

---

## **Programmer's Guide**

# **Function/Arbitrary Generators**

## 1. SCPI command language

SCPI (Standard Commands for Programmable Instruments) is a standardized set of commands and be based on ASCII code, through the remote interface to programmer control the instruments.

**1.1 Command format:** The SCPI adopt the following format:

[SOURce[1|2]:]FREQuency {<Value>[MHz|kHz|Hz|mHz]|MINimum|MAXimum}

Brace { } contains the parameter options of command string.

Separating character|: separate several parameter options, only one parameter option could be chosen at one time.

Angular bracket < > indicates this option is a parameter value

Square bracket [ ]: parameters contains in this character is optional and could be omitted.

The sign { }, [ ], < > and | are shown for convenient expression in example, but not sent with command string and not allowed in practical applications.

**1.2 Command abbreviation:** command for instrument is either can be used in abbreviation format which is simple and brief for program writing, or be used in full format which is clear for program meaning and easy to read. The commands given by this guide is full format, among which written by uppercase stands for abbreviation format, abbreviation format is shorter than 4 letters. The other format besides that will result in mistake.

The command will not distinguish uppercase and lowercase, and it allows both uppercase and lowercase, or mixing uppercase and lowercase is also allowed. For example, "FREQ 1KHZ", "freq 1kHz" and "FReq 1kHz" are all acceptable with same running result. But for unit, letter M is different from letter m and be easily mix-used. So we make an improvement to distinguish, for example, 1MHz =1000000Hz and 1mHz=0.001Hz.

**1.3 Command separator:** with hierarchical structure, SCPI command can be divided into Root, Subnode and Endnode commands. Using colon ":" to separate the keywords, and separate command keyword and parameters with space, if the command contains many parameters, you can use comma "," to separate.

For example, Apply:Sin 1 kHz, 5.2 Vpp, -0.2Vdc

Use semicolon “;” to link several commands with same scale and under one subsystem, in this case the higher scale command could be omitted and the program becomes simpler.

For example, two commands of ‘AM:DEPT<sub>H</sub> 95’ and ‘AM:INT<sub>ernal</sub>:FREQuency 3kHz’, could be linked with semicolon “;” and are simplified as one command as:

AM:DEPT<sub>H</sub> 95; INT<sub>ernal</sub>:FREQuency 3kHz

Use a semicolon and a colon “;:” to link several commands under different subsystems, every command should start with root command.

For example: AM:STAT<sub>e</sub> ON;; FREQuency 100kHz;; AM:DEPT<sub>H</sub>?

**1.4 Parameter type:** parameter type has 4 formats as following.

**1.4.1 Numeric value parameter:** numeric value parameter is presented by decimal number, composed by digits, minus and decimal point, such as -8.253. Floating-point number could be used to indicate either, such as 1.5E6. You also can use two special values Minimum and Maximum to instead parameter value of command. Min set the parameter to be the allowed minimum value, and Max set the parameter to be the allowed maximum value. You can add the unit in the end of parameter value, such as kHz, V<sub>rms</sub> and so on. If not adding, you can use the default unit Hz, V, s, deg and so on.

If not adding unit after amplitude value, use command ‘VOLT<sub>age</sub>:UNIT {V<sub>pp</sub>|V<sub>rms</sub>}’ to set unit.

The unit omitting parameter can make the program simple, but sometimes adding unit can be more convenient, for example, Freq 10MHz is more simple than 10000000.

Command of value parameter, take Frequency 1000 or Amplitude Max for example.

**1.4.2 Discrete parameter:** discrete parameter only has a few of value, and be same as commands, you can use full or abbreviation format, or mixing uppercase and lowercase is also allowed.

The command of discrete parameter such as FSKey:Source Internal.

**1.4.3 Boolean parameter:** a Boolean parameter specifies a single binary condition which is either true or false. For “True”, the parameter value is “ON” or “1”, and for “False”, the parameter value is “OFF” or “0”.

The command of Boolean parameter such as FM: State On.

**1.4.4 Character string parameter:** this parameter is constituted by ASCII characters, and enclosed by a pair of quotation marks. Such as ‘No error’.

**1.5 Parameter query:** users can add interrogation “?” in the end of commands, so you can query the current value of most parameters. For example, the present function is sine wave, send query command Function?, and return “SIN”.

For value parameter, basic unit is default when query or return a value without a unit with it, the format of the value is floating-point.

For discrete parameter, both the query and return are abbreviation format with the uppercase.

For Boolean parameter, query or return “1” or “0”.

For character string parameter, query or return a character string enclosed in a pair of quotation marks.

**1.6 Universal command:** universal commands start with \*, with length of three characters, and could have parameters.

Example of universal command: \*RST

**1.7 Command end character:** total characters in a command string should not be more than 60. Each end of character string should be added an end character (shift character of ASCII code 10), indicates an end of character string in order to avoid a mistake. It is suggested that the end character is written in the sending function when programmable so that it is not necessary for adding it at the end of each command and it will never be lost carelessly.

## 2. Command set

The instrument set the SCPI command of most functions, but since the arbitrary edition and parameter calibration function and the operations of the two are complex and remote, so the programmer commands are not set.

The command keyword in the command set is written in full format, the uppercase is the abbreviation of this command, the unit of value is not distinguished full or abbreviation format.

If omit SOURCE, or use SOURCE, SOURCE1, OUTPUT or OUTPUT1 in command set, CHA is available.

If using SOURCE2 or OUTPUT2, CHB is available.

### 2.1 Continuous waveform command

[SOURCE[1|2]:]APPLY:<function>[<frequency>[,<amplitude>[,<offset>]]]

[SOURCE[1|2]:]APPLY?

Users may use Apply command to configure directly the four parameters of the function generator: function, frequency, amplitude and offset, the sequence of which could not be changed. Where frequency, amplitude and offset three parameters could be omitted from the last to the first, the omitted parameters keeps as the current value.

For example: APPL:SIN 10kHz,1.2,0.5

Configuration continuously outputs sine, frequency 10kHz, amplitude 1.2Vpp and offset 0.5Vdc,

APPL?

Query or return present function, frequency, amplitude or offset value.

'SIN,1.000000E+04,1.200000E+00,5.000000E-01'.

```
[SOURce[1|2]:]FUNCTION {SINusoid|SQUare|RAMP|PULSe|NOISe|PRBS
                        |REXP|FEXP|RLOG|TANGent|SINC|CIRCle|GAUSSian
                        |CARDiac|QUAKe|USER1|USER2|USER3|USER4|USER5}
```

[SOURce[1|2]:]FUNCTION?

**Examples: FUNC SQU**

**FUNC?**

```
[SOURce[1|2]:]FUNCTION:SQUare:DCYCl {<Value in percent>|MINimum|MAXimum}
```

```
[SOURce[1|2]:]FUNCTION:SQUare:DCYCl? [{MINimum|MAXimum}]
```

**Examples: FUNC:SQU:DCYC 25**

**FUNC:SQU:DCYC?**

```
[SOURce[1|2]:]FUNCTION:RAMP:SYMMetry {<Value in percent>|MINimum|MAXimum}
```

```
[SOURce[1|2]:]FUNCTION:RAMP:SYMMetry? [{MINimum|MAXimum}]
```

**Examples: FUNC:RAMP:SYMM 100**

**FUNC:RAMP:SYMM?**

```
[SOURce[1|2]:]FUNCTION:PULSe:PERiod {<Value[s|ms]>|MINimum|MAXimum}
```

```
[SOURce[1|2]:]FUNCTION:PULSe:PERiod? [{MINimum|MAXimum}]
```

**Examples: FUNC:PULS:PER 1us**

**FUNC:PULS:PER?**

---

[SOURce[1|2]:]FUNCtion:PULSe:WIDTh {<Value>[s|ms]>|MINimum|MAXimum}

[SOURce[1|2]:]FUNCtion:PULSe:WIDTh? [{MINimum|MAXimum}]

**Examples: FUNC:PULS:WIDT 200ns**

**FUNC:PULS:WIDT?**

[SOURce[1|2]:]FREQuency {<Value>[MHz|kHz|Hz|mHz|uHz]|MINimum|MAXimum}

[SOURce[1|2]:]FREQuency? [{MINimum|MAXimum}]

**Examples: FREQ 2E6**

**FREQ?**

[SOURce[1|2]:]PERiod {<Value>[s|ms|us|ns]|MINimum|MAXimum}

[SOURce[1|2]:]PERiod? [{MINimum|MAXimum}]

**Examples: PER 100ns**

**PER?**

[SOURce[1|2]:]VOLTage {<Value>[Vrms|mVrms|Vpp|mVpp|dBm]|MINimum|MAXimum}

[SOURce[1|2]:]VOLTage? [{MINimum|MAXimum}]

**Examples: VOLT 1.2**

**VOLT?**

[SOURce[1|2]:]VOLTage:HIGH {<Value>[V|mV]|MINimum|MAXimum}

[SOURce[1|2]:]VOLTage:HIGH? [{MINimum|MAXimum}]

**Examples: VOLT:HIGH 2V**

**VOLT:HIGH?**

[SOURce[1|2]:]VOLTage:LOW {<Value>[V|mV]|MINimum|MAXimum}

[SOURce[1|2]:]VOLTage:LOW? [{MINimum|MAXimum}]

**Examples: VOLT:LOW -3**

**VOLT:LOW?**

[SOURce[1|2]:]VOLTage:LIMit:HIGH {<Value>[V|mV]|MINimum|MAXimum}

[SOURce[1|2]:]VOLTage:LIMit:HIGH? [{MINimum|MAXimum}]

**Examples: VOLT:LIM:HIGH 5**

**VOLT:LIM:HIGH?**

[SOURce[1|2]:]VOLTage:LIMit:LOW {<Value>[V|mV]|MINimum|MAXimum}

[SOURce[1|2]:]VOLTage:LIMit:LOW? [{MINimum|MAXimum}]

**Examples: VOLT:LIM:LOW 0V**

**VOLT:LIM:LOW?**

[SOURce[1|2]:]VOLTage:OFFSet {<Value>[V|mV]|MINimum|MAXimum}

[SOURce[1|2]:]VOLTage:OFFSet? [{MINimum|MAXimum}]

**Examples: VOLT:OFFS 100mv**

**VOLT:OFFS?**

[SOURce[1|2]:]VOLTage:RANGe:AUTO {ON|1|OFF|0}

[SOURce[1|2]:]VOLTage:RANGe:AUTO?

**Examples: VOLT:RANG:AUTO ON**

**VOLT:RANG:AUTO?**

[SOURce[1|2]:]VOLTage:UNIT {Vpp|Vrms|dBm}

[SOURce[1|2]:]VOLTage:UNIT?

**Examples: VOLT:UNIT Vrms**

**VOLT:UNIT?**

[SOURce[1|2]:]PHASe {<Value>[deg]|MINimum|MAXimum}

[SOURce[1|2]:]PHASe? [{MINimum|MAXimum}]

**Examples: PHAS 90deg**

**PHAS?**

[SOURce[1|2]:]PHASe:SYNChronize

Examples: PHAS:SYNC

**2.2 Output configuration command**

OUTPut[1|2]:POLarity {NORMal|INVerted}

OUTPut[1|2]:POLarity?

**Examples: OUTP:POL INV****OUTP:POL?**

OUTPut[1|2]:LOAD {&lt;Value&gt;[Ohm]|INFinity|MINimum|MAXimum}

OUTPut{1|2}:LOAD? [{MINimum|MAXimum}]

**Examples: OUTP1:LOAD 600****OUTP1:LOAD?**

OUTPut[1|2][:STATe] {ON|1|OFF|0}

OUTPut[1|2][:STATe]?

**Examples: OUTP2 1****OUTP2:STAT?**

OUTPut:SYNC[:STATe] {ON|1|OFF|0}

OUTPut:SYNC[:STATe]?

**Examples: OUTP:SYNC:STAT OFF****OUTP:SYNC?**

### 2.3 Frequency modulation (FM) command

[SOURce[1]:]FM[:DEViation] {&lt;Value&gt;[MHz|kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]FM[:DEViation]? [{MINimum|MAXimum}]

**Examples: FM:DEV 150Hz****FM?**

[SOURce[1]:]FM:INTernal:FREQuency{&lt;Value&gt;[kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]FM:INTernal:FREQuency? [{MINimum|MAXimum}]

**Examples: FM:INT:FREQ 100****FM:INT:FREQ?**

[SOURce[1]:]FM:INTernal:FUNCTion {SINusoid|SQUare|RAMP|PULSe|NOISe

|PRBS|REXP|FEXP|RLOG|TANGent|SINC

|CIRCle|GAUSSian|CARDiac|QUAKE

|USER1|USER2|USER3|USER4|USER5}



---

[SOURce[1]:]FM:INTernal:FUNCtion?

**Examples: FM:INT:FUNC RAMP**  
**FM:INT:FUNC?**

[SOURce[1]:]FM:SOURce {INTernal|EXTernal}

[SOURce[1]:]FM:SOURce?

**Examples: FM:SOUR INT**  
**FM:SOUR?**

[SOURce[1]:]FM:STATe {ON|1|OFF|0}

[SOURce[1]:]FM:STATe?

**Examples: FM:STAT ON**  
**FM:STAT?**

## 2.4 Amplitude modulation (AM) command

[SOURce[1]:]AM[:DEPT] {<Value in percent>|MINimum|MAXimum}

[SOURce[1]:]AM[:DEPT]? [{MINimum|MAXimum}]

**Examples: AM 100**  
**AM:DEPT?**

[SOURce[1]:]AM:INTernal:FREQuency {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]AM:INTernal:FREQuency? [{MINimum|MAXimum}]

**Examples: AM:INT:FREQ 10**  
**AM:INT:FREQ?**

[SOURce[1]:]AM:INTernal:FUNCtion {SINusoid|SQUare|RAMP|PULSe|NOISe  
 |PRBS|REXP|FEXP|RLOG|TANGent|SINC  
 |CIRCle|GAUSSian|CARDiac|QUAKe  
 |USER1|USER2|USER3|USER4|USER5}

[SOURce[1]:]AM:INTernal:FUNCtion?

**Examples: AM:INT:FUNC SIN**  
**AM:INT:FUNC?**

---

[SOURce[1]:]FM:SOURce {INTernal|EXTernal}

[SOURce[1]:]FM:SOURce?

**Examples: AM:SOUR INT**

**AM:SOUR?**

[SOURce[1]:]AM:STATe {ON|1|OFF|0}

[SOURce[1]:]AM:STATe?

**Examples: aM:STAT ON**

**AM:STAT?**

## 2.5 Phase modulation (PM) command

[SOURce[1]:]PM[:DEViation] {<Value>[deg]|MINimum|MAXimum}

[SOURce[1]:]PM:DEViation? [{MINimum|MAXimum}]

**Examples: PM:DEV 90deg**

**PM:DEV?**

[SOURce[1]:]PM:INTernal:FREQuency {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]PM:INTernal:FREQuency? [{MINimum|MAXimum}]

**Examples: PM:INT:FREQ 100**

**PM:INT:FREQ?**

[SOURce[1]:]PM:INTernal:FUNCTion {SINusoid|SQUare|RAMP|PULSe|NOISe  
|PRBS|REXP|FEXP|RLOG|TANGent|SINC  
|CIRCle|GAUSSian|CARDiac|QUAKE  
|USER1|USER2|USER3|USER4|USER5}

[SOURce[1]:]PM:INTernal:FUNCTion?

**Examples: PM:INT:FUNC SQU**

**PM:INT:FUNC?**

[SOURce[1]:]PM:SOURce {INTernal|EXTernal}

[SOURce[1]:]PM:SOURce?

**Examples: PM:SOUR INT**

**PM:SOUR?**

---

[SOURce[1]:]PM:STATe {ON|1|OFF|0}

[SOURce[1]:]PM:STATe?

**Examples: PM:STAT ON**

**PM:STAT?**

## 2.6 Pulse width modulation (PWM) command

[SOURce[1]:]PWM[:DEViation]:DCYCle {<Value in percent>|MINimum|MAXimum}

[SOURce[1]:]PWM:DEViation:DCYCle? [{MINimum|MAXimum}]

**Examples: PWM:DEV:DCYC 50**

**PWM:DEV:DCYC?**

[SOURce[1]:] PWM:INTernal:FREQuency {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:] PWM:INTernal:FREQuency? [{MINimum|MAXimum}]

**Examples: PWM:INT:FREQ 100**

**PWM:INT:FREQ?**

[SOURce[1]:] PWM:INTernal:FUNCTion {SINusoid|SQUare|RAMP|PULSe|NOISe

|PRBS|REXP|FEXP|RLOG|TANGent|SINC

|CIRCle|GAUSSian|CARDiac|QUAKE

|USER1|USER2|USER3|USER4|USER5}

[SOURce[1]:] PWM:INTernal:FUNCTion?

**Examples: PWM:INT:FUNC RAMP**

**PWM:INT:FUNC?**

[SOURce[1]:]PWM:SOURce {INTernal|EXTernal}

[SOURce[1]:]PWM:SOURce?

**Examples: PWM:SOUR INT**

**PWM:SOUR?**

[SOURce[1]:]PWM:STATe {ON|1|OFF|0}

[SOURce[1]:]PWM:STATe?

**Examples: PWM:STAT ON**

**PWM:STAT?**

## 2.7 Sum modulation command

---

[SOURce[1]:]SUM[:AMPLitude] {<Value in percent>|MINimum|MAXimum}

[SOURce[1]:]SUM:AMPLitude? [{MINimum|MAXimum}]

**Examples: SUM:AMPL 50**

**SUM:AMPL?**

[SOURce[1]:] SUM:INTernal:FREQuency {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:] SUM:INTernal:FREQuency? [{MINimum|MAXimum}]

**Examples: SUM:INT:FREQ 100**

**SUM:INT:FREQ?**

[SOURce[1]:] SUM:INTernal:FUNCTion {SINusoid|SQUare|RAMP|PULSe|NOISe  
|PRBS|REXP|FEXP|RLOG|TANGent|SINC  
|CIRCle|GAUSSian|CARDiac|QUAKE  
|USER1|USER2|USER3|USER4|USER5}

[SOURce[1]:] SUM:INTernal:FUNCTion?

**Examples: SUM:INT:FUNC RAMP**

**SUM:INT:FUNC?**

[SOURce[1]:]SUM:SOURce {INTernal|EXTernal}

[SOURce[1]:]SUM:SOURce?

**Examples: SUM:SOUR INT**

**SUM:SOUR?**

[SOURce[1]:]SUM:STATe {ON|1|OFF|0}

[SOURce[1]:]SUM:STATe?

**Examples: SUM:STAT 0**

**SUM:STAT?**

## 2.8 Frequency shift keying (FSK) command

[SOURce[1]:]FSKey[:FREQuency] {<Value>[MHz|kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]FSKey:FREQuency? [{MINimum|MAXimum}]

**Examples: FSK:FREQ 3kHz**

**FSK:FREQ?**

---

[SOURce[1]:]FSKey:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]FSKey:INTernal:RATE? [{MINimum|MAXimum}]

**Examples: FSK:INT:RATE 100**

**FSK:INT:RATE?**

[SOURce[1]:]FSKey:SOURce {INTernal|EXTernal}

[SOURce[1]:]FSKey:SOURce?

**Examples: FSK:SOUR EXT**

**FSK:SOUR?**

[SOURce[1]:]FSKey:STATe {ON|1|OFF|0}

[SOURce[1]:]FSKey:STATe?

**Examples: FSK:STAT OFF**

**FSK:STAT?**

## 2.9 Phase shift keying command

[SOURce[1]:]BPSK[:PHASe] {<Value>[deg]|MINimum|MAXimum}

[SOURce[1]:] BPSK[:PHASe]? [{MINimum|MAXimum}]

**Examples: BPSK 180**

**FSK:PHAS?**

[SOURce[1]:]BPSK:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]BPSK:INTernal:RATE? [{MINimum|MAXimum}]

**Examples: BPSK:INT:RATE 100**

**BPSK:INT:RATE?**

[SOURce[1]:]BPSK:SOURce {INTernal|EXTernal}

[SOURce[1]:]BPSK:SOURce?

**Examples: BPSK:SOUR INT**

**BPSK:SOUR?**

[SOURce[1]:]BPSK:STATe {ON|1|OFF|0}

[SOURce[1]:]BPSK:STATe?

**Examples: BPSK:STAT OFF**

**BPSK:STAT?****2.10 Frequency sweeping command**

[SOURce[1]:]FREQuency:STARt {<Value>[MHz|kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]FREQuency:STARt? [{MINimum|MAXimum}]

**Examples: FREQ:STAR 120**

**FREQ:STAR?**

[SOURce[1]:]FREQuency:STOP {<Value>[MHz|kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]FREQuency:STOP? [{MINimum|MAXimum}]

**Examples: FREQ:STOP 25kHz**

**FREQ:STOP?**

[SOURce[1]:]MARKer:FREQuency {<Value>[MHz|kHz|Hz|mHz]|MINimum|MAXimum}

[SOURce[1]:]MARKer:FREQuency? [{MINimum|MAXimum}]

**Examples: MARK:FREQ 5kHz**

**MARK:FREQ?**

[SOURce[1]:]SWEep:SPACing {LINear|LOGarithmic}

[SOURce[1]:]SWEep:SPACing?

**Examples: SWE:SPAC LOG**

**SWE:SPAC?**

[SOURce[1]:]SWEep:TIME {<Value>[s|ms]|MINimum|MAXimum}

[SOURce[1]:]SWEep:TIME? [{MINimum|MAXimum}]

**Examples: SWE:TIME 5s**

**SWE:TIME?**

[SOURce[1]:]SWEep:HTIME {<Value>[s|ms]|MINimum|MAXimum}

[SOURce[1]:]SWEep:HTIME? [{MINimum|MAXimum}]

**Examples: SWE:HTIM 2s**

**SWE:HTIM?**

[SOURce[1]:]SWEep:RTIME {<Value>[s|ms]|MINimum|MAXimum}

---

[SOURce[1]:]SWEep:RTIME? [{MINimum|MAXimum}]

**Examples: SWE:RTIM 0**  
**SWE:RTIM?**

[SOURce[1]:]SWEep[:STATe] {ON|1|OFF|0}

[SOURce[1]:]SWEep[:STATe]?

**Examples: SWE:STAT ON**  
**SWE:STAT?**

TRIGger[1]:SOURce {IMMediate|EXTernal}

TRIGger[1]:SOURce?

**Examples: TRIG:SOUR IMM**  
**TRIG:SOUR?**

\*TRG

Example: \*TRG

## 2.11 List sweep command

[SOURce[1]:]LIST:FREQuency:POINts {<Value>|MINimum|MAXimum}

[SOURce[1]:]LIST:FREQuency:POINts? [{MINimum|MAXimum}]

**Example: LIST:FREQ:POIN 100**  
**LIST:FREQ:POIN?**

[SOURce[1]:]LIST:DWELl {<Value[s|ms]|MINimum|MAXimum}

[SOURce[1]:]LIST:DWELl? [{MINimum|MAXimum}]

**Example: LIST:DWEL 1s**  
**LIST:DWEL?**

[SOURce[1]:]LIST[:STATe] {ON|OFF}

[SOURce[1]:]LIST[:STATe]?

**Example: LIST ON**  
**LIST:STAT?**

TRIGger[1]:SOURce {IMMediate|EXTernal}

---

TRIGger[1]:SOURce?

**Examples: TRIG:SOUR IMM**  
**TRIG:SOUR?**

\*TRG

**Example: \*TRG**

## 2.12 Burst command

[SOURce[1]:]BURSt:NCYCles {<Value>|MINimum|MAXimum}

[SOURce[1]:]BURSt:NCYCles? [{MINimum|MAXimum}]

**Examples: BURS:NCYC 2**  
**BURS:NCYC?**

[SOURce[1]:]BURSt:INTernal:PERiod {<Value>[s|ms]|MINimum|MAXimum}

[SOURce[1]:]BURSt:INTernal:PERiod? [{MINimum|MAXimum}]

**Examples: BURS:INT:PER 20ms**  
**BURS:INT:PER?**

[SOURce[1]:]BURSt:PHASe {<Value>[deg]|MINimum|MAXimum}

[SOURce[1]:]BURSt:PHASe? [{MINimum|MAXimum}]

**Examples: BURS:PHAS 180**  
**BURS:PHAS?**

BURSt:MODE {TRIGgered|GATed}

BURSt:MODE?

**Examples: BURS:MODE GAT**  
**BURS:MODE?**

[SOURce[1]:]BURSt[:STATe] {ON|1|OFF|0}

[SOURce[1]:]BURSt[:STATe]?

**Examples: BURS:STAT ON**  
**BURS?**

TRIGger[1]:SOURce {IMMediate|EXTernal}



TRIGger[1]:SOURce?

**Examples: TRIG:SOUR EXT**  
**TRIG:SOUR?**

\*TRG

**Example: \*TRG**

### 2.13 Dual channel command

[SOURce[1]:]FREQuency:COUPle:RATio {<Value>|MINimum|MAXimum}

[SOURce[1]:]FREQuency:COUPle:RATio? [{MINimum|MAXimum}]

**Examples: FREQ:COUP:RAT 2**  
**FREQ:COUP:RAT?**

[SOURce[1]:]FREQuency:COUPle:OFFSet {<Value>[MHz|kHz|Hz|mHz]  
|MINimum|MAXimum}

[SOURce[1]:]FREQuency:COUPle:OFFSet? [{MINimum|MAXimum}]

**Examples: FREQ:COUP:OFFS 2kHz**  
**FREQ:COUP:OFFS?**

[SOURce[1]:]FREQuency:COUPle[:STATe] {ON|1|OFF|0}

[SOURce[1]:]FREQuency:COUPle[:STATe]?

**Examples: FREQ:COUP ON**  
**FREQ:COUP?**

[SOURce[1]:]VOLTage:COUPle:OFFSet {<Value Vpp|mVpp>}

[SOURce[1]:]VOLTage:COUPle:OFFSet?

**Examples: VOLT:COUP:OFFS 1Vpp**  
**VOLT:COUP:OFFS?**

[SOURce[1]:]VOLTage:COUPle[:STATe] {ON|1|OFF|0}

[SOURce[1]:]VOLTage:COUPle[:STATe]?

**Examples: VOLT:COUP:STAT ON**  
**VOLT:COUP?**

[SOURce[1]:]COMBine:AMPLitude {<Value in percent>|MINimum|MAXimum}

[SOURce[1]:]COMBine:AMPLitude? [{MINimum|MAXimum}]

**Examples: COMB:AMPL 100**

**COMB:AMPL?**

[SOURce[1]:]COMBine:FEED {CH2|NONE}

[SOURce[1]:]COMBine:FEED?

**Examples: COMB:FEED CH2**

**COMB:FEED?**

## 2.14 Save & Recall command

\*SAV {1|2|3|4}

**Example: \*SAV 1**

\*RCL {0|1|2|3|4}

**Example: \*RCL 0**

## 2.15 System command

\*RST /\* System reset

**Example: \*RST**

\*CLS /\* Clear error queue

**Example: \*CLS**

SYSTem:ERRor? /\*Query error queue

**Example: SYST:ERR?**

SYSTem:LOCal /\*Return to local

**Example: SYST:LOC**

DISPlay {ON|1|OFF|0} /\*Screen display

DISPlay?

**Example: DISP ON**

**DISP?**

### 3. Error information

**3.1 Error queue:** Instrument has an error memory, with an error queue arraying in a first-in-first-out (FIFO) way. Each time an error occurred, the queue will store it and the generator will make an alarm at the same time. The queue stores 20 errors at most, and when exceed, the last error information will be defined as "Queue overflow", indicating queue overflow and will not store new error.

**3.2 Error reading:** use error query command SYSTem:ERRor? to read an error, the first error read is also the first one to be stored. When an error is read, it will be cleared at the same time. If there is no stored error in this queue, or all the stored errors have been read out, "No error" will be returned.

**3.3 Error clearance:** use \*CLS command or cut off the power of the generator to clear error queue, which will be not cleared when using resetting command \*RST.

**3.4 Error message:** the form of error message is "-100, Queue overflow". There are two error messages: "-1xx" indicates command error, "-2xx" indicates run error, such as:

"-100, Queue overflow":error queue overflow, the quantity of stored error exceeds 20.

"-101, First level command error": First scale (root) command error, may be writing error, such as, Swep:time 3s

"-102, Second level command error":Second scale command error, may be writing error, or this command is not matched with the first scale command, such as, FM:depth 20

"-103, Third level command error": Third scale command error, may be writing error, or this command is not matched with the second scale command, such as, Fskey:Internal:Frequency 3kHz.

"-104, Invalid parameter": invalid parameter, may be parameter writing error, or the parameter type is not accordance with current option, such as, FM:State OM

"-105, Invalid suffix(unit)":invalid postfix (data unit), may be writing error of unit, or this unit is not in accordance with the current data, such as, Burst:Ncycles 3 cyc

"-106, Syntax error":syntax error, of which the possibility is complex, such as using comma instead of space: Frequency, 6kHz

"-107, Missing parameter": miss parameter, the command should be with the commands but without, such as: Voltage:Offset

“-202, Current waveform not able to use Vrms”: Current waveform could not be described by Vrms, such as: Voltage:unit Vrms

“-203, Trigger only use in sweep or burst”: single trigger command is only applicable to frequency sweeping or burst functions. Such as: Trigger:source external

‘-204, Data out of range,value clipped to limit’: Data exceed range and limited the value to limit.