

# **Function/Arbitrary Generators**

**P 4046**

05/2023

---

## **1. SCPI command language**

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

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

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

Brace { } contains the parameter value and its unit.

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 signs { }, [ ], < > 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, "Mhz" represents MilliHertz (not distinguish uppercase and lowercase), "MAHz" represents Megahertz (not distinguish uppercase and lowercase). For example, 1MAHz = 1000000Hz, 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:DEPTH 95' and 'AM:INTERNAL:FREQUENCY 3kHz', could be linked with semicolon ";" and are simplified as one command as:

AM:DEPTH 95; INTERNAL: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:STATE ON;; FREQUENCY 100kHz;; AM:DEPTH?

**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, Vrms 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 'VOLTage:UNIT {Vpp|Vrms}' 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 Z exponential scheme.

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 IEEE-488 command:** universal commands based on IEEE-488 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 255. 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, Channel A is available.

If using SOURce2 or OUTPut2, Channel B is available.

## 2.1 Continuous waveform command

### 2.1.1 APPLY command

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. The <function> can be any built-in waveform: ARB# as well as standard waveforms. Wherein, the pound sig # represents the number of built-in waveform. The APPLY command can be used to set and enquiry.

```
[SOURce[1|2]:]APPLy:<function>[<frequency>[,<amplitude>[,<offset>]]]
```

```
[SOURce[1|2]:]APPLy?
```

Examples: APPL:SIN 10kHz,1.2,0.5

```
APPLY:ARB0 10kHz,1.2,0.5
```

Both the above two commands can output 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'
```

### 2.1.2 FUNCTION command

The Function command can be used to set the function of instrument. Both the waveform name (just apply to standard waveforms) and waveform number can be directly apply to the Function command.

Function setting command:

```
[SOURce[1|2]:]FUNctIon {SINusoid|SQUare|RAMP|PULSe|NOISe|<arb number>}
```

Wherein, <arb number> means the arbitrary waveform number. When value of the arb number equal to any one among 0,1, 2, 3, 4, the above function is the same as following:

```
[SOURce[1|2]:]FUNctIon {SINusoid|SQUare|RAMP|PULSe|NOISe}
```

#### Examples: FUNC SQU

##### FUNC 1

Waveform setting command:

This command can facilitate the work of user-defined waveform greatly. The waveform name (User\_arb\*,\*=1,2,3...7), together with waveform number (1, 2, 3...7) can be used in the command parameter.

```
[SOURce[1|2]:]FUNctIon :USER{User_arb1| User_arb2| User_arb3| User_arb4|
User_arb5| User_arb6| User_arb7|<user number> }
```

#### Examples: FUNC:USER User\_arb1

Query command:

This command can return the function the user required to know. For example, waveform Inquiry command for command can confirm the waveform the user is using.

[SOURce[1|2]:]FUNctIon?

**Examples:FUNC?**

[SOURce[1|2]:]FUNctIon:USER?

**Examples:FUNC:USER?**

Query and return the user-defined waveform. If such one is not used, the instrument would show no user-defined is used.

### 2.1.3 Duty Cycle command for square wave

Set command:

[SOURce[1|2]:]FUNctIon:SQUare:DCYClE {<Value in percent>|MINimum|MAXimum}

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

Inquiry command:

[SOURce[1|2]:]FUNctIon:SQUare:DCYClE? [ {MINimum|MAXimum} ]

**Examples: FUNC:SQU:DCYC?**

### 2.1.4 Symmetry command for sawtooth wave

Set command:

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

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

Inquiry command:

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

**Examples: FUNC:RAMP:SYMM?**

### 2.1.5 Pulse period command for Pulse wave:

Set and Inquiry command:

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

**Examples: FUNC:PULS:PER 1ms**

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

**Examples:FUNC:PULS:PER?**

### 2.1.6 Pulse width command for pulse wave

Set command:

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

**Examples: FUNC:PULS:WIDT 200n**

Inquiry command:

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

**Examples: FUNC:PULS:WIDT?**

### 2.1.7 Edge time command for pulse wave

Set command:

SOURce[1|2]:FUNCtion:PULSe:TRANsition {<seconds>|MINimum |MAXimum }

**Examples: FUNC:PULS:TRAN 100ns**

Inquiry command:

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

**Examples: FUNC:PULS:TRAN?**

### 2.1.8 Waveform frequency command

Set command:

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

**Examples: FREQ 1KHz**

Inquiry command

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

**Examples: FREQ?**

### 2.1.9 Waveform amplitude command

Set command:

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

Wherein, the unit dBm can only be used in non- impedance state.

**Examples: VOLT 1.2**

Inquiry command

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

**Examples: VOLT?**

### 2.1.10 Waveform level command

#### 2.1.10.1 Waveform high level command

Set command:

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

**Examples: VOLT:HIGH 2V**

Inquiry command:

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

**Examples: VOLT:HIGH?**

#### 2.1.10.2 Waveform low level command

Set command:

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

**Examples: VOLT:LOW -3**

Inquiry command:

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

**Examples: VOLT:LOW?**

## 2.1.11 Limit command of waveform level

### 2.1.11.1 Limit command of waveform high level

Set command:

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

**Examples: VOLT:LIM:HIGH 8**

Inquiry command

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

**Examples: VOLT:LIM:HIGH?**

### 2.1.11.2 Limit command of waveform low level

Set command:

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

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

Inquiry command

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

**Examples: VOLT:LIM:LOW?**

## 2.1.12 Waveform offset command

Set command:

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

**Examples: VOLT:OFFS 100mv**

Inquiry command:

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

**Examples: VOLT:OFFS?**

## 2.1.13 Amplitude unit command:

Set command:

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

**Examples: VOLT:UNIT Vrms**

Inquiry command:

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

**Examples: VOLT:UNIT?**

## 2.1.14 Waveform phase command:

Set command:

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

**Examples: PHAS 90deg**

Inquiry command

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

**Examples: PHAS?**

### 2.1.15 phase synchronization command

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

Examples: PHAS:SYNC

## 2.2 Output configuration command

### 2.2.1 Waveform polarity command

Set command

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

**Examples: OUTP:POL INV**

Inquiry command

OUTPut[1|2]:POLarity?

**Examples: OUTP:POL?**

### 2.2.2 Output load command

Set command

OUTPut[1|2]:LOAD {<Value>[Ohm]|INFinity|MINimum|MAXimum}

**Examples: OUTP1:LOAD 50**

Inquiry command:

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

**Examples: OUTP1:LOAD?**

### 2.2.3 State command of output port

Set command

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

**Examples: OUTP 1**

Inquiry command

OUTPut[1|2]?

**Examples: OUTP?**

### 2.2.4 Synchronization output port command

Set command

OUTPut:SYNC {ON|1|OFF|0}

**Examples: OUTP:SYNC: OFF**

Inquiry command

OUTPut: SYNC?

**Examples: OUTP:SYNC?****2.2.5 Trigger output command****2.2.5.1 Trigger output state command**

Set command

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

**Examples: OUTP:TRIG 1**

Inquiry command:

OUTPut[1|2]: TRIGger?

**Examples: OUTP:TRIG?****2.2.5.2 Trigger output edge command**

Set command

OUTPut[1|2]:TRIGger: SLOPe {POSitive|NEGative}

**Examples:OUTP:TRIG:SLOPe Neg**

Inquiry command

**Examples:OUTP:TRIG:SLOPe?****2.3 Modulation command****2.3.1 Frequency modulation (FM) command****2.3.1.1 FM state command**

Set command

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

**Examples: FM:STAT ON**

Inquiry command:

[SOURce[1|2]:]FM:STATe?

**Examples: FM:STAT?****2.3.1.2 FM modulation source command**

Set command

[SOURce[1|2]:]FM:SOURce {INTernal|EXTernal}

**Examples: FM:SOUR INT**

Inquiry command

[SOURce[1|2]:]FM:SOURce?

**Examples: FM:SOUR?****2.3.1.3 FM frequency deviation command**

Set command:

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

---

**Examples: FM:DEV 150Hz**

```
[SOURce[1|2]:]FM[:DEVIation]? [{MINimum|MAXimum}]
```

**Examples: FM:DEV?****2.3.1.4 FM modulation frequency command**

Set command:

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

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

Inquiry command:

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

**Examples: FM:INT:FREQ?****2.3.1.5 FM modulation waveform command**

Set command:

```
[SOURce[1|2]:] FM: INTernal: FUNCtion {SINusoid|SQUare|RAMP|PULSe|NOISe
                                     <arb numbers>}
```

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

```
FM: INT: FUNC 20
```

Inquiry command:

```
[SOURce[1|2]:]FM: INTernal:FUNCtion?
```

**Examples: FM:INT:FUNC?****2.3.2 Amplitude modulation (AM) command****2.3.2.1 AM state command**

Set command:

```
[SOURce[1|2]:]AM:STATe {ON|1|OFF|0}
```

**Examples: AM:STAT ON**

Inquiry command:

```
[SOURce[1|2]:]AM:STATe?
```

**Examples: AM:STAT?****2.3.2.2 AM modulation source command**

Set command:

```
[SOURce[1|2]:]AM:SOURce {INTernal|EXTernal}
```

**Examples:AM:SOUR INT**

Inquiry command:

```
[SOURce[1|2]:]AM:SOURce?
```

### 2.3.2.3 AM modulation depth command

Set command:

```
[SOURce[1|2]:]AM[:DEPT]h {<Value in percent>|MINimum|MAXimum}
```

**Examples: AM 100**

Inquiry command:

```
[SOURce[1|2]:]AM[:DEPT]h? [{MINimum|MAXimum}]
```

**Examples: AM?**

### 2.3.2.4 AM modulation frequency command

Set command:

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

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

Inquiry command:

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

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

### 2.3.2.5 FM modulation waveform command

Set command:

```
[SOURce[1|2]:]AM:INTernal:FUNcTion {SINusoid|SQUare|RAMP|PULSe|NOISe  

                                     <arb numbers>}
```

**Examples: AM:INT:FUNC RAMP**

**AM:INT:FUNC 20**

Inquiry command:

```
[SOURce[1|2]:]AM:INTernal:FUNcTion?
```

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

## 2.3.3 Phase modulation (PM) command

### 2.3.3.1 PM state command

Set command:

```
[SOURce[1|2]:]PM:STATe {ON|1|OFF|0}
```

**Examples: PM:STAT ON**

Inquiry command:

```
[SOURce[1|2]:]PM:STATe?
```

**Examples: PM:STAT?**

### 2.3.3.2 PM modulation source command

Set command:

