UNI-T®

Instruments.uni-trend.com

Programming Manual
UTE310 Digital Power Meter

Chapter 1 Communication Interface Description

1.1 USB Interface

The user can send the command to the power meter via the USB interface and perform the function that corresponding to the Function key on the front panel;

After the power meter receives the command, it will return the measured and calculated data, the setting parameter and state byte of the control panel and error code.

Note: Do not use other communication interface to control the power meter when using the USB interface.

Table 1.1 Interface Explanations

Item	Description
Port number	1
Connector	B type connector (plug)
Electrical and mechanical specifications	USB Rev.2.0
Transmission mode	Full speed
Protocol	USBTMC-USB488
PC requirement	Windows10, Windows8 with USB, Windows 7 (32-bit /64-bit), Vista (32-bit), XP (SP2 version and above, 32-bit)

1.2 GPIB Interface

The user can send the command to the power meter via GPIB interface and perform the function that corresponding to the Function key on the front panel;

After the power meter receives the command, it will return the measured and calculated data, the setting parameter and state byte of the control panel and error code.

Table 1.2 GPIB Interface

Applicable equipment	American National Instrument ● PCI-GPIB or PCI-GPIB+、 PCIe-GPIB or PCIe-GPIB+ ● PCMCIA-GPIB or PCMCIA-GPIB+ (Windows Vista or Windows 7 are not support) ● GPIB-USB-HS uses NI-488.2M Ver. 2.8.1 or update drive
Electrical and mechanical specifications	IEEE St' d 488-1978 (JIS C 1901-1987)

Note: To ensure reliable and stable GPIB communication, please use genuine GPIB cables.

1.2.1 GPIB Setting

Each GPIB device has a unique GPIB address, which used to distinguish different GPIB devices. Therefore, when using the GPIB interface of the power meter, the user needs to set

the GPIB address of the power meter first.

Note: Do not change the GPIB address when the power meter is using the GPIB interface to communicate. In addition, do not use other communication interfaces to control the power meter when using the GPIB interface for remote control.

1.3 RS-232 Interface

The user can send the command to the power meter via RS-232 interface and perform the function that corresponding to the Function key on the front panel;
After the power meter receives the command, it will return the measured and calculated data, the setting parameter and state byte of the control panel and error code.

Table 1.3 RS-232 interface description

Interface type	D-Sub 9-pin (plug)
Electrical and mechanical specifications	EIA-574 (EIA-232(RS-232) 9-pin)
Baud rate	1200, 2400, 4800, 9600, 19200, 57600, 115200

Note: When using RS-232 interface to communicate, please select the correct communication protocol and baud rate, and refer to "UTE310 User Manual Section 7.2.3" and "UTE310 User Manual Section 7.3.1" for setting methods.

1.4 Ethernet Interface

The user can send the command to the power meter via Ethernet interface and perform the function that corresponding to the Function key on the front panel;

After the power meter receives the command, it will return the measured and calculated data, the setting parameter and state byte of the control panel and error code.

Table 1.4 Ethernet Interface

Port number	1
Interface type	RJ-45
Electrical and mechanical specifications	IEEE802.3
Transmission system	Ethernet (100BASE-TX, 10BASE-T)
Transmission rate	Maximum 100Mbps
Communication protocol	TCP/IP
Support services	DHCP, remote control

Chapter 2 Programming Overview

2.1 Message

The message is used for communication between the controller and the power meter.
The message sent by the controller to the power meter called command message, and the message sent back by the power meter to the controller called response message
When the power meter receives an inquiry command, it will return a response message immediately.
For the power meter, one response message corresponds to one command message.

2.1.1 Command Message

The format of command message is shown in Figure 2.1

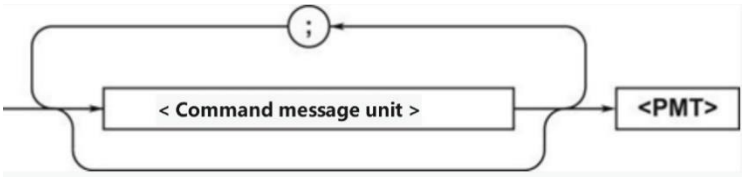


Figure 2.1

- Command Message Unit

A command message consists of one or more command message units. Each command message unit corresponds to one command, the command message units are separated with a semicolon ";", and the power meter executes the command received first.

