30

#### Setting example

\*SAV 8

#### Remarks

The setting is error during the System Lock/warning.

The operation cannot be done in the Sequence control state or Simulation control state.

The operation cannot be done during the output on.

**4.3.1.3 Panel Display Setting: Set/Get Measurement Display Phase Setting (Phase Number)**

:DISPlay[:WINDow]:INSTrument:NSElect

**Description**

Sets/gets the measurement display phase setting (phase number)

Switches the measurement display phase displayed in the LCD

**Setting parameter**

<phase number> | MINimum | MAXimum

<phase number> ::= <INT>

<INT> → Phase number: 0 to 5

0: L1 phase | 1: L2 phase | 2: L3 phase | 3: ALL | 4: LINE

MINimum → 0

MAXimum → 4

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

DISP:INST:NSEL 1

**Query example**

DISP:INST:NSEL?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.



**4.3.1.4 Panel Display Setting: Set/Get Measurement Display Phase Setting (Phase)**`:DISPlay[:WINDow]:INSTrument:SElect`**Description**

Sets/gets the measurement display phase setting (phase)

Switches the measurement phase displayed in the LCD

**Setting parameter**`<phase name> ::= <DISC>``<DISC> ::= L1 | L2 | L3 | L12 | L23 | L31`

L1 → L1 phase

L2 → L2 phase

L3 → L3 phase

LINE → LINE

ALL → ALL phase

**Query parameter**

None

**Response format**`<DISC>`**Setting example**`DISP:INST:SEL L1`**Query example**`DISP:INST:SEL?`**Response example**

L1

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.5 Panel Display Setting: Set/Get Measurement Display Selection**

```
:DISPlay[:WINDow]:MEASure:MODE
```

**Description**

Sets/gets the measurement display selection

It sets the measurement phase displayed during the Continuous, Sequence control, and Simulation control.

**Setting parameter**

```
<disp meas mode> ::= <DISC>
```

```
<DISC> ::= RMS | RMS2 | AVG | AVG2 | PEAK | PEAK2 | HC1 | HC2 | HC3  
| HC4
```

RMS → Effective value

RMS2 → Effective value2

AVG → Average value

AVG2 → Average value2

PEAK → Peak value

PEAK2 → Peak value2

HC1 → Harmonic current 1

HC2 → Harmonic current 2

HC3 → Harmonic current 3

HC4 → Harmonic current 4

**Query parameter**

None

**Response format**

```
<DISC>
```

**Setting example**

```
DISP:MEAS:MODE RMS
```

**Query example**

```
DISP:MEAS:MODE?
```

**Response example**

```
RMS
```

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.6** Output Setting (for All Phases): Set/Get Output Frequency

[:SOURce]:FREQUency[:IMMediate]

**Description**

Sets/gets the output frequency

**Setting parameter**

&lt;frequency&gt; | MINimum | MAXimum

&lt;frequency&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Frequency: The range that can be used are as follows.

40.00 to 550.00 (AC)

40.00 to 5000 (ACHF)

1.00 to 1500 (ACDC)

Resolution 0.01(<100Hz), 0.1(<1000Hz), 1( $\leq$ 5000Hz)

MINimum → 40.00 (AC, ACHF)/1.00 (ACDC)

MAXimum → 1500(AC, ACDC)/5000(ACHF)

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

FREQ 50

**Query example**

FREQ?

**Response example**

50.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting can be made only in the AC-INT, AC-VCA, AC-ADD, ACHF-INT, ACHF-VCA, ACDC-INT, and ACDC-ADD modes.

**4.3.1.7** Output Setting (for All Phases): Set/Get Output On Phase

[:SOURce]:PHASe:STARt[:IMMediate]

**Description**

Sets/gets the output on phase

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

PHAS:STAR 90

**Query example**

PHAS:STAR?

**Response example**

90.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting cannot be done in the AC-EXT, ACDC-EXT, DC-INT, and DC-VCA modes.

**4.3.1.8** Output Setting (for All Phases): Enable/Disable Output Off Phase Specification

[:SOURce]:PHASe:STOP:ENABLE

**Description**

Sets/gets whether to enable/disable the output off phase

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

PHAS:STOP:ENAB ON

**Query example**

PHAS:STOP:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting cannot be done in the AC-EXT, ACDC-EXT, DC-INT, and DC-VCA modes.

**4.3.1.9** Output Setting (for All Phases): Set/Get Output Off Phase

[:SOURce]:PHASe:STOP[:IMMEDIATE]

**Description**

Sets/gets the output off phase

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

PHAS:STOP 90

**Query example**

PHAS:STOP?

**Response example**

90.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting cannot be done in the AC-EXT, ACDC-EXT, DC-INT, and DC-VCA modes.

**4.3.1.10** Output Setting (for All Phases): Set/Get Balanced/Unbalanced Mode

[:SOURce]:PHASe:MODE

**Description**

Sets/gets the balanced mode

**Setting parameter**

&lt;mode&gt; ::= &lt;DISC&gt;

BALance → Balanced mode

UNBalance → Unbalanced mode

**Query parameter**

None

**Response format**

&lt;DISC&gt;

**Setting example**

PHAS:MODE BAL

**Query example**

PHAS:MODE?

**Response example**

BAL

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the single-phase.

**4.3.1.11 Output Setting (for All Phases): Set/Get Remote Sensing Function**

:MEASure:CONFigure:SENSing

**Description**

Sets/gets the remote sensing

**Setting parameter**

&lt;state&gt; ::= &lt;INT&gt;

&lt;INT&gt; → 0: Disable, 1: Enable

**Query parameter**

None

**Response format**

&lt;INT&gt;

**Setting example**

MEAS:CONF:SENS 1

**Query example**

MEAS:CONF:SENS?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The remote sensing can be set only when the waveform is a sine wave for the AC mode in the AC-INT, AC-VCA, AC-SYNC, ACHF-INT, ACHF-VCA, DC-INT, and DC-VCA modes.

When the Sequence or Simulation function is selected, the remote sensing is forcibly set to off.

When the Sequence function is selected, it can be established on at the following condition.

Waveform is a sine wave

One of the AC voltage or the DC voltage is 0V setting.

When the output function is set to the Simulation, the remote sensing function cannot be enabled.



**4.3.1.12 Output Setting (for Specific Phase): Set/Get Phase To Set (Phase Number)**

:INSTrument:NSELect

**Description**

Sets/gets the output phase setting (phase number)

It specifies the phase to set/get by the SOURce subsystem command.

It does not affect the phase specification by the SOURce:SEQuence subsystem.

**Setting parameter**

&lt;phase number&gt; | MINimum | MAXimum

&lt;phase number&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Phase number: 0 to 4

0: All phases at once | 1: L1 phase | 2: L2 phase | 3: L3 phase | 4: Line

MINimum → 0

MAXimum → 4

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

INST:NSEL 1

**Query example**

INST:NSEL?

**Response example**

1

**Remarks**

For single-phase, only 0 and 1 can be set.

For single-phase three-wire, 0, 1, 2, and 4 can be set.

**4.3.1.13** Output Setting (for Specific Phase): Set/Get Phase To Set (Phase)`:INSTrument:SELEct`**Description**

Sets/gets the output phase setting (phase)

It specifies the phase to set/get by the SOURce subsystem command.

It does not affect the phase specification by the SOURce:SEQuence subsystem.

**Setting parameter**

`<phase name> ::= <DISC>`

`<DISC> ::= ALL | L1 | L2 | L3 | LINE`

ALL → All phases at once

L1 → L1 phase, L2 → L2 phase, L3 → L3 phase, LINE → Line

**Query parameter**

None

**Response format**

`<DISC>`

**Setting example**

`INST:SEL L1`

**Query example**

`INST:SEL?`

**Response example**

L1

**Remarks**

For single-phase, only ALL and L1 can be set.

For single-phase three-wire, ALL, L1, L2, and LINE can be set.

**4.3.1.14** Output Setting (for Specific Phase): Set/Get AC Output Voltage

[:SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]

**Description**

Sets/gets the AC output voltage

**Setting parameter**

&lt;volt&gt; | MINimum | MAXimum

&lt;volt&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → AC output voltage

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

VOLT 100

**Query example**

VOLT?

**Response example**

100.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

**4.3.1.15** Output Setting (for Specific Phase): Set/Get DC Output Voltage

[:SOURce]:VOLTage[:LEVel][:IMMEdiate]:OFFSet

**Description**

Sets/gets the DC output voltage

**Setting parameter**

&lt;volt offs&gt; | MINimum | MAXimum

&lt;volt offs&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → DC output voltage

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

VOLT:OFFS 10

**Query example**

VOLT:OFFS?

**Response example**

10.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

**4.3.1.16** Output Setting (for Specific Phase): Set/Get AC Output Current

[:SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]

**Description**

Sets/gets the AC output current

**Setting parameter**

&lt;cur&gt; | MINimum | MAXimum

&lt;cur&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → AC output current

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CURR 10

**Query example**

CURR?

**Response example**

10.00

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

This command can be set in CC mode.

**4.3.1.17** Output Setting (for Specific Phase): Set/Get DC Output Current

[:SOURce]:CURRent[:LEVel][:IMMediate]:OFFSet

**Description**

Sets/gets the DC output current

**Setting parameter**

&lt;curr offs&gt; | MINimum | MAXimum

&lt;curr offs&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → DC output current

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CURR:OFFS 10

**Query example**

CURR:OFFS?

**Response example**

10.00

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

This command can be set in CC mode.

**4.3.1.18** Output Setting (for Specific Phase): Set/Get Output Waveform

[:SOURce]:FUNCtion[:SHAPE][:IMMediate]

**Description**

Sets/gets the output waveform

**Setting parameter**

&lt;func shap&gt; ::= &lt;DISC&gt;

<DISC> ::= SIN | ARB1 | ARB2 | ARB3 | ARB4 | ARB5 | ARB6 | ARB7 | ARB8  
 | ARB9 | ARB10 | ARB11 | ARB12 | ARB13 | ARB14 | ARB15 | ARB16 | CLP1  
 | CLP2 | CLP3

SIN → Sine wave

ARB1 → Arbitrary wave 1

:

ARB16 → Arbitrary wave 16

CLP1 → Clipped sine wave 1

CLP2 → Clipped sine wave 2

CLP3 → Clipped sine wave 3

**Query parameter**

None

**Response format**

&lt;DISC&gt;

**Setting example**

FUNC SIN

**Query example**

FUNC?

**Response example**

SIN

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

**4.3.1.19** Output Setting (for Specific Phase): Set/Get Phase Angle at Polyphase System

[:SOURce]:PHASe:PHASe

**Description**

Sets/gets the phase angle

**Setting parameter**

&lt;target&gt;,{&lt;phase&gt; | MINimum | MAXimum}

&lt;target&gt; ::= &lt;DISC&gt;

L1 → L1 Phase angle

L2 → L2 Phase angle

L3 → L3 Phase angle

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase angle: 0.0 to 359.9, Resolution 0.1

MINimum → Minimum value

MAXimum → Maximum value

**Query parameter**

&lt;target&gt; [, MINimum | MAXimum]

&lt;target&gt; ::= &lt;DISC&gt;

L1 → L1 Phase angle

L2 → L2 Phase angle

L3 → L3 Phase angle

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

PHAS:PHAS L1,60

**Query example**

PHAS:PHAS? L1

**Response example**

60.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the single-phase.



**4.3.1.20** Output Setting (for Specific Phase): Set/Get DC Offset Voltage Adjustment Value (AC and ACHF mode)

[:SOURce]:VOLTage:ADJ:OFFSet:AC

**Description**

Sets/gets the DC offset voltage adjustment value [mV] in the AC and ACHF mode

**Setting parameter**

&lt;adjust voltage&gt; | MINimum | MAXimum

&lt;adjust voltage&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Voltage adjustment value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

VOLT:ADJ:OFFS:AC 10

**Query example**

VOLT:ADJ:OFFS:AC?

**Response example**

10.0

**Remarks**

The setting is error during the System Lock/warning.

When the set phase is ALL, only L1 is set.

**4.3.1.21** Output Setting (for Specific Phase): Set/Get DC Offset Voltage Adjustment Value (ACDC and DC modes)

[:SOURce]:VOLTage:ADJ:OFFSet:DC

**Description**

Sets/gets the DC offset voltage adjustment value [mV] in the ACDC and DC modes

**Setting parameter**

&lt;adjust voltage&gt; | MINimum | MAXimum

&lt;adjust voltage&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Voltage adjustment value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

VOLT:ADJ:OFFS:DC 10

**Query example**

VOLT:ADJ:OFFS:DC?

**Response example**

10

**Remarks**

The setting is error during the System Lock/warning.

When the set phase is ALL, only L1 is set.

**4.3.1.22** Output Setting (for Specific Phase): Set/Get DC Offset Current Adjustment Value (AC and ACHF mode)

[:SOURce]:CURRent:ADJust:OFFSet:AC

**Description**

Sets/gets the DC offset current adjustment value [mV] in the AC and ACHF mode

**Setting parameter**

&lt;adjust current&gt; | MINimum | MAXimum

&lt;adjust current&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Current adjustment value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CURR:ADJ:OFFS:AC 10

**Query example**

CURR:ADJ:OFFS:AC?

**Response example**

10

**Remarks**

The setting is error during the System Lock/warning.

When the set phase is ALL, only L1 is set.

This command can be set in CC mode.

**4.3.1.23** Output Setting (for Specific Phase): Set/Get DC Offset Current Adjustment Value (ACDC and DC modes)

[:SOURce]:CURRent:ADJust:OFFSet:DC

**Description**

Sets/gets the DC offset current adjustment value [mV] in the ACDC and DC modes

**Setting parameter**

&lt;adjust current&gt; | MINimum | MAXimum

&lt;adjust current&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Current adjustment value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

CURR:ADJ:OFFS:DC 10

**Query example**

CURR:ADJ:OFFS:DC?

**Response example**

10

**Remarks**

The setting is error during the System Lock/warning.

When the set phase is ALL, only L1 is set.

This command can be set in CC mode.

**4.3.1.24 Set Output Voltage Limit: Set/Get Voltage Limit (Effective Value)**

[:SOURce]:VOLTage:LIMit:RMS

**Description**

Sets/gets the effective voltage setting range limit (effective value)

**Setting parameter**

&lt;volt lim low&gt; | MINimum | MAXimum

&lt;volt lim low&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Voltage upper limit value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

VOLT:LIM:RMS 100

**Query example**

VOLT:LIM:RMS?

**Response example**

100.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

**4.3.1.25 Set Output Voltage Limit: Set/Get Voltage Limit (Peak Value <Maximum>)**

[:SOURce]:VOLTage:LIMit:HIGH

**Description**

Sets/gets the voltage setting range limit (peak value &lt;maximum&gt;)

**Setting parameter**

&lt;volt lim high&gt; | MINimum | MAXimum

&lt;volt lim high&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Output voltage upper limit

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

VOLT:LIM:HIGH 200

**Query example**

VOLT:LIM:HIGH?

**Response example**

200.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

**4.3.1.26** Set Output Voltage Limit: Set/Get Voltage Limit (Peak Value <Minimum>)

[:SOURce]:VOLTage:LIMit:LOW

**Description**

Sets/gets the voltage setting range limit (peak value &lt;minimum&gt;)

**Setting parameter**

&lt;volt lim low&gt; | MINimum | MAXimum

&lt;volt lim low&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Output voltage lower limit

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

VOLT:LIM:LOW -200

**Query example**

VOLT:LIM:LOW?

**Response example**

-200.0

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

**4.3.1.27 Set Output Current Limit: Set/Get Current Limit (Effective Value)**

[:SOURce]:CC:CURRent:RANGe:LIMit:RMS

**Description**

Sets/gets the effective current setting range limit (effective value)

**Setting parameter**

<curr lim low> | MINimum | MAXimum

<curr lim low> ::= <REAL>

<REAL> → Current upper limit value

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<NR2>

**Setting example**

CC:CURR:RANG:LIMit:RMS 10

**Query example**

CC:CURR:RANG:LIMit:RMS?

**Response example**

10.00

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

This command can be set in CC mode.



**4.3.1.28** Set Output Current Limit: Set/Get Current Limit (Peak Value <Maximum>)

[:SOURce]:CC:CURRent:RANGe:LIMit:HIGH

**Description**

Sets/gets the current setting range limit (peak value &lt;maximum&gt;)

**Setting parameter**

&lt;curr lim high&gt; | MINimum | MAXimum

&lt;curr lim high&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Output current upper limit

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CC:CURR:RANG:LIM:HIGH 50

**Query example**

CC:CURR:RANG:LIM:HIGH?

**Response example**

50.00

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

This command can be set in CC mode.

**4.3.1.29** Set Output Current Limit: Set/Get Current Limit (Peak Value <Minimum>)

[:SOURce]:CC:CURRent:RANGe:LIMit:LOW

**Description**

Sets/gets the current setting range limit (peak value &lt;minimum&gt;)

**Setting parameter**

&lt;curr lim low&gt; | MINimum | MAXimum

&lt;curr lim low&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Output current lower limit

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

CC:CURR:RANG:LIM:LOW -50

**Query example**

CC:CURR:RANG:LIM:LOW?

**Response example**

-50.00

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

This command can be set in CC mode.

#### 4.3.1.30 Set Output Frequency Setting Range Limit: Set/Get Upper Limit Value of Frequency Limit [:SOURce]:FREQUency:LIMit:HIGH

##### Description

Sets/gets the upper limit value [Hz] of the frequency setting range limit

##### Setting parameter

<frequency> | MINimum | MAXimum

<frequency> ::= <REAL>

<REAL> → Frequency: The range that can be used are as follows.

40.00 to 550.00 (AC)

40.00 to 5000 (ACHF)

1.00 to 1500 (ACDC)

Resolution 0.01(<100Hz), 0.1(<1000Hz), 1( $\leq$ 5000Hz)

MINimum → 40.00 (AC, ACHF)/1.00 (ACDC)

MAXimum → 1500(AC, ACDC)/5000(ACHF)

##### Query parameter

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

##### Response format

<NR2>

##### Setting example

FREQ:LIM:HIGH 60

##### Query example

FREQ:LIM:HIGH?

##### Response example

60.00

##### Remarks

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting can be made only in the AC-INT, AC-VCA, AC-ADD, ACDC-INT, and ACDC-ADD modes.

#### 4.3.1.31 Set Output Frequency Setting Range Limit: Set/Get Lower Limit Value of Frequency Limit [:SOURce]:FREQUency:LIMit:LOW

##### Description

Sets/gets the lower limit value [Hz] of the frequency setting range limit

##### Setting parameter

<frequency> | MINimum | MAXimum

<frequency> ::= <REAL>

<REAL> → Frequency: The range that can be used are as follows.

40.00 to 550.00 (AC)

40.00 to 5000 (ACHF)

1.00 to 1500 (ACDC)

Resolution 0.01(<100Hz), 0.1(<1000Hz), 1( $\leq$ 5000Hz)

MINimum → 40.00 (AC, ACHF)/1.00 (ACDC)

MAXimum → 1500(AC, ACDC)/5000(ACHF)

##### Query parameter

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

##### Response format

<NR2>

##### Setting example

FREQ:LIM:LOW 40

##### Query example

FREQ:LIM:LOW?

##### Response example

40.00

##### Remarks

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting can be made only in the AC-INT, AC-VCA, AC-ADD, ACDC-INT, and ACDC-ADD modes.

**4.3.1.32** Get Output Measured Value (for Specific Phase): Set/Get Measurement Query Phase (Phase Number)

:MEASure:INSTrument:NSElect

**Description**

Sets/gets the measurement query phase (phase number)

It specifies the phase to query the measured value for by the MEASure subsystem command.

\* This command does not affect the measured value display phase displayed in the LCD.

**Setting parameter**

&lt;phase number&gt; | MINimum | MAXimum

&lt;phase number&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Phase number: 0 to 5

0: L1 phase | 1: L2 phase | 2: L3 phase |

3: L1-L2 line | 4: L2-L3 line | 5: L3-L1 line

MINimum → 0

MAXimum → 5

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

MEAS:INST:NSEL 1

**Query example**

MEAS:INST:NSEL?

**Response example**

1

**Remarks**

For single-phase, only 0 can be set.

For single-phase three-wire, 0, 1, and 3 can be set.

#### 4.3.1.33 Get Output Measured Value (for Specific Phase): Set/Get Measurement Query Phase (Phase) :MEASure:INSTrument:SElect

##### Description

Sets/gets the measurement query phase (phase)

It specifies the phase to query the measured value for by the MEASure subsystem command.

\* This command does not affect the measured value display phase displayed in the LCD.

##### Setting parameter

<phase name> ::= <DISC>

<DISC> ::= L1 | L2 | L3 | L12 | L23 | L31

L1 → L1 phase, L2 → L2 phase, L3 → L3 phase

L12 → L1-L2 line, L23 → L2-L3 line, L31 → L3-L1 line

##### Query parameter

None

##### Response format

<DISC>

##### Setting example

MEAS:INST:SEL L1

##### Query example

MEAS:INST:SEL?

##### Response example

L1

##### Remarks

For single-phase, only L1 can be set.

For single-phase three-wire, L1, L2, and L12 can be set.

**4.3.1.34** Get Output Measured Value: Clear Current Peak-Hold Value

:MEASure[:SCALar]:CURRent:PEAK:CLEAr

**Description**

Clears the current peak-hold value

**Setting parameter**

None

**Setting example**

MEAS:CURR:PEAK:CLE

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.35** Get Output Measured Value: Current Peak-Hold Value

:MEASure[:SCALar]:CURRent:PEAK:HOLD?

**Description**

Gets the current peak-hold value [Apk]

**Query parameter**

None

**Response format**

<NR2>→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR:PEAK:HOLD?

**Response example**

26.55

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.36** Get Output Measured Value: Clear Voltage Peak-Hold Value`:MEASure[:SCALar]:VOLTage:PEAK:CLEar`**Description**

Clears the voltage peak-hold value

**Setting parameter**

None

**Setting example**`MEAS:VOLT:PEAK:CLE`**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.37** Get Output Measured Value: Voltage Peak-Hold Value`:MEASure[:SCALar]:VOLTage:PEAK:HOLD?`**Description**

Gets the voltage peak-hold value [Vpk]

**Query parameter**

None

**Response format**

<NR2>→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**`MEAS:VOLT:PEAK:HOLD?`**Response example**

106.0

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.



**4.3.1.38** Get Output Measured Value: Synchronization Signal Source Frequency

:MEASure[:SCALar]:FREQuency?

## Description

Gets the synchronization signal source frequency [Hz]

## Query parameter

None

## Response format

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

## Query example

MEAS:FREQ?

## Response example

50.4

## Remarks

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.39** Get Output Measured Value (for Specific Phase): Output Voltage (Effective Value)

:MEASure[:SCALar]:VOLTage[:RMS]?

## Description

Gets the output voltage effective value [Vrms]

## Query parameter

None

## Response format

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

## Query example

MEAS:VOLT?

## Response example

100.4

## Remarks

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.40** Get Output Measured Value (for Specific Phase): Output Voltage (Average Value)

:MEASure[:SCALar]:VOLTage:AVErage?

**Description**

Gets the output voltage average value [V]

**Query parameter**

None

**Response format**

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:VOLT:AVE?

**Response example**

100.0

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.41** Get Output Measured Value (for Specific Phase): Output Voltage (Peak Value <Maximum>)

:MEASure[:SCALar]:VOLTage:HIGh?

**Description**

Gets the voltage maximum value [Vpk]

**Query parameter**

None

**Response format**

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:VOLT:HIGh?

**Response example**

141.4

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.42** Get Output Measured Value (for Specific Phase): Output Voltage (Peak Value <Minimum>  
:MEASure[:SCALar]:VOLTage:LOW?

**Description**

Gets the voltage minimum value [Vpk]

**Query parameter**

None

**Response format**

<NR2>→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:VOLT:LOW?

**Response example**

-141.4

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.43** Get Output Measured Value (for Specific Phase): Output Voltage Crest Factor  
:MEASure[:SCALar]:VOLTage:CFACtor?

**Description**

Gets the output voltage crest factor

**Query parameter**

None

**Response format**

<NR2>→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:VOLT:CRES?

**Response example**

1.41

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.44** Get Output Measured Value (for Specific Phase): Output Current (Effective Value)

:MEASure[:SCALar]:CURRent[:RMS]?

**Description**

Gets the output current effective value [Arms]

**Query parameter**

None

**Response format**

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR?

**Response example**

12.75

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.45** Get Output Measured Value (for Specific Phase): Output Current (Average)

:MEASure[:SCALar]:CURRent:AVErage?

**Description**

Gets the output current average value [A]

**Query parameter**

None

**Response format**

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR:AVE?

**Response example**

1.20

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.46** Get Output Measured Value (for Specific Phase): Output Current (Peak Value <Maximum>)  
:MEASure[:SCALar]:CURRent:HIGH?**Description**

Gets the output current maximum peak value [Apk]

**Query parameter**

None

**Response format**

<NR2>→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR:HIGH?

**Response example**

30.0

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.47** Get Output Measured Value (for Specific Phase): Output Current (Peak Value <Minimum>)  
:MEASure[:SCALar]:CURRent:LOW?**Description**

Gets the output current minimum peak value [Apk]

**Query parameter**

None

**Response format**

<NR2>→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR:LOW?

**Response example**

-30.0

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.48** Get Output Measured Value (for Specific Phase): Set/Get Control Harmonic Measurement Type  
:MEASure[:SCALar]:HARMonic:TYPE

**Description**

Sets/gets the harmonic measurement type

**Setting parameter**

<type> ::= <DISC>

<DISC> ::= CURRent | VOLTage

CURRent → Harmonic current type

VOLTage → Harmonic voltage type

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

MEAS:HARM:TYPE CURR

**Query example**

MEAS:HARM:TYPE CURR?

**Response example**

CURR

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.49** Get Output Measured Value (for Specific Phase): Harmonic Voltage Value  
 :MEASure[:SCALar]:VOLTage:HARMonic[:RMS]?

#### Description

Gets the harmonic voltage [Vrms]

#### Query parameter

< page > ::= <INT> | MINimum | MAXimum

<INT> → Page: 1 to 5

1 (order 1 to 10) | 2 (order 11 to 20) | 3 (order 21 to 30) |

4 (order 31 to 40) | 5 (order 41 to 50)

MINimum → 1

MAXimum → 5

#### Response format

<NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>,<NR2>

→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

#### Query example

MEAS:VOLT:HARM? 1

#### Response example

2.24,0.02,0.01,0.15,0.06,0.08,0.01,0.01,0.02,0.06

#### Remarks

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.50** Get Output Measured Value (for Specific Phase): Harmonic Voltage Percentage

:MEASure[:SCALar]:VOLTage:HARMonic:RATio?

**Description**

Gets the harmonic voltage percentage [%]

**Query parameter**

&lt; page &gt; ::= &lt;INT&gt; | MINimum | MAXimum

&lt;INT&gt; → Page: 1 to 5

1 (order 1 to 10) | 2 (order 11 to 20) | 3 (order 21 to 30) |

4 (order 31 to 40) | 5 (order 41 to 50)

MINimum → 1

MAXimum → 5

**Response format**

&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;

→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:VOLT:HARM:RAT? 1

**Response example**

100.0,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.



**4.3.1.51** Get Output Measured Value (for Specific Phase): Set/Get Control Harmonic Measurement Function

:MEASure[:SCALar]:CURRent:HARMonic:ENABle

**Description**

Sets/gets whether to enable/disable the harmonic current measurement.

If harmonic current measurement is enabled with this command, harmonic voltage measurement will be disabled.

This command was prepared for the past DP series.

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Do nothing, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

MEAS:CURR:HARM:ENAB ON

**Query example**

MEAS:CURR:HARM:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.52** Get Output Measured Value (for Specific Phase): Harmonic Current Value

:MEASure[:SCALar]:CURRent:HARMonic[:RMS]?

**Description**

Gets the harmonic current [Arms]

**Query parameter**

&lt; page &gt; ::= &lt;INT&gt; | MINimum | MAXimum

&lt;INT&gt; → Page: 1 to 5

1 (order 1 to 10) | 2 (order 11 to 20) | 3 (order 21 to 30) |

4 (order 31 to 40) | 5 (order 41 to 50)

MINimum → 1

MAXimum → 5

**Response format**

&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;

→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR:HARM? 1

**Response example**

2.24,0.02,0.01,0.15,0.06,0.08,0.01,0.01,0.02,0.06

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.53** Get Output Measured Value (for Specific Phase): Harmonic Current Percentage

:MEASure[:SCALar]:CURRent:HARMonic:RATio?

**Description**

Gets the harmonic current percentage [%]

**Query parameter**

&lt; page &gt; ::= &lt;INT&gt; | MINimum | MAXimum

&lt;INT&gt; → Page: 1 to 5

1 (order 1 to 10) | 2 (order 11 to 20) | 3 (order 21 to 30) |

4 (order 31 to 40) | 5 (order 41 to 50)

MINimum → 1

MAXimum → 5

**Response format**

&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;,&lt;NR2&gt;

→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR:HARM:RAT? 1

**Response example**

100.0,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

.

**4.3.1.54** Get Output Measured Value (for Specific Phase): Output Current Crest Factor

:MEASure[:SCALar]:CURRent:CFACtor?

**Description**

Gets the output current crest factor

**Query parameter**

None

**Response format**

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:CURR:CRES?

**Response example**

1.41

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.55** Get Output Measured Value (for Specific Phase): Apparent Power

:MEASure[:SCALar]:POWer[:AC]:APParent?

**Description**

Gets the apparent power [VA]

**Query parameter**

None

**Response format**

&lt;INT&gt;→ (1000 [VA] or more)

&lt;NR2&gt;→ (Less than 1000 [VA])

Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:POW:APP?

**Response example**

367.0

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.56** Get Output Measured Value (for Specific Phase): Power Factor

:MEASure[:SCALar]:POWer[:AC]:PFACtor?

**Description**

Gets the power factor

**Query parameter**

None

**Response format**

&lt;NR2&gt;→ Query response range

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:POW:PFAC?

**Response example**

0.68

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.57** Get Output Measured Value (for Specific Phase): Active Power

:MEASure[:SCALar]:POWer[:AC][:REAL]?

**Description**

Gets the active power [W]

**Query parameter**

None

**Response format**

&lt;INT&gt;→ (1000 [W] or more)

&lt;NR2&gt;→ (Less than 1000 [W])

For the query response range and resolution, see the instruction manual of the main unit.

**Query example**

MEAS:POW?

**Response example**

249.0

**Remarks**

If overrange occurs, returns 9999999.

If the measurement conditions are not met, 99999999 is returned.

**4.3.1.58** Set Output Correction Function: Enable/Disable AGC Function

:OUTPut:AGC

**Description**

Sets/gets the AGC

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

OUTP:AGC ON

**Query example**

OUTP:AGC?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The AGC function can be used only when the output function is the Continuous function and the output mode is AC-INT/AC-VCA/AC-SYNC/ACHF-INT/ACHF\_VCA/DC-INT/DC-VCA.

When an AC voltage waveform other than SIN is selected, the AGC function cannot be enabled.

When the AGC function is enabled, and the output function is set to the Sequence or Simulation, the AGC function is forcibly disabled.

When the output function is set to the Sequence or Simulation, the AGC function cannot be enabled.

**4.3.1.59** Set Output Correction Function: Enable/Disable Autocal Function

:OUTPut:ACALibration

**Description**

Sets/gets the Autocal

**Setting parameter**

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

&lt;NBOL&gt;

**Setting example**

OUTP:ACAL ON

**Query example**

OUTP:ACAL?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The Autocal function can be used only when the output function is the Continuous function and the output mode is AC-INT/AC-VCA/AC-SYNC/ACHF-INT/ACHF-VCA/DC-INT/DC-VCA.

When an AC voltage waveform other than SIN is selected, the Autocal function cannot be enabled.

When the Autocal function is enabled, and the output function is set to the Sequence or Simulation, the Autocal function is forcibly disabled.

When the output function is set to the Sequence, it can be established on at the following condition.

AC voltage waveform is a sine wave.

One of the AC voltage or the DC voltage is 0V setting.

When the output function is set to the Simulation, the Autocal function cannot be enabled.

**4.3.1.60 Set External Input Signal: Set/Get External Input Gain**

:INPut:GAIN

**Description**

Sets/gets the external input gain

The setting can be made in the following modes:

AC-EXT, AC+DC-EXT, AC-VCA, ACHF-VCA, DC-VCA, AC-ADD, AC+DC-ADD

**Setting parameter**

<inp gain> | MINimum | MAXimum

<inp gain> ::= <REAL>

<REAL> → External input gain

MINimum → Minimum value

MAXimum → Maximum value

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<NR2>

**Setting example**

INP:GAIN 10

**Query example**

INP:GAIN?

**Response example**

10.0

**Remarks**

The setting is error during the System Lock/warning.



**4.3.1.61 Set External Input Signal: Set/Get External Synchronization Signal Source**`:INPut:SYNC:SOURce`**Description**

Sets/gets the external synchronization signal source

The setting can be made in the AC-SYNC and AC+DC-SYNC modes

**Setting parameter**`<cloc> ::= <DISC>``<DISC> ::= LINE | EXT``LINE` → Line synchronization`EXT` → External synchronization**Query parameter**`None`**Response format**`<DISC>`**Setting example**`INP:SYNC:SOUR EXT`**Query example**`INP:SYNC:SOUR?`**Response example**`EXT`**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting cannot be done during the output on.

**4.3.1.62** Set External Control Signal: Permit/Inhibit External Control Input`:SYSTem:CONFigure:EXTio[:STATe]`**Description**

Sets/gets the external control input

Note that the external control input is ignored in the Remote state.

**Setting parameter**`<state> ::= <INT>``<INT> → 0: Disable, 1: Enable`**Query parameter**

None

**Response format**`<INT>`**Setting example**`SYST:CONF:EXT 1`**Query example**`SYST:CONF:EXT?`**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.63** Set External Control Signal: Set/Get External Control Output Polarity

:SYSTem:CONFigure:EXTio:POLarity

**Description**

Sets/gets the external control output polarity

**Setting parameter**

<polarity> ::= <DISC>

<DISC> ::= POSitive | NEGative

POSitive → Positive logic

NEGative → Negative logic

**Query parameter**

None

**Response format**

<DISC>

**Setting example**

SYST:CONF:EXT:POL POSITIVE

**Query example**

SYST:CONF:EXT:POL?

**Response example**

POS

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.64** Output Setting (for All Phases): Enable/disable Soft Start Function`:OUTPut:SStart[:STATe][:RISE]`**Description**

Sets/gets whether to enable/disable soft start function.

**Setting parameter**`<state> ::= <BOL>``<BOL> → 0/OFF: Disable, 1/ON: Enable`**Query parameter**`None`**Response format**`<NBOL>`**Setting example**`OUTP:SST ON`**Query example**`OUTP:SST?`**Response example**`1`**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting can be made only in the AC-INT, AC-SYNC, ACHF-INT, ACDC-SYNC, and DC-INT modes.

The setting cannot be done if output on phase is other than 0.

**4.3.1.65** Output Setting (for All Phases): Set/get Soft Start Time`:OUTPut:SSTart:TIME[:RISE]`**Description**

Set/get soft start time[s].

**Setting parameter**`<time> | MINimum | MAXimum``<time> ::= <REAL>``<REAL>` → soft start time: 0.1 to 30.0 Resolution 0.1

MINimum → 0.1

MAXimum → 30.0

**Query parameter**`[MINimum | MAXimum]`

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**`<REAL>`**Setting example**`OUTP:SST:TIME 5`**Query example**`OUTP:SST:TIME?`**Response example**

5.0

**Remarks**

The setting is error during the System Lock/warning.

**4.3.1.66** Output Setting (for All Phases): Enable/disable Soft Stop Function`:OUTPut:SStart[:STATe]:FALL`**Description**

Sets/gets whether to enable/disable soft stop function.

**Setting parameter**`<state> ::= <BOL>``<BOL> → 0/OFF: Disable, 1/ON: Enable`**Query parameter**`None`**Response format**`<NBOL>`**Setting example**`OUTP:SST:FALL ON`**Query example**`OUTP:SST:FALL?`**Response example**`1`**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done in the Sequence and Simulation.

The setting can be made only in the AC-INT, AC-SYNC, ACHF-INT, ACDC-SYNC, and DC-INT modes.

The setting cannot be done if output off phase is enable.

**4.3.1.67** Output Setting (for All Phases): Set/get Soft Stop Time

:OUTPut:SStart:TIME:FALL

**Description**

Set/get soft stop time[s].

**Setting parameter**

<time> | MINimum | MAXimum

<time> ::= <REAL>

<REAL> → soft stop time: 0.1 to 30.0 Resolution 0.1

MINimum → 0.1

MAXimum → 30.0

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<REAL>

**Setting example**

OUTP:SST:TIME:FALL 5

**Query example**

OUTP:SST:TIME:FALL?

**Response example**

5.0

**Remarks**

The setting is error during the System Lock/warning.

## 4.3.2 Sequence Function Commands

### 4.3.2.1 Set Panel Display: Set/Get Step Time Display Unit

:DISPlay[:WINDow]:TIME:UNIT

#### Description

Sets/gets the Sequence/Simulation Step Time unit

It is reflected to the display and setting on the panel

The setting/getting from the remote command is made in seconds regardless of the DISP:TIME:UNIT setting

#### Setting parameter

<unit> ::= <DISC>

<DISC> ::= MS | S

MS → millisecond

S → second

#### Query parameter

None

#### Response format

<DISC>

#### Setting example

DISP:TIME:UNIT MS

#### Query example

DISP:TIME:UNIT?

#### Response example

MS

#### Remarks

The setting is error during the System Lock/warning.



**4.3.2.2** Sequence Function Control: Get State

[:SOURce]:SEQuence:CONTRol[:STATe]?

**Description**

Gets the sequence control state

**Query parameter**

None

**Response format**

&lt;DISC&gt;

**Query example**

SEQ:CONT?

**Response example**

|         |               |
|---------|---------------|
| EDIT    | Edit state    |
| CONTROL | Control state |

**4.3.2.3** Sequence Function Control: Execution Control

:TRIGger:SEQuence:SELEcted:EXECute

**Description**

Sets the sequence control

**Setting parameter**

&lt;ctrl&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= STOP | START | HOLD | BRAN1 | BRAN2

STOP → Stop running

START → Start/resume running

HOLD → Suspend

BRAN1 → Branch 1

BRAN2 → Branch 2

**Setting example**

TRIG:SEQ:SEL:EXEC HOLD

**Remarks**

The setting is error during the System Lock/warning.

**4.3.2.4** Sequence Function Control: Switch to Edit Mode`[[:SOURce]:SEQuence:EDIT`**Description**

Sets the Sequence state to the Edit state

Switches it from the Control state to the Edit state

**Setting parameter**

None

**Setting example**`SEQ:EDIT`**Remarks**

The setting is error during the System Lock/warning.

The setting can be made only in the Sequence control state.

**4.3.2.5** Sequence Function Control: Compile Setting Parameter and Switch to Control Mode`:TRIGger:SEQuence:COMPile`**Description**

Compiles the sequence data

**Setting parameter**

None

**Setting example**`TRIG:SEQ:COMP`**Remarks**

The setting is error during the System Lock/warning.

The operation can be made only in the Sequence edit state.

**4.3.2.6** Sequence Function Control: Get Running Step Number

[[:SOURce]:]SEQ:ence:CSTep?

Description

Gets the running step number

Query parameter

None

Response format

<INT>

Query example

SEQ:CST?

Response example

2

#### 4.3.2.7 Set Parameter Common to Steps (Enabled at Edit Mode): Set/Get Parameter Common to Steps [:SOURce]:SEQuence:CPARAmeter

##### Description

Sets/gets the sequence step-control parameter

Dummy data should be set for parameters excluded depending on the step.

Dummy data are not set.

##### Setting parameter

<time>,<start phs>,<start phs enable>,<stop phs >,<stop phs enable>,<step term>,<branch1>,<branch1 enable>,<branch2>,<branch2 enable>,<trig out>

<jump to>,<jump enable>,<jump cnt>,<code>,<branch1>,<branch1 enable>,<branch2>,<branch2 enable>,<trig out>

<time> ::= <REAL>

<REAL> → Step Time: 0.0010 to 999.9999, Resolution 0.0001

<start phs> ::= <REAL>

<REAL> → Start Phase: 0.0 to 359.9, Resolution 0.1

<start phs enable> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

<stop phs> ::= <REAL>

<REAL> → Stop Phase: 0.0 to 359.9, Resolution 0.1

<stop phs enable> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

<step term> ::= <DISC>

<DISC> ::= CONTInue | END | HOLD

CONTInue → Continue

END → Exit

HOLD → Hold

<jump to> ::= <INT>

<INT> → Jump to step number: 0 to 255

<jump enable> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

<jump cnt> ::= <INT>

<INT> → Jump count: 0 to 9999

Jump count 0 means infinite

<code> ::= <INT>

<INT> → Step sync output: 0 to 3

<branch1> ::= <INT>

<INT> → Branch 1 destination: 0 to 255

<branch1 enable> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

<branch2> ::= <INT>

<INT> → Branch 2 destination: 0 to 255

<branch2 enable> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

<trig out> ::= < BOL >

<BOL> → 0/OFF: Disable, 1/ON: Enable

#### Query parameter

None

#### Response format

<NR2>,<NR2>,<NBOL>,<NR2>,<NBOL>,<DISC>,<INT>,<NBOL>,<INT>,<INT>,  
<INT>,<NBOL>,<INT>,<NBOL>,<NBOL>

#### Setting example

SEQ:CPAR 10,90,ON,270,ON,CONT,3,ON,5,2,5,ON,6,ON,ON

#### Query example

SEQ:CPAR?

#### Response example

10.0000,90.0,1,270.0,1,CONT,3,1,5, 2,5,1,6,1,1

#### Remarks

The setting is error during the System Lock/warning.

The setting can be made only in the Sequence edit state.

**4.3.2.8** Set Parameter for Specified Step (Enabled at Edit Mode): Set/Get Step Number to Set  
[:SOURce]:SEQuence:STEP**Description**

Sets/gets the Sequence edit target step number

**Setting parameter**

<seq step> | MINimum | MAXimum

<seq step> ::= <INT>

<INT> → Sequence edit target step number: 0 to 255

MINimum → 0

MAXimum → 255

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

SEQ:STEP 1

**Query example**

SEQ:STEP?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting can be made only in the Sequence edit state.

**4.3.2.9** Set Specified Step Parameter (Enabled at Edit Mode): Set/Get Phase To Set (Phase Number)

[:SOURce]:SEQuence:INSTrument:NSELect

**Description**

Sets/gets the phase to set sequence (phase number)

**Setting parameter**

&lt;phase number&gt; | MINimum | MAXimum

&lt;phase number&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Phase number: 0 to 2

0: L1 phase | 1: L2 phase | 2: L3 phase

MINimum → 0

MAXimum → 2

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

SEQ:INST:NSEL 1

**Query example**

SEQ:INST:NSEL?

**Response example**

1

**4.3.2.10** Set Specified Step Parameter (Enabled at Edit Mode): Set/Get Phase To Set (Phase)

[:SOURce]:SEQUence:INSTrument:SELect

## Description

Sets/gets the phase to set sequence (phase)

## Setting parameter

&lt;phase name&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= L1 | L2 | L3

L1 → L1 phase, L2 → L2 phase, L3 → L3 phase

## Query parameter

None

## Response format

&lt;DISC&gt;

## Setting example

SEQ:INST:SEL L1

## Query example

SEQ:INST:SEL?

## Response example

L1



**4.3.2.11 Set Parameter for Specified Step (Enabled at Edit Mode): Set/Get Step Parameter**

[:SOURce]:SEQuence:SPARAmeter

**Description**

Sets/gets the sequence intra-Step parameter

Dummy data should be set for parameters excluded depending on the mode, step, and phase.

Dummy data are not set.

**Setting parameter**

<acv>,<acv mode>,<dcv>,<dcv mode>,<freq>,<freq mode>,<wave>,<phase>

<acv> ::= <REAL>

<REAL> → AC voltage

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

<acv mode> ::= <DISC>

<DISC> ::= CONST | KEEP | SWEEP

CONST → Constant

SWEEP → Sweep

KEEP → Keep

<dcv> ::= <REAL>

<REAL> → DC voltage

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

<dcv mode> ::= <DISC>

<DISC> ::= CONST | KEEP | SWEEP

CONST → Constant

SWEEP → Sweep

KEEP → Keep

<freq> ::= <REAL>

<REAL> → Frequency

For the setting value range, maximum value, minimum value, and resolution, see the instruction manual of the main unit.

<freq mode> ::= <DISC>

<DISC> ::= CONST | KEEP | SWEEP

CONST → Constant

SWEEP → Sweep

KEEP → Keep

<wave> ::= <DISC>

<DISC> ::= SIN | ARB1 | ARB2 | ARB3 | ARB4 | ARB5 | ARB6 | ARB7 | ARB8  
| ARB9 | ARB10 | ARB11 | ARB12 | ARB13 | ARB14 | ARB15 | ARB16 | CLP1  
| CLP2 | CLP3

SIN → Sine wave  
 ARB1 → Arbitrary wave 1  
 :  
 ARB16 → Arbitrary wave 16  
 CLP1 → Clipped sine wave 1  
 CLP2 → Clipped sine wave 2  
 CLP3 → Clipped sine wave 3

<phase> ::= <REAL>

<REAL> → Phase angle: 0.0 to 359.9, Resolution 0.1

#### Query parameter

None

#### Response format

<NR2>,<DISC>,<NR2>,<DISC>,<NR2>,<DISC>,<DISC>,<NR2>

#### Setting example

SEQ:SPAR 10,SWEEP,20,SWEEP,50,SWEEP,SIN,120

#### Query example

SEQ:SPAR?

#### Response example

10.0,SWEEP,20.0,SWEEP,50.00,SWEEP,SIN,120.0

#### Remarks

The setting is error during the System Lock/warning.

The setting can be made only in the Sequence edit state.

**4.3.2.12 Initialize/Read/Save Sequence Data (Enabled at Edit Mode): Initialize Sequence Data**

:TRACe|DATA:SEQuence:CLEAr

**Description**

Initializes the sequence data

**Setting parameter**

<seq memory> | MINimum | MAXimum

<seq memory> ::= <INT>

<INT> → 0 to 5

0: Sequence data being edited

1 to 5: Sequence Memory numbers 1-5

MINimum → 0

MAXimum → 5

**Setting example**

TRAC:SEQ:CLE 1

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

#### 4.3.2.13 Initialize/Read/Save Sequence Data (Enabled at Edit Mode): Set/Get Sequence Data Name :TRACe|DATA:SEQuence:NAME

##### Description

Sets/gets the sequence data name

##### Setting parameter

{<seq memory> | MINimum | MAXimum},<name>

<seq memory> ::= <INT>

<INT> → 1 to 5: Sequence Memory numbers 1-5

MINimum → 1

MAXimum → 5

<name> ::= <STR>

<STR> → Sequence data name

Arbitrary wave name (16 characters or less)

The arbitrary wave name should contain 16 one-byte characters

(alphanumeric characters) or less. The characters that can be used are as follows.

| valid character       |   |
|-----------------------|---|
| Number                | 0~9   |
| Alphabet (upper case) | A-Z   |
| Alphabet (lower case) | a-z   |
| Symbol                | !#\$%&'()+, - . : ; @ [ \ ] ^ _ ` { } ~ (space) |

##### Query parameter

<seq memory> | MINimum | MAXimum

<seq memory> ::= <INT>

<INT> → 1 to 5: Sequence Memory numbers 1-5

MINimum → 1

MAXimum → 5

##### Setting example

TRAC:SEQ:NAME 1,"SEQ1"

##### Query example

TRAC:SEQ:NAME? 1

##### Response example

"SEQ1"

##### Remarks

The setting is error during the System Lock/warning.

The setting/query cannot be done unless the output function is the Sequence.

**4.3.2.14 Initialize/Read/Save Sequence Data (Enabled at Edit Mode): Read Sequence Data**

:TRACe|DATA:SEQuence:RECall

**Description**

Reads the sequence data

**Setting parameter**

&lt;seq memory&gt; | MINimum | MAXimum

&lt;seq memory&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Sequence Memory number: 1 to 5

MINimum → 1

MAXimum → 5

**Setting example**

TRAC:SEQ:REC 1

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

**4.3.2.15 Initialize/Read/Save Sequence Data (Enabled at Edit Mode): Save Sequence Data**

:TRACe|DATA:SEQuence:STORe

**Description**

Saves the sequence data

**Setting parameter**

&lt;seq memory&gt; | MINimum | MAXimum

&lt;seq memory&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Sequence Memory number: 1 to 5

MINimum → 1

MAXimum → 5

**Setting example**

TRAC:SEQ:STOR 1

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

**4.3.2.16** Set Parameter for Step 0: Set/Get Output Frequency

[:SOURce]:SEQuence:FREQuency[:IMMediate]

**Description**

Sets/gets the output frequency, for sequence step 0

**Setting parameter**

&lt;frequency&gt;

&lt;frequency&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Frequency: The range that can be used are as follows.

40.00 to 550.00 (AC)

1.00 to 1500 (ACDC)

Resolution 0.01(<100Hz), 0.1(<1000Hz), 1( $\leq$ 5000Hz)

MINimum → 40.00 (AC, ACHF)/1.00 (ACDC)

MAXimum → 1500(AC, ACDC)

**Query parameter**

None

**Response format**

&lt;NR2&gt;

**Setting example**

:SEQ:FREQ 50

**Query example**

:SEQ:FREQ?

**Response example**

50.00

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

The setting can be made only in the Sequence edit state.

The setting can be made only in the Sequence edit step equal 0.

The setting can be made only in the AC-INT and ACDC-INT modes.

**4.3.2.17 Set Parameter for Step 0: Set/Get AC Output Voltage**

[:SOURce]:SEQUence:VOLTage[:LEVel][:IMMEDIATE][:AMPLitude]

**Description**

Sets/gets the AC output voltage, for sequence step 0

**Setting parameter**

&lt;volt&gt;

&lt;volt&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; →AC output voltage

For the setting value range and resolution, see the instruction manual of the main unit.

**Query parameter**

None

**Response format**

&lt;NR2&gt;

**Setting example**

:SEQ:VOLT 100

**Query example**

:SEQ:VOLT?

**Response example**

100.0

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

The setting can be made only in the Sequence edit state.

The setting can be made only in the Sequence edit step equal 0.

The setting can be made only in the AC-INT and ACDC-INT modes.

In the case of polyphase system, It is reflected in the set phase.

(See. 4.3.2.9 or 4.3.2.10)

**4.3.2.18** Set Parameter for Step 0: Set/Get DC Output Voltage

[:SOURce]:SEQuence:VOLTage[:LEVel][:IMMediate]:OFFSet

**Description**

Sets/gets the DC output voltage, for sequence step 0

**Setting parameter**

<volt offs>

<volt offs> ::= <REAL>

<REAL> → DC output voltage

For the setting value range and resolution, see the instruction manual of the main unit.

**Query parameter**

None

**Response format**

<NR2>

**Setting example**

:SEQ:VOLT:OFFS 10

**Query example**

:SEQ:VOLT:OFFS?

**Response example**

10.0

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

The setting can be made only in the Sequence edit state.

The setting can be made only in the Sequence edit step equal 0.

The setting can be made only in the ACDC-INT and DC-INT modes.



**4.3.2.19** Set Parameter for Step 0: Set/Get AC Output Current

[:SOURce]:SEQuence:CURRent[:LEVel][:IMMediate][:AMPLitude]

**Description**

Sets/gets the AC output current, for sequence step 0

**Setting parameter**

&lt;volt&gt;

&lt;volt&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; →AC output current

For the setting value range and resolution, see the instruction manual of the main unit.

**Query parameter**

None

**Response format**

&lt;NR2&gt;

**Setting example**

:SEQ:CURR 10

**Query example**

:SEQ:CURR?

**Response example**

10.00

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

The setting can be made only in the Sequence edit state.

The setting can be made only in the Sequence edit step equal 0.

The setting can be made only in the AC-INT and ACDC-INT modes.

In the case of polyphase system, it is reflected in the set phase.

(See. 4.3.2.9 or 4.3.2.10)

**4.3.2.20** Set Parameter for Step 0: Set/Get DC Output Current

[:SOURce]:SEQuence:CURRent[:LEVel][:IMMediate]:OFFSet

**Description**

Sets/gets the DC output current, for sequence step 0

**Setting parameter**

<curr offs>

<curr offs> ::= <REAL>

<REAL> → DC output current

For the setting value range and resolution, see the instruction manual of the main unit.

**Query parameter**

None

**Response format**

<NR2>

**Setting example**

:SEQ:CURR:OFFS 10

**Query example**

:SEQ:CURR:OFFS?

**Response example**

10.00

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Sequence.

The setting can be made only in the Sequence edit state.

The setting can be made only in the Sequence edit step equal 0.

The setting can be made only in the ACDC-INT and DC-INT modes.

**4.3.2.21** Output Setting (for All Phases): Set/Get Remote Sensing Function

:MEASure:CONFigure:SENSing

**Description**

Sets/gets the remote sensing

**Setting parameter**

&lt;state&gt; ::= &lt;INT&gt;

&lt;INT&gt; → 0: Disable, 1: Enable

**Query parameter**

None

**Response format**

&lt;INT&gt;

**Setting example**

MEAS:CONF:SENS 1

**Query example**

MEAS:CONF:SENS?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The remote sensing can be set only when the waveform is a sine wave for the AC mode in the AC-INT, AC-VCA, AC-SYNC, ACHF-INT, ACHF-VCA, DC-INT, and DC-VCA modes.

When the Sequence or Simulation function is selected, the remote sensing is forcibly set to off.

When the Sequence function is selected, it can be established on at the following condition.

Waveform is a sine wave

One of the AC voltage or the DC voltage is 0V setting.

When the output function is set to the Simulation, the remote sensing function cannot be enabled.

\* Same as the Continuous function command.

**4.3.2.22** Set Output Correction Function: Enable/Disable Autocal Function

:OUTPut:ACALibration

**Description**

Sets/gets the Autocal

**Setting parameter**

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

&lt;NBOL&gt;

**Setting example**

OUTP:ACAL ON

**Query example**

OUTP:ACAL?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The Autocal function can be used only when the output function is the Continuous function and the output mode is AC-INT/AC-VCA/AC-SYNC/ACHF-INT/ACHF-VCA/DC-INT/DC-VCA.

When an AC voltage waveform other than SIN is selected, the Autocal function cannot be enabled.

When the Autocal function is enabled, and the output function is set to the Sequence or Simulation, the Autocal function is forcibly disabled.

When the output function is set to the Sequence, it can be established on at the following condition.

AC voltage waveform is a sine wave.

One of the AC voltage or the DC voltage is 0V setting.

When the output function is set to the Simulation, the Autocal function cannot be enabled.

\* Same as the Continuous function command.

### 4.3.3 Simulation Function Commands

#### 4.3.3.1 Set Panel Display: Set/Get Step Time Display Unit

:DISPlay[:WINDow]:TIME:UNIT

##### Description

Sets/gets the Sequence/Simulation Step Time unit

It is reflected to the display and setting on the panel

The setting/getting from the remote command is made in seconds regardless of the DISP:TIME:UNIT setting

##### Setting parameter

<unit> ::= <DISC>

<DISC> ::= MS | S

MS → millisecond

S → second

##### Query parameter

None

##### Response format

<DISC>

##### Setting example

DISP:TIME:UNIT MS

##### Query example

DISP:TIME:UNIT?

##### Response example

MS

\* Same as the Sequence function command.

**4.3.3.2** Simulation Function Control: Get State

[:SOURce]:SIMulation:CONTrol[:STATe]?

## Description

Gets the simulation control state

## Query parameter

None

## Response format

&lt;DISC&gt;

## Query example

SIM:CONT?

## Response example

|         |               |
|---------|---------------|
| EDIT    | Edit state    |
| CONTROL | Control state |

**4.3.3.3** Simulation Function Control: Execution Control

:TRIGger:SIMulation:SElected:EXECute

## Description

Sets the simulation control

## Setting parameter

&lt;ctrl&gt; ::= &lt;DISC&gt;

&lt;DISC&gt; ::= STOP | START

STOP → Stop running

START → Start running

## Setting example

TRIG:SIM:SEL:EXEC STAR

## Remarks

The command is error during the System Lock/warning.

The operation can be made only in the Simulation control state.

**4.3.3.4** Simulation Function Control: Switch to Edit Mode`[[:SOURce]:SIMulation:EDIT`**Description**

Sets the Simulation state to the Edit state

Switches it from the Control state to the Edit state

**Setting parameter**

None

**Setting example**`SIM:EDIT`**Remarks**

The command is error during the System Lock/warning.

The setting can be made only in the Simulation control state.

**4.3.3.5** Simulation Function Control: Compile Setting Parameter and Switch to Control Mode`:TRIGger:SIMulation:COMPIle`**Description**

Compiles the simulation data

**Setting parameter**

None

**Setting example**`TRIG:SIM:COMP`**Remarks**

The command is error during the System Lock/warning.

The operation can be made only in the simulation edit state.

**4.3.3.6** Simulation Function Control: Get Running Step Number

[:SOURce]:SIMulation:CSTep?

## Description

Gets the running step number

## Query parameter

None

## Response format

&lt;INT&gt;

## Query example

SIM:CST?

## Response example

|   |                   |
|---|-------------------|
| 0 | Initial Step      |
| 1 | Normal 1 step     |
| 2 | Transition 1 step |
| 3 | Abnormal step     |
| 4 | Transition 2 step |
| 5 | Normal 2 step     |



**4.3.3.7 Set Repeat Execution (Enabled at Edit Mode): Enable/Disable Repeat Execution**

```
[[:SOURce]:SIMulation:REPeat:ENABle
```

**Description**

Sets/gets whether to enable/disable the Simulation repeat count

**Setting parameter**

```
<state> ::= <BOL>
```

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

```
<BOL>
```

**Setting example**

```
SIM:REP:ENAB ON
```

**Query example**

```
SIM:REP:ENAB?
```

**Response example**

```
1
```

**Remarks**

The command is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.8** Set Repeat Execution (Enabled at Edit Mode): Set/Get Repeat Execution Count

[:SOURce]:SIMulation:REPeat:COUNT

**Description**

Sets/gets the Simulation repeat count

**Setting parameter**

&lt;sim count&gt; | MINimum | MAXimum

&lt;sim count&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Repeat count: 0 to 9999 (0 means the infinite repeat)

MINimum → 0

MAXimum → 9999

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

SIM:REP:COUN 10

**Query example**

SIM:REP:COUN?

**Response example**

10

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.9** Set Parameter for Initial Step (Enabled at Edit Mode): Set/Get Output Voltage

[:SOURce]:SIMulation:INITial:VOLTage

**Description**

Sets/gets the Initial Step amplitude

**Setting parameter**

&lt;volt&gt; | MINimum | MAXimum

&lt;volt&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → AC output voltage

0.0 to 160.0 (100V range), Resolution 0.1

0.0 to 320.0 (200V range), Resolution 0.1

MINimum → 0.0

MAXimum → 160.0 (100V range) | 320.0 (200V range)

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:INIT:VOLT 100

**Query example**

SIM:INIT:VOLT?

**Response example**

100.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.10** Set Parameter for Initial Step (Enabled at Edit Mode): Set/Get Frequency

[:SOURce]:SIMulation:INITial:FREQuency

**Description**

Sets/gets the Initial Step frequency

**Setting parameter**

&lt;frequency&gt; | MINimum | MAXimum

&lt;frequency&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Frequency: 1.00 to 1500

Resolution 0.01(<100Hz), 0.1(<1000Hz), 1( $\leq$ 5000Hz)

MINimum → 1.00

MAXimum → 1500

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:INIT:FREQ 50

**Query example**

SIM:INIT:FREQ?

**Response example**

50.00

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

---

**4.3.3.11** Set Parameter for Initial Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Start Phase Specification  
[:SOURce]:SIMulation:INITial:PHASe:STARt:ENABLE

**Description**

Sets/gets whether to enable/disable the Initial Step start phase

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SIM:INIT:PHAS:STAR:ENAB ON

**Query example**

SIM:INIT:PHAS:STAR:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.12** Set Parameter for Initial Step (Enabled at Edit Mode): Set/Get Start Phase

[:SOURce]:SIMulation:INITial:PHASe:STARt[:IMMEdiate]

**Description**

Sets/gets the Initial Step start phase

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:INIT:PHAS:STAR 0

**Query example**

SIM:INIT:PHAS:STAR?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.13** Set Parameter for Initial Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Stop

Phase Specification

[:SOURce]:SIMulation:INITial:PHASe:STOP:ENABle

**Description**

Sets/gets whether to enable/disable the Initial Step stop phase

**Setting parameter**

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

&lt;NBOL&gt;

**Setting example**

SIM:INIT:PHAS:STOP:ENAB ON

**Query example**

SIM:INIT:PHAS:STOP:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.14** Set Parameter for Initial Step (Enabled at Edit Mode): Set/Get Stop Phase

[:SOURce]:SIMulation:INITial:PHASe:STOP[:IMMediate]

**Description**

Sets/gets the Initial Step stop phase

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:INIT:PHAS:STOP 0

**Query example**

SIM:INIT:PHAS:STOP?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.



**4.3.3.15** Set Parameter for Initial Step (Enabled at Edit Mode): Set/Get Synchronization Output

[:SOURce]:SIMulation:INITial:CODE

**Description**

Sets/gets the Initial Step - step sync output

**Setting parameter**

&lt;code&gt; | MINimum | MAXimum

&lt;code&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Step sync output: 0 to 3, Resolution 1

MINimum → 0

MAXimum → 3

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

SIM:INIT:CODE 1

**Query example**

SIM:INIT:CODE?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.16** Set Parameter for Normal 1 Step Parameter (Enabled at Edit Mode): Set/Get Execution Time  
[:SOURce]:SIMulation:NORMal1:TIME

**Description**

Sets/gets the Normal 1 Step Step Time

The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

<sim step time> | MINimum | MAXimum

<sim step time> ::= <REAL>

<REAL> → Step Time: 0.0010 to 999.9999, Resolution 0.0001

MINimum → 0.0010

MAXimum → 999.9999

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<NR2>

**Setting example**

SIM:NORM1:TIME 10

**Query example**

SIM:NORM1:TIME?

**Response example**

10.0000

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.17** Set Parameter for Normal 1 Step Parameter (Enabled at Edit Mode): Set/Get Output Voltage

[:SOURce]:SIMulation:NORMal1:VOLTage

**Description**

Sets/gets the Normal 1 Step amplitude

The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

&lt;volt&gt; | MINimum | MAXimum

&lt;volt&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → AC output voltage:

0.0 to 160.0 (100V range), Resolution 0.1

0.0 to 320.0 (200V range), Resolution 0.1

MINimum → 0.0

MAXimum → 160.0 (100V range) | 320.0 (200V range)

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:NORM1:VOLT 100

**Query example**

SIM:NORM1:VOLT?

**Response example**

100.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.18** Set Parameter for Normal 1 Step Parameter (Enabled at Edit Mode): Set/Get Frequency

[:SOURce]:SIMulation:NORMal1:FREQuency

**Description**

Sets/gets the Normal 1 Step frequency

The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

&lt;frequency&gt; | MINimum | MAXimum

&lt;frequency&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Frequency: 1.00 to 1500

Resolution 0.01(<100Hz), 0.1(<1000Hz), 1( $\leq$  5000Hz)

MINimum → 1.00

MAXimum → 1500.00

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:NORM1:FREQ 50

**Query example**

SIM:NORM1:FREQ?

**Response example**

50.00

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

---

**4.3.3.19** Set Parameter for Normal 1 Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Start Phase Specification  
[:SOURce]:SIMulation:NORMal1:PHASe:STARt:ENABle

**Description**

Sets/gets whether to enable/disable the Normal 1 Step start phase  
The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

<state> ::= <BOL>  
<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SIM:NORM1:PHAS:STAR:ENAB ON

**Query example**

SIM:NORM1:PHAS:STAR:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.  
The setting/query can be made only in the simulation edit state.

**4.3.3.20** Set Parameter for Normal 1 Step (Enabled at Edit Mode): Set/Get Start Phase

[:SOURce]:SIMulation:NORMal1:PHASe:STARt[:IMMEDIATE]

**Description**

Sets/gets the Normal 1 Step start phase

The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:NORM1:PHAS:STAR 0

**Query example**

SIM:NORM1:PHAS:STAR?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

---

**4.3.3.21** Set Parameter for Normal 1 Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Stop Phase Specification  
[:SOURce]:SIMulation:NORMal1:PHASe:STOP:ENABLE

**Description**

Sets/gets whether to enable/disable the Normal 1 Step stop phase  
The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

<state> ::= <BOL>  
<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SIM:NORM1:PHAS:STOP:ENAB ON

**Query example**

SIM:NORM1:PHAS:STOP:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.  
The setting/query can be made only in the simulation edit state.

**4.3.3.22** Set Parameter for Normal 1 Step Parameter (Enabled at Edit Mode): Set/Get Stop Phase

[:SOURce]:SIMulation:NORMal1:PHASe:STOP[:IMMEDIATE]

**Description**

Sets/gets the Normal 1 Step stop phase

The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:NORM1:PHAS:STOP 0

**Query example**

SIM:NORM1:PHAS:STOP?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.



**4.3.3.23** Set Parameter for Normal 1 Step Parameter (Enabled at Edit Mode): Set/Get Synchronization

Output

[:SOURce]:SIMulation:NORMal1:CODE

**Description**

Sets/gets the Normal 1 Step step sync output

The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

&lt;code&gt; | MINimum | MAXimum

&lt;code&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Step sync output: 0 to 3, Resolution 1

MINimum → 0

MAXimum → 3

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;INT&gt;

**Setting example**

SIM:NORM1:CODE 1

**Query example**

SIM:NORM1:CODE?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.24** Set Parameter for Normal 1 Step (Enabled at Edit Mode): Set/Get Trigger Output (G1)

[:SOURce]:SIMulation:NORMal1:TRIGger[:STATe]

**Description**

Sets/gets the Normal 1 Step trigger (G1)

The short form of NORMal1 is NORM1. "1" cannot be omitted.

**Setting parameter**

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

&lt;NBOL&gt;

**Setting example**

SIM:NORM1:TRIG ON

**Query example**

SIM:NORM1:TRIG?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.25** Set Parameter for Trans 1 Step Parameter (Enabled at Edit Mode): Set/Get Execution Time

[:SOURce]:SIMulation:TRANsition1:TIME

**Description**

Sets/gets the Trans 1 Step - Step Time

**Setting parameter**

&lt;sim step time&gt; | MINimum | MAXimum

&lt;sim step time&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Step Time: 0.0000, 0.0010 to 999.9999, Resolution 0.0001

MINimum → 0.0000

MAXimum → 999.9999

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:TRAN1:TIME 10

**Query example**

SIM:TRAN1:TIME?

**Response example**

10.0000

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.26** Set Parameter for Trans 1 Step Parameter (Enabled at Edit Mode): Set/Get Synchronization

Output

[[:SOURce]:SIMulation:TRANsition1:CODE

**Description**

Sets/gets the Trans 1 Step step sync output

**Setting parameter**

<code> | MINimum | MAXimum

<code> ::= <INT>

<INT> → Step sync output: 0 to 3, Resolution 1

MINimum → 0

MAXimum → 3

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

SIM:TRAN1:CODE 1

**Query example**

SIM:TRAN:CODE?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.27** Set Parameter for Trans 1 Step (Enabled at Edit Mode): Set/Get Trigger Output (G2)

[:SOURce]:SIMulation:TRANsition1:TRIGger[:STATe]

## Description

Sets/gets the Trans 1 Step trigger (G2)

## Setting parameter

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

## Query parameter

None

## Response format

&lt;NBOL&gt;

## Setting example

SIM:TRAN1:TRIG ON

## Query example

SIM:TRAN1:TRIG?

## Response example

1

## Remarks

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.28** Set Parameter for Abnormal Step Parameter (Enabled at Edit Mode): Set/Get Execution Time  
[:SOURce]:SIMulation:ABNormal:TIME**Description**

Sets/gets the Abnormal Step - Step Time

**Setting parameter**

<sim step time> | MINimum | MAXimum

<sim step time> ::= <REAL>

<REAL> → Step Time: 0.0010 to 999.9999, Resolution 0.0001

MINimum → 0.0010

MAXimum → 999.9999

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<NR2>

**Setting example**

SIM:ABN:TIME 10

**Query example**

SIM:ABN:TIME?

**Response example**

10.0000

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

### 4.3.3.29 Set Parameter for Abnormal Step Parameter (Enabled at Edit Mode): Set/Get Output Voltage [:SOURce]:SIMulation:ABNormal:VOLTage

#### Description

Sets/gets the Abnormal Step abnormal amplitude

#### Setting parameter

<volt> | MINimum | MAXimum

<volt> ::= <REAL>

<REAL> → AC output voltage

0.0 to 160.0 (100V range), Resolution 0.1

0.0 to 320.0 (200V range), Resolution 0.1

MINimum → 0.0

MAXimum → 160.0 (100V range) | 320.0 (200V range)

#### Query parameter

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

#### Response format

<NR2>

#### Setting example

SIM:ABN:VOLT 100

#### Query example

SIM:ABN:VOLT?

#### Response example

100.0

#### Remarks

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.30** Set Parameter for Abnormal Step Parameter (Enabled at Edit Mode): Set/Get Frequency

[:SOURce]:SIMulation:ABNormal:FREQuency

**Description**

Sets/gets the Abnormal Step abnormal frequency

**Setting parameter**

&lt;frequency&gt; | MINimum | MAXimum

&lt;frequency&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Frequency: 1.00 to 1500

Resolution 0.01(<100Hz), 0.1(<1000Hz), 1( $\leq$ 5000Hz)

MINimum → 1.00

MAXimum → 1500

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:ABN:FREQ 50

**Query example**

SIM:ABN:FREQ?

**Response example**

50.00

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.



---

**4.3.3.31** Set Parameter for Abnormal Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Start Phase Specification  
[:SOURce]:SIMulation:ABNormal:PHASe:STARt:ENABle

**Description**

Sets/gets whether to enable/disable the Abnormal Step start phase

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SIM:ABN:PHAS:STAR:ENAB ON

**Query example**

SIM:ABN:PHAS:STAR:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.32** Set Parameter for Abnormal Step (Enabled at Edit Mode): Set/Get Start Phase

[:SOURce]:SIMulation:ABNormal:PHASe:STARt[:IMMEdiate]

**Description**

Sets/gets the Abnormal Step start phase

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:ABN:PHAS:STAR 0

**Query example**

SIM:ABN:PHAS:STAR?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

---

**4.3.3.33** Set Parameter for Abnormal Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Stop Phase Specification  
[:SOURce]:SIMulation:ABNormal:PHASe:STOP:ENABle

**Description**

Sets/gets whether to enable/disable the Abnormal Step stop phase

**Setting parameter**

<state> ::= <BOL>

<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SIM:ABN:PHAS:STOP:ENAB ON

**Query example**

SIM:ABN:PHAS:STOP:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.34** Set Parameter for Abnormal Step Parameter (Enabled at Edit Mode): Set/Get Stop Phase  
[:SOURce]:SIMulation:ABNormal:PHASe:STOP[:IMMediate]**Description**

Sets/gets the Abnormal Step stop phase

**Setting parameter**

<phase> | MINimum | MAXimum

<phase> ::= <REAL>

<REAL> → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<NR2>

**Setting example**

SIM:ABN:PHAS:STOP 0

**Query example**

SIM:ABN:PHAS:STOP?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.35** Set Parameter for Abnormal Step Parameter (Enabled at Edit Mode): Set/Get Synchronization

Output

[[:SOURce]:SIMulation:ABNormal:CODE

**Description**

Sets/gets the Abnormal Step - step sync output

**Setting parameter**

<code> | MINimum | MAXimum

<code> ::= <INT>

<INT> → Step sync output: 0 to 3, Resolution 1

MINimum → 0

MAXimum → 3

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

SIM:ABN:CODE

**Query example**

SIM:ABN:CODE?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.36** Set Parameter for Abnormal Step (Enabled at Edit Mode): Set/Get Trigger Output (G3)

[:SOURce]:SIMulation:ABNormal:TRIGger[:STATe]

## Description

Sets/gets the Abnormal Step trigger (G3)

## Setting parameter

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

## Query parameter

None

## Response format

&lt;NBOL&gt;

## Setting example

SIM:ABN:TRIG ON

## Query example

SIM:ABN:TRIG?

## Response example

1

## Remarks

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.37** Set Parameter for Trans 2 Step Parameter (Enabled at Edit Mode): Set/Get Execution Time

[:SOURce]:SIMulation:TRANsition2:TIME

**Description**

Sets/gets the Trans 2 Step - Step Time

**Setting parameter**

&lt;sim step time&gt; | MINimum | MAXimum

&lt;sim step time&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Step Time: 0.0000, 0.0010 to 999.9999, Resolution 0.0001

MINimum → 0.0000

MAXimum → 999.9999

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:TRAN2:TIME 10

**Query example**

SIM:TRAN2:TIME?

**Response example**

10.0000

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.38** Set Parameter for Trans 2 Step Parameter (Enabled at Edit Mode): Set/Get Synchronization

Output

[:SOURce]:SIMulation:TRANsition2:CODE

## Description

Sets/gets the Trans 2 Step - step sync output

## Setting parameter

&lt;code&gt; | MINimum | MAXimum

&lt;code&gt; ::= &lt;INT&gt;

&lt;INT&gt; → Step sync output: 0 to 3, Resolution 1

MINimum → 0

MAXimum → 3

## Query parameter

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

## Response format

&lt;INT&gt;

## Setting example

SIM:TRAN2:CODE 1

## Query example

SIM:TRAN2:CODE?

## Response example

1

## Remarks

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.



**4.3.3.39** Set Parameter for Trans 2 Step (Enabled at Edit Mode): Set/Get Trigger Output (G4)

[:SOURce]:SIMulation:TRANsition2:TRIGger[:STATe]

## Description

Sets/gets the Trans 2 Step trigger (G4)

## Setting parameter

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

## Query parameter

None

## Response format

&lt;NBOL&gt;

## Setting example

SIM:TRAN2:TRIG ON

## Query example

SIM:TRAN2:TRIG?

## Response example

1

## Remarks

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.40** Set Parameter for Normal 2 Step Parameter (Enabled at Edit Mode): Set/Get Execution Time  
[:SOURce]:SIMulation:NORMal2:TIME**Description**

Sets/gets the Normal 2 Step Step Time

The short form of NORMal2 is NORM2. "2" cannot be omitted.

**Setting parameter**

<sim step time> | MINimum | MAXimum

<sim step time> ::= <REAL>

<REAL> → Step Time: 0.0010 to 999.9999, Resolution 0.0001

MINimum → 0.0010

MAXimum → 999.9999

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<NR2>

**Setting example**

SIM:NORM2:TIME 10

**Query example**

SIM:NORM2:TIME?

**Response example**

10.0000

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

---

**4.3.3.41** Set Parameter for Normal 2 Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Start Phase Specification  
[:SOURce]:SIMulation:NORMal2:PHASe:STARt:ENABle

**Description**

Sets/gets whether to enable/disable the Normal 2 Step start phase  
The short form of NORMAl2 is NORM2. "2" cannot be omitted.

**Setting parameter**

<state> ::= <BOL>  
<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SIM:NORM2:PHAS:STAR:ENAB ON

**Query example**

SIM:NORM2:PHAS:STAR:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.  
The setting/query can be made only in the simulation edit state.

**4.3.3.42** Set Parameter for Normal 2 Step (Enabled at Edit Mode): Set/Get Start Phase

[:SOURce]:SIMulation:NORMal2:PHASe:STARt[:IMMEDIATE]

**Description**

Sets/gets the Normal 2 Step start phase

The short form of NORMal2 is NORM2. "2" cannot be omitted.

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:NORM2:PHAS:STAR 0

**Query example**

SIM:NORM2:PHAS:STAR?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

---

**4.3.3.43** Set Parameter for Normal 2 Step (Enabled at Edit Mode): Set/Get Whether to Enable/Disable Stop Phase Specification  
[:SOURce]:SIMulation:NORMal2:PHASe:STOP:ENABLE

**Description**

Sets/gets whether to enable/disable the Normal 2 Step stop phase  
The short form of NORMAl2 is NORM2. "2" cannot be omitted.

**Setting parameter**

<state> ::= <BOL>  
<BOL> → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

<NBOL>

**Setting example**

SIM:NORM2:PHAS:STOP:ENAB ON

**Query example**

SIM:NORM2:PHAS:STOP:ENAB?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.  
The setting/query can be made only in the simulation edit state.

**4.3.3.44** Set Parameter for Normal 2 Step Parameter (Enabled at Edit Mode): Set/Get Stop Phase

[:SOURce]:SIMulation:NORMal2:PHASe:STOP[:IMMEDIATE]

**Description**

Sets/gets the Normal 2 Step stop phase

The short form of NORMal2 is NORM2. "2" cannot be omitted.

**Setting parameter**

&lt;phase&gt; | MINimum | MAXimum

&lt;phase&gt; ::= &lt;REAL&gt;

&lt;REAL&gt; → Phase: 0.0 to 359.9, Resolution 0.1

MINimum → 0.0

MAXimum → 359.9

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

&lt;NR2&gt;

**Setting example**

SIM:NORM2:PHAS:STOP 0

**Query example**

SIM:NORM2:PHAS:STOP?

**Response example**

0.0

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.45** Set Parameter for Normal 2 Step Parameter (Enabled at Edit Mode): Set/Get Synchronization

Output

[[:SOURce]:SIMulation:NORMal2:CODE

**Description**

Sets/gets the Normal 2 Step step sync output

The short form of NORMal2 is NORM2. "2" cannot be omitted.

**Setting parameter**

<code> | MINimum | MAXimum

<code> ::= <INT>

<INT> → Step sync output: 0 to 3, Resolution 1

MINimum → 0

MAXimum → 3

**Query parameter**

[MINimum | MAXimum]

MINimum → Gets the minimum value

MAXimum → Gets the maximum value

**Response format**

<INT>

**Setting example**

SIM:NORM2:CODE 1

**Query example**

SIM:NORM2:CODE?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.46** Set Parameter for Normal 2 Step (Enabled at Edit Mode): Set/Get Trigger Output (G5)

[:SOURce]:SIMulation:NORMal2:TRIGger[:STATe]

**Description**

Sets/gets the Normal 2 Step trigger (G5)

The short form of NORMal2 is NORM2. "2" cannot be omitted.

**Setting parameter**

&lt;state&gt; ::= &lt;BOL&gt;

&lt;BOL&gt; → 0/OFF: Disable, 1/ON: Enable

**Query parameter**

None

**Response format**

&lt;NBOL&gt;

**Setting example**

SIM:NORM2:TRIG ON

**Query example**

SIM:NORM2:TRIG?

**Response example**

1

**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.



**4.3.3.47** Initialize/Read/Save Simulation (Enabled at Edit Mode): Initialize Simulation Data`:TRACe|DATA:SIMulation:CLEar`**Description**

Initializes the simulation data

**Setting parameter**`<sim memory> | MINimum | MAXimum``<sim memory> ::= <INT>``<INT> → 0 to 5``0` : Simulation data being edited`1 to 5` : Simulation Memory number 1-5`MINimum` → 0`MAXimum` → 5**Setting example**`TRAC:SIM:CLE 1`**Remarks**

The setting is error during the System Lock/warning.

The setting/query can be made only in the simulation edit state.

**4.3.3.48** Initialize/Read/Save Simulation (Enabled at Edit Mode): Simulation Data Name

:TRACe|DATA:SIMulation:NAME

**Description**

Sets/gets the simulation data name

**Setting parameter**

{&lt;sim memory&gt; | MINimum | MAXimum},&lt;name&gt;

&lt;sim memory&gt; ::= &lt;INT&gt;

&lt;INT&gt; → 1 to 5: Simulation Memory number 1-5

MINimum → 1

MAXimum → 5

&lt;name&gt; ::= &lt;STR&gt;

&lt;STR&gt; → Sequence data name

Arbitrary wave name (16 characters or less)

The arbitrary wave name should contain 16 one-byte characters (alphanumeric characters) or less.

The characters that can be used are as follows.

| valid character       |                                    |
|-----------------------|------------------------------------|
| Number                | 0~9                                |
| Alphabet (upper case) | A-Z                                |
| Alphabet (lower case) | a-z                                |
| Symbol                | !#\$%&'()+,.-:;=@[]^_`{ }~ (space) |

**Query parameter**

&lt;sim memory&gt; | MINimum | MAXimum

&lt;sim memory&gt; ::= &lt;INT&gt;

&lt;INT&gt; → 1 to 5: Simulation Memory number 1-5

MINimum → 1

MAXimum → 5

**Setting example**

TRAC:SIM:NAME 1,"SIM1"

**Query example**

TRAC:SIM:NAME? 1

**Response example**

"SIM1"

**Remarks**

The setting is error during the System Lock/warning.

The setting cannot be done unless the output function is the Simulation.

**4.3.3.49** Initialize/Read/Save Simulation (Enabled at Edit Mode): Read Simulation Data

:TRACe|DATA:SIMulation:RECall

**Description**

Reads the simulation data

**Setting parameter**

<sim memory> | MINimum | MAXimum

<sim memory> ::= <INT>

<INT> → Simulation Memory number: 1 to 5

MINimum → 1

MAXimum → 5

**Setting example**

TRAC:SIM:REC 1

**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Simulation.

**4.3.3.50** Initialize/Read/Save Simulation (Enabled at Edit Mode): Save Simulation Data`:TRACe|DATA:SIMulation:STORe`**Description**

Saves the simulation data

**Setting parameter**`<sim memory> | MINimum | MAXimum``<sim memory> ::= <INT>``<INT>` → Simulation Memory number: 1 to 5`MINimum` → 1`MAXimum` → 5**Setting example**`TRAC:SIM:STOR 1`**Remarks**

The setting is error during the System Lock/warning.

The operation cannot be done unless the output function is the Simulation.

## 5. Programming Sample

|                                       |     |
|---------------------------------------|-----|
| 5.1 DP020AS Control Programming ..... | 252 |
| 5.2 Continuous Function Setting ..... | 253 |
| 5.3 Sequence Function Control .....   | 256 |
| 5.4 Simulation Function Control ..... | 258 |
| 5.5 Obtain Status Register .....      | 260 |
| 5.6 Cautions in Programming .....     | 262 |

---

## 5.1 DP020AS Control Programming

This section explains how to remote-control the DP020AS from the personal computer through the USB interface.

This section assumes an understanding of the VISA (Virtual Instrument System Architecture) programming interface and programming language supported by the VISA library.

This program is checking the operation in the following environment.

VISA library : National Instruments' NI-VISA.net 23.3 Library.

Programming Environment: Microsoft Visual Studio 2015 .Net Framework 4.8 with C# language

This section describes:

### a) Continuous output setting

Switch the output function mode to Continuous, make the output setting, then turn on the output. After that, use the measurement function to read the measured output value.

### b) Sequence function control

Switch the output function mode to Sequence, read the sequence data pre-stored in the main unit, and start the sequence execution. After that, use the measurement function to read the measured output value.

### c) Simulation function control

Switch the output function mode to Simulation, read the simulation data pre-stored in the main unit, and start the power fluctuation simulation. After that, use the measurement function to read the measured output value.

### d) Obtain status register

After setting the warning status transition filter and the warning event enable register, obtain the status byte to check for the warning state.

These explanations are intended for helping you understand the remote command control procedure, and thus omit general programming considerations such as error detection processing and variable initialization processing.

For the VISA library installation, see the document from the VISA library distributors.

## 5.2 Continuous Function Setting

```

using System;
using System.Windows.Forms;
using System.Linq;

// <Process flow>
// ■Initialize the communication state.
// Namespace declaration of NI-VISA library
using Ivi.Visa;
using NationalInstruments.Visa;

// Search for DP020AS using the resource string
// Generate the VISA session of the NI-VISA library
// Use the VISA session to specify the DP020AS serial number and
// establish the communication session with the DP020AS.
// In your program, write the exception handling.
// Use catch to write the error handling when communication session is established.
// Whenever an error occurs, NI-VISA library throws it to the exception.
//
System.Collections.Generic.IEnumerable<string> strResNames;
ResourceManager rm;
rm = new ResourceManager();
strResNames = rm.Find("USB0::0x0D4A::0x0077::00000000::INSTR");

IMessageBasedSession dp;
dp = (MessageBasedSession)rm.Open(strResNames.ElementAt(0));

// Set to Terminator LF(=0x0a)
dp.TerminationCharacter = 0x0A;

// Set to the Remote State
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Remote);

// Clear the device
dp.Clear();

// Clear the error status
dp.RawIO.Write("*CLS\n");
System.Threading.Thread.Sleep(1000);

```

```
// ■Return the DP020AS to the initial state just after the startup.
// Return the output setting to the state just after the startup.
dp.RawIO.Write("*RST\n");
System.Threading.Thread.Sleep(1000);

// ■Set the output function to Continuous.
dp.RawIO.Write(":SYSTem:CONFIgure:MODE CONTInuous\n");
System.Threading.Thread.Sleep(1000);

// ■Set the output mode.
dp.RawIO.Write(":SOURce:MODE AC_INT\n");
System.Threading.Thread.Sleep(1000);

// ■Set the output range.
dp.RawIO.Write(":SOURce:VOLTagE:RANGe R100V\n");
System.Threading.Thread.Sleep(1000);

// ■Set the output waveform.
dp.RawIO.Write(":SOURce:FUNCTion:SHAPE:IMMediate SIN\n");
System.Threading.Thread.Sleep(1000);

// ■Set the output frequency.
dp.RawIO.Write(":SOURce:FREQUency:IMMediate 50.00\n");
System.Threading.Thread.Sleep(1000);

// ■Set the output voltage.
dp.RawIO.Write(":SOURce:VOLTagE:LEVel:IMMediate:AMPLitude 100.0\n");
System.Threading.Thread.Sleep(1000);

// ■Turn on the output.
dp.RawIO.Write(":OUTPut:STATe ON\n");
System.Threading.Thread.Sleep(1000);

// ■Obtain the measured output voltage value.
string strMeasureVoltageRMS = "";
dp.RawIO.Write(":MEASure:SCALar:VOLTagE:RMS?\n");
strMeasureVoltageRMS = dp.RawIO.ReadString();

// ■Obtain the measured output current value.
string strMeasureCurrentRMS = "";
dp.RawIO.Write(":MEASure:SCALar:CURREnt:RMS?\n");
strMeasureCurrentRMS = dp.RawIO.ReadString();
```



// ■ Turn off the output.

```
dp.RawIO.Write(":OUTPut:STATe OFF\n");
```

```
System.Threading.Thread.Sleep(1000);
```

// Clear the Remote state

```
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Local);
```

// ■ Terminate the communication and release the session.

```
dp.Dispose();
```

---

## 5.3 Sequence Function Control

```
using System;
using System.Windows.Forms;
using System.Linq;

// <Process flow>
// ■Initialize the communication state.
// Namespace declaration of NI-VISA library
using Ivi.Visa;
using NationalInstruments.Visa;

// Search for DP020AS using the resource string
// Generate the VISA session of the NI-VISA library
// Use the VISA session to specify the DP020AS serial number and
// establish the communication session with the DP020AS.
// In your program, write the exception handling.
// Use catch to write the error handling when communication session is established.
// Whenever an error occurs, NI-VISA library throws it to the exception.
//
System.Collections.Generic.IEnumerable<string> strResNames;
ResourceManager rm;
rm = new ResourceManager();
strResNames = rm.Find("USB0::0x0D4A::0x0077::00000000::INSTR");

IMessageBasedSession dp;
dp = (MessageBasedSession)rm.Open(strResNames.ElementAt(0));

// Set to Terminator LF(=0x0a)
dp.TerminationCharacter = 0x0A;

// Set to the Remote State
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Remote);

// Clear the device
dp.Clear();

// Clear the error status
dp.RawIO.Write("*CLS\n");
System.Threading.Thread.Sleep(1000);
```

```
// ■Set the output function to Sequence.
dp.RawIO.Write(":SYSTem:CONFigure:MODE SEQUENCE\n");
System.Threading.Thread.Sleep(1000);

// ■Recall the sequence data from the Sequence Memory 1.
dp.RawIO.Write(":TRACe:SEQuence:RECall1\n");
System.Threading.Thread.Sleep(1000);

// ■Compile the sequence data.
dp.RawIO.Write(":TRIGger:SEQuence:COMPIle\n");
System.Threading.Thread.Sleep(1000);

// ■Turn on the output.
dp.RawIO.Write(":OUTPut:STATe ON\n");
System.Threading.Thread.Sleep(1000);

// ■Start the sequence.
dp.RawIO.Write(":TRIGger:SEQuence:SElected:EXECute START\n");
System.Threading.Thread.Sleep(1000);

// ■Obtain the measured output voltage value.
string strMeasureVoltageRMS = "";
dp.RawIO.Write(":MEASure:SCALar:VOLTage:RMS?\n");
strMeasureVoltageRMS = dp.RawIO.ReadString();

// ■Obtain the measured output current value.
string strMeasureCurrentRMS = "";
dp.RawIO.Write(":MEASure:SCALar:CURRent:RMS?\n");
strMeasureCurrentRMS = dp.RawIO.ReadString();

// ■Turn off the output.
dp.RawIO.Write(":OUTPut:STATe OFF\n");
System.Threading.Thread.Sleep(1000);

// ■Change to the Edit mode.
dp.RawIO.Write(":SOURce:SEQuence:EDIT\n");

// Clear the Remote state
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Local);

// ■Terminate the communication and release the session.
dp.Dispose();
```

## 5.4 Simulation Function Control

```

using System;
using System.Windows.Forms;
using System.Linq;

// <Process flow>
// ■Initialize the communication state.
// Namespace declaration of NI-VISA library
using Ivi.Visa;
using NationalInstruments.Visa;

// Search for DP020AS using the resource string
// Generate the VISA session of the NI-VISA library
// Use the VISA session to specify the DP020AS serial number and
// establish the communication session with the DP020AS.
// In your program, write the exception handling.
// Use catch to write the error handling when communication session is established.
// Whenever an error occurs, NI-VISA library throws it to the exception.
//
System.Collections.Generic.IEnumerable<string> strResNames;
ResourceManager rm;
rm = new ResourceManager();
strResNames = rm.Find("USB0::0x0D4A::0x0077::00000000::INSTR");

IMessageBasedSession dp;
dp = (MessageBasedSession)rm.Open(strResNames.ElementAt(0));

// Set to Terminator LF(=0x0a)
dp.TerminationCharacter = 0x0A;

// Set to the Remote State
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Remote);

// Clear the device
dp.Clear();

// Clear the error status
dp.RawIO.Write("*CLS\n");
System.Threading.Thread.Sleep(1000);

```

```
// ■Set the output function to Simulation.
dp.RawIO.Write(":SYSTem:CONFIgure:MODE SIMulation\n");
System.Threading.Thread.Sleep(1000);

// ■Recall the power fluctuation simulation data from the Simulation Memory 1.
dp.RawIO.Write(":TRACe:SIMulation:RECall 1\n");
System.Threading.Thread.Sleep(1000);

// ■Compile the simulation data.
dp.RawIO.Write(":TRIGger:SIMulation:COMPIle\n");
System.Threading.Thread.Sleep(1000);

// ■Turn on the output.
dp.RawIO.Write(":OUTPut:STATe ON\n");
System.Threading.Thread.Sleep(1000);

// ■Starts the Simulation.
dp.RawIO.Write(":TRIGger:SIMulation:SElected:EXECute START\n");
System.Threading.Thread.Sleep(1000);

// ■Obtain the measured output voltage value.
string strMeasureVoltageRMS = "";
dp.RawIO.Write(":MEASure:SCALAR:VOLTage:RMS?\n");
strMeasureVoltageRMS = dp.RawIO.ReadString();

// ■Obtain the measured output current value.
string strMeasureCurrentRMS = "";
dp.RawIO.Write(":MEASure:SCALAR:CURRent:RMS?\n");
strMeasureCurrentRMS = dp.RawIO.ReadString();

// ■Turn off the output.
dp.RawIO.Write(":OUTPut:STATe OFF\n");
System.Threading.Thread.Sleep(1000);

// ■Change to the Edit mode.
dp.RawIO.Write(":SOURce:SIMulation:EDIT\n");

// Clear the Remote state
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Local);

// ■Terminate the communication and release the session.
dp.Dispose();
```

## 5.5 Obtain Status Register

```

using System;
using System.Windows.Forms;
using System.Linq;

// <Process flow>
// ■Initialize the communication state.
// NameSpace declaration of NI-VISA library
using Ivi.Visa;
using NationalInstruments.Visa;

// Search for DP020AS using the resource string
// Generate the VISA session of the NI-VISA library
// Use the VISA session to specify the DP020AS serial number and
// establish the communication session with the DP020AS.
// In your program, write the exception handling.
// Use catch to write the error handling when communication session is established.
// Whenever an error occurs, NI-VISA library throws it to the exception.
//
System.Collections.Generic.IEnumerable<string> strResNames;
ResourceManager rm;
rm = new ResourceManager();
strResNames = rm.Find("USB0::0x0D4A::0x0077::00000000::INSTR");

IMessageBasedSession dp;
dp = (MessageBasedSession)rm.Open(strResNames.ElementAt(0));

// Set to Terminator LF(=0x0a)
dp.TerminationCharacter = 0x0A;

// Set to the Remote State
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Remote);

// Clear the device
dp.Clear();

// Clear the error status
dp.RawIO.Write("*CLS\n");
System.Threading.Thread.Sleep(1000);

```

```

// Set the warning transition filter (positive logic)
dp.RawIO.Write(":STATus:WARNing:PTRansition 65535\n");
System.Threading.Thread.Sleep(1000);

// Set permission of the warning event register
dp.RawIO.Write(":STATus:WARNing:ENABle 65535\n");
System.Threading.Thread.Sleep(1000);

// ■Obtain the status byte to check if the warning state is detected

// ReadStatusByte() needs to be repeatedly queried in order to detect the change
// in each bit of the register.
// In actual programming, the status byte is acquired by worker-threading or
// other means.

StatusByteFlags sbFlag = 0;
sbFlag = dp.ReadStatusByte();
short sFlag = (short)sbFlag;

// Status changes
if (sFlag != 0)
{
    // Is the warning register detected?
    if ((sFlag & 2) == 2)
    {
        // Obtain the warning state
        string strWarn = "";
        dp.RawIO.Write(":STATus:WARNing:CONDition?\n");
        strWarn=dp.RawIO.ReadString();
        System.Threading.Thread.Sleep(1000);

        // Release the warning
        // This also clears the warning screen display on the DP020AS.
        dp.RawIO.Write(":SYSTem:WRELease\n");
        System.Threading.Thread.Sleep(1000);
    }
}

// Clear the Remote state
new UsbSession(dp.ResourceName).SendRemoteLocalCommand(RemoteLocalMode.Local);

// ■Terminate the communication and release the session.
dp.Dispose();

```

## 5.6 Cautions in Programming

- a) When a program command send from the computer is interrupted on the way, an error may occur in the next program command. If there was an interruption on the way, perform the device clear.
- b) When the computer interrupts receiving on the way after the computer sent a query message then data transfer was started with the DP020AS specified as talker, the DP020AS may freeze awaiting for send. If there was an interruption on the way, perform the device clear.
- c) After the computer sent a query message, if the computer sends another query message without specifying the DP020AS as talker, the response to the first query may be returned with the trailing response to the next one (without delimiter).

When the DP020AS does not work as talker after a query message is sent, perform the device clear.



## 6. Error Message List

|                              |     |
|------------------------------|-----|
| 6.1 Error Message List ..... | 264 |
|------------------------------|-----|

## 6.1 Error Message List

Table 6.1 shows the list of error messages.

**Table 6.1 Error Message List**

| ID   | Message                                      | Meaning  |
|------|--|--|
| 0    | No error                                     | There is no error.   |
| -100 | Command error                                | An illegal command was received.   |
| -102 | Syntax error                                 | An undefined command or parameter was received.  |
| -103 | Invalid separator                            | An illegal separator was received.   |
| -104 | Data type error                              | The received command or parameter was undefined, and thus was recognized as an illegal command.                              |
| -108 | Parameter not allowed                        | There are too many parameters.   |
| -109 | Missing parameter                            | One or more parameters are missing.  |
| -110 | Command header error                         | There is an error in the command header.   |
| -111 | Header separator error                       | There is an error in the keyword separator of the command header.  |
| -113 | Undefined header                             | The received string contains an invalid header.  |
| -120 | Numeric data error                           | There is an error in the numerical parameter.  |
| -130 | Suffix error                                 | There is an error in the suffix of the numerical parameter.  |
| -140 | Character data error                         | There is an error in the discrete parameter.   |
| -144 | Character data too long                      | The discrete parameter is too long.  |
| -150 | String data error                            | There is an error in the string parameter.   |
| -160 | Block data error                             | There is an error in the block parameter.  |
| -200 | Execution error                              | An error was detected during command execution.  |
| -222 | Data out of range                            | A command parameter is out of the valid range.   |
| -300 | Device-specific error                        | This is a device-specific error.   |
| -350 | Queue overflow                               | The error queue overflowed and thus can no longer hold a new error.  |
| -363 | Input buffer overrun                         | The command input buffer overflowed.   |
| -410 | Query INTERRUPTED                            | A new response message entered the output queue, and thus the previous message was lost.                                     |
| -420 | Query UNTERMINATED                           | The talker was specified, but the output queue has no response message.  |
| -430 | Query DEADLOCKED                             | Send/receive came to a deadlock. Sending is canceled.  |
| -440 | Query UNTERMINATED after indefinite response | The response request order is wrong.   |
| 2    | Invalid in This Output Mode                  | This output mode does not allow the setting (execution). Change the output mode.   |
| 3    | Invalid with Output ON                       | The output on state does not allow the setting (execution). Turn the output off, then set (execute).                         |
| 4    | Invalid with Output OFF                      | The output off state does not allow the setting (execution). Turn the output on, then set (execute).                         |
| 5    | Busy   | The busy state does not allow the setting (execution). Execute after the busy icon disappears.                               |
| 7    | Invalid with Sweep                           | The soft start or soft stop state does not allow the setting(execution). Please set (execute) after exiting the sweep state. |
| 11   | Under Error State                            | Cannot be set (executed) because it is in a protected state.   |

## 6.1 Error Message List

|    |                                   |  |
|----|-----------------------------------|--|
| 13 | Auto Cal Disabled                 | The Autocal setting becomes off because of out of the calibration range.   |
| 15 | AGC Disabled                      | The AGC setting becomes off because of out of the calibration range.   |
| 16 | Invalid in Sequence Edit          | The sequence edit mode does not allow the setting (execution).   |
| 17 | Invalid in Sequence Control       | The sequence control mode does not allow the setting (execution).  |
| 18 | Invalid in Simulation Edit        | The simulation edit mode does not allow the setting (execution).   |
| 19 | Invalid in Simulation Control     | The simulation control mode does not allow the setting (execution).  |
| 20 | Invalid                           | The condition cannot be set (executed).  |
| 23 | Option not Installed              | Setting (execution) is not possible because the option is invalid.   |
| 24 | USB Memory Unconnected            | No USB memory is connected. Connect a USB memory.  |
| 25 | USB Memory Removed illegally      | The USB memory was removed before the eject operation. Perform the eject operation before removing a USB memory.   |
| 26 | USB Memory Access Error           | The USB memory has an access error.  |
| 27 | Too Many Files                    | There are too many files in the USB memory. Each folder can contain 500 files or less.   |
| 29 | Sync Frequency Unlocked           | The output cannot be turned on because of not locking to the synchronization frequency.  |
| 31 | Configure Memory Error            | The internal memory has an error.  |
| 33 | Line Overvoltage                  | The power input voltage is excessive. Check if the power input is appropriate.   |
| 34 | Line Undervoltage                 | The power input voltage is insufficient. Check if the power input is appropriate.  |
| 35 | Line Frequency Error              | The power input frequency is abnormal. Check if the power input is appropriate.  |
| 36 | Communication Failure Power Unit  | There is a communication error between the firmware and the power unit. Cycle the power. If it still occurs, contact the seller (NF Corporation or our agent) from whom you purchased the product. |
| 37 | Internal Communication Failure    | There is a communication error in the firmware. Cycle the power. If it still occurs, contact the seller (NF Corporation or our agent) from whom you purchased the product.                         |
| 38 | Communication Failure SC          | The polyphase system communication is abnormal. Make sure the System Cable is connected properly.  |
| 41 | Mismatched Firmware Versions      | In the polyphase system, the main firmware versions of the cabinets do not match.  |
| 42 | Mismatched Internal Versions      | The versions do not match within the DP020AS. Please update again.   |
| 43 | Mismatched Pair of HW/SW Versions | The firmware and internal hardware versions do not match. Please update again.   |
| 44 | Unsupported Power Unit Version    | The power unit version is not supported. Please update again.  |
| 45 | Mismatched Cabinet Connect        | In the polyphase system, the output capacities of the phases do not match.   |
| 48 | Outyput Overvoltage               | An output overvoltage has occurred.  |

## 6.1 Error Message List

|    |  |  |
|----|--|--|
| 50 | Output Overcurrent[RMS]                | An output overcurrent[RMS] has occurred.   |
| 51 | Output Overcurrent[Peak]               | An output overcurrent[Peak] has occurred.  |
| 52 | Power Unit DCPS Error                  | A PFC error has occurred.  |
| 53 | Power Unit DCPS Overvoltage            | A DC power supply overvoltage has occurred.  |
| 54 | Power Unit DCPS Undervoltage           | A DC power supply undervoltage has occurred.   |
| 55 | Overheat                               | Overheating has occurred.  |
| 56 | Sensing Voltage Error                  | A sensing voltage error has occurred.  |
| 57 | Sync Frequency Error                   | A sync frequency error has occurred.   |
| 58 | Limiter[RMS]                           | Output OFF occurred due to effective value limiter.  |
| 59 | Limiter[Peak]                          | Output OFF occurred due to peak limiter.   |
| 60 | Power Unit Internal Error              | An error has occurred inside the power unit.   |
| 67 | Battery Depleted                       | The built-in battery is exhausted.   |
| 68 | Current Parameters Rounded             | Data with different numbers of connected devices was read.   |
| 69 | Polyphase Connection Updated           | A phase configuration is detected that is different from the previous one.   |
| 75 | Calibration Data Error in Control Unit | There is an error in the adjustment data. Cycle the power. If it still occurs, contact the seller (NF Corporation or our agent) from whom you purchased the product. |
| 76 | Property Data Error in Control Unit    | There is an error in the option data. Cycle the power. If it still occurs, contact the seller (NF Corporation or our agent) from whom you purchased the product.     |
| 77 | USB Memory Connected                   | USB memory connection was detected.  |
| 78 | USB Memory Disconnected                | USB memory was disconnected.   |
| 80 | Invalid Character                      | An invalid character was specified.  |
| 82 | Sequence Compile Error                 | Sequence compilation failed.   |
| 83 | Simulation Compile Error               | Simulation compilation failed.   |
| 84 | System Updated Please Reboot           | The firmware was updated.  |
| 85 | System Update Fialed                   | Firmware update failed.  |
| 88 | Auxiliary Power Failure                | The internal auxiliary power is abnormal.  |
| 90 | Mismatched System Setting              | An inconsistency in system settings between cabinet was detected.  |
| 91 | Shut Down                              | Shutdown detected.   |
| 92 | Power off Detected                     | A power outage was detected.   |

## 6.1 Error Message List

|    |                                  |   |
|----|----------------------------------|---|
| 93 | Power off Waiting                | Waiting for power off operation.                                    |
| 95 | Memory Data Error                | The read memory data is abnormal.                                   |
| 96 | Update Data Error                | The update data is invalid.   |
| 97 | Fail with<br>Duplicated FileName | Save the same file name in the USB memory.                          |
| 98 | No update Needed                 | This is a version that does not require updates.                    |
| 99 | Output Overpower                 | Overpower protection occurred when multiple cabinet were installed. |



## 7. Specifications

|   |     |
|---|-----|
| 7.1 Interface Functions.....            | 270 |
| 7.2 GPIB Bus Driver .....               | 270 |
| 7.3 Code Used .....                     | 271 |
| 7.4 Response to Interface Message ..... | 271 |
| 7.5 Multi-Line Interface Message.....   | 272 |

## 7.1 Interface Functions

**Table 7.1 Interface Functions**

| Function           | Subset | Description  |
|--------------------|--------|--|
| Source handshake   | SH1    | All send handshake functions provided                    |
| Acceptor handshake | AH1    | All receive handshake functions provided                 |
| Talker             | T6     | Basic talker functions,<br>Talker release by the MLA     |
| Listener           | L4     | Basic listener functions,<br>Listener release by the MTA |
| Service request    | SR1    | All service request functions provided                   |
| Remote/Local       | RL1    | All remote/local functions provided                      |
| Parallel poll      | PP0    | Parallel poll function not provided                      |
| Device clear       | DC1    | All device clear functions provided                      |
| Device trigger     | DT0    | Device trigger function not provided                     |
| Controller         | C0     | Controller function not provided                         |

- RS232 and LAN interface do not correspond to all the above functions.

## 7.2 GPIB Bus Driver

**Table 7.2 Bus Driver Specifications**

|                |                 |                          |
|----------------|-----------------|--------------------------|
| Data bus       | DIO1 to 8       | Open collector           |
| Handshake bus  | NRFD, NDAC, DAV | Open collector tri-state |
| Management bus | SRQ<br>EOI      | Open collector tri-state |



## 7.3 Code Used

The DP020AS as the listener can accept ISO 7-bit codes (JIS/ASCII), except when receiving binary data, and ignores MSB of 8-bit data (e.g., parity). Any program codes are interpreted and executed with no case sensitivity. However, the control characters except <CR>, <LF>, and <TAB> (codes from 0 to 8, B, D to 1F, and 7F in hexadecimal) are ignored.

As the talker, it sends ISO 7-bit codes (JIS or ASCII) with no parity except when sending binary data (MSB of 8-bit data = 0). All alphabetical characters in the program code are sent in uppercase.

When binary data is sent and received, all 8-bit patterns are valid and the standard IEEE floating point format (including any control codes) is supported.

## 7.4 Response to Interface Message

**Table 7.3 Response to Interface Message**

|             |   |
|-------------|---|
| IFC         | <ul style="list-style-type: none"> <li>• Initialize the GPIB interface message.</li> <li>• Release the specified listener or talker.</li> </ul> |
| DCL and SDC | <ul style="list-style-type: none"> <li>• Clear the I/O buffer for GPIB.</li> </ul>  |
| LLO         | <ul style="list-style-type: none"> <li>• Disable the LOCAL key on the panel.</li> </ul>   |
| GTL         | <ul style="list-style-type: none"> <li>• Turn to the Local state.</li> </ul>  |

## 7.5 Multi-Line Interface Message

| *2                          |    |    |    |     | b7                          | 0         | *1  | 0                             | 0   | 0  | 0                            | 1 | 1   | 1                          | 1   | 1  | MSG                           |   |   |     |   |
|-----------------------------|----|----|----|-----|-----------------------------|-----------|-----|-------------------------------|-----|----|------------------------------|---|-----|----------------------------|-----|----|-------------------------------|---|---|-----|---|
|                             |    |    |    |     | b6                          | 0         | MSG | 0                             | MSG | 1  | MSG                          | 0 | MSG | 0                          | MSG | 1  | MSG                           |   |   |     |   |
|                             |    |    |    |     | b5                          | 0         |     | 1                             |     | 0  |                              | 0 |     | 1                          |     | 1  |                               |   |   |     |   |
| b4                          | b3 | b2 | b1 | Row | Column                      | 0         |     | 1                             |     | 2  |                              | 3 |     | 4                          |     | 5  |                               | 6 |   | 7   |   |
| 0                           | 0  | 0  | 0  | 0   | NUL                         |           |     | DLE                           |     | SP |                              | ↑ | 0   | @                          | ↑   | P  | ↑                             | ` | ↑ | p   | ↑ |
| 0                           | 0  | 0  | 1  | 1   | SOH                         | GTL       | DC1 | LLO                           | !   |    |                              | ↑ | 1   | A                          | ↑   | Q  | ↑                             | a | ↑ | q   | ↑ |
| 0                           | 0  | 1  | 0  | 2   | STX                         |           | DC2 |                               | "   |    |                              | ↑ | 2   | B                          | ↑   | R  | ↑                             | b | ↑ | r   | ↑ |
| 0                           | 0  | 1  | 1  | 3   | ETX                         |           | DC3 |                               | #   |    |                              | ↑ | 3   | C                          | ↑   | S  | ↑                             | c | ↑ | s   | ↑ |
| 0                           | 1  | 0  | 0  | 4   | EOT                         | SDC       | DC4 | DCL                           | \$  |    |                              | ↑ | 4   | D                          | ↑   | T  | ↑                             | d | ↑ | t   | ↑ |
| 0                           | 1  | 0  | 1  | 5   | ENQ                         | *3<br>PPC | NAK | PPU                           | %   |    |                              | ↑ | 5   | E                          | ↑   | U  | ↑                             | e | ↑ | u   | ↑ |
| 0                           | 1  | 1  | 0  | 6   | ACK                         |           | SYN |                               | &   |    |                              | ↑ | 6   | F                          | ↑   | V  | ↑                             | f | ↑ | v   | ↑ |
| 0                           | 1  | 1  | 1  | 7   | BEL                         |           | ETB |                               | '   |    |                              | ↑ | 7   | G                          | ↑   | W  | ↑                             | g | ↑ | w   | ↑ |
| 1                           | 0  | 0  | 0  | 8   | BS                          | GET       | CAN | SPE                           | (   |    |                              | ↑ | 8   | H                          | ↑   | X  | ↑                             | h | ↑ | x   | ↑ |
| 1                           | 0  | 0  | 1  | 9   | HT                          | TCT       | EM  | SPD                           | )   |    |                              | ↑ | 9   | I                          | ↑   | Y  | ↑                             | i | ↑ | y   | ↑ |
| 1                           | 0  | 1  | 0  | 10  | LF                          |           | SUB |                               | *   |    |                              | ↑ | :   | J                          | ↑   | Z  | ↑                             | j | ↑ | z   | ↑ |
| 1                           | 0  | 1  | 1  | 11  | VT                          |           | ESC |                               | +   |    |                              | ↑ | :   | K                          | ↑   | [  | ↑                             | k | ↑ | {   | ↑ |
| 1                           | 1  | 0  | 0  | 12  | FF                          |           | FS  |                               | ,   |    |                              | ↑ | <   | L                          | ↑   | *4 | ↑                             | l | ↑ |     | ↑ |
| 1                           | 1  | 0  | 1  | 13  | CR                          |           | GS  |                               | -   |    |                              | ↑ | =   | M                          | ↑   | ]  | ↑                             | m | ↑ | }   | ↑ |
| 1                           | 1  | 1  | 0  | 14  | SO                          |           | RS  |                               | .   |    |                              | ↑ | >   | N                          | ↑   | ^  | ↑                             | n | ↑ | ~   | ↑ |
| 1                           | 1  | 1  | 1  | 15  | SI                          |           | US  |                               | /   |    |                              | ↑ | ?   | UNL                        | ↑   | O  | ↑                             | _ | ↑ | DEL | ↑ |
|                             |    |    |    |     | Address command group (ACG) |           |     | Universal command group (UCG) |     |    | Listener address group (LAG) |   |     | Talker address group (TAG) |     |    | Secondary command group (SCG) |   |   |     |   |
| Primary command group (PCG) |    |    |    |     |                             |           |     |                               |     |    |                              |   |     |                            |     |    |                               |   |   |     |   |

Figure 7.1 Multi-Line Interface Message

- Note: \*1 MSG means an interface message  
 \*2 b1=DIO1...b7=DIO7. DIO8 is not used  
 \*3 Accompanied by the secondary command  
 \*4 "\" for IEC standard, "¥" for JIS
- |                               |                                 |
|-------------------------------|---------------------------------|
| GTL : Go To Local             | TCT : Take Control              |
| SDC : Selected Device Clear   | LLO : Local Lockout             |
| PPC : Parallel Poll Configure | DCL : Device Clear              |
| GET : Group Execute Trigger   | PPU : Parallel Poll Unconfigure |
|                               | SPE : Serial Poll Enable        |
|                               | SPD : Serial Poll Disable       |
|                               | UNL : Unlisten                  |
|                               | UNT : Untalk                    |

WABUN: (DA00102293)

#### NOTES

- Reproduction of the contents of this manual is forbidden by applicable laws.
- The contents of this manual may be revised without notice.
- Information provided in this manual is intended to be accurate and reliable. However, we assume no responsibility for any damage regarding the contents of this manual.
- Should you notice any doubtful or wrong statement, or lack of description, please contact us or our agent.

©Copyright 2023, **NF Corporation**

Programmable AC Power Source DP020AS  
Instruction Manual  
(Remote Control)

#### **NF Corporation**

6-3-20, Tsunashima Higashi, Kohoku-ku, Yokohama  
223-8508 JAPAN  
Phone +81-45-545-8128 Fax +81-45-545-8187