```
[SOURce[1|2]:]PM:SOURce {INTernal|EXTernal}
```

**Examples:PM:SOUR INT**

Inquiry command:

[SOURce[1|2]:]PM:SOURce?

**2.3.3.3 PM phase deviation command**

Set command:

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

**Examples:PM:DEV 90deg**

Inquiry command:

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

**Examples: PM:DEV?****2.3.3.4 PM modulation frequency command**

Set command:

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

**Examples: PM:INT:FREQ 10**

Inquiry command:

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

**Examples: PM:INT:FREQ?****2.3.3.5 PM modulation waveform command**

Set command:

[SOURce[1|2]:]PM:INTernal:FUNCTion {SINusoid|SQUare|RAMP|PULSe|NOISe  
<arb numbers>}

**Examples: PM:INT:FUNC RAMP**

**PM:INT:FUNC 20**

Inquiry command:

[SOURce[1|2]:]PM:INTernal:FUNCTion?

**Examples: PM:INT:FUNC?****2.3.4 PWM command****2.3.4.1 PWM state command**

Set command:

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

**Examples: PWM:STAT ON**

Inquiry command:

[SOURce[1|2]:]PWM:STATe?

**Examples: PWM:STAT?**

### 2.3.4.2 PWM modulation source command

Set command:

```
[SOURce[1|2]:]PWM:SOURce {INTernal|EXTernal}
```

**Examples: PWM:SOUR INT**

Inquiry command:

```
[SOURce[1|2]:]PWM:SOURce?
```

### 2.3.4.3 PWM Pulse width deviation command

Set command:

```
[SOURce[1|2]:]PWM:DEViation {<Value in percent>|MINimum|MAXimum}
```

**Examples: PWM:DEV 60**

Inquiry command:

```
[SOURce[1|2]:]PWM:DEViation? [{MINimum|MAXimum}]
```

**Examples: PWM:DEV?**

### 2.3.4.4 PWM modulation frequency command

Set command:

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

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

Inquiry command:

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

**Examples: PWM:INT:FREQ?**

### 2.3.4.5 PWM modulation waveform command

Set command:

```
[SOURce[1|2]:]PWM:INTernal:FUNcTion {SINusoid|SQUare|RAMP|PULSe|NOISe  

<arb numbers>}
```

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

**PWM:INT:FUNC 20**

Inquiry command:

```
[SOURce[1|2]:] PWM:INTernal:FUNcTion?
```

**Examples: PWM:INT:FUNC?**

## 2.3.5 Superposition modulation (SM) command

### 2.3.5.1 SM state command

Set command:

```
[SOURce[1|2]:]SUM:STATe {ON|1|OFF|0}
```

**Examples: SUM:STAT ON**

Inquiry command:

[SOURce[1|2]:]SUM:STATe?

**Examples: SUM:STAT ?****2.3.5.2 SM modulation source command**

Set command:

[SOURce[1|2]:]SUM:SOURce {INTernal|EXTernal}

**Examples:SUM:SOUR INT**

Inquiry command:

[SOURce[1|2]:]SUM:SOURce?

**2.3.5.3 SM Superposition amplitude command**

Set command:

[SOURce[1|2]:]SUM:AMPLitude {<Value in percent>|MINimum|MAXimum}

**Examples: SUM:AMPL 50**

Inquiry command:

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

**Examples: SUM:AMPL 50****2.3.5.4 SM modulation frequency command**

Set command:

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

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

Inquiry command:

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

**Examples: SUM:INT:FREQ?****2.3.5.5 SM modulation waveform command**

Set command:

[SOURce[1|2]:]SUM:INTernal:FUNcTion {SINusoid|SQUare|RAMP|PULSe|NOISe  
<arb numbers>}

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

**SUM:INT:FUNC 20**

Inquiry command:

[SOURce[1|2]:] SUM:INTernal:FUNcTion?

**Examples: SUM:INT:FUNC?**

### 2.3.6 FSK Modulation command

#### 2.3.6.1 FSK state command

Set command:

```
[SOURCE[1|2]:]FSKey:STATe {ON|1|OFF|0}
```

**Examples: FSK:STAT OFF**

Inquiry command:

```
[SOURCE[1|2]:]FSKey:STATe?
```

Examples: FSK:STAT?

#### 2.3.6.2 FSK modulation source command:

Set command:

```
[SOURCE[1|2]:]FSKey:SOURce {INTernal|EXTernal}
```

**Examples: FSK:SOUR EXT**

Inquiry command:

```
[SOURCE[1|2]:]FSKey:SOURce?
```

**Examples: FSK:SOUR?**

#### 2.3.6.3 FSK hopping frequency command

Set command:

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

**Examples: FSK:FREQ 3kHz**

Inquiry command:

```
[SOURCE[1|2]:]FSKey:FREQuency? [{MINimum|MAXimum}]
```

**Examples: FSK:FREQ?**

#### 2.3.6.4 FSK hopping rate command

Set command:

```
[SOURCE[1|2]:]FSKey:INTernal:RATE{<Value>[kHz|Hz|mHz]|MINimum|MAXimum}
```

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

Inquiry command:

```
[SOURCE[1|2]:]FSKey:INTernal:RATE? [{MINimum|MAXimum}]
```

**Examples: FSK:INT:RATE?**

### 2.3.7 QFSK Modulation command

#### 2.3.7.1 QFSK state command

Set command:

```
[SOURCE[1|2]:]QFSKey:STATe {ON|1|OFF|0}
```

**Examples: QFSK:STAT OFF**

Inquiry command:

[SOURce[1|2]:]QFSKey:STATe?

**Examples: QFSK:STAT?**

### 2.3.7.2 QFSK modulation source command:

Set command:

[SOURce[1|2]:]QFSKey:SOURce {INTernal|EXTernal}

**Examples: QFSK:SOUR EXT**

Inquiry command:

[SOURce[1|2]:]QFSKey:SOURce?

**Examples: QFSK:SOUR?**

### 2.3.7.3 QFSK hopping frequency command

Set command:

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

The three hopping frequency included in QFSK can be distinguished by adding the numerical suffix # (#=1,2,3) behind FREQuency.

**Examples: QFSK:FREQ2 3kHz**

Inquiry command:

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

**Examples: QFSK:FREQ2?**

### 2.3.7.4 QFSK hopping rate command

Set command:

[SOURce[1|2]:]QFSKey:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

**Examples: QFSK:INT:RATE 100**

Inquiry command:

[SOURce[1|2]:]QFSKey:INTernal:RATE? [ {MINimum|MAXimum} ]

**Examples: QFSK:INT:RATE?**

## 2.3.8 4FSK Modulation command

### 2.3.8.1 4FSK state command

Set command:

[SOURce[1|2]:]4FSKey:STATe {ON|1|OFF|0}

**Examples: 4FSK:STAT OFF**

Inquiry command:

[SOURce[1|2]:]4FSKey:STATe?

**Examples: 4FSK:STAT?**

**2.3.8.2 4FSK modulation source command:**

Set command:

[SOURce[1|2]:]4FSKey:SOURce {INTernal|EXTernal}

**Examples: 4FSK:SOUR EXT**

Inquiry command:

[SOURce[1|2]:]4FSKey:SOURce?

**Examples: 4FSK:SOUR?****2.3.8.3 4FSK hopping frequency command**

Set command:

[SOURce[1|2]:]4FSKey:FREQuency# {&lt;Value&gt; [MHz|kHz|Hz|mHz]|MINimum|MAXimum}

Three hopping frequencies included in 4FSK can be distinguished by adding the numerical suffix # (#=1,2,3) behind FREQuency.

**Examples: 4FSK:FREQ2 3kHz**

Inquiry command:

[SOURce[1|2]:]4FSKey:FREQuency#? [ {MINimum|MAXimum} ]

**Examples: 4FSK:FREQ2?****2.3.8.4 4FSK hopping rate command**

Set command:

[SOURce[1|2]:]4FSKey:INTernal:RATE {&lt;Value&gt; [kHz|Hz|mHz]|MINimum|MAXimum}

**Examples: 4FSK:INT:RATE 100**

Inquiry command:

[SOURce[1|2]:]4FSKey:INTernal:RATE? [ {MINimum|MAXimum} ]

**Examples: 4FSK:INT:RATE?****2.3.9 PSK Modulation command****2.3.9.1 PSK state command**

Set command:

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

**Examples: PSK:STAT OFF**

Inquiry command:

[SOURce[1|2]:]PSKey:STATe?

**Examples: PSK:STAT?****2.3.9.2 PSK modulation source command:**

Set command:

[SOURCE[1|2]:]PSKey:SOURce {INTernal|EXTernal}

**Examples: PSK:SOUR EXT**

Inquiry command:

[SOURCE[1|2]:]PSKey:SOURce?

**Examples: PSK:SOUR ?**

### 2.3.9.3 PSK hopping phase command

Set command:

[SOURCE[1|2]:]PSKey:PHASe {<Value>|MINimum|MAXimum}

**Examples: PSK:PHASe 90deg**

Inquiry command:

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

**Examples: PSK:PHASe?**

### 2.3.9.4 PSK hopping rate command

Set command:

[SOURCE[1|2]:]PSKey:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

**Examples: PSK:INT:RATE 100**

Inquiry command:

[SOURCE[1|2]:]PSKey:INTernal:RATE? [ {MINimum|MAXimum} ]

**Examples: PSK:INT:RATE ?**

## 2.3.10 QPSK Modulation command

### 2.3.10.1 QPSK state command

Set command:

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

**Examples: QPSK:STAT OFF**

Inquiry command:

[SOURCE[1|2]:]QPSKey:STATe?

**Examples: QPSK:STAT ?**

### 2.3.10.2 QPSK modulation source command:

Set command:

[SOURCE[1|2]:]QPSKey:SOURce {INTernal|EXTernal}

**Examples: QPSK:SOUR EXT**

Inquiry command:

[SOURCE[1|2]:]QPSKey:SOURce?

**Examples: QPSK:SOUR ?**

### 2.3.10.3 QPSK hopping phase command

Set command:

```
[SOURCE[1|2]:]QPSKey:PHASE# {<Value>|MINimum|MAXimum}
```

Three hopping phase included in QPSK can be distinguished by adding the value of hopping rate behind the numerical suffix # (#=1,2,3)

**Examples: QPSK:PHASE2 90**

Inquiry command:

```
[SOURCE[1|2]:]QPSKey:PHASE? [{MINimum|MAXimum}]
```

**Examples:QPSK:PHASE2?**

### 2.3.10.4 QPSK hopping rate command

Set command:

```
[SOURCE[1|2]:]QPSKey:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}
```

**Examples: QPSK:INT:RATE 100**

Inquiry command:

```
[SOURCE[1|2]:]QPSKey:INTernal:RATE? [{MINimum|MAXimum}]
```

**Examples:QPSK:INT:RATE?**

## 2.3.11 4PSK Modulation command

### 2.3.11.1 4PSK state command

Set command:

```
[SOURCE[1|2]:]4PSKey:STATE {ON|1|OFF|0}
```

**Examples: 4PSK:STAT OFF**

Inquiry command:

```
[SOURCE[1|2]:]4PSKey:STATE?
```

**Examples: 4PSK:STAT?**

### 2.3.11.2 4PSK modulation source command:

Set command:

```
[SOURCE[1|2]:]4PSKey:SOURce {INTernal|EXTernal}
```

**Examples: 4PSK:SOUR EXT**

Inquiry command:

```
[SOURCE[1|2]:]4PSKey:SOURce?
```

**Examples: 4PSK:SOUR?**

### 2.3.11.3 4PSK hopping phase command

Set command:

---

```
[SOURCE[1|2]:]4PSKey:PHASe# {<Value>|MINimum|MAXimum}
```

Three hopping phase included in 4PSK can be distinguished by adding the value of hopping rate behind the numerical suffix # (#=1,2,3)

**Examples: 4PSK:PHASE2 90**

Inquiry command:

```
[SOURCE[1|2]:]4PSKey:PHASe? [{MINimum|MAXimum}]
```

**Examples:4PSK:PHASE2?**

#### 2.3.11.4 4PSK hopping rate command

Set command:

```
[SOURCE[1|2]:]4PSKey:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}
```

**Examples: 4PSK:INT:RATE 100**

Inquiry command:

```
[SOURCE[1|2]:]4PSKey:INTernal:RATE? [{MINimum|MAXimum}]
```

**Examples:4PSK:INT:RATE?**

#### 2.3.12 ASK Modulation command

##### 2.3.12.1 ASK state command

Set command:

```
[SOURCE[1|2]:]ASKey:STATe {ON|1|OFF|0}
```

**Examples: ASK:STAT OFF**

Inquiry command:

```
[SOURCE[1|2]:]ASKey:STATe?
```

**Examples: ASK:STAT?**

##### 2.3.12.2 ASK modulation source command:

Set command:

```
[SOURCE[1|2]:]ASK:SOURce {INTernal|EXTernal}
```

**Examples: ASK:SOUR EXT**

Inquiry command:

```
[SOURCE[1|2]:]ASK:SOURce?
```

**Examples: ASK:SOUR?**

##### 2.3.12.3 ASK hopping amplitude command

Set command:

```
[SOURCE[1|2]:]ASK:AMPLitude {<Value>|MINimum|MAXimum}
```

**Examples:ASK:AMPL 60**

Inquiry command:

---

---

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

**Examples: ASK:AMPL?**

### 2.3.12.4 ASK hopping rate command

Set command:

[SOURce[1|2]:]ASK:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}

**Examples: ASK:INT:RATE 100**

Inquiry command:

[SOURce[1|2]:]ASK:INTernal:RATE? [{MINimum|MAXimum}]

**Examples: ASK:INT:RATE?**

### 2.3.13 OSK Modulation command

#### 2.3.13.1 OSK state command

Set command:

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

**Examples: OSK:STAT OFF**

Inquiry command:

[SOURce[1|2]:]OSK:STATe?

**Examples: OSK:STAT?**

#### 2.3.13.2 OSK modulation source command:

Set command:

[SOURce[1|2]:]OSK:SOURce {INTernal|EXTernal}

**Examples: OSK:SOUR EXT**

Inquiry command:

[SOURce[1|2]:]OSK:SOURce?

**Examples: OSK:SOUR?**

#### 2.3.13.3 OSK oscillation time command

Set command:

[SOURce[1|2]:]OSK:TIME {<Value>|MINimum|MAXimum}

**Examples: OSK:TIME 30us**

Inquiry command:

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

**Examples: OSK:TIME?**

#### 2.3.13.4 OSK hopping rate command

Set command:

---

---

```
[SOURce[1|2]:]OSK:INTernal:RATE {<Value>[kHz|Hz|mHz]|MINimum|MAXimum}
```

**Examples: OSK:INT:RATE 100**

Inquiry command:

```
[SOURce[1|2]:]OSK:INTernal:RATE? [{MINimum|MAXimum}]
```

**Examples: OSK:INT:RATE?**

## 2.4 Sweeping command

### 2.4.1 Frequency sweeping

#### 2.4.1.1 Starting frequency command

Set command:

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

**Examples: FREQ:STAR 120**

Inquiry command:

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

**Examples: FREQ:STAR?**

#### 2.4.1.2 Ending frequency command

Set command:

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

**Examples: FREQ:STAR 25KHz**

Inquiry command for ending frequency:

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

**Examples: FREQ:STOP?**

#### 2.4.1.3 Mark frequency command

Set command:

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

**Examples: MARK:FREQ 5kHz**

Inquiry command:

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

**Examples: MARK:FREQ?**

#### 2.4.1.4 Sweeping type command

Set command:

```
[SOURce[1|2]:]SWEep:SPACing {LINear|LOGarithmic|LIST}
```

**Examples: SWE:SPAC LOG**

Inquiry command:

```
[SOURce[1|2]:]SWEep:SPACing?
```

**Examples: SWE:SPAC?****2.4.1.5 Sweeping time command**

Set command:

```
[SOURCE[1|2]:]SWEep:TIME {<Value>[s|ms]|MINimum|MAXimum}
```

**Examples: SWE:TIME 5s**

Inquiry command:

```
[SOURCE[1|2]:]SWEep:TIME? [{MINimum|MAXimum}]
```

**Examples: SWE:TIME?**

**2.4.1.6 Hold time command**

Set command:

```
[SOURCE[1|2]:]SWEep:HTIME {<Value>[s|ms]|MINimum|MAXimum}
```

**Examples: SWE:HTIME 5s**

Inquiry command:

```
[SOURCE[1|2]:]SWEep:HTIME? [{MINimum|MAXimum}]
```

**Examples: SWE:HTIME?**

**2.4.1.7 Return time command**

Set command:

```
[SOURCE[1|2]:]SWEep:RTIME {<Value>[s|ms]|MINimum|MAXimum}
```

**Examples: SWE:RTIME 5s**

Inquiry command:

```
[SOURCE[1|2]:]SWEep:RTIME? [{MINimum|MAXimum}]
```

**Examples: SWE:RTIME?**

**2.4.1.8 Interval time command**

Set command:

```
[SOURCE[1|2]:]SWEep:INTerval {<Value>[s|ms]|MINimum|MAXimum}
```

**Examples: SWE:INT 5s**

Inquiry command:

```
[SOURCE[1|2]:]SWEep:INT? [{MINimum|MAXimum}]
```

**Examples: SWE:INT?**

**2.4.1.9 Sweeping state command**

Set command:

```
[SOURCE[1|2]:]SWEep[:]STATe {ON|1|OFF|0}
```

**Examples: SWE:STAT ON**

Inquiry command:

[SOURce[1|2]:]SWEep[:STATe]?

**Examples: SWE:STAT?**

#### 2.4.1.10 Sweeping source command

Set command:

TRIGger[1|2]:SOURce {IMMEDIATE|EXTernal}

**Examples: TRIG:SOUR IMM**

Inquiry command:

TRIGger[1|2]:SOURce?

**Examples: TRIG:SOUR?**

#### 2.4.1.11 Trigger slope command

Set command:

TRIGger[1|2]:SLOPe {POSitive|NEGative}

**Examples: TRIG:SLOP POS**

Inquiry command:

TRIGger:SLOPe?

**Examples: TRIG:SLOP?**

### 2.4.2 List sweeping command

#### 2.4.2.1 Starting point command

Set command:

[SOURce[1|2]:]LIST:POINt:STARt {<Value>|MINimum|MAXimum}

**Example: LIST:POIN:STAR 100**

Inquiry command:

[SOURce[1|2]:]LIST:POINt:STARt?

**Example: LIST:POIN:STAR?**

#### 2.4.2.2 Ending point command

Set command:

[SOURce[1|2]:]LIST:POINt:STOP {<Value>|MINimum|MAXimum}

**Example: LIST:POIN:STOP 100**

Inquiry command:

[SOURce[1|2]:]LIST:POINt:STOP?

**Example: LIST:POIN:STOP?**

#### 2.4.2.3 Total point numbers command of list sweeping

---

Set command:

[SOURce[1|2]:]LIST:FREQuency:POINts {value|MINimum|MAXimum}

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

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

Inquiry command:

**Example: LIST:FREQ:POINts?**

#### 2.4.2.4 Frequency set for specified point of list sweeping

[SOURce[1|2]:]LIST:FREQuency {<num>,<freq>}

**Example: LIST:FREQ 10,3khz**

#### 2.4.2.5 List sweeping dwell time command

Set command:

[SOURce[1|2]:]LIST:DWELl {<Value[s|ms]|MINimum|MAXimum}

**Example: LIST:DWEL 1s**

Inquiry command:

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

**Example: LIST:DWEL?**

### 2.5 Burst command

#### 2.5.1 Burst mode command

Set command:

[SOURce[1|2]:]BURSt:MODE {TRIGgered|GATed}

**Examples: BURS:MODE GAT**

Inquiry command:

[SOURce[1|2]:]BURSt:MODE?

**Examples: BURS:MODE GAT**

**BURS:MODE?**

#### 2.5.2 N cycle command

Set command:

[SOURce[1|2]:]BURSt:NCYCles {<Value>|MINimum|MAXimum}

**Examples: BURS:NCYC 2**

Inquiry command:

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

**Examples: BURS:NCYC?**

#### 2.5.3 Burst period command

Set command:

---

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

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

Inquiry command:

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

**Examples: BURS:INT:PER?**

#### 2.5.4 Burst start phase command

Set command:

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

**Examples: BURS:PHAS 180**

Inquiry command:

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

**Examples: BURS:PHAS?**

#### 2.5.5 Burst state command

Set command:

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

**Examples: BURS:STAT ON**

Inquiry command:

[SOURce[1|2]:]BURSt:STATe?

**Examples: BURS:STAT?**

#### 2.5.6 Trigger source command

Set command:

TRIGger[1|2]:SOURce {IMMediate|EXTernal}

**Examples: TRIG:SOUR EXT**

Inquiry command:

TRIGger[1|2]:SOURce?

**Examples: TRIG:SOUR?**

### 2.6 Dual channel command

#### 2.6.1 Frequency coupling command

##### 2.6.1.1 Frequency ratio command

Set command:

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

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

Inquiry command:

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

**Examples: `FREQ:COUP:RAT?`****2.6.1.2 Frequency difference command**

Set command:

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

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

Inquiry command:

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

**Examples: `FREQ:COUP:OFFS?`****2.6.1.3 Frequency coupling state command**

Set command:

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

**Examples: `FREQ:COUP ON`**

Inquiry command:

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

**Examples: `FREQ:COUP?`****2.6.2 Voltage coupling command****2.6.2.1 Amplitude coupling command**

Set command for amplitude difference:

```
[SOURce:]VOLTage:COUPle:AMPLitude {<Value >}
```

**Examples: `VOLT:COUP:AMPL 1VRMS`**

Inquiry command for amplitude difference:

```
[SOURce:]VOLTage:COUPle:AMPL?
```

**Examples: `VOLT:COUP:AMPL?`****2.6.2.2 Offset coupling command**

Set command for offset difference:

```
[SOURce:]VOLTage:COUPle:OFFSet {<Value >}
```

**Examples: `VOLT:COUP: OFFS 1VRMS`**

Inquiry command for offset difference:

```
[SOURce:]VOLTage:COUPle:OFFS?
```

**Examples: `VOLT:COUP:OFFS?`****2.6.2.3 Voltage coupling state command**

Set command:

[SOURCE:]VOLTage:COUPlE[:STATe] {ON|1|OFF|0}

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

Inquiry command:

[SOURCE:]VOLTage:COUPlE[:STATe]?

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

## 2.6.3 Waveform combination command

### 2.6.3.1 Combination amplitude command

Set command:

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

**Examples: COMB:AMPL 50**

Inquiry command:

[SOURCE:]COMBine:AMPLitude? [{MINimum|MAXimum}]

**Examples: COMB:AMPL?**

### 2.6.3.2 Coupling direction command

Set command:

[SOURCE[1|2]:]COMBine:FEED {CHA|CHB}

Wherein, 1|2 means the specified basic channel.

**Examples: COMB:FEED CHB**

Inquiry command:

[SOURCE[1|2]:]COMBine:FEED?

**Examples: COMB:FEED?**

### 2.6.3.2 Waveform combination state command

Set command:

[SOURCE:]COMBine:COUPlE[:STATe] {ON|1|OFF|0}

**Examples: COMB:COUP ON**

Inquiry command:

**Examples: COMB:COUP?**

## 2.7 Arbitrary waveform editing command

### 2.7.1 Arbitrary waveform sampling rate command

Set command:

ARB:SET:SAMPRate {<Value>[MASa/s|kSa/s|Sa/s|mSa/s|uSa/s]|MINimum|MAXimum}

**Examples: ARB:SET:SAMPRate 1.1MASa/s**

Inquiry command:

ARB:SET:SAMPRate?

**Examples: ARB:SET:SAMPRate?**

### 2.7.2 Arbitrary waveform amplitude

Look at 2.1.9

### 2.7.3 Arbitrary waveform offset

Look at 2.1.12

### 2.7.4 Arbitrary waveform length

Inquiry command:

```
ARB:SET:POINts {<Value>MINimum|MAXimum}
```

**Examples: ARB:SET:POINts 1000**

Inquiry command:

```
ARB:SET:POINts?
```

**Examples: ARB:SET:POINts?**

### 2.7.5 Arbitrary Long Waveform Data

Set command:

```
ARB:DATA {space} {#bytes} {,} {24 bytes} {Data payload} {0x0a}
```

Wherein, #bytes = len(Data payload)+24+1

#### Introduction of the command

Frame Header	Space mark	Length domain	Space mark	Reserve domain	Data domain	Frame tail
ARB:DATA	Space	2073	,	24 bytes	1024*2 bytes	\n

#### Frame Header

“ARB:DATA” printable character, fixed character

#### Length domain

Len indicates the data length which is transmitted from PC software to generator including information behind ‘,’ till \n . For example, the data length that PC software transmits to generator is 475, then the character string of len is 475.

Note:

The waveform length that PC software transmits to generator should not surpass 1\*1024\*1024, otherwise, all the data will not be processed.

Data length is 24+points\*2+1, in which, 24 is reserve data domain length, points\*2 means each point takes 2 bytes, the other is the ending character.

#### 24 bytes reserve domain

Reserved, meaningless

#### Data

Each point in the waveform takes 2 bytes, and the range is 0-16383, low bytes at the front.

Calculating method:

Assume the voltage of some point is  $x$ (mV), amplitude pp value of waveform is  $a$ (mVpp), offset of waveform is  $o$ (mV).

Then the value in this point in the waveform data should be  $(x - o) / a * 16383 + 8192$ , which should be more than 0, less than 16383. The calculation result saves as short type taking 2 bytes.

**Frame tail**

End character '\n'.

**Example**

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	41	52	42	3A	44	41	54	41	20	35	37	2C	AA	55	AA	55	ARB:DATA	57, ^U^U
00000010	AA	55	AA	55	AA	55	AA	55	AA	55	AA	55	AA	55	AA	55	^U^U^U^U^U^U^U^U^U^U	
00000020	AA	55	AA	55	46	1D	FD	54	F5	4F	6F	E5	4F	5E	4F	51	^U^UF	ýTõoãO^OQ
00000030	F6	54	DF	64	DF	54	54	54	54	46	1D	FD	54	F5	4F	6F	öTßdßTTTTF	ýTõo
00000040	E5	4F	5E	94	0A												ãO^"	

This group data indicates an arbitrary waveform with 16 points.

Green background is length domain, 57 means there are totally 57 bytes after ','(0x2C).

Blue background AA55 parts reserves for now, and can be input any data.

Purple background is data domain, which is calculated according to the former formula.

0x0A ('\n') is frame tail end character.

**2.7.6 Arbitrary Short Waveform Data**

Set command:

```
ARB:SHRT[1|2]{space} {#bytes} {,} {24 bytes} {Data payload} {0x0a}
```

Wherein, #bytes = len(Data payload)+24+1

Introduction of command

Frame Header	Space mark	Length domain	Space mark	Reserve domain	Data domain	Frame tail
ARB:SHRT1	Space	2073	,	24 bytes	1024*2 bytes	\n

For instrument with dual channels, ARB:SHRT1 is for CHA, and ARB:SHRT2 is for CHB.

Other domain introduction about the command is same with section 2.7.5.

**2.8 System command**

**2.8.1 Store status command**

```
*SAV {0|1|2|3|4}
```

Example: \*SAV 1

**2.8.2 Recall state command**

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

Example: \*RCL 0

**2.8.3 Return to local command**

```
SYSTem:LOCal
```

Example: SYST:LOC

### 2.8.4 System reset command

\*RST

Example: \*RST

### 2.8.5 Trigger command

\*TRG

Example: \*TRG

### 2.8.6 System error command

SYSTem:ERRor?

Examples:SYST:ERR?

### 2.8.7 Error message queue clearance

\*CLS

Examples:\*CLS

## 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, "0, 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:

Error code	Explanation
"0,No error"	No error
"-100, Queue overflow"	error queue overflow
"-101,Invalid Command"	Command error
"-102,Invalid parameters count"	Params count error
"-104,Invalid parameters unit type"	Parameter unit error
"-105,Invalid parameter value"	Parameter value error
"-106,Too many number suffix"	Too many number suffix

"-107,Invalid number suffix"	Suffix error
"-108,Need more data"	Need more data
"-200,Frequency out of range"	Frequency out of range
"-201,Amplitude out of range"	Amplitude out of range
"-202,Offset out of range"	Offset out of range
"-203,No user wave is used"	No customer-defined waveform is used
"-204,No such user waveform"	There is no specified customer-defined waveform.
"-205,No unit \"dBm\" can be used when the load is highz"	The unit of amplitude can't be dBm in high impedance state
"-206,High level out of range"	High level out of range
"-207,Low level out of range"	Low level out of range
"-208,Start phase out of range"	Start phase out of range
"-209,Ramp symmetry out of range"	Ramp symmetry out of range
"-210,Square duty out of range"	Square duty cycle out of range
"-211,Load out of range"	load out of range
"-212,Pulse period out of range"	Pulse period out of range
"-213,Pulse width out of range"	Pulse width out of range
"-214,Pulse edge time out of range"	Pulse edge time out of range
"-215,No such modulate waveform"	No such modulate waveform
"-216,Trigger out can not be turned on while the source is external"	Trigger out can not be turned on while the source is external
"-217,Mod freq out of range"	Mod freq out of range
"-218,Sum ampl out of range"	Sum ampl out of range
"-219,Pulse width deviation out of range"	Pulse width deviation out of range
"-220,Am depth out of range"	AM depth out of range
"-221,Fm deviation out of range"	FM deviation out of range
"-222,Pm deviation out of range"	PM deviation out of range
"-223,Fskey hop freq out of range"	FSK hopping frequency out of range
"-224,Fskey hop rate out of range"	FSK hopping rate out of range
"-225,QFSK hop frequency out of range"	QFSK hopping frequency out of range
"-226,QFSK hop rate out of range"	QFSK hopping rate out of range

"-227,4FSK hop frequency out of range"	4FSK hopping frequency out of range
"-228,4FSK hop rate out of range"	4FSK hopping rate out of range
"-229,PSK hop phase out of range"	PSK hopping phase out of range
"-230,PSK hop rate out of range"	PSK hopping rate out of range
"-231,QPSK hop phase out of range"	QPSK hopping phase out of range
"-232,4PSK hop phase out of range"	4PSK hopping phase out of range
"-233,QPSK hop rate out of range"	QPSK hopping rate out of range
"-234,4PSK hop rate out of range"	4PSK hopping rate out of range
"-235,ASK hop rate out of range"	ASK hopping rate out of range
"-236,ASK hop ampl out of range"	ASK hopping amplitude out of range
"-237,OSK hop rate out of range"	OSK hopping rate out of range
"-238,OSK hop time out of range"	OSK hopping time out of range
"-240,Sweep start frequency out of range"	Start sweeping frequency out of range
"-241,Sweep stop frequency out of range"	End sweeping frequency out of range
"-242,Sweep marker frequency out of range"	Sweeping marking frequency out of range
"-243,List num out of range"	Sequence number of list sweeping out of range
"-244,List freq out of range"	Sweeping frequency of list sweeping out of range
"-245,Sweep time out of range"	Sweeping time of frequency sweeping out of range
"-246,Sweep hold time out of range"	Retention time of frequency sweeping out of range
"-247,Sweep return time out of range"	Return time of frequency sweeping out of range
"-248,Sweep dwell time out of range"	Dwell time of list sweeping out of range
"-249,Sweep interval out of range"	Interval time of frequency

	sweeping out of range
"-250,List start number out of range"	Starting number of list sweeping out of range
"-251,List stop number out of range"	Stop number of list sweeping out of range
"-252,Sweep list total points out of range"	Total points of list sweeping out of range
"-253,Trigger source can be set only in burst or sweep function"	Trigger source can be set only in burst or sweep function
"-254,Current source is not manual"	Current source is not manual
"-255,Burst ncycle out of range"	Number of burst cycle out of range
"-256,Dual comb ampl out of range"	Dual comb ampl out of range
"-257,Dual same channel is used"	Dual same channel is used
"-258,Unit "Vrms"can only be used in standard waveform except noise"	Unit 'Vrms' can only be used in standard waveform except noise"
"-259,No parameters can be set when the channel is in dual state."	No parameters can be set when the channel is in dual state.
"-260,High level limit out of range. "	High level limit out of range
"-261,Low level limit out of range. "	Low level limit out of range
"-262,Burst period out of range. "	Burst period out of range
"-263,Frequency dual is not open. "	Frequency dual is not open
"-264,Voltage dual is not open. "	Voltage dual is not open
"-265,Voltage dual amplitude difference out of range. "	Voltage dual amplitude difference out of range
"-266,Voltage dual offset difference out of range. "	Voltage dual offset difference out of range
"-267,Voltage dual amplitude only support unit \"Vpp\""	The unit of Amplitude deviation of voltage coupling can only be Vpp.
"-268,Combine dual is not open. "	Waveform combination is not open.