The syntax of command message units is shown in Figure 5.2, an example is shown in Figure 2.2.

The command header describes the command type, and the command data is the data parameter of the command, which is separated from the command header by a space, and a comma, an example of command header, separates the data of multiple commands and command data are shown in Figure 2.3

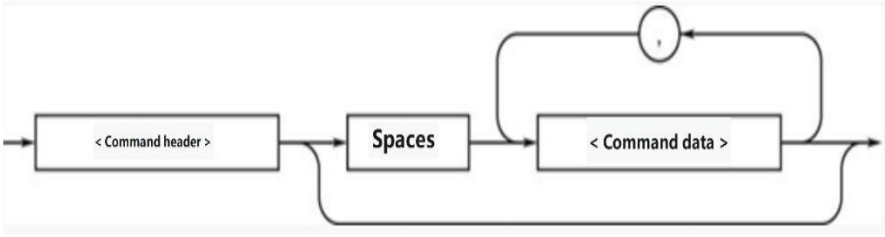


Figure 2.2 Syntax of command message unit

Figure 2.3 Example of command message unit

Figure 2.4 Example of command header and command data

- PMT

  PMT is the end mark of command message. There are three kinds of end mark
  - NL (new line) and LF (line feed) is the same, ASCII code is "0AH";
  - ^END: ^END follow a command data is the last data byte in the command message;
  - NL^END: NL and END message should send together.

2.1.2 Response Message

The syntax of response message is shown in Figure 2.5.

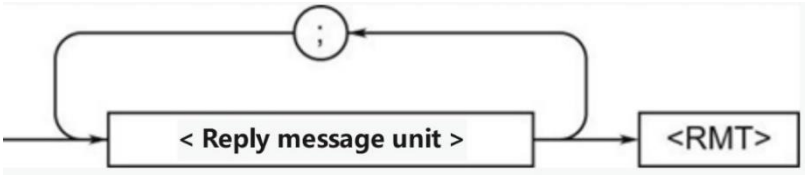


Figure 2.5 Syntax of response message

- Response Message Unit

A response message has one or more response message units are separated by semicolons, as shown in Figure 2.6.

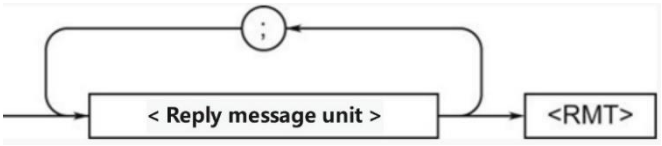


Figure 2.6 Example of response message

The syntax of response message is shown in Figure 2.7

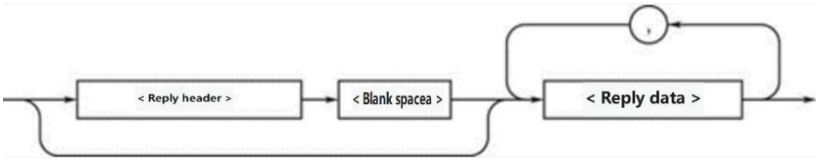


Figure 2.7 Syntax of response message

The response header appears before the response data, and the response header is separated from the response data by a space. The response data contains the content of the response, and multiple sets of response data are separated by a comma ",".

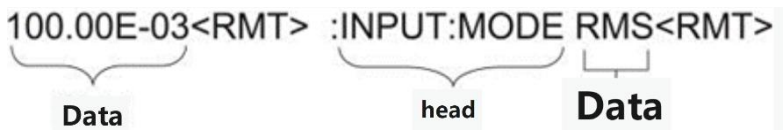


Figure 2.8 Example of command header and command data

If the command message has multiple query requests, the power meter will respond the first query request. In most cases, single query returns single response, while there are some queries need to return multiple responses; thus, the first response corresponds to the first query, but the Nth response does not necessarily correspond to the Nth query. Therefore, if the user wants to be sure to receive every response, a command message can contain only one query request.

