Remote Control Manual

SDG800 Series Function/Arbitrary Waveform Generator

RC02008-E02A

Catalogue

Introduction to the SCPI Language

1.1. About Commands & Queries

This section lists and describes the remote control commands and queries recognized by the instrument. All commands and queries can be executed in either local or remote state.

The description for each command or query, with syntax and other information, begins on a new page. The name (header) is given in both long and short form at the top of the page, and the subject is indicated as a command or query or both. Queries perform actions such as obtaining information, and are recognized by the question mark (?) following the header.

1.1.1. How they are listed?

The descriptions are listed in alphabetical order according to their short form.

1.1.2. How they are described?

In the descriptions themselves, a brief explanation of the function performed is given. This is followed by a presentation of the formal syntax, with the header given in Upper-and-Lower-Case characters and the short form derived from it in ALL UPPER-CASE characters. Where applicable, the syntax of the query is given with the format of its response.

1.1.3. When can they be used?

The commands and queries listed here can be used for SDGxxxx Series digital instruments.

1.1.4.Command Notation

The following notation is used in the commands:

< > Angular brackets enclose words that are used placeholders, of which there are two types: the header path and the data parameter of a command.

:= A colon followed by an equals sign separates a placeholder from the description of the type and range of values that may be used in a command instead of the placeholder.

{ } Braces enclose a list of choices, one of which one must be made.

[ ] Square brackets enclose optional items.

… An ellipsis indicates that the items both to its left and right may be repeated a number of times.

1.2. Table of Commands & Queries

Short	Long Form	Subsystem	What Command/Query does
*IDN	*IDN	SYSTEM	Get identification from device.
*OPC	*OPC	SYSTEM	Get or set the OPC bit (0) in the Event Status Register (ESR).
*CLS	*CLS	SYSTEM	Clears all the status data registers.
*ESE	*ESE	SYSTEM	set or get the Standard Event Status Enable register.
*ESR	*ESR	SYSTEM	reads and clears the contents of the Event Status Register.
*RST	*RST	SYSTEM	Initiates a device reset.
*SRE	*SRE	SYSTEM	Set the Service Request Enable register.
*STB	*STB	SYSTEM	Get the contents of the 488.1 defined status register.
*TST	*TST	SYSTEM	Get performs an internal self-test.
*WAI	*WAI	SYSTEM	wait to continue command
CHDR	COMM_HEADER	SYSTEM	Set the format of Return data（Long, short, off）.
BSWV	BASIC_WAVE	SIGNAL	Set or get basic wave parameters.
ARWV	ARBWAVE	Data SYSTEM	Change arbitrary wave type.
BUZZ	BUZZER	SYSTEM	Set or get buzzer State.
SCFG	SYSTEM_CONFIG	SYSTEM	Set or get power on initializing parameter way
ROSC	ROSCILLATOR	SIGNAL	Set or get clock source.
OUTP	OUTPUT	SIGNAL	Set or get output state.
PACP	CHANNEL_COPY	SIGNAL	Copy parameters from channel one to channel two, or from channel two to channel one.
SCSV	SCREEN_SAVE	SYSTEM	Set or get screen save State.
SWWV	SWEEP	SIGNAL	Set or get sweep wave.
SYNC	SYNC	SIGNAL	Set or get in-phase signal.
BTWV	BURSTWAVE	SIGNAL	Set or get burst wave parameters.
MDWV	MODULATEWAVE	SIGNAL	Set or get modulate wave parameters.

NBFM	NUMBER_FORMAT	SYSTEM	Set Data Format.
STL	STORE_LIST	SIGNAL	Get the all of device wave date name.
VKEY	VIRTUALKEY	SYSTEM	Set the virtual key.
DDR	DDR	SYSTEM	Reads and clears the Device Dependent Register
CMR	CMR	SYSTEM	Reads and clears the Command error Register

1.3. IEEE 488.2 Common Command Introduction

IEEE standard defines the common commands used for querying the basic information of the instrument or executing basic operations. These commands usually start with "*" and the length of the keywords of the command is usually 3 characters.

1.3.1 IDN

DESCRIPTION

The *IDN? Query causes the instrument to identify itself. The response comprises manufacturer, scope model, serial number, software version and firmware version.

QUERY SYNTAX

*IDN?

RESPONSE FORMAT

*IDN, <device id>,<model>,<serial number>,<software version>,<firmware version>
<device id>：="WST"is used to identify instrument.
<model>:= A model identifier less than 14 characters.
<software version>:= A serial numbers about software version.
<firmware version>:= two digits giving the major release level followed by a period, then one digit giving the minor release level followed by a period and a single-digit update level (xx.y.z).

EXAMPLE

Reads version information.
*IDN?
Return:
*IDN WST,WaveStation 3162,120465,5.01.02.05,02-00-00-21-24(Return may differ from each version)

1.3.2 OPC

DESCRIPTION

The *OPC (Operation Complete) command sets to true the OPC bit (bit 0) in the standard Event Status Register (ESR).
This command has no other effect on the operation of the device because the instrument starts parsing a command or query only after it has completely processed the previous command or query.
The *OPC? query always responds with the ASCII character 1 because the device only responds to the query when the previous command has been entirely executed.

COMMAND SYNTAX

*OPC

QUERY SYNTAX

*OPC?

RESPONSE FORMAT

*OPC 1

RELATED COMMANDS

*WAI

1.3.3 CLS

DESCRIPTION

The *CLS command clears all the status data registers.

COMMAND SYNTAX

*CLS

EXAMPLE

The following command causes all the status data registers to be cleared:

```
*CLS
```

RELATED COMMANDS

ALL_STATUS, CMR, DDR, *ESR, EXR, *STB, URR

1.3.4 ESE

DESCRIPTION

The *ESE command sets the Standard Event Status Enable register (ESE). This command allows one or more events in the ESR register to be reflected in the ESB summary message bit(bit 5) of the STB register.

The *ESE? query reads the contents of the ESE register.

COMMAND SYNTAX

*ESE <value>
<value> : = 0 to 255

QUERY SYNTAX

*ESE?

RESPONSE FORMAT

*ESE <value>

EXAMPLE

The following instruction allows the ESB bit to be set if a user request (URQ bit 6, i.e. decimal 64) and/or a device dependent error (DDE bit 3, i.e. decimal 8) occurs. Summing these values yields the ESE register mask 64+8=72.

*ESE?
Return:
*ESE 72

RELATED COMMANDS

*ESR

1.3.5 ESR

DESCRIPTION The *ESR? query reads and clears the contents of the Event Status Register (ESR). The response represents the sum of the binary values of the register bits 0 to 7. The table below gives an overview of the ESR register structure.

QUERY SYNTAX *ESR?

RESPONSE FORMAT *ESR <value>
<value> : = 0 to 255

EXAMPLE The following instruction reads and clears the contents of the ESR register:
*ESR?
Return：
*ESR 0

RELATED COMMANDS ALL_STATUS, *CLS, *ESE

1.3.6 RST

DESCRIPTION The *RST command initiates a device reset. The *RST sets all eight traces to the GND line and recalls the default setup.

COMMAND SYNTAX * RST

EXAMPLE This example resets the signal generator:
*RST

RELATED COMMANDS *CAL, *RCL

1.3.7 SRE

DESCRIPTION

The *SRE command sets the Service Request Enable register (SRE). This command allows the user to specify which summary message bit(s) in the STB register will generate a service request. Refer to the table on page 167 for an overview of the available summary messages.

A summary message bit is enabled by writing a ‘1’ into the corresponding bit location. Conversely, writing a ‘0’ into a given bit location prevents the associated event from generating a service request (SRQ). Clearing the SRE register disables SRQ interrupts.

The *SRE? query Returns a value that, when converted to a binary number, represents the bit settings of the SRE register. Note that bit 6 (MSS) cannot be set and its Returned value is always zero.

COMMAND SYNTAX

*SRE <value>

<value> : = 0 to 255

QUERY SYNTAX

*SRE?

RESPONSE FORMAT

*SRE <value>

EXAMPLE

The following instruction allows an SRQ to be generated as soon as the MAV summary bit (bit 4, i.e. decimal 16) or the INB summary bit (bit 0, i.e. decimal 1) in the STB register, or both,are set. Summing these two values yields the SRE mask 16+1 = 17.

*SRE?

Return:

*SRE 17

1.3.8 STB

DESCRIPTION

The *STB? query reads the contents of the 488.1 defined status register (STB), and the Master Summary Status (MSS).
The response represents the values of bits 0 to 5 and 7 of the Status Byte register and the MSS summary message.
The response to a *STB? query is identical to the response of a serial poll except that the MSS summary message appears in bit 6 in place of the RQS message. Refer to the table on page 167 for further details of the status register structure.

QUERY SYNTAX

*STB?

RESPONSE FORMAT

*STB <value>
<value> : = 0 to 255

EXAMPLE

The following reads the status byte register:
*STB?
Return:
*STB 0

RELATED COMMANDS

ALL_STATUS, *CLS, *PRE, *SRE

1.3.9 TST

DESCRIPTION

The *TST? query performs an internal self-test, the response indicating whether the self-test has detected any errors. The selftest includes testing the hardware of all channels, the timebase and the trigger circuits.
Hardware failures are identified by a unique binary code in the Returned <status> number. A “0” response indicates that no failures occurred.

QUERY SYNTAX

*TST?

RESPONSE FORMAT

*TST <status>
<status> : = 0 self-test successful

EXAMPLE

The following causes a self-test to be performed:
TST?
Return(if no failure):
*TST 0

RELATED COMMANDS

*CAL

1.3.10 WAI

DESCRIPTION

The *WAI (WAIt to continue) command, required by the IEEE 488.2 standard, has no effect on the instrument, as the signal generator only starts processing a command when the previous command has been entirely executed.

COMMAND SYNTAX

*WAI

RELATED COMMANDS

*OPC

1.3.11 DDR

DESCRIPTION The DDR? query reads and clears the contents of the Device Dependent or device specific error Register (DDR). In the case of a hardware failure, the DDR register specifies the origin of the failure.

QUERY SYNTAX DDR?

RESPONSE FORMAT DDR <value>
<value> : = 0 to 65535

EXAMPLE DDR?
Response message:
DDR 0

The following table gives details：

Bit	Bit Value	Description
15...14		Reserved
13	8192	Timebase hardware failure detected
12	4096	Trigger hardware failure detected
11		Reserved
10		Reserved
9	512	Channel 2 hardware failure detected
8	256	Channel 1 hardware failure detected
7	128	External input overload condition detected
6…4		Reserved
3		Reserved
2		Reserved
1	2	Channel 2 overload condition detected
0	1	Channel 1 overload condition detected

1.3.12 CMR

DESCRIPTION

The CMR? query reads and clears the contents of the CoMmand error Register (CMR) —see table next —which specifies the last syntax error type detected by the instrument.

QUERY SYNTAX

CMR?

RESPONSE FORMAT

CMR <value>
<value> : = 0 to 14

EXAMPLE

CMR?
Response message:
CMR 0

Value	Description
0	
1	Unrecognized command/query header
2	Invalid character
3	Invalid separator
4	missing parameter
5	Unrecognized keyword
6	String error
7	parameter cannot allowed
8	Command String Too Long
9	Query cannot allowed
10	missing Query mask
11	Invalid parameter
12	parameter syntax error
13	filename too long
14	directory not exist

1.4. Comm_Header Command

DESCRIPTION

This Command is used to change query command Return format.
SHORT parameter is Return short format. LONG parameter is Return long format. Off is that command header and parameter unit will not Return.

COMMAND SYNTAX

Comm_HeaDeR <parameter>
<parameter>:= {SHORT,LONG,OFF}

QUERY SYNTAX

Comm_HeaDeR?

RESPONSE FORMAT

SYNC <parameter>

EXAMPLE

Set query command format to long.
CHDR LONG

Read query command format.
CHDR?
Return:
COMM_HEADER LONG

1.5. Output Command

DESCRIPTION

Enable or disable the output of the [Output] connector at the front panel corresponding to the channel.
The query Returns ON or OFF.

COMMAND SYNTAX

<channel>: OUTPut <parameter>
<channel>:={C1,C2}
<parameter >:= {a parameter from the table below}

Parameters	Value	Description
ON	---	Turn on channel
OFF	---	Turn off channel
LOAD	<load>	Value of load
PLRT	<NOR,INVT>	Value of polarity parameters

where: <load>:= {50(default unit is ohm), HZ}

QUERY SYNTAX

<channel>: OUTPut?

RESPONSE FORMAT

<channel>:OUTPut <load>

EXAMPLE

Turns on channel one.
C1:OUTP ON

Reads channel one output state.
C1:OUTP?
Return:
C1:OUTP ON,LOAD,HZ,PLRT,NOR

Set the load to 500
C1:OUTP LOAD,50

Set the load to HiZ
C1:OUTP LOAD,HZ

Set the polarity normal
C1:OUTP PLRT,NOR

Set the polarity inverted
C1:OUTP PLRT,INVT

1.6. Basic Wave Command

DESCRIPTION Set or get basic wave parameters.

COMMAND SYNTAX <channel>:BaSicWaVe <parameter>

<channel>:={C1, C2}

<parameter>:= {a parameter from the table below}

Parameters	Value	Description
WVTP	<type>	Type of wave
FRQ	<frequency>	Value of frequency. If wave type is Noise, you can't set this parameter.
AMP	<amplifier>	Value of amplifier. If wave type is Noise, you can't set this parameter.
OFST	<offset>	Value of offset. If wave type is Noise, you can't set this parameter.
SYM	<symmetry>	Value of symmetry. Only wave type is Ramp, you can set this parameter.
DUTY	<duty>	Value of duty cycle. Only wave type is Square and Pulse, you can set this parameter.
PHSE	<phase>	Value of phase. If wave type is Noise and Pulse, you can't set this parameter.
STDEV	<stdev>	Value of Noise wave stdev. Only wave type is Noise, you can set this parameter.
MEAN	<mean>	Value of Noise wave mean. Only wave type is Noise, you can set this parameter.
WIDTH	<width>	Value of width, Only wave type is Pulse, you can set this parameter.
RISE（WST3000）	<rise>	Value of rise, Only wave type is Pulse, you can set this parameter.
FALL（WST3000）	<fall>	Value of fall, Only wave type is Pulse, you can set this parameter.
DLY	<delay>	Value of delay, Only wave type is Pulse ,you can set this parameter.
PERI	<period>	Value of period, If wave type is Noise, you cann' set this parameter.
HLEV	<high level>	Value of high level, If wave type is Noise, you cann' set this parameter.
LLEV	<low level>	Value of low level, If wave type is Noise, you cann' set this parameter.

Note: if the command don't set basic wave type, the parameter will set parameters to current device wave type default.

where: <type>:={SINE, SQUARE, RAMP, PULSE, NOISE, ARB ,DC}

<frequency>:= { Default unit is "HZ". Minimum value is 1xe-6 HZ, maximal value depends on the version.}

<amplifier>:= {Default unit is "V". Channel one minimum value 0.004V, Maximal is 6V. Channel two minimum value 0.004V, Maximal is 20V. }

<offset>:= { Default unit is "V". maximal value depends on the maximal value depends on the version.}

<duty>:= { if wave type is Square, range is from 20% to 80%.
if wave type is pulse, range is from 0.0012% to 99.9988%}

<symmetry> :={ 0% to 100%}

<phase>:= {0={ 0 to 360}

<stdev>:= Maximal is 2.222V, minimum is 0.5mV.The default unit is "V".

<mean>:= The range depends on Variance . The default unit is "V".

<width>:= Max_width < (Max_duty * 0.01) * period and Min_width > (Min_duty * 0.01) * period.

<rise>:= Maximal is less than width ,Minimal is 6ns.

<fall>:= Maximal is less than width ,Minimal is 6ns.

<delay>:= Maximal is Pulse Period, minimum value is 0.Unit is S.

QUERY SYNTAX <channel>:BaSicWaVe?
<channel>:={C1, C2}

RESPONSE FORMAT <channel>:BSWV<type>,<frequency>,<amplifier>,<offset>, <duty>,<symmetry>, <phase>,<variance>,<mean>,<width>, <rise>,<fall>,<delay>.

EXAMPLE change channel one current wave type to ramp.

```
C1:BSWV WVTP,RAMP
```

Changes current signal frequency of channel one to 2000 Hz.

```
C1: BSWV FRQ, 2000HZ
```

set current signal amplifier of channel one.

```
C1: BSWV AMP, 3V
```

reads channel basic wave parameters from device.

```
C1:BSWV?
```

Return:

```
C1: BSWV WVTP,SINE,FRQ,100HZ,PERI,0.01S,AMP,2V,
OFST,0V,HLEV,1V,LLEV,-1V,PHSE,0
```

RELATED COMMANDS ARWV, BTWV, CFG, CPL, MDWV, SWWV

1.7. Modulate Wave Command

DESCRIPTION Set or get modulated wave parameters.

COMMAND SYNTAX <channel>:MoDulateWaVe <parameter>
<channel>:={C1, C2}
<parameter>:= {a parameter from the table below}

Parameters	Value	Description
STATE	<state>	Turn on or off modulated wave. Note if you want set or read modulate wave other parameter you must turn on the state first.
AM,SRC	<src>	AM signal source.
AM,MDSP	<mod wave shape>	AM modulation wave. . Only AM signa l source is set to INT.
AM,FRQ	<am frequency>	AM frequency. Only AM signal source is set to INT.
AM,DEPTH	<depth>	AM deep. Only AM signal source is set to INT.
DSBAM,SRC	<src>	DSBAM signal source
DSBAM,MDSP	<mod wave shape>	DSBAM modulation wave. . Only AM signal source is set to INT.
DSBAM,FRQ	<dsbam frequency>	DSBAM frequency. Only AM signal source is set to INT.
FM,SRC	<src>	FM signal source
FM, MDSP	<mod wave shape>	FM modulation wave. Only FM signal source is set to INT.
FM,FRQ	<fm frequency>	FM frequency. Only FM signal source is set to INT.
FM,DEVI	<fm frequency offset>	FM frequency offset. Only FM signal source is set to INT.
PM,SRC,	<src>	PM signal source
PM,MDSP	<mod wave shape>	PM modulation wave. Only PM signal source is set to INT.
PM,FRQ	<pm frequency>	PM frequency. Only PM signal source is set to INT.
PWM,FRQ	<pwm frequency>	PWM frequency. Only carry wave is PULSE wave.

PWM,DEVI	<pwm devi>	Duty cycle deviation. Only carry wave is Pulse Wave.
PWM,MDSP	<mod wave shape>	PWM modulation wave. Only carry wave is PULSE wave.
PWM,SRC	<src>	PWM signal source.
PM,DEVI	<pm phase offset>	PM phase offset. Only PM signal source is set to INT.
ASK,SRC	<src>	ASK signal source.
ASK,KFRQ	<ask key frequency>	ASK key frequency. Only ASK signal source is set to INT.
FSK,KFRQ	<fsk frequency>	FSK frequency. Only FSK signal source is set to INT.
FSK,HFRQ	<fsk hop frequency>	FSK jump frequency
FSK,SRC	<src>	FSK signal source
CARR,WVTP	<wave type>	Value of carrier wave type.
CARR,FRQ	<frequency>	Value of frequency.
CARR,AMP	<amplifier>	Value of amplifier.
CARR,OFST	<offset>	Value of offset.
CARR,SYM	<symmetry>	Value of symmetry.
CARR,DUTY	<duty>	Value of duty cycle. Only Square can set this parameter.
CARR,PHSE	<phase>	Value of phase.
CARR RISE	<rise>	Value of rise.
CARR FALL	<fall>	Value of fall.
CARR DLY	<delay>	Value of delay.

Note: If Carrier wave is Pulse or Noise you can’t set turn on modulate wave.
If you want set AM, FM, PM, CARR and STATE the first parameter have to one of them.

where:
<state>:={ON,OFF}
<src>:= {INT,EXT}
<mod wave shape>:={SINE,SQUARE,TRIANGLE,
UPRAMP,DNRAMP,NOISE,ARB}
<am frequency>:= {0.002Hz to 20000Hz}
<deep>:= {0% to 120%}
<fm frequency>:= {0.002Hz to 20000Hz}
<fm frequency offset> :={0 to basic wave frequency / 2}
<pm frequency> :={ 0.002Hz to 20000Hz}
<pm phase offset>:= {000000to 20
<pwm frequency>:= {0Hz to 4kHz }
<pwm devi>:= {depends on carry wave duty}

<ask key frequency>:= {0.002Hz to 20000Hz}
<fsk frequency>:= {0.002Hz to 50000Hz}
<fsk jump frequency>:= { the same witch basic wave frequency}
<wave type>:={SINE ,SQUARE, RAMP, ARB, PULSE }
<frequency>:= { Default unit is "HZ". Minimum value is 1xe-6 HZ, maximal value depends on the version.}
<amplifier>:={Default unit is "V". Channel one
minimum value 0.004V, Maximal is 6V. Channel two minimum value 0.004V, Maximal is 20V. }
<offset>:={ Default unit is "V".}
<duty>:={ If wave type is square, range is from 20% to 80%.
If wave type is pulse, range is from 0.1% to 99.9%.}
<symmetry>:={ 0% to 100%}
<rise>:={ Maximal is less than width ,Minimal is 6ns.}
<fall>:={ Maximal is less than width ,Minimal is 6ns.}
<delay>:= {default unit is S; minimum 0, maximum is value of Pulse Period}.

QUERY SYNTAX

<channel>:MoDulateWaVe?
<channel>:={C1, C2}

RESPONSE FORMAT

<channel>:MoDulateWaVe <parameter>
<parameter>:={Return all parameter of the current modulate wave parameters.}

(MOD STATE IS ON, AND THE OTHER COMMAND IS VALID)

EXAMPLE

Set channel one modulation type to AM.
C1:MDWV AM

Set modulation shape to AM, and set AM modulating wave type to sine wave.
C1:MDWV AM, MDSP, SINE

Reads channel one modulate wave parameters that STATE is ON.
C1:MDWV?
Return:
C1:MDWV
STATE,ON,AM,MDSP,SINE,SRC,INT,FRQ,100HZ,DEPTH,100,
CARR,WVTP,RAMP,FRQ,1000HZ,AMP,4V,OFST,0V,SYM,50

Reads channel one modulate wave parameters that STATE is OFF.
C1:MDWV?

Return:
C1:MDWV STATE,OFF

Set channel one FM frequency to 1000HZ
C1:MDWV FM, FRQ, 1000HZ

Set the Value of channel one carrier wave shape to SINE.
C1:MDWV CARR,WVTP,SINE

Set the Value of channel one carrier wave frequency to 1000hZ.
C1:MDWV CARR,FRQ,1000HZ

RELATED COMMANDS ARWV, BTWV, CFG, CPL, SWWV, BSWV

1.8. Sweep Wave Command

DESCRIPTION Set or get sweep wave parameters.

COMMAND SYNTAX <channel>:SWeepWaVe <parameter>
<channel>:={C1, C2}
<parameter>:= {a parameter from the table below}

Parameters	Value	Description
STATE	<state>	Turn on or off sweep wave. Note if you want set or read sweep wave other parameter you must turn on the state first.
TIME	<time>	Value of sweep time
STOP	<stop frequency>	Value of stop frequency
START	<start frequency>	Value of start frequency
TRSR	<trigger src>	Trigger source
TRMD	<trigger mode>	Value of trigger output. If TRSR is EXT, the parameter is invalid.
SWMD	<sweep mode>	Sweep way
DIR	<direction>	Sweep direction
EDGE	<edge>	Value of edge. Only TRSR is EXT, the parameter is valid.
MTRIG	<manual trigger>	Make the device once manual trigger. Only TRSR is MAN, the parameter is valid.
CARR,WVTP	<wave type>	Value of carrier wave type.
CARR,FRQ	<frequency>	Value of frequency.
CARR,AMP	<amplifier>	Value of amplifier.
CARR,OFST	<offset>	Value of offset.
CARR,SYM	<symmetry>	Value of symmetry.
CARR,DUTY	<duty>	Value of duty cycle. Only Square can set this parameter.
CARR,PHSE	<phase>	Value of phase.

Note: If Carrier wave is Pulse or Noise you can’t set turn on modulate wave.
If you want set CARR and STATE the first parameter have to one of them.

where: <state>:= {ON|OFF}
<time>:= {0.001S to 500S}

<stop frequency> :={ the same with basic wave frequency}
<start frequency> :={ the same with basic wave frequency}
<trigger src>:= {EXT,INT,MAN}
<trigger mode>:= {ON,OFF}
<sweep mod>:= {LINE,LOG}
<direction>:= {UP,DOWN}
<edge>:={RISE, FALL}
<wave type>:={SINE ,SQUARE, RAMP, ARB}
<frequency>:= { Default unit is "HZ". Minimum value is 1xe-6 HZ, maximal value depends on the version.}
<amplifier>:={Default unit is "V". Channel one
minimum value 0.004V, Maximal is 6V. Channel two minimum value 0.004V, Maximal is 20V. }
<offset>:={ Default unit is "V". maximal value depends on the maximal value depends on the version.}
<duty>:={ 20% to 80%. }
<symmetry>:={ 0% to 100%}

QUERY SYNTAX

<channel>:SWeepWaVe?
<channel>:={C1, C2}

RESPONSE FORMAT

<parameter>:={Return all parameter of the current sweep wave parameters.}

EXAMPLE

Set channel one sweep time to 1 S.
C1:SWWV TIME, 1S

Set channel one sweep stop frequency to 1000hz.
C1: SWWV STOP, 1000HZ

Reads channel one modulate wave parameters that STATE is ON.
C2:SWWV?
Return:
C2:SWWV
STATE,ON,TIME,1S,STOP,100HZ,START,100HZ,TRSR,MAN,
TRMD,OFF,SWMD,LINE,DIR,UP,CARR,WVTP,SQUARE,
FRQ,1000HZ,AMP,4V,OFST,0V,DUTY,50

Reads channel two modulate wave parameters that STATE is OFF.
C2:SWWV?
Return:
C2:SWWV STATE,OFF

1.9. Burst Wave Command

DESCRIPTION Set or get burst wave parameters.

COMMAND SYNTAX <channel>:BursTWaVe <parameter>
<channel>:={C1, C2}
<parameter>:= {a parameter from the table below}

Parameters	Value	Description
STATE	<state>	Turn on or off burst wave. Note if you want set or read burst wave other parameter you must turn on the state first.
PRD	<period>	When carrier wave is NOISE wave, yo u can't set it. When GATE was chose n, you can't set it. And only trig sourc e is IN, you can set it.
STPS	<start phase>	When carrier wave is NOISE or PULS E wave, you can't set it.
GATE_NCYC	<gate ncycle>	When carrier wave is NOISE, you ca n't set it.
TRSR	<trigger>	When carrier wave is NOISE wave, you can't set it. When NCYC was chosen you can set it.
DLAY	<delay>	When carrier wave is NOISE wave, you can't set it. When NCYC was chosen you can't set it.
PLRT	<polarity>	When GATE was chosen you can set it. When carrier wave is NOISE, it is the only parameter.
TRMD	<trig mode>	When carrier wave is NOISE wave, you can't set it. When NCYC was chosen you can set it. When TRSR is set to EXT, you can't set is.
EDGE	<edge>	When carrier wave is NOISE wave, you can't set it. When NCYC was chosen and TRSR is set to EXT, you can set it.
TIME	<circle time>	When carrier wave is NOISE wave, you can't set it. When NCYC was chosen you can set it.
MTRIG		When TRSR's parameter be chosen to MAN, that it can be set.

CARR,WVTP	<wave type>	Value of carrier wave type.
CARR,FRQ	<frequency>	Value of frequency.
CARR,AMP	<amplifier>	Value of amplifier.
CARR,OFST	<offset>	Value of offset.
CARR,SYM	<symmetry>	Value of symmetry.
CARR,DUTY	<duty>	Value of duty cycle. Only Square can set this parameter.
CARR,PHSE	<phase>	Value of phase.
CARR,RISE	<rise>	Value of carrier wave rise edge. Only Carrier wave is Pulse, the Value is valid.
CARR,FALL	<fall>	Value of carrier wave fall edge. Only Carrier wave is Pulse, the Value is valid.
CARR,STDEV	<stdev>	Value of carrier wave stdev. Only Carrier wave is Noise ,the Value is valid.
CARR,MEAN	<mean>	Value of carrier wave mean. Only Carrier wave is Noise ,the Value is valid.
CARR,DLY	<delay>	Parameter is valid only when CARR,WVTP is PULSE.

Note: If you want set CARR and STATE the first parameter have to one of them.

where: <state>:= {ON,OFF}
<period>:= { Default unit is “S ” . 1us to 500s }
<start phase>:= {0 to 360}
<gate ncycle>:= {GATE,NCYC}
<trigger>:= {EXT,INT,MAN}
<delay>:= { Default unit is "S". 0s to 500s }
<polarity>:= {NEG,POS}
<trig mode >:= {RISE,FALL,OFF}
<edge>:= { RISE,FALL}
<circle time> :={ 1cycle to 50000 cycle}
<wave type>:={SINE ,SQUARE, RAMP,PULSE,NOISE, ARB}
<frequency>:= { Default unit is "HZ". Minimum value is 1xe-6 HZ, maximal value depends on the version.}
<amplifier>:={Default unit is "V". Channel one
minimum value 0.004V, Maximal is 6V. Channel two minimum value 0.004V, Maximal is 20V. }
<offset>:={ Default unit is "V". the maximal value depends on the version.}
<duty>:={ If wave type is Square, range is from 20% to 80%
If wave type is pulse, range is from 0.1% to 99.9%}
<symmetry>:={ 0% to 100%}
<rise>:= {Maximal is less than width, minimum value

is 6ns.}
<fall>:= {Maximal is less than width, minimum value
is 6ns.}
<stdev>:={0.4mv to 666.6mv}
<mean>:={ the maximal value depends on the variance}
<delay>:= {default unit is S; minimum 0, maximum is value of Pulse Period}.}

QUERY SYNTAX

<channel>:BursTWaVe? <parameter>
<channel>:={C1, C2}
<parameter>:=<period>······

RESPONSE FORMAT

<channel>:BursTWaVe <type>|<state>|<period>······

EXAMPLE

Set channel one burst wave period to 1S.
C1:BTWV PRD, 1S

Set channel one burst wave delay to 0S
C1:BTWV DLAY, 0S

Reads channel two burst wave parameters that STATE is ON.
C2: BTWV?
Return:
C2:BTWV STATE,ON,PRD,0.01S,STPS,0,TRSR,INT,
TRMD,OFF,TIME,1,DLAY,2.4e-07S,,GATE_NCYC,NCYC,
CARR,WVTP,SINE,FRQ,1000HZ,AMP,4V,OFST,0V,PHSE,0

Reads channel two modulate wave parameters that STATE is OFF.
C2: BTWV?
Return:
C2: BTWV STATE,OFF

1.10. Parameter Copy Command

DESCRIPTION	Copy channel data.

COMMAND SYNTAX	PAraCoPy <destinat channel>, <src channel>
	<destinat channel>:= {C1, C2}
	<src channel>:= {C1, C2}
	Note: the parameters C1 and C2 must be set to device together.

EXAMPLE	Copy parameters from channel one to channel two. PACP C2,C1
RELATED COMMANDS	ARWV, BTWV, CFG, CPL, MDWV, SWWV, BSWV

1.11. Arbitrary Wave Command

DESCRIPTION Change arbitrary wave type.

COMMAND SYNTAX
<channel>:ARbWaVe {INDEX, NAME}
<channel>:={C1, C2}
<index>: 0 to 67 (see blow table what the index number mean.)
<name>: see blow table.

QUERY SYNTAX
<channel>:ARbWaVe?
<channel>:={C1, C2}

RESPONSE FORMAT <channel>:ARbWaVe <index>

EXAMPLE
Set StarUp arbitrary wave output by index.
ARWV INDEX, 2

Reads system current wave.
ARWV?
Return:
ARWV INDEX,2,NAME, Gauspuls

Set Atan arbitrary wave output by name.
ARWV NAME, Cardiac

RELATED COMMANDS BSWV

Index	Name	Index	Name	Index	Name	Index	Name
0	StairUp	10	Sinc	20	SNR	30	Sec
1	StairDn	11	Gaussian	21	Hamming	31	Csc
2	StairUD	12	Dlorentz	22	Hanning	32	Asin
3	Trapezia	13	Haversine	23	Kaiser	33	Acos
4	ExpFall	14	Lorentz	24	Blackman	34	Atan
5	ExpRise	15	Gauspuls	25	GaussiWin	35	ACot
6	LogFall	16	Gmonopuls	26	Harris	36~59	User Store 16K Wave data
7	LogRise	17	Cardiac	27	Bartlett	60~67	User Store 512K

							Wave data
8	Sqrt	18	Quake	28	Tan		
9	X^2	19	TwoTone	29	Cot		

Note: There are two ways to set current arbitrary wave. In one command must use one way to set arbitrary wave.

1.12. Sync Command

DESCRIPTION

Set signal output from backward panel in phase with forward.

COMMAND SYNTAX

<channel>: SYNC <parameter>
<channel>:={C1,C2}
<parameter>:= {ON,OFF}

QUERY SYNTAX

<channel>:SYNC?
<channel>:={C1, C2}

RESPONSE FORMAT

<channel>:SYNC <parameter>

EXAMPLE

Sync function on defend of channel one
C1:SYNC ON

Reads channel one sync state.
C1:SYNC?
Return:
C1:SYNC OFF\n

1.13. Number Format Comand

DESCRIPTION

Set the display Number format.

COMMAND SYNTAX

NBFM <parameter>

<parameter>:={ a parameter from the table below.}

Parameters	Value	Description
PNT	<pnt>	Point format
SEPT	<sept>	Separator format

Where:

<pnt>:={Dot,Comma}.

<sept>:={Space,Off,On}.

QUERY SYNTAX

NBFM?

RESPONSE FORMAT

NBFM <parameter>

EXAMPLE

Set Point format is able belo

NBFM PNT,DOT

Set Separator format is ON

NBFM SEPT，ON

Read data format

NBFM?

Return:

NBFM PNT,DOT,SEPT,ON

1.14. Language Command

DESCRIPTION	Set system display language
COMMAND SYNTAX	LAGG <parameter> <parameter>:={EN,CH}
QUERY SYNTAX	LAGG?
RESPONSE FORMAT	LAGG <parameter>
EXAMPLE	Set language is English LAGG EN Reads language format LAGG? Return: LAGG EN

1.15. Configuration Command

DESCRIPTION	Changes system load data of power on.
COMMAND SYNTAX	Sys_CFG<parameter> <parameter>:= {DEFAULT,LAST}
QUERY SYNTAX	Sys_CFG?
RESPONSE FORMAT	Sys_CFG <parameter>
EXAMPLE	Set system load data of power on to last time data. SCFG LAST

1.16. Buzzer Command

DESCRIPTION	Turns on or off buzzer.
COMMAND SYNTAX	BUZZer <parameter> <parameter>:= {ON,OFF}
QUERY SYNTAX	BUZZer?
RESPONSE FORMAT	BUZZer <parameter>
EXAMPLE	Turns on buzzer. BUZZ ON

1.17. Screen Save Command

DESCRIPTION	Turns on or off Screen Save.
COMMAND SYNTAX	SCreen_SaVe <parameter> <parameter>:= {OFF,1,5,15,30,60,120,300, Unit is minute}
QUERY SYNTAX	SCreen_SaVe?
RESPONSE FORMAT	SCreen_SaVe <parameter>
EXAMPLE	Set screen save time 5 minutes. SCSV 5 Read the current screen save time SCreen_SaVe? Rerurn: SCSV 5

1.18. Clock Source Command

DESCRIPTION	Set or get signal oscillator resource.
COMMAND SYNTAX	ROSCillator <parameter> <parameter>:= {INT, EXT }
QUERY SYNTAX	ROSCillator?
RESPONSE FORMAT	ROSC <parameter>
EXAMPLE	Uses system clock source. ROSC INT SCSV 5

1.19. Store list command

DESCRIPTION

This command used to read the device wave data name if the store unit is empty, the command will Return “EMPTY” string.

Note: M50~ M59 is user defined memory. The name will Return what you defined. is you not defined arbitrary name will turn “EMPTY”.

QUERY SYNTAX

SToreList?

RESPONSE FORMAT

STL M0, StairUp, M1, StairDn, M2, StairUD, M3, Trapezia, M4, ExpFall, M5, ExpRise, M6, LogFall, M7, LogRise, M8, Sqrt, M9, X^2, M10, Sinc, M11, Gaussian, M12, Dlorentz, M13, Haversine, M14, Lorentz, M15, Gauspuls, M16, Gmonopuls, M17, Cardiac, M18, Quake, M19, TwoTone, M20, SNR, M21, Hamming, M22, Hanning, M23, Kaiser, M24, Blackman, M25, GaussiWin, M26, Harris, M27, Bartlett, M28, Tan, M29, Cot, M30, Sec, M31, Csc, M32, Asin, M33, Acos, M34, Atan, M35, ACot, M36, EMPTY, M37, EMPTY, M38, EMPTY, M39, EMPTY, M40, EMPTY, M41, EMPTY, M42, EMPTY, M43, EMPTY, M44, AS EMPTY IN, M45, EMPTY, M46, EMPTY, M47, EMPTY, M48, EMPTY, M49, EMPTY, M50, EMPTY, M51, EMPTY, M52, EMPTY, M53, EMPTY, M54, EMPTY, M55, EMPTY, M56, EMPTY, M57, EMPTY, M58, EMPTY, M59, EMPTY, M60, EMPTY, M61, EMPTY, M62, EMPTY, M63, EMPTY, M64, EMPTY, M65, EMPTY, M66, EMPTY, M67, EMPTY

EXAMPLE

Read device memory saved arbitrary data.
STL?
Return:
STL M0, StairUp, M1, StairDn, M2, StairUD, M3, Trapezia, M4, ExpFall, M5, ExpRise, M6, LogFall, M7, LogRise, M8, Sqrt, M9, X^2, M10, Sinc, M11, Gaussian, M12, Dlorentz, M13, Haversine, M14, Lorentz, M15, Gauspuls, M16, Gmonopuls, M17, Cardiac, M18, Quake, M19, TwoTone, M20, SNR, M21, Hamming, M22, Hanning, M23, Kaiser, M24, Blackman, M25, GaussiWin, M26, Harris, M27, Bartlett, M28, Tan, M29, Cot, M30, Sec, M31, Csc, M32, Asin, M33, Acos, M34, Atan, M35, ACot, M36, EMPTY, M37, DSFDSFDWAVE2DFSDFWSDF, M38, HJRGFKSGKUWAVE3, M39, SDASIN, M40, LOGRISE, M41, STAIRUP, M42, SDEXPFALL, M43, ACOT, M44, ASIN, M45, SDSTAIRDN, M46, STAIRUD, M47, ASTRAPEZIA, M48,

EXPRISE, M49, ASDLOGFALL, M50, LORENTZ, M51, GAUSPULS, M52, GMONOPULS, M53, CARDIAC, M54, QUAKE, M55, TWOTONE, M56, SNR, M57, HAMMING, M58, WAVE7, M59, WAVE12, M60, WAVE11, M61, WAVE10, M62, WAVE9, M63, WAVE8, M64, EMPTY, M65, EMPTY, M66, EMPTY, M67, EMPTY

1.20. Virtual key command

DESCRIPTION

The Command is to send key word to device.

COMMAND SYNTAX

VirtualKEY VALUE,<value>,STATE,<sate>
<value>:= {a parameter from the table below. }
<state>:=<0,1>(The “1” is effective to virtual value, and the “0” is useless)

EXAMPLE

VKEY VALUE,15,STATE,1

VKEY VALUE,KB_SWEEP,STATE,1

KB_FUNC1	28	KB_NUMBER_4	52
KB_FUNC2	23	KB_NUMBER_5	53
KB_FUNC3	18	KB_NUMBER_6	54
KB_FUNC4	13	KB_NUMBER_7	55
KB_FUNC5	8	KB_NUMBER_8	56
KB_FUNC6	3	KB_NUMBER_9	57
KB_MOD	15	KB_POINT	46
KB_SWEEP	16	KB_NEGATIVE	43
KB_BURST	17	KB_LEFT	44
KB_WAVES	4	KB_RIGHT	40
KB_UTILITY	11	KB_OUTPUT1	153
KB_PARAMETER	5	KB_OUTPUT2	152
KB_NUMBER_0	48	KB_KNOB_RIGHT	175
KB_NUMBER_1	49	KB_KNOB_LEFT	177
KB_NUMBER_2	50	KB_KNOB_DOWN	176
KB_NUMBER_3	51	KB_HELP	12
		KB_CHANNEL	33

1.21. Index