UNI-T

Programming Manual

UTG9000T Seris Programmable Signal Manual

January 1, 2020

UNI-Trend Technology (China) Co., Ltd

Warranty and Statement

Copyright

Brand Information

UNI-T is the registered trademark of Uni-Trend Technology (China) Co., Ltd.

Software Version

00.00.01

Software upgrade may have some change and add more function, please subscribe **UNI-T** website to get the new version or contact **UNI-T.**

Statement

- UNI-T products are protected by patents (including obtained and pending) in China and other countries and regions.
- UNI-T reserves the right to change specifications and prices.
- The information provided in this manual supersedes all previous publications.
- Information provided in this manual is subject to change without prior notice.
- UNI-T shall not be liable for any errors that may be contained in this manual. For any incidental or consequential damages, arising out of the use or the information and deductive functions provided in this manual.
- Without the written permission of UNI-T, this manual cannot photocopied, reproduced or adapted.

Product Certification

UNI-T has certified that the product conforms to China national product standard and industry product standard as well as ISO9001:2008 standard and ISO14001:2004 standard. UNI-T will go further to certificate product to meet the standard of other member of the international standards organization.

Contact Us

If you have any question or problem, you can contact us,

Website：https://www.uni-trend.com

SCPI

SCPI was defined as an additional layer on top of the IEEE 488.2-1987 specification "Standard Codes, Formats, Protocols, and Common Commands".The standard specifies a common syntax, command structure, and data formats, to be used with all instruments. It introduced generic commands (such as CONFigure and MEASure) that could be used with any instrument. These commands are grouped into subsystems. SCPI also defines several classes of instruments. For example, any controllable power supply would implement the same DCPSUPPLY base functionality class. Instrument classes specify which subsystems they implement, as well as any instrument-specific features.

The physical hardware communications link is not defined by SCPI. While it was originally created for the IEEE-488.1 (GPIB) bus, SCPI can also be used with RS-232, RS-422, Ethernet, USB, VXIbus, HiSLIP, etc.

SCPI commands are ASCII textual strings, which are send to the instrument over the physical layer (e.g., IEEE-488.1). Commands are a series of one or more keywords, many of which take parameters. In the specification, keywords are written CONFigure: The entire keyword can be used, or it can be abbreviated to just the uppercase portion. Responses to query commands are typically ASCII strings. However, for bulk data, binary formats can be used.

This section introduces the format, symbols, parameters, and abbreviations of the SCPI command.

Instruction Format

The SCPI command is a tree-like hierarchy consisting of multiple subsystems, each consisting of a root keyword and one or more hierarchical key words. **The command line usually begins with a colon ":"; Keywords are separated by the colon ":", followed by optional parameter settings. The command keyword is separated by spaces from the first parameter. The command string must end with a newline <NL> character. Add the question mark "? " after the command line. It is usually indicated that this feature is being queried.**

Symbol Description

The following four symbols are not part of SCPI command, it cannot send with the command. It usually used as supplementary description of command parameter.

- **Brace { }** usually contains multiple optional parameters, it should select one parameter when send command.

Such as DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE} command

- **Vertical Bar |** used to separated multiple parameters, it should select one parameter when send command.

  Such as DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE} command

- **Square Brackets [ ]** the contents in square brackets (command keywords) can omissible. If the parameter is ignoret, the instrument will set the parameter as the default value.

  Such as MEASure:NDUTy? [<source>] command, it presents current channel

- **Triangular Brackets < >** The parameter in the brackets must be replaced with a valid value.

  Such as use DISPlay:GRID:BRIGhtness 30 form to send DISPlay:GRID:BRIGhtness <count> command.

Parameter Description

The parameter in this manual can divide into five types: Boolean, Integer, Real, Discrete, ASCII string

- **Boolean**

  Parameter value can set “ON” (1) or “OFF” (0)

  Such as SYSTem:LOCK {{1 | ON} | {0 | OFF}}

- **Integer**

  Parameter can take any valid integer value unless there have some other descriptions.

  Such as command: DISPlay:GRID:BRIGhtness <count> , parameter of <count> can take integer from 0~100

  Note: Do not set decimal as parameter, otherwise it may occur error.

- **Real**

  Parameter can take any valid integer value unless there have some other descriptions.

  Such as for command CH1, CHANnell: OFFSet <offset> , parameter of <offset> can take integer value.

- **Discrete**

  Parameter can only take some specified numbers or characters.

  Such as command DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE} parameter can only take FULL, GRID, CROSS, NONE

- **ASCII Character String**

  String parameter contain all ASCII string sets. Strings must begin and end with paired quotes; it can use single or double quotation marks. The quotation and delimiter can also be part of a string by typing it twice and not adding any characters.

  Such as set IP SYST:COMM:LAN:IPAD "192.168.1.10"

Shorthand Rule

All command can identify capital and small letter, if command need enter shorthand, it should be all capital letter.

Data Return

Data return is divided into single data and batch data. The single data return is the corresponding parameter type, in which the real return type is presents by the scientific notation method. The part before e retains three figure behind the decimal point, and the e part retains three figure; the batch return must be obey IEEE 488.2# string data format, '#'+ the length of character bits[fixed to one character]+ ASCII valid value+ valid data+ end string['\n']

Such as #3123xxxxxxxxxxxxxxxxxxx\n presents 123 strings batch data return format, '3' presents "123" occupies three character bits.

SCPI Command

IEEE488.2 Common Command

*IDN?

- **Command format:**

  *IDN?

- **Functional description：**

  For query manufacture name, model, product serial number and software version.

- **Return format:**

  Manufacture name, model, product serial number, software version separated by dot mark.

  Notes: returned model information must be the same with nameplate.

- **For example:**

  UNI-T Technologies, UTG9000T, 000000001, 00.00.01

*RST

- **Command format:**

  *RST

- **Functional description：**

  Restore factory settings and clear the entire error message, send and receive queue buffers.

SYSTem Command

The command used for the basic operation of signal source, including lock full qwerty and system parameter setup.

:SYSTem:INFo?

- Command format:
  :SYSTem:INFo?
- Functional description:
  For query built-in version number of communication protocol- system information.
- Return format:
  {protocol version number},{software version number },{hardware version number },{instrument model},{model internal code},{bandwidth},{sampling rate},{channel quantity},{display name},{channel information},{serial number}.
  Explanation: separated by dot mark, fill in the specified information by position.
- For example:
  :SYSTem:INFo? 1,1.02.3,1.00.1,DSO,BG,100MHz,1GSa/s,2CH,UTD7000BG,**,12345678

  Detailed description of return format:
  - **1: Protocol version number**
    Built-in version number, each time correction need to progressive increase, to maintain integer value instead of 1.02.3 traditional mode. Management system estimate whether is upgrade or hint some function is enable by this version number.
  - **1.02.3: Software version number**
  - **1.00.1: Hardware version number**
  - **DSO: Instrument model**
    For example: DSO present oscilloscope
  - **BG: built-in model code, each serial model has fixed model code, for upper computer to distinguish.**
    For example: BG presents UTD7000BG, UPO0001 presents UPO3000CS series, UPO0002 presents new UPO2000CS series
  - **100MHz: bandwidth**
    For example: 100MHz
  - **1GSa/s : samoling rate**
    For example: 1GS/s
  - **2CH: channel quantity**
    For example: 2 channel
  - **UTG7000BG: display name**
    For example: UTG7000BG
  - **** : channel information**
    For example: ** presents export sales, EN presents educational channel
  - **12345678 : serial number**

✧ **For example： 12345678**

:SYSTem:LOCK

- Command format:

  :SYSTem:LOCK {{1 | ON} | {0 | OFF}}

  :SYSTem:LOCK?

- Functional description:

  For lock or unlock full qwerty.

- Return format:

  Query return full qwerty locked status, 0 presents unlock, 1 presents locke.

- For example:

  :SYSTem:LOCK ON full qwerty locked

  :SYSTem:LOCK OFF unlock full qwerty

  :SYSTem:LOCK? query return 1, it presents locked

:SYSTem:CONFigure

- Command format:

  :SYSTem:CONFigure <file>

  :SYSTem:CONFigure?

- Functional description:

  For read and write configuration file, send this command firstly, then send configuration file data to singal source.

  <file> presents configuration file.

- Return format:

  Query return the current configuration file data of signal source.

- For example:

  :SYSTem:CONFigure written configuration file data in signal source and load it

  :SYSTem:CONFigure? query return binary system of the current configuration file data of signal source

:SYSTem:PHASe:MODe

- Command format:

  :SYSTem:PHASe:MODe {INDependent | SYNChronization}

  :SYSTem:PHASe:MODe?

- Functional description:

  Phase mode between in channel, if it is sync mode, it presents the initial phase of two channels is sync.Otherwise, phase is independent.

- Return format:

  Query return phase mode between in channel.

- For example:

  :SYSTem:PHASe:MODe INDependent set independent phase mode between in channel

:SYSTem:PHASe:MODe? query returnINDependent

:SYSTem:LANGuage

- Command format:

  :SYSTem:LANGuage {ENGLish|CHINese}

  :SYSTem:LANGuage?

- Functional description:

  Control system language.

- Return format:

  Query return system display language.

- For example:

  :SYSTem:LANGuage ENGLish set system language as English

  :SYSTem:LANGuage? query returnENGLish

:SYSTem:BEEP

- Command format:

  :SYSTem:BEEP {{1 | ON} | {0 | OFF}}

  :SYSTem:BEEP?

- Functional description:

  Control beeper switch.

- Return format:

  Query return the status of beeper switch.

- For example:

  :SYSTem:BEEP ON turn on beeper

  :SYSTem:BEEP? query return1

:SYSTem:NUMBer:FORMat

- Command format:

  :SYSTem:NUMBer:FORMat {COMMa|SPACe|NONe}

  :SYSTem:NUMBer:FORMat?

- Functional description:

  Control separator of system number format.

- Return format:

  Query return separator of system number format.

- For example:

  :SYSTem:NUMBer:FORMat NONe set no system number format

  :SYSTem:NUMBer:FORMat? query return NONe

:SYSTem:BRIGhtness

- Command format:

  :SYSTem:BRIGhtness { 30|40|50|60|70|80|90|100}

  :SYSTem:BRIGhtness?

- Functional description:

  Control system backlight brightness.

- Return format:

  Query return backlight brightness.

- For example:

  :SYSTem:BRIGhtness 30 set system backlight brightness as 30%

  :SYSTem:BRIGhtness? query return30

:SYSTem:SLEEP:TIMe

- Command format:

  :SYSTem:SLEEP:TIMe { CLOSe | 1 |5 | 15 |30 |60}

  :SYSTem:SLEEP:TIMe?

- Functional description:

  Control system sleeping time, unit is minute.

- Return format:

  Query return sleeping time.

- For example:

  :SYSTem:SLEEP:TIMe 1 set automatic sleeping time at 1 minute

  :SYSTem:SLEEP:TIMe? query return 1

:SYSTem:CYMometer

- Command format:

  :SYSTem:CYMometer {{1 | ON} | {0 | OFF}}

  :SYSTem:CYMometer?

- Functional description:

  Control the switch of frequency meter.

  Notes：when this function is operating, sync output function will be turned off.

- Return format:

  Query return the switch staus of frequency meter，0 presents off, 1 presents on.

- For example:

  :SYSTem:CYMometer ON turn on frequency meter

  :SYSTem:CYMometer? query return1

:SYSTem:CYMometer:FREQuency?

- Command format:

  :SYSTem:CYMometer:FREQuency?

- Functional description:

  Acquire the current measured frequency of frequency meter.

- Return format:

  Query return the current measured frequency of frequency meter in Hz，return data by scientific notation method.

- For example:

  :SYSTem:CYMometer:FREQuency? query return 2e+3

:SYSTem:CYMometer:PERiod?

- Command format:

  :SYSTem:CYMometer:PERiod?

- Functional description:

  Acquire the current measured period of frequency meter.

- Return format:

  Query return the current measured period of frequency meter in s, return data by scientific notation method.

- For example:

  :SYSTem:CYMometer:PERiod? query return 2e-3

:SYSTem:CYMometer:DUTY?

- Command format:

  :SYSTem:CYMometer:DUTY?

- Functional description:

  Acquire the current measured duty ratio of frequency meter.

- Return format:

  Query return the current measured duty ratio of frequency meter in %.

- For example:

  :SYSTem:CYMometer:DUTY? query return 20，it presents duty ratio is 20%

:SYSTem:CYMometer:PWIDTh?

- Command format:

  :SYSTem:CYMometer: PWIDTh?

- Functional description:

  Acquire the current measured positive pulse width of frequency meter.

- Return format:

Query return the current measured positive pulse width of frequency meter in s.

- For example:

  :SYSTem:CYMometer: PWIDTh? query return 1e-3，it presents duty ratio is 1 ms.

:SYSTem:CYMometer:NWIDTh?

- Command format:

  :SYSTem:CYMometer: NWIDTh?

- Functional description:

  Acquire the current measured negative pulse width of frequency meter.

- Return format:

  Query return the current measured negative pulse width of frequency meter in s.

- For example:

  :SYSTem:CYMometer: NWIDTh? query return 1e-3，it presents duty ratio is 1 ms.

CHANnel Command

This command is to set channel function of signal source.

:CHANnel<n>:MODE

- Command format:

  :CHANnel<n>:MODE {CONTinue | MODulation| SWEep| BURSt }

  :CHANnel<n>:MODE?

- Functional description:

  Set signal mode of the specified channel, they are CONTinue, MODulation, SWEep and BURSt.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return signal mode of the specified channel.

- For example:

  :CHANnel1:MODe AM　　set channel 1 signal mode as amplitude modulating output

  :CHANnel1:MODe?　　query return AM

:CHANnel<n>:OUTPut

- Command format:

  :CHANnel<n>:OUTPut　{{1 | ON} | {0 | OFF}}

  :CHANnel<n>:OUTPut?

- Functional description:

  Set the switch of the specified channel output.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the status of the specified channel output，0 presents off,，1 presents on.

- For example:

  :CHANnel1:OUTPut ON　　turn on channel 1 output

  :CHANnel1:OUTPut?　　query return1

:CHANnel<n>:INVersion

- Command format:

  :CHANnel<n>:INVersion　{{1 | ON} | {0 | OFF}}

  :CHANnel<n>:INVersion?

- Functional description:

  Set the switch of the specified channel inversion.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the status of the specified channel inversion，0 presents off,，1 presents on.

- For example:

:CHANnel1:INVersion ON　　turn on channel 1 inversion output

:CHANnel1:INVersion?　　query return1

:CHANnel<n>:OUTPut:SYNC

- Command format:

  :CHANnel<n>:OUTPut:SYNC {{1 | ON} | {0 | OFF}}

  :CHANnel<n>:OUTPut:SYNC?

- Functional description:

  Set channel sync output status.

  Notes: the instrument has only one sync output interface and can only turn on one channel sync output.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the output status of the specified channel, 0 presents off,, 1 presents on.

- For example:

  :CHANnel1:OUTPut:SYNC ON　　turn on sync output of channel 1

  :CHANnel1:OUTPut:SYNC?　　query return1

:CHANnel<n>:OUTPut:SYNC:INVersion

- Command format:

  :CHANnel<n>:OUTPut:SYNC:INVersion {{1 | ON} | {0 | OFF}}

  :CHANnel<n>:OUTPut:SYNC:INVersion?

- Functional description:

  Set channel sync inversion output.

  Notes: the instrument has only one sync output interface and can only turn on one channel sync output.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return inversion status of the specified channel, 0 presents off,, 1 presents on.

- For example:

  :CHANnel1:OUTPut:SYNC:INVersion ON　　set sync inversion output of channel 1

  :CHANnel1:OUTPut:SYNC:INVersion?　　query return1

:CHANnel<n>:LIMit:ENABle

- Command format:

  :CHANnel<n>:LIMit:ENABle　{{1 | ON} | {0 | OFF}}

  :CHANnel<n>:LIMit:ENABle?

- Functional description:

  Set the switch of amplitude-limiting of the specified channel.

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

  Query return amplitude limiting status of the specified channel.

- For example：

  :CHANnel1:LIMit:ENABle ON　　turn on channel 1 amplitude limiting

  :CHANnel1:LIMit:ENABle?　　query return1

:CHANnel<n>:LIMit:LOWer

- Command format：

  :CHANnel<n>:LIMit:LOWer {<voltage>}

  :CHANnel<n>:LIMit:LOWer?

- Functional description：

  Set the lower limit of the amplitude limiting of the specified channel.

  <voltage>　unit is the specified channel channel unit.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

  Query return the lower limit of the amplitude limiting of the specified channel, return by scientific notation method.

- For example：

  :CHANnel1:LIMit:LOWer 2　　set the lower limit as 2V

  :CHANnel1:LIMit:LOWer?　　query return 2e+0

:CHANnel<n>:LIMit:UPPer

- Command format：

  :CHANnel<n>:LIMit:UPPer　{<voltage>}

  :CHANnel<n>:LIMit:UPPer?

- Functional description：

  Set the upper limit of the amplitude limiting of the specified channel.

  <voltage> unit is the specified channel channel unit.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

  Query return the upper limit of the amplitude limiting of the specified channel, return by scientific notation method.

- For example：

  :CHANnel1:LIMit:UPPer 2　　set the upper limit as 2V

  :CHANnel1:LIMit:UPPer?　　query return2e+0

:CHANnel<n>:AMPLitude:UNIT

- Command format：

:CHANnel<n>:AMPLitude:UNIT {VPP | VRMS | DBM}

:CHANnel<n>:AMPLitude:UNIT?

- Functional description:

Set the amplitude unit of the specified channel output.

<n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

Query return the amplitude unit of the specified channel output.

- For example:

:CHANnel1:AMPLitude:UNIT VPP set channel 1 amplitude output unit as VPP

:CHANnel1:AMPLitude:UNIT? query returnVPP

:CHANnel<n>:LOAD

- Command format:

:CHANnel<n>:LOAD <resistance>

:CHANnel<n>:LOAD?

- Functional description:

Set the load of the specified channel output.

<resistance> presents load resistance value, unit is Ω.

<n>: channel number, n take value as 1, 2, 3, 4.

Notes: resistance value 1~10000, 10000 is correspondingto high resistance.

- Return format:

Query return the load resistance value of the specified channel output, return by scientific notation method.

- For example:

:CHANnel1:LOAD 50 set channel 1 load output as 50Ω

:CHANnel1:LOAD? query return50e+0

:CHANnel<n>:PNCode

- Command format:

:CHANnel<n>:PNCode <code>

:CHANnel<n>:PNCode?

- Functional description:

Set the specified channel PN code, this command is only valid for waveform with PN code function.

<code>: presents PN code, for example:

{PN3|PN5|PN7|PN9|PN11|PN13|PN15|PN17|PN21|PN23|PN25|PN27|PN29|PN31|PN33}

<n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

Query return the specified channel PN code.

- For example:

:CHANnel1:PNCode PN9 set channel 1 PN code as PN9

:CHANnel1:PNCode? query return PN9

:CHANnel<n>:TRIGger:SOURce

- Command format:

  :CHANnel<n>:TRIGger:SOURce {INTernal|EXTRise|EXTFall|MANUal}

  :CHANnel<n>:TRIGger:SOURce?

- Functional description:

  Set trigger source of the specified channel, this command is for sweep frequency and burst mode.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the trigger source of the specified channel.

- For example:

  :CHANnel1:TRIGger:SOURce INTernal set channel 1 as internal trigger source

  :CHANnel1:TRIGger:SOURce? query return INTernal

:CHANnel<n>:TRIGger:OUTPut

- Command format:

  :CHANnel<n>:TRIGger:OUTPut {CLOSe|RISe|FALL}

  :CHANnel<n>:TRIGger:OUTPut?

- Functional description:

  Set tigger output mode of the specified channel, this command is for sweep frequency and burst.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return tigger output mode of the specified channel.

- For example:

  :CHANnel1:TRIGger:OUTPut RISe set channel 1 trigger output mode as rise edge trigger

  :CHANnel1:TRIGger:OUTPut? query returnRISe

:CHANnel<n>:NS

- Command format:

  :CHANnel<n>:NS {{1 | ON} | {0 | OFF}}

  :CHANnel<n>:NS?

- Functional description:

  Set the switch of channel noise superposition, this command is only valid for waveform with noise superposition function.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the output status of the specified channel, 0 presents off, 1 presents on.

- For example:

  :CHANnel1:NS ON　　turn on channel 1 noise superposition

  :CHANnel1:NS?　　query return1

:CHANnel<n>:NS:SNR:UNIT

- Command format:

  :CHANnel<n>:NS:SNR:UNIT　{DBM | RATIO }

  :CHANnel<n>:NS:SNR:UNIT?
- Functional description:

  Set signal-to-noise ratio unit of channel noise superposition, this command is only valid for waveform with noise superposition function.

  <n>：channel number，n take value as 1, 2, 3, 4.
- Return format:

  Query return signal-to-noise ratio unit of noise superposition of the specified channel.
- For example:

  :CHANnel1:NS:SNR:UNIT DBM　　set signal-to-noise ratio unit as dB

  :CHANnel1:NS:SNR:UNIT?　　query return DBM

:CHANnel<n>:NS:SNR

- Command format:

  :CHANnel<n>:NS:SNR　<value>

  :CHANnel<n>:NS:SNR?
- Functional description:

  Set signal-to-noise ratio of channel noise superposition, this command is only valid for waveform with noise superposition function.

  <n>：channel number，n take value as 1, 2, 3, 4.
- Return format:

  Query return signal-to-noise ratio of channel noise superposition, return by scientific notation method.
- For example:

  :CHANnel1:NS:SNR 3　　set channel 1 noise superposition signal-to-noise ratio as 3dB

  :CHANnel1:NS:SNR?　　query return 3e+0

:CHANnel<n>:MERge

- Command format:

  :CHANnel<n>:MERge {{1 | ON} | {0 | OFF}}

  :CHANnel<n>:MERge?
- Functional description:

  Set the merging switch of the specified channel.

Channel 1, 2 merging channel only can output from channel 1, 2.

Channel 3, 4 merging channel only can output from channel 3, 4.

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return channel merging status of the specified channel.

- For example：

:CHANnel1:MERge ON　　channel 1,2 channel merging signal output from channel 1

:CHANnel1:MERge?　　return 1

:CHANnel:COUPle<m>:FREQuency

- Command format：

:CHANnel:COUPle<m>:FREQuency　{{1 | ON} | {0 | OFF}}

:CHANnel:COUPle<m>:FREQuency?

- Functional description：

Set the channel frequency coupling switch, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

<m>：channel number，m take value as 1, 2.

1 presents channel 1 coupling with channel 2；2 presents channel 3 coupling with channel 4.

- Return format：

Query return channel coupling status，0 presents off,，1 presents on.

- For example：

:CHANnel:COUPle1:FREQuency ON　　turn on channel 1, 2 channel coupling

:CHANnel:COUPle1:FREQuency?　　return 1

:CHANnel:COUPle<m>:FREQuency:SCALe

- Command format：

:CHANnel:COUPle<m>:FREQuency:SCALe　<scale>

:CHANnel:COUPle<m>:FREQuency:SCALe?

- Functional description：

Set channel coupling frequency ratio, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

<scale >：coupling frequency

<m>：channel number，m take value as 1, 2.

1 presents channel 1 coupling with channel 2；2 presents channel 3 coupling with channel 4.

- Return format：

Query return channel coupling frequency ratio, return by scientific notation method.

- For example：

:CHANnel:COUPle1:FREQuency:SCALe 0.1

set frequency ratio of channel 2 and channel 1 as 0.1

:CHANnel:COUPle1:FREQuency:SCALe? return 1e-1

:CHANnel:COUPle<m>:FREQuency:DEV

- Command format:

  :CHANnel:COUPle<m>:FREQuency:DEV <dev >

  :CHANnel:COUPle<m>:FREQuency:DEV?

- Functional description:

  Set channel coupling frequency deviation, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

  <scale >：coupling frequency deviation，in Hz.

  <m>：channel number，m take value as 1, 2.

  1 presents channel 1 coupling with channel 2；2 presents coupling with channel 4.

- Return format:

  Query return channel coupling frequency deviation，return by scientific notation method.

- For example:

  :CHANnel:COUPle1:FREQuency:DEV 100

  set coupling frequency ratio of channel 2 and channel 1 as 100Hz

  :CHANnel:COUPle1:FREQuency:DEV? return 1e+2

:CHANnel:COUPle<m>:PHASe

- Command format:

  :CHANnel:COUPle<m>:PHASe {{1 | ON} | {0 | OFF}}

  :CHANnel:COUPle<m>:PHASe?

- Functional description:

  Set channel coupling phase, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

  <m>：channel number，m take value as 1, 2.

  1 presents channel 1 coupling with channel 2；2 presents channel 3 coupling with channel 4.

- Return format:

  Query return channel coupling phase status，0 presents off, 1 presents on.

- For example:

  :CHANnel:COUPle1:PHASe ON turn on channel 1, 2 coupling

  :CHANnel:COUPle1:PHASe? return 1

:CHANnel:COUPle<m>:PHASe:SCALe

- Command format:

:CHANnel:COUPle<m>:PHASe:SCALe <scale>

:CHANnel:COUPle<m>:PHASe:SCALe?

- Functional description:

Set channel coupling phase ratio, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

<scale >: coupling phase ratio.

<m>: channel number, m take value as 1, 2.

1 presents channel 1 coupling with channel 2; 2 presents channel 3 coupling with channel 4.

- Return format:

Query return channel coupling phase ratio, return by scientific notation method.

- For example:

:CHANnel:COUPle1:PHASe:SCALe 0.1

set frequency ratio of channel 2 and channel 1 as 0.1

:CHANnel:COUPle1:PHASe:SCALe? return 1e-1

:CHANnel:COUPle<m>:PHASe:DEV

- Command format:

:CHANnel:COUPle<m>:PHASe:DEV <dev >

:CHANnel:COUPle<m>:PHASe:DEV?

- Functional description:

Set channel coupling phase deviation, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

<scale >: coupling phase deviation, unit °.

<m>: channel number, m take value as 1, 2.

1 presents channel 1 coupling with channel 2; 2 presents channel 3 coupling with channel 4.

- Return format:

Query return channel coupling phase deviation, return by scientific notation method.

- For example:

:CHANnel:COUPle1:PHASe:DEV 100

set coupling phase deviation of channel 2 and channel 1 as 100°.

:CHANnel:COUPle1:PHASe:DEV? return 1e+2

:CHANnel:COUPle<m>:AMPLitude

- Command format:

:CHANnel:COUPle<m>:AMPLitude {{1 | ON} | {0 | OFF}}

:CHANnel:COUPle<m>:AMPLitude?

- Functional description:

Set channel amplitude coupling switch, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

<m>：channel number，m take value as 1, 2.

1 presents channel 1 coupling with channel 2；2 presents channel 3 coupling with channel 4.

- Return format：

Query return status of channel coupling amplitude switch，0 presents off,，1 presents on.

- For example：

:CHANnel:COUPle1:AMPLitude ON　　turn on channel 1, 2 coupling amplitude

:CHANnel:COUPle1:AMPLitude?　　return 1

:CHANnel:COUPle<m>:AMPLitude:SCALe

- Command format：

:CHANnel:COUPle<m>:AMPLitude:SCALe　<scale>

:CHANnel:COUPle<m>:AMPLitude:SCALe?

- Functional description：

Set channel coupling amplitude ratio, there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

<scale >：coupling amplitude ratio.

<m>：channel number，m take value as 1, 2.

1 presents channel 1 coupling with channel 2；2 presents channel 3 coupling with channel 4.

- Return format：

Query return channel coupling amplitude ratio，return by scientific notation method.

- For example：

:CHANnel:COUPle1:AMPLitude:SCALe 0.1

set channel coupling of channel 2 and channel 1 as 0.1

:CHANnel:COUPle1:AMPLitude:SCALe?　　return 1e-1

:CHANnel:COUPle<m>:AMPLitude:DEV

- Command format：

:CHANnel:COUPle<m>:AMPLitude:DEV　<dev >

:CHANnel:COUPle<m>:AMPLitude:DEV?

- Functional description：

Set channel coupling amplitude deviation，there are two channel coupling mode: channel 1 coupling with channel 2, channel 3 coupling with channel 4.

<scale >：coupling amplitude deviation, unit V.

<m>：channel number，m take value as 1, 2.

1 presents channel 1 coupling with channel 2；2 presents channel 3 coupling with channel 4.

- Return format：

Query return channel coupling amplitude deviation, return by scientific notation method.

- For example：

:CHANnel:COUPle1:AMPLitude:DEV 100

set channel coupling amplitude of channel 2 and channel 1 as 100Hz

:CHANnel:COUPle1:AMPLitude:DEV? return 1e+2

Continuity

:CHANnel<n>:BASE:WAVe

- Command format:

  :CHANnel<n>:BASE:WAVe { SINe | SQUare | PULSe | RAMP | ARB | NOISe | DC | HARMonic | PRBS }

  :CHANnel<n>:BASE:WAVe?

- Functional description:

  Set the fundamental waveform type of the specified channel. They are sine wave, square wave, pulse wave, triangle wave, arbitrary wave, noise, DC, harmonic wave and pseudo-random binary sequence.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the fundamental waveforn type of the specified channel.

- For example:

  :CHANnel1:BASE:WAVe SINe　　set channel 1 fundamental waveform as sine wave

  :CHANnel1:BASE:BWAVe?　　query returnSINe

:CHANnel<n>:BASE:FREQuency

- Command format:

  :CHANnel<n>:BASE:FREQuency {<freq>}

  :CHANnel<n>:BASE:FREQuency?

- Functional description:

  Set the specified channel frequency output.

  <freq> presents frequency value, in Hz（1e-6s ~ the maximum allowable frequency of the current waveform）

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the specified channel frequency output，return by scientific notation method.

- For example:

  :CHANnel1:BASE:FREQuency 2000　　set channel 1 frequency output as 2KHz

  :CHANnel1:BASE:FREQuency?　　Query return2e+3

:CHANnel<n>:BASE:PERiod

- Command format:

  :CHANnel<n>:BASE:PERiod { <period>}

  :CHANnel<n>:BASE:PERiod?

- Functional description:

  Set output periode of the specified channel.

  <period> unit is S.

  If it is sine wave: range（the current allowance time tops ~ 1e3s）

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the upper limit of amplitude limit, return by scientific notation method.

- For example:

  :CHANnel1:BASE:PERiod 0.002 set channel 1 output period as 2ms

  :CHANnel1:BASE:PERiod? query return 2e-3

:CHANnel<n>:BASE:PHASe

- Command format:

  :CHANnel<n>:BASE:PHASe { <phase>}

  :CHANnel<n>:BASE:PHAse?

- Functional description:

  Set output phase of the specified channel.

  <phase> unit°, range -360~360.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return output phase of the specified channel.

- For example:

  :CHANnel1:BASE:PHAse 20 set channel 1 output phase as 20°

  :CHANnel1:BASE:PHAse? query return 20

:CHANnel<n>:BASE:AMPLitude

- Command format:

  :CHANnel<n>:BASE:AMPLitude { <amp>}

  :CHANnel<n>:BASE:AMPLitude?

- Functional description:

  Set output amplitude of the specified channel.

  <amp> presents voltage, unit is the current channel unit. 1mVpp ~ the maximum output under the current load.

  If the current unit is VPP， the maximum output under the current load = the current load*20/(50+ the current load)

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return output amplitude of the specified channel，return by scientific notation method.

- For example:

  :CHANnel1:BASE:AMPLitude 2　　set channel 1 output amplitude as 2V

  :CHANnel1:BASE:AMPLitude?　　query return 2e+0

:CHANnel<n>:BASE:OFFSet

- Command format:

  :CHANnel<n>:BASE:OFFSet { <voltage>}

  :CHANnel<n>:BASE:OFFSet?

- Functional description:

  Set output DC offset of the specified channel.

  <voltage> unit is V. range：0~± the maximum DC under the current load

  the maximum DC under the current load = the current load *10/(50+ the current load) –the minimum AC under the current load /2;

  minimum AC value is 2mVpp, DC mode is 0;

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return DC offset of the specified channel，return by scientific notation method.

- For example:

  :CHANnel1:BASE:OFFSet 2　　set channel 1 output DC offset as 2V

  :CHANnel1:BASE:OFFSet?　　query return 2e+0

:CHANnel<n>:BASE:HIGH

- Command format:

  :CHANnel<n>:BASE:HIGH { <voltage>}

  :CHANnel<n>:BASE:HIGH?

- Functional description:

  Set the high value of the specified channel signal output.

  <voltage> unit is the current channel unit.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the high value of the specified channel signal output, return by scientific notation method.

- For example:

  :CHANnel1:BASE:HIGH 2　　set channel 1 signal output high valus as 2V

  :CHANnel1:BASE:HIGH?　　query return 2e+0

:CHANnel<n>:BASE:LOW

- Command format:

  :CHANnel<n>:BASE:LOW { <voltage>}

  :CHANnel<n>:BASE:LOW?

- Functional description:

  Set the low value of the specified channel signal output.

  <voltage> unit is the current channel unit.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the low value of the specified channel signal output, return by scientific notation method.

- For example:

  :CHANnel1:BASE:LOW 2　　set channel 1 signal output low valus as 2V

  :CHANnel1:BASE:LOW?　　query return 2e+0

:CHANnel<n>:BASE:DUTY

- Command format:

  :CHANnel<n>:BASE:DUTY { <duty>}

  :CHANnel<n>:BASE:DUTY?

- Functional description:

  Set the duty ratio of the specified channel signal output.

  <duty> presents duty ratio, unit %，range 0~100.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the duty ratio of the specified channel signal output.

- For example:

  :CHANnel1:BASE:DUTY 20　　set channel 1 signal output duty ratio as 20%

  :CHANnel1:BASE:DUTY?　　query return 20

:CHANnel<n>:BASE:BITRatio

- Command format:

  :CHANnel<n>:BASE:BITRatio　<ratio>

  :CHANnel<n>:BASE:BITRatio?

- Functional description:

  Set bit rate value of the specified channel, this command is only valid for waveform with bit rate function.

  < ratio > presents bit rate，unit is bps.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return bit rate value of the specified channel，return by scientific notation method.

- For example:

  :CHANnel1:BASE:BITRatio 1000000　　set channel 1 bit rate as 100Kbps

  :CHANnel1:BASE:BITRatio?　　query return 1e+6

:CHANnel<n>:BASE:ARB

- Command format:

  :CHANnel<n>:BASE:ARB <source>,<filename>

  :CHANnel<n>:BASE:ARB?

- Functional description:

  Set the specified channel load arbitrary wave data from a fundamental wave file.

  <n>：channel number，n take value as 1, 2, 3, 4.

  <source>：{INTernal|EXTernal|USER}，it divided into internal, external and custom.

<filename>：arbitrary waveform file name

- For example：

:CHANnel1:BASE:ARB INTernal, "test.bsv"

:CHANnel<n>:RAMP:SYMMetry

- Command format：

:CHANnel<n>:RAMP:SYMMetry { < symmetry >}

:CHANnel<n>:RAMP:SYMMetry?

- Functional description：

Set ramp wave outputsymmetry of the specified channel.

< symmetry > unit %，range 0~100.

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return ramp wave outputsymmetry of the specified channel.

- For example：

:CHANnel1:RAMP:SYMMetry 20	set channel 1 ramp wave symmetry as 20%
:CHANnel1:RAMP:SYMMetry?	query return 20

:CHANnel<n>:PULSe:RISe

- Command format：

:CHANnel<n>:PULSe:RISe {<width>}

:CHANnel<n>:PULSe:RISe?

- Functional description：

Set the specified channel pulse wave as rise edge pulse width.

<width> presents puls width，unit is S.

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return rise edge pulse width of the specified channel pulse wave, return by scientific notation method.

- For example：

:CHANnel1:PULSe:RISe 0.002	set channel 1 signal rise edge pulse width as 2ms
:CHANnel1:PULSe:RISe?	query return 2e-3

:CHANnel<n>:PULSe:FALL

- Command format:

  :CHANnel<n>:PULSe:FALL {<width>}

  :CHANnel<n>:PULSe:FALL?

- Functional description:

  Set the fall edge pulse width of the specified channel pulse wave.

  <width> presents puls width，unit is S.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return fall edge pulse width of the specified channel pulse wave, return by scientific notation method.

- For example:

  :CHANnel1:PULSe:FALL 0.002　　set channel 1 signal rise edge pulse width as 2ms

  :CHANnel1:PULSe:FALL?　　query return 2e-3

:CHANnel<n>:NOISe:BANDwith

- Command format:

  :CHANnel<n>:NOISe:BANDwith {<width>}

  :CHANnel<n>:NOISe:BANDwith?

- Functional description:

  Set the noise signal bandwidth of the specified channel.

  <width> presents bandwidth, in Hz.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the noise signal bandwidth of the specified channel, return by scientific notation method.

- For example:

  :CHANnel1:NOISe:BANDwith 2MHz　　set channel 1 noise signal width as 2MHz

  :CHANnel1:NOISe:BANDwith?　　query return 2e+6

:CHANnel<n>:HARMonic:TYPe?

- Command format:

  :CHANnel<n>:HARMonic:TYPe {ODD|EVEN|ALL|USER}

  :CHANnel<n>:HARMonic:TYPe?

- Functional description:

  Set the harmonic wave type of the specified channel.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the harmonic wave type of the specified channel.

- For example:

  :CHANnel1:HARMonic:TYPe ODD set channek 1harmonic wave type as ODD

  :CHANnel1:HARMonic:TYPe? query return ODD

:CHANnel<n>:HARMonic:MAX:ORDer?

- Command format:

  :CHANnel<n>:HARMonic:MAX:ORDer <order>

  :CHANnel<n>:HARMonic:MAX:ORDer?

- Functional description:

  Set the maximum harmonic order of the specified channel.

  < order >: harmonic order, range 2~16.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the maximum harmonic order of the specified channel, return integer data type.

- For example:

  :CHANnel1:HARMonic:TOTal:ORDer 2

  set channel 1 the maximum harmonic order as 2 time

  :CHANnel1:HARMonic:TOTal:ORDer? query return 2

:CHANnel<n>:HARMonic:USER:TYPe?

- Command format:

  :CHANnel<n>:HARMonic:USER:TYPe #H<order>

  :CHANnel<n>:HARMonic:USER:TYPe?

- Functional description:

  Set the custom harmoninc wave type of the specified channel.

  < order >: the custom harmoninc wave type, #H presents hexadecimal number, X0111 1111 1111 1111 presents the swtich of harmonic wave.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the custom harmoninc wave type of the specified channel, return integer data type.

- For example:

  :CHANnel1:HARMonic:USER:ORDer #H7FFF set channel 1 as the custom harmoninc wave

  :CHANnel1:HARMonic:USER:ORDer? query return 32767

:CHANnel<n>:HARMonic:ORDer<m>:AMPLitude?

- Command format:

  :CHANnel<n>:HARMonic:ORDer<m>:AMPLitude <amp>

  :CHANnel<n>:HARMonic:ORDer<m>:AMPLitude?

- Functional description:

  Set the amplitude value of harmonic order in the specified channel.

  < amp >: amplitude value, in Vpp.

  <n>: channel number, n take value as 1, 2, 3, 4.

  <m>: harmonic order, m take value as 2~16.

- Return format:

  Query return the amplitude value of harmonic order in the specified channel, return by scientific notation method.

- For example:

  :CHANnel1:HARM:ORDER2:AMP 0.02

  set channel 1 amplitude value of 2 time harmonic order as 20mVpp

  :CHANnel1:HARM:ORDER2:AMP? query return 2e-2

:CHANnel<n>:HARMonic:ORDER<m>:PHASe?

- Command format:

  :CHANnel<n>:HARMonic:ORDER<m>:PHASe <phase>

  :CHANnel<n>:HARMonic:ORDER<m>:PHASe?

- Functional description:

  Set the phase valu of harmonic order in the specified channel.

  <phase>: phase value, unit° .

  <n>: channel number, n take value as 1, 2, 3, 4.

  <m>: harmonic order, mtake value as 2~16

- Return format:

  Query return the phase valu of harmonic order in the specified channel, return by scientific notation method.

- For example:

  :CHANnel1:HARM:ORDER2:PHASe 20

  set channel 1 phase value of 2 time harmonic order as 20°

  :CHANnel1:HARM:ORDER2:PHASe? query return 2e+1

:CHANnel<n>:ARB:MODe

- Command format:

  :CHANnel<n>:ARB:MODe {DDS | POINTS }

  :CHANnel<n>:ARB:MODe?

- Functional description:

  Set the arbitrary output mode of the specified channel, it divided into DDS and point-to-point mode.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the arbitrary output mode of the specified channel.

- For example:

  :CHANnel1:ARB:MODe DDS　　set channel 1 arbitray mode as DDS

  :CHANnel1:ARB:MODe?　　query return DDS

:CHANnel<n>:ARB:FILTer

- Command format:

  :CHANnel<n>:ARB:FILTer {ZEROHOLD | LINE }

  :CHANnel<n>:ARB:FILTer?

- Functional description:

  Set interpolation method of the specified channel arbitrary wave output, it divided into retain zero and linear interpolation.

  <n>：channel number，n take value as 1, 2.

- Return format:

  Query return interpolation method of the specified channel arbitrary wave output.

- For example:

  :CHANnel1:ARB:FILTer LINE

  set channel 1 arbitrary wave output as linear interpolation method

  :CHANnel1:ARB:FILTer?　　query return LINE

Modulation

:CHANnel<n>:MODulate:TYPe

- Command format:

  :CHANnel<n>:MODulate:TYPe　<type>

  :CHANnel<n>:MODulate:TYPe?

- Functional description:

  Set the modulation type of the specified signal.

  <type>: {AM|BAM|QAM|ASK|FM|FSK|ThreeFSK|FourFSK|PM|PSK|BPSK|QPSK|OSK|PWM|SUM}

  They are AM, DSB-AM, QAM, ASK, FM, FSK, 3FSK, 4FSK, PSK, BPSK, QPSK, OSK, PWM, SUM.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the modulation type of the specified signal.

- For example:

  :CHANnel1:MODulate:TYPe AM set channel 1 signal as AM modulation

  :CHANnel1:MODulate:TYPe? query return AM

:CHANnel<n>:MODulate:WAVe

- Command format:

  :CHANnel<n>:MODulate:WAVe { SINe|SQUare|UPRamp|DNRamp|ARB|NOISe }

  :CHANnel<n>:MODulate:WAVe?

- Functional description:

  Set the modulating waveform type of the specified signal, they are sine wave, square wave, upper triangle, down triangle, arbitrary wave and noise.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the modulating waveform type of the specified signal.

- For example:

  :CHANnel1:MODulate:WAVe SINe

  set channel 1 signal modulating waveform type as sine wave

  :CHANnel1:MODulate:WAVe? query return SINe

:CHANnel<n>:MODulate:SOURce

- Command format:

  :CHANnel<n>:MODulate:SOURce { INTernal|EXTernal }

  :CHANnel<n>:MODulate:SOURce?

- Functional description:

  Set the modulating source of the specified channel, it divided into internal and external.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the modulating source of the specified channel.

- For example:

  :CHANnel1:MODulate:SOURce INTernal set channel 1 modulating source as interanl

  :CHANnel1:MODulate:SOURce? query return INTernal

:CHANnel<n>:MODulate:FREQuency

- Command format:

  :CHANnel<n>:MODulate:FREQuency {<freq>}

  :CHANnel<n>:MODulate:FREQuency?

- Functional description:

  Set the modulating frequency of the specified channel signal.

  <freq> presents frequency value, in Hz.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the modulating frequency of the specified channel signal, return by scientific notation method.

- For example:

  :CHANnel1:MODulate:FREQuency 2000

  set channel 1 signal modulaing frequency as 2KHz

  :CHANnel1:MODulate:FREQuency? query return 2e+3

:CHANnel<n>:MODulate:IQMap

- Command format:

  :CHANnel<n>:MODulate: IQMap {<IQ TYPE>}

  :CHANnel<n>:MODulate: IQMap?

- Functional description:

  IQ type of the specified QAM can divid into:

  4QAM, 8QAM, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM.

  < IQ TYPE > presents IO map type.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return IQ type of the specified channel.

- For example:

  :CHANnel1:MODulate:IQMap 32QAM set channel 1 modulating IQ map as 32QAM

  :CHANnel1:MODulate:IQMap? query return 32QAM

:CHANnel<n>:MODulate:ARB

- Command format:

  :CHANnel<n>:MODulate:ARB <source>,<filename>

  :CHANnel<n>:MODulate:ARB?

- Functional description:

  Set the specified channel load arbitrary wave form modulating wave file.

<n>：channel number，n take value as 1, 2, 3, 4.

<source>：{INTernal|EXTernal|USER}，it divided into internal, external and custom.

<filename>：arbitrary wave file name.

- For example：

:CHANnel1:MODulate:ARB INTernal, "test.bsv"

:CHANnel<n>:MODulate:DEPTh

- Command format：

:CHANnel<n>:MODulate:DEPTh { <depth>}

:CHANnel<n>:MODulate:DEPTh?

- Functional description：

Set the modulating depth of the specified channel.

<depth> presents modulating depth, unit %。0% ~ 100%，AM modulating depth is 0% ~ 120%

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return the modulating depth of the specified channel.

- For example：

:CHANnel1:MODulate:DEPTh 50　　set channel 1 modulating depth as 50%

:CHANnel1:MODulate:DEPTh?　　query return 50

:CHANnel<n>:MODulate:BITRatio

- Command format：

:CHANnel<n>:MODulate:BITRatio　<ratio>

:CHANnel<n>:MODulate:BITRatio?

- Functional description：

Set the modulating bit ratio of the specified channel, this command is only valid for the modulating type with bit ratio function.

< ratio > presents bit ratio，in bps.

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return the modulating bit ratio of the specified channel，return by scientific notation method.

- For example：

:CHANnel1:MODulate:BITRatio 1000000　　set channel 1 bit ratio as 100Kbps

:CHANnel1:MODulate:BITRatio?　　query return 1e+6

:CHANnel<n>:OSK:TRIGger:SOURce

- Command format:

  :CHANnel<n>:OSK:TRIGger:SOURce { INTernal|EXTernal }

  :CHANnel<n>:OSK:TRIGger:SOURce?

- Functional description:

  Set the specified channel oscillation keying trigger source, it divided into internal and external.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the specified channel oscillation keying trigger source.

- For example:

  Set channel 1 oscillation keying trigger source as internal

  :CHANnel1:OSK:TRIGger:SOURce? query return INTernal

:CHANnel<n>:FM:FREQuency:DEV

- Command format:

  :CHANnel<n>:FM:FREQuency:DEV { <freq>}

  :CHANnel<n>:FM:FREQuency:DEV?

- Functional description:

  Set the frequency deviation of the specified channel.

  <freq> presents frequency deviation, in Hz. 0Hz ~ the current fundamental frequency.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the frequency deviation of the specified channel, return data by scientific notation method.

- For example:

  :CHANnel1:FM:FREQuency:DEV 2000 set channel 1 frequency deviation as 2KHz

  :CHANnel1:FM:FREQuency:DEV? query return 2e+3

:CHANnel<n>:PM:PHASe:DEV

- Command format:

  :CHANnel<n>:PM:PHASe:DEV { <phase>}

  :CHANnel<n>:PM:PHASe:DEV?

- Functional description:

  Set the phase deviation of the specified channel output.

  < phase > presents phase deviation，unit°, range 0~360.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the phase deviation of the specified channel output.

- For example:

  :CHANnel1:PM:PHASe:DEV 30　　set channel 1 phase deviation as 30°

  :CHANnel1:PM:PHASe:DEV?　　qery return 30

:CHANnel<n>:PWM:DUTY:DEV

- Command format:

  :CHANnel<n>:PWM:DUTY:DEV { <duty>}

  :CHANnel<n>:PWM:DUTY:DEV?

- Functional description:

  Set pulse width deviation of the specified channel output in pulse width modulation.

  < duty > presents pulse width deviation，unit %，range 0~100.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return pulse width deviation of the specified channel output in pulse width modulation, return data by scientific notation method.

- For example:

  :CHANnel1:PWM:DUTY:DEV 10　　set channel 1 pulse width deviation as 10%

  :CHANnel1:PWM:DUTY:DEV?　　query return 1e+1

:CHANnel<n>:FSK:FREQuency<m>

- Command format:

  :CHANnel<n>:FSK:FREQuency<m> { <freq>}

  :CHANnel<n>:FSK:FREQuency<m>?

- Functional description:

  Set the sweep frequency of multifrequency frequency shift keying of the specified channel output, the modulation method must be specified in advance for this command to take effect.

  < freq > presents frequency, in Hz.

<n>：channel number，n take value as 1, 2, 3, 4.

<m>: frequency serial number，2FSK take value as 1；3FSK take value as 1, 2；4FSK take value as 1, 2, 3；

- Return format：

Query return the sweep frequency of the specified channel, return data by scientific notation method.

- For example：

:CHANnel1:FSK:FREQ1 2000	set channel 1 output sweep frequency as 2KHz
:CHANnel1:FSK:FREQ1?	query return 2e+3

:CHANnel<n>:PSK:PHASe<m>

- Command format：

:CHANnel<n>:PSK:PHASe<m> { < phase >}

:CHANnel<n>:PSK:PHASe<m>?

- Functional description：

Set phase value of multifrequency phase shift keying of the specified channel output, the modulation method must be specified in advance for this command to take effect.

< phase> unit ° , range 0~360

<n>：channel number，n take value as 1, 2, 3, 4.

<m>: phase serial number，BPSK take value as 1；QPSK take value as 1, 2, 3；

- Return format：

Query return the phase value of the specified channel phase shift keying, return data by scientific notation method.

- For example：

:CHANnel1:PSK:PHAS1 90	set channel 1 output phase as 90°
:CHANnel1:PSK:PHAS1?	query return 9e+1

:CHANnel<n>:OSK:TIMe

- Command format：

:CHANnel<n>:OSK:TIMe { <time>}

:CHANnel<n>:OSK:TIMe?

- Functional description：

Set oscillation time of oscillation shift keying in modulating mode of the specified channel.

< time > presents oscillation time, unit S.

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return oscillation time of oscillation shift keying in modulating mode of the specified channel, return data by scientific notation method.

- For example：

:CHANnel1:OSK:TIMe 2ms

set channel 1 oscillation time of oscillation shift keying as 2ms

:CHANnel1:OSK:TIMe? query return 2e-3

Sweep Frequency

:CHANnel<n>:SWEep:TYPe

- Command format:

  :CHANnel<n>:SWEep:TYPe { LINe|LOG|STEP|LIST}

  :CHANnel<n>:SWEep:TYPe?

- Functional description:

  Set the sweep frequency mode of the specified channel, they are linear sweep, logarithm sweep, step sweep and list sweep.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the sweep frequency mode of the specified channel

- For example:

  :CHANnel1:SWEep:TYPe LINe set channel 1 as linear sweep

  :CHANnel1:SWEep:TYPe? query return LINe

:CHANnel<n>:SWEep:FREQuency:STARt

- Command format:

  :CHANnel<n>:SWEep:FREQuency:STARt <freq>

  :CHANnel<n>:SWEep:FREQuency:STARt?

- Functional description:

  Set the initial frequency of the specified channel sweep frequency.

  < freq > presents frequency, in Hz.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the initial frequency of the specified channel sweep frequency，return data by scientific notation method.

- For example:

  :CHANnel1:SWE:FREQ:STAR 2000

  set channel 1 initial frequency of sweep frequency output as 2KHz

  :CHANnel1:SWE:FREQ:STAR? query return 2e+3

:CHANnel<n>:SWEep:FREQuency:STOP

- Command format:

  :CHANnel<n>:SWEep:FREQuency:STOP <freq>

  :CHANnel<n>:SWEep:FREQuency:STOP?

- Functional description:

  Set the stop frequency of the specified channel sweep frequency.

  < freq > presents frequency, in Hz.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return the stop frequency of the specified channel sweep frequency, return data by scientific notation method.

- For example:

  :CHANnel1:SWE:FREQ:STOP 2000

  set channel 1 stop frequency of sweep frequency output as 2KHz

  :CHANnel1:SWE:FREQ:STOP? query return 2e+3

:CHANnel<n>:SWEep:TIMe

- Command format:

  :CHANnel<n>:SWEEP:TIMe <time>

  :CHANnel<n>:SWEEP:TIMe?

- Functional description:

  Set scanning time of the specified channel sweep frequency.

  < time > unit S. range：1ms ~ 500s

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return scanning time of the specified channel sweep frequency, return data by scientific notation method.

- For example:

  :CHANnel1:SWEEP:TIMe 2　　set channel 1 scanning time as 2S

  :CHANnel1:SWEEP:TIMe?　　query return 2e+0

:CHANnel<n>:SWEep:HOLD

- Command format:

  :CHANnel<n>:SWEEP:HOLD <time>

  :CHANnel<n>:SWEEP:HOLD?

- Functional description:

  Set dwell time of the specified channel sweep frequency, this command is only valid for step sweeo and linear sweep.

  < time > unit S.

  <n>：channel number，n take value as 1, 2, 3, 4.

- Return format:

  Query return dwell time of the specified channel sweep frequency，return data by scientific notation method.

- For example:

  :CHANnel1:SWEEP:HOLD 2　　set channel 1 dwell time as 2S

  :CHANnel1:SWEEP:HOLD?　　query return 2e+0

:CHANnel<n>:SWEep:STEPs

- Command format:

  :CHANnel<n>:SWEEP:STEPs <steps>

  :CHANnel<n>:SWEEP:STEPs?

- Functional description:

  Set the total step number of the specified channel in step sweep mode, this command is only valid

for step sweep.

< steps >：step number

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return scanning time of the specified channel sweep frequency, return integer data.

- For example：

:CHANnel1:SWEEP:STEPs 10 set the step number as 10

:CHANnel1:SWEEP:STEPs? query return 10

:CHANnel<n>:SWEep:TRIGger

- Command format：

:CHANnel<n>:SWEep:TRIGger

- Functional description：

Set sweep frequency output of the specified channel, this parameter is only valid when in manual trigger mode.

- For example：

:CHANnel1:SWEep:TRIGger trigger ont-time sweep frequency signal output

Burst

:CHANnel<n>:BURSt:TYPe

- Command format：

:CHANnel<n>:BURSt:TYPe {NCYC|GATe|INFinit}

:CHANnel<n>:BURSt:TYPe?

- Functional description：

Set burst type of the specified channel, they are N cycle gating and infinite pulse string.

<n>：channel number，n take value as 1, 2, 3, 4.

- Return format：

Query return burst type of the specified channel.

- For example：

:CHANnel1:BURSt:TYPe NCYC set channel 1 as N cycle burst mode

:CHANnel1:BURSt:TYPe? query return 2e+0

:CHANnel<n>:BURSt:PERiod

- Command format：

:CHANnel<n>:BURSt:PERiod <period >

:CHANnel<n>:BURSt:PERiod?

- Functional description:

Set the burst period of the specified channel.

< period > presents time in S.

<n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

Query return the burst period of the specified channel, return data by scientific notation method.

- For example:

:CHANnel1:BURSt:PERiod 5ms	set channel 1 burst period as 5ms
:CHANnel1:BURSt:PERiod?	query return 5e-3

:CHANnel<n>:BURSt:PHASe

- Command format:

:CHANnel<n>:BURSt:PHASe <phase>

:CHANnel<n>:BURSt:PHASe?

- Functional description:

Set the burst phase of the specified channel.

< phase >unit °, range: 0 ~ 360

<n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

Query return the burst phase of the specified channel, return data by scientific notation method.

- For example:

:CHANnel1:BURSt:PHASe 18	set channel 1 burst phase as 18°
:CHANnel1:BURSt:PHASe?	query return 1.8e+1

:CHANnel<n>:BURSt:CYCLes

- Command format:

:CHANnel<n>:BURSt:CYCLes <cycles>

:CHANnel<n>:BURSt:CYCLes?

- Functional description:

Set burst cycle number of the specified channel.

< cycles > presents cycle number, integer data.

<n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

Query return burst cycle number of the specified channel.

- For example:

:CHANnel1:BURSt:CYCLes 2	set burst cycle number as 2
:CHANnel1:BURSt:CYCLes?	query return 2

:CHANnel<n>:BURSt:GATe:POLarity

- Command format:

  :CHANnel<n>:BURSt:GATe:POLarity {POSitive|NEGative}

  :CHANnel<n>:BURSt:GATe:POLarity?

- Functional description:

  Set the burst polarity of gating mode of the specified channel, it divided into positive polarity and negative polarity.

  <n>: channel number, n take value as 1, 2, 3, 4.

- Return format:

  Query return the burst polarity of gating mode of the specified channel.

- For example:

  :CHANnel1:BURSt:GATe:POLarity POSitive set channel 1 gating mode as positive polarity

  :CHANnel1:BURSt:GATe:POLarity? qery return POSitive

:CHANnel<n>:BURSt:TRIGger

- Command format:

  :CHANnel<n>:BURSt:TRIGger

- Functional description:

  Trigger the burst output of the specified channel, this parameter is only valid when manual trigger mode.

- For example:

  :CHANnel1:BURSt:TRIGger trigger burst signal one-time

WARB Command

This command is to write arbitrary wave file command, including configuration of fundamental arbitrary wave and modulating arbitray wave.

:WARB<n>:MODulate

- Command format:

  :WARB<n>:MODulate <arb file>

- Functional description:

  To write arbitray wave, send this command firstly, and then send arbitrary wave file data to signal source.

  <arb file> presents arbitrary wave file name, only support bsv file format.

- For example:

  :WARB1:MODulate "test.bsv" write channel 1 modulating arbitray wave file

:WARB<n>:CARRier

- Command format:

  :WARB<n>:CARRier <arb file>

- Functional description:

  To write fundamental arbitray wave, send this command firstly, and then send arbitrary wave file data to signal source.

  <arb file> presents arbitrary wave file name, only support bsv file format.

- For example:

  :WARB1:CARRier "test.bsv" write channel 1 fundamental arbitray wave file

:WFREQLIST<n>

- Command format:

  :WFREQLIST<n> <arb file>

- Functional description:

  To write frequency list file, send this command firstly, and then sen frequency list file data to signal source.

  <arb file> presents frequency list file name, only support csv file format.

- For example:

  :WFREQLIST1 "freq.csv" write channel 1 frequency list file

DIGital Command

This command is to output digital communication signal, like UART, SPI and I2C etc.

:DIGital

- Command format:

  :DIGital {{1 | ON} | {0 | OFF}}

  :DIGital?

- Functional description:

  Set the switch of digital communication signal function of the specified channel.

- Return format:

  Query return the switch of digital communication signal function, 0 presents off, 1 presents on.

- For example:

  :DIGital ON　　turn on digital communication signal

  :DIGital?　　query return 1

:DIGital:TYPe

- Command format:

  :DIGital:TYPe {UART|I2C|SPI }

  :DIGital:TYPe?

- Functional description:

  set digital communication signal type of the specified channel output, it divided into UART, I2C and SPI.

- Return format:

  Query return digital communication signal type of the specified channel output.

- For example:

  :DIGital:TYPe UART　　set output UART communication signal

  :DIGital:TYPe?　　query return UART

:DIGital:AMPLitude

- Command format:

  :DIGital:AMPLitude <amp>

  :DIGital:AMPLitude?

- Functional description:

  Set communication signal amplitude of the specified channel output.

  <amp>: presents amplitude in V.

- Return format:

  Query return digital communication signal type of the specified channel output, return by scientific notation method.

- For example:

  :DIGital:AMPLitude 3 set channel output 3V communication signal

  :DIGital:AMPLitude? query return 3e+0

:DIGital:FORMat

- Command format:

  :DIGital:FORMat { HEX|CHAR }

  :DIGital:FORMat?

- Functional description:

  Set data format of digital communication signal of the specified channel output, it divided into hexadecimal data and ASCII data.

- Return format:

  Query return data format of digital communication signal of the specified channel output.

- For example:

  :DIGital:FORMat HEX

  set data format of digital communication signal as hexadecimal data

  :DIGital:FORMat? query return HEX

:DIGital:AS

- Command format:

  :DIGital:AS {{1 | ON} | {0 | OFF}}

  :DIGital:AS?

- Functional description:

  Set send mode of digital communication signal of the specified channel output, OFF presents manually send; ON presents automatically send.

- Return format:

  Query return whether digital communication signal of the specified channel output is turn on, 1 presents on, 0 presents off.

- For example:

  :DIGital:AS ON set send mode as automatically send

  :DIGital:AS? query return 1

:DIGital:AS:INTerval

- Command format:

  :DIGital:AS:INTerval <time>

  :DIGital:AS:INTerval?

- Functional description:

  Set the interval time of the specified channel automatically output digital communication signal

  <time>: interval time, unit S.

- Return format:

Query return the interval time of the specified channel automatically output digital communication signal，return by scientific notation method.

- For example：

:DIGital:AS:INTerval 10ms　　set interval time as 10ms

:DIGital:AS:INTerval?　　query return 1e-2

:DIGital:TRIGger

- Command format：

:DIGital:TRIGger

- Functional description：

Trigger the digital communication signal send of the specified channel output, this signal is onl valid when manually send.

- For example：

:DIGital:TRIGger　　trigger digital signal send one-time

:DIGital:DATA

- Command format：

:DIGital:DATA <data>

:DIGital:DATA?

- Functional description：

Set digital communication signal data of the specified channel automatically send.

<data>：binary byte stream data

- Return format：

Query return digital communication signal data of the specified channel output, return binary byte stream data.

- For example：

:DIGital:DATA　　write digital communication signal data into signal source

:DIGital:DATA?　　query return binary byte stream data

UART

:DIGital:UART:BAUDrate

- Command format：

:CHANnel:UART:BAUDrate <baudrate>

:CHANnel:UART:BAUDrate?

- Functional description：

Set baud rate of the specified channel output digital UART communication signal.

<baudrate>：baud rat in bps，integer data

- Return format：

Query return baud rate of the specified channel output digital UART communication signal，return integer data.

- For example：

  :DIGital:UART:BAUDrate 115200

  set baud rate of UART communicaton signal as 115200

  :DIGital:UART:BAUDrate? query return 115200

:DIGital:UART:DATa

- Command format：

  :CHANnel:UART:DATa <bit >

  :CHANnel:UART:DATa?

- Functional description：

  Set data bit of UART communication signal of the specified channel output.

  <bit >：data bit，integer data，range 4~8

- Return format：

  Query return data bit of UART communication signal of the specified channel output, return integer data.

- For example：

  :DIGital:UART:DATa 4 set data bit of UART communication signal as 4

  :DIGital:UART:DATa? query return 4

:DIGital:UART:STOP

- Command format：

  :CHANnel:UART:STOP <bit >

  :CHANnel:UART:STOP?

- Functional description：

  Set stop bit of UART communication signal of the specified channel output.

  <bit >：stop bit，integer，range 1~2

- Return format：

  Query return stop bit of UART communication signal of the specified channel output, return integer data.

- For example：

  :DIGital:UART:STOP 1 set data bit of UART communication signal as 1

  :DIGital:UART:STOP? query return 1

:DIGital:UART:PARity

- Command format：

:CHANnel:UART:PARity {NONE|EVEN|ODD}

:CHANnel:UART:PARity?

- Functional description:

Set the check bit of UART communication signal of the specified channel output, it divided into none parity check, odd parity check and even parity check.

- Return format:

Query return the check bit of UART communication signal of the specified channel output, return integer data.

- For example:

:DIGital:UART:PARity NONE　　set check bit as none parity check

:DIGital:UART:PARity?　　query return NONE

IIC

:DIGital: IIC:CLOCk

- Command format:

:CHANnel:IIC:CLOCk　<freq>

:CHANnel:IIC:CLOCk?

- Functional description:

Set digital IIC communication clock of the specified channel output.

<freq>：clock，in Hz

- Return format:

Query return digital IIC communication clock of the specified channel output, return by scientific notation method.

- For example:

:DIGital:IIC:CLOCk 1000　　set digital IIC communication clock as 1KHz

:DIGital:IIC:CLOCk?　　query return 1e+3

:DIGital:IIC:ADDRess

- Command format:

:CHANnel:IIC:ADDRess　<address>

:CHANnel:IIC:ADDRess?

- Functional description:

Set the digital IIC communication signal of the specified channel output.

<address>：integer data

- Return format:

Query return digital IIC communication signal address of the specified channel output.

- For example:

:DIGital:IIC:ADDRess 3　　set IIC communication signal address as 3

:DIGital:IIC:ADDRess? query return 3

SPI

:DIGital:SPI:CLOCk

- Command format:

  :CHANnel:SPI:CLOCk <freq>

  :CHANnel:SPI:CLOCk?

- Functional description:

  Set SPI communication signal clock of the speicifed channel output.

  <freq>: clock, in Hz

- Return format:

  Query return SPI communication signal clock of the speicifed channel output, return by scientific notation method.

- For example:

  :DIGital:SPI:CLOCk 1000 set digital SPI communication clock as 1KHz

  :DIGital:SPI:CLOCk? query return 1e+3

DISPlay Command

This command is to display signal source information.

:DISPlay:DATA?

- Command format:

  :DISPlay:DATA?

- Functional description:

  To query image data display on the screen.

- Return format:

  Query return image data, return data conform to binary data IEEE 488.2 #.

- For example:

  :DISPlay:DATA? query return image data

  data format: #800012345+bitmap data

KEY Command

This command is to control key and knob on the operating panel.

:KEY:<key>

- Command format:

  :KEY:<key>

  :KEY:<key>:LOCK { {1 | ON} | {0 | OFF} }

  :KEY:<key>:LOCK?

  :KEY:<key>:LED?

- Functional description:

  Set key function and lock/unlock status. <key> definition and description see Appendix 1:<Key> Table.

- Return format:

  Query return key lock status or LED light status.

  Lock status：0 presents unlock, 1 presents lock；

  LED light status：0 presents light off, 1 presents light on（green）, 2 presents light on（red）.

- For example:

:KEY:AUTO	automatically set oscilloscope's control parameter
:KEY:AUTO:LOCK ON/OFF	lock/unlock key
:KEY:AUTO:LOCK?	query return key lock status, 1 presents lock
:KEY:AUTO:LED?	query return LED light status, 0 presents light off

:KEY:LOCK?

- Command format:

  :KEY:LOCK?

- Functional description:

  Query all key status

- Return format:

  Query return all key status, return character sequency, each character presents one key lock status, ASCII'1'measns locked, ASCII'0'means unlock，return lock status according Appendix 2:<Key Lock Status> Table sequency.

- For example:

  10 key in total，only the fourth and fifith key is locked，return ASCII character string 0001100000

:KEY:LED?

- Command format:

  :KEY:LED?

- Functional description:

  Query status of key with light function.

- Return format:

  Query return light staus of key, return character sequency, each character presents one key's light status, ASCII'1' means light on , ASCII'0' means light off, return light status according to Appendix 1: <Key> Table.

- For example:

  Three key with light function, only CH1, CH2 key light on, CH3、CH4 key light off, return ASCII character string 1100

Appendix 1：<Key> Table

Key name	Functional description	LED light
COUNtinue	continuity（CW）	
MOD	Modulation	
SWEep	sweep freqeueny	
BURSt	Burst	
SINe	sine wave	
SQUare	square wave	
RAMP	triangle wave	
PULSe	pulse wave	
ARB	arbitrary wave	
HARMonic	harmonic wave	
NOISe	noise	
DC	DC	
PRBS	pseudo-random binary sequence	
WARB	written arbitrary wave	
CH1	channel 1 key	√
CH2	channel 2 key	√
CH3	channel 3 key	√
CH4	channel 4 key	√
Utility	system	
Right	right direction key	
Left	left direction key	
OK	confirm key	
Up	up direction key	
Down	down direction key	
NUM0	numeric key 0	
NUM1	numeric key 1	
NUM2	numeric key 2	
NUM3	numeric key 3	
NUM4	numeric key 4	
NUM5	numeric key 5	
NUM6	numeric key 6	
NUM7	numeric key 7	
NUM8	numeric key 8	
NUM9	numeric key 9	
Dot	numeric key decimal point	
Symbol	numeric key symbol	

~~BACKspace~~	~~backspace~~	

Appendix 2：Key Lock Status

Order of bit	Key	Status
0	COUNtinue	0 presents unlock, 1 presents lock
1	MOD	0 presents unlock, 1 presents lock
2	SWEep	0 presents unlock, 1 presents lock
3	BURSt	0 presents unlock, 1 presents lock
4	SINe	0 presents unlock, 1 presents lock
5	SQUare	0 presents unlock, 1 presents lock
6	RAMP	0 presents unlock, 1 presents lock
7	PULSe	0 presents unlock, 1 presents lock
8	ARB	0 presents unlock, 1 presents lock
9	HARMonic	0 presents unlock, 1 presents lock
10	NOISe	0 presents unlock, 1 presents lock
11	DC	0 presents unlock, 1 presents lock
12	PRBS	0 presents unlock, 1 presents lock
13	WARB	0 presents unlock, 1 presents lock
14	CH1	0 presents unlock, 1 presents lock
15	CH2	0 presents unlock, 1 presents lock
16	CH3	0 presents unlock, 1 presents lock
17	CH4	0 presents unlock, 1 presents lock
18	Utility	0 presents unlock, 1 presents lock
19	Right	0 presents unlock, 1 presents lock
20	Left	0 presents unlock, 1 presents lock
21	OK	0 presents unlock, 1 presents lock
22	Up	0 presents unlock, 1 presents lock
23	Down	0 presents unlock, 1 presents lock
24	NUM0	0 presents unlock, 1 presents lock
25	NUM1	0 presents unlock, 1 presents lock
26	NUM2	0 presents unlock, 1 presents lock
27	NUM3	0 presents unlock, 1 presents lock
28	NUM4	0 presents unlock, 1 presents lock
29	NUM5	0 presents unlock, 1 presents lock
30	NUM6	0 presents unlock, 1 presents lock
31	NUM7	0 presents unlock, 1 presents lock
32	NUM8	0 presents unlock, 1 presents lock
33	NUM9	0 presents unlock, 1 presents lock
34	Dot	0 presents unlock, 1 presents lock
35	Symbol	0 presents unlock, 1 presents lock

36	BACKspace	0 presents unlock, 1 presents lock