- RMT
  RMT is the end mark of a response message, which is NL^END

2.1.3 Notes

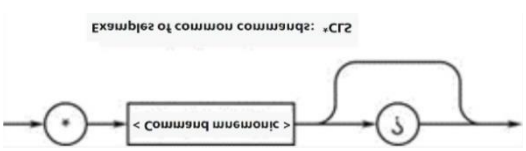
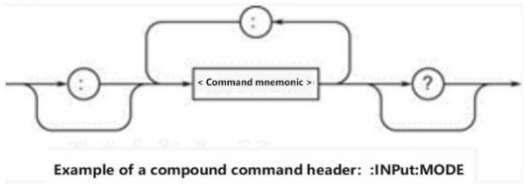
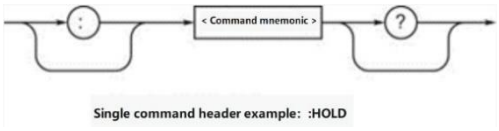
- If the controller sends a command message without a query, then another command message can be sent at any subsequent moment.
- If the controller sends a command message containing a query, the next command message can only be sent when the response message are all received. Otherwise, an error will occur.
- If the controller receives a non-existent response, an error will occur.
- If the controller receives a response when the command message is not sent, an error will occur.
- If the controller sends an incomplete command message, an error will occur.

2.2 Command

2.2.1 Type

According to the different format of the command header, the commands sent by the controller to the power meter can be categorized into three kinds, as shown in Table 2.1.

Table 2.1 Format of command header

Format of Command Header	Description
	The command defines by the standard IEEE 488.2-1992 is common command. The format of common command is shown on the left figure, and "*" should attach before the common command.
	Some special command is represent by different level according to its function. A colon separates different level ":", and the format and example is shown on the left figure.
	The function of some commands is independent, and the format of its command header is simple and not hierarchical, as shown on the left figure.

2.2.2 Command Connection

1. Connecting Different Command Group
   If two different command groups need to be separated by a colon ":", and the colon cannot be omitted.
   For example,
   : INTEGrate: MODE NORMal; INPut: MODE RMS<PMT>

2. Single Command Connecting to Other Command
   If a single command follows another command, a colon ":" must be attach in front of the single command.
   For example,
   : INTEGrate: MODE NORMal; : HOLD ON<PMT>

3. Connecting Common Command
   The common command is defined by IEEE488.2-1992. When connecting the common command, a colon ":" is not required before the command.
   For example,
   : INTEGrate: MODE NORMal; :*CLS; INTEGrate: TIMer 1, 0, 0<PMT>

4. <PMT> for Separating Command

When using <PMT> to separate two commands, two command messages will be sent firstly. In this case, the command header cannot be omitted, even if the two commands are in the same command group and with the same command header.

2.2.3 Upper Level Request

Add a question mark"?" after the highest-level command of the command group, it can query the information queried by all the lower-level query commands in the command group.

For example

INTEGrate?<PMT>->:INTEGRATE:MODE NORMAL;TIMER 0,0,0; ; STATE START<RMT>

The response information from the higher-level query can also be sent back to the controller as a command message, so that the setting information returned at that time can be re-validated. It should be noted that not all messages from a command group will be returned. For example, for some upper level query requests, setting information that is not currently used will not be returned.

2.2.4 Writing of Command Header

The instructions for the command header are as follows:

- The command mnemonic is not case-sensitive. For example, INPut can write to input or INPUT.
- The lowercase in the command mnemonic can be omitted. For example, INPut can write to INPu or INP.
- The question mark at the end of the command header is used to indicate the request function; the question mark cannot be omitted. For example, INPut? can be abbreviated as INP? .
- If the numerical value following the command mnemonic is omitted when writing, then the value is considered to be 1. For example, ELEMent writes to ELEM, that recognizes "ELEMent1".
- The command or parameter written in boxes can be omitted. For example, [:INPut]:SCALing[:STATe]ON" can write to "SCAL ON.", but in the upper level command, the command or parameter in the last box cannot be omitted.

  For example, "SCALing?" is different from "SCALing: STATe?"

2.3 Response

The response message returned by the power meter, which has two forms.

- The response message consists of the head and data. This response message has command header, so it can be used as the command message directly.

  For example,

  : INTEGrate: MODE? <PMT> -> :INTEGRATE:MODE NORMAL<RMT>;

- The response message consist of data only, and no the command header, so this response message cannot be used as the command message.
  For example,
  INTEGrate: STATe? <PMT> -> RESET<RMT>.

  If you want to set the response message returned by the power meter not be accompanied by a command header, you can use the "COMMunicate: HEADer" command to configure the power meter. The returned response header usually omits the part of lowercase and content in the box.

2.4 Data

The data is located after the command header and is separated from the command header by a space. The data contains the conditions and values, as shown in Table 2.2

Table 2.2 Data Explanations

Data	Description	Example
<Decimal>	The value expressed in decimals	VT ratio setting ->[:INPut]:SCALing:VT 100)
<Voltage><Current><Time>	Value of the physical quantity	Voltage range setting ->[:INPut]:VOLTage:RANge 150V)
<Register>	Register value can express in binary system, octonary number system, decimals and hexadecimal	Extend event register value ->:STATUS:EESE #HFE
<Character data>	Predefined character string	Measurement mode ->[:INPut]:MODE {RMS|VMEan|DC}
<Boolean>	To indicate turn on or off. It usually is ON, OFF or a value.	Set data to hold mode ->:HOLD ON
<String data>	Arbitrary character string	Return model ->:SYSTEM:MODEL "UTE310")
<Block data>	Including multiple 8-bit data	Measured data in response message -> #40012ABCDEFGHIJKL)

2.4.2 Multiplication and Unit Symbol

The user can use the multiplication and its unit symbol, as shown in Table 2.3.
The multiplier symbols are not case sensitive.

Table 2.3 Multiplication and Unit Symbol

Symbol	Multiplier
EX	10^{18}
PE	10^{15}
T	10^{12}
G	10^{9}

MA	10^{6}
K	10^{3}
M	10^{-3}
U	10^{-6}
N	10^{-9}
P	10^{-12}
F	10^{-15}

The unit symbol is shown in Table 2.4.

Table 2.4 Unit and Symbol

Symbol	Unit	Description
V	volt	voltage
A	ampere	Current
S	second	Time

If the multiplication and unit are both omitted, then use the basic unit by default (U, A, S). The response message returned by the power meter usually uses the format of <NR3>, and uses the basic format instead of multiplication and unit.

。

2.4.3 <Decimal> Value

<Decimal> indicates the data is decimals. In ANSI X3.42-1975., use "NR" expresses the different format of decimals, as shown in Table 2.5.

Table 2.5 Decimal Value

Symbol	Meaning	Example
<NR1>	Integer	125, - 1, + 100
<NR2>	Fixed number	125.0, - .90, + 001.
<NR3>	Floating number	125.0E+0 , -9E-1 , +.1E4
<NRf>	Any one of <NR1> to <NR3>	

Explanations

- The power meter can receive decimal values sent by the controller in the format of NR1~NR3;
- The format of <Decimal> data value returned by the power meter is the same as the format of <Decimal> data in the command request;
- The plus mark "+" in the format of <NR3>Format can be omitted, but the minus mark "-" cannot be omitted.
- If input data is over the range, then the data will be recognize to the value that close to the range;
- If there are too many valid bits of data, the excess valid bits of data will be discarded directly.

2.4.4 Register Data

<Register> indicates the data is an integer that can be expressed in binary, decimal, octal and hexadecimal system. When each bit of an integer data has a special meaning, you can use <Register> to represent the data.

Table 2.6 Register Data

Format	Example
<NRf>	1
#H<hexadecimal, including 0 ~ 9 and A~ F>	#H0F
#Q<octonary number system consists of 0~7>	#Q777
#B<binary system consists of 0~1>	#B001100

The <Register> writing is not case sensitive, and the <Register> data in a response message usually uses the <NR1> format.

2.4.5 <Character Data>

<Character data> is a character string with a special meaning that is typically used to represent an operation or a function parameter that is available for selection. The writing format of <Character Data> see the section "2.2.4 writing of command header".

2.4.6 <Boolean Data>

<Boolean Data > is used to indicate the state of ON and OFF, it can be ON, OFF or an integer. If <Boolean data> uses the integer format, then when <Boolean> data round off number is 0, it is considered to be OFF state, otherwise it is considered to be ON state.
The response message usually returns 1 for ON and 0 for OFF.

2.4.7 <string Data>

<String data> is different from <Character data>, which no special meaning, it can be any one of character string. <String data> must be within a pair of single or double quotes, as shown in Table 2.7.

Table 2.7<String data>

Format	Example
<String data>	'ABC' "IEEE488.2-1992"

2.4.8 <Block Data>

<Block data> consists of 8-bit, only for response message. The syntax of <Block data> is shown in Table 2.8.

Table 2.8 Syntax

Format	Example
#N<Byte number of Data> < Byte sequence of Data >	#800000010ABCDEFGHIJ

Format explanation of Table 2.8

- "#N" indicates the length of data byte in <Block data>.
  For example, 8 in "#800000010ABCDEFGHIJ" means the length of "00000010" is 8 digits, and "00000010" means the byte number of data, indicating the following byte sequence of data A-J is 10 characters of data;
- "Byte number of data" indicates the byte number in data and express in decimals;
- "Byte sequence of data" indicates the actual data byte, which is "ABCDEFGHIJ".

Chapter 3 SCPI Command Set

3.1 Introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) is a standard command set that builds on the existing standards IEEE 488.1 and IEEE 488.2, The standard instrument programming language follows the floating-point arithmetic rules in the IEEE754 standard, the ISO646 Information exchange 7-bit encoding symbol (equivalent to ASCII programming) and other standards. This section describes the format, notation, parameters, and abbreviation rules for SCPI commands.

3.2 D/A Output Command

This command set is to process the D/A output

3.2.1 ADOUTput?

1. Function
   Query all the D/A output settings.
1. Syntax
   : AOUTput?

3.2.2 AOUTput[:NORMal]:PRESet

1. Function
   Set the D/A output setting to its default value.
2. Syntax
   : AOUTput[:NORMal]:PRESet {NORMal|INTEGrate}
2. Example
   : AOUTPUT:NORMAL:PRESET NORMAL

3.2.3 AOUTput[:NORMal]:CHANnel<x>

1. Function
   Set or query the D/A output function.
2. Syntax
   :AOUTput[:NORMal]:CHANnel<x> {NONE|<Function>}
   :AOUTput[:NORMal]:CHANnel<x>?
   <x> = 1 to 12 (output channel)
   NONE = no output item
   <Function>= {U|I|P|S|Q|LAMBda|PHI|FU|FI|WH|WHP|WHM|AH|AHP|AHM|MATH|UPeak|IPeak}
3. Example
   :AOUTPUT:NORMAL:CHANNEL1 U
   :AOUTPUT:NORMAL:CHANNEL1?
   -> :AOUTPUT:NORMAL:CHANNEL1 U

3.2.4 AOUTput[:NORMal]:IRTime

1. Function
   Set or query the integration time in the integral value D/A output.
2. Syntax
   : AOUTput[: NORMal]: IRTime {<NRf>, <NRf>, <NRf>}
   : AOUTput[: NORMal]: IRTime?
   {<NRf>, <NRf>, <NRf>} = 0, 0, 0 to 10000, 0, 0
   From the first <NRf> = 0 to 10000 (hour)
   From the second <NRf> = 0 to 59 (minute)
   From the third <NRf> = 0 to 59 (second)
3. Example
   : AOUTPUT: NORMAL: IRTIME 1, 0, 0
   : AOUTPUT: NORMAL: IRTIME?
   ->: AOUTPUT: IRTIME 1, 0, 0

3.2.5 AOUTput[:NORMal]:MODE<x>

1. Function
   Set or query the D/A range mode.
2. Syntax
   : AOUTput[: NORMal]: MODE<x> {FIXed|MANual|COMPare}
   : AOUTput[: NORMal]: MODE<x>?
   From <x> = 1 to 4 (output channel)
3. Example
   : AOUTPUT: NORMAL: MODE1 FIXED
   : AOUTPUT: NORMAL: MODE1?

->: AOUTPUT: NORMAL: MODE1 FIXED

Explanations

- FIXed fixed range (default value)

When receiving the rated value of each measurement function, it outputs +5V.

- MANual (manual mode)

When +5 V and -5 V used as the D/A output, the display value of measurement function can be set to any value you choose.

It makes the the D/A output can zoom out the D/A of each channel (zoom-in or zoom-out).

- COMPare (compare mode)

By comparing with the comparator limits, the instrument outputs +5 V, 0 V or -5 V. The relay can be driven by using the output analog voltage

3.2.6 AOUTput[:NORMal]:RATE<x>

1. Function

Set or query the maximum and minimum value when the D/A output in manual mode.

Set or query the upper and lower limit when the D/A output in comparator mode.

2. Syntax

: AOUTput[:NORMal]:RATE<x> {<NRf>, <NRf>}

: AOUTput[:NORMal]:RATE<x>?

<x> = 1 to 4 (output channel)

<NRf> = -9.999E+12 ~ 9.999E+12

3. Example

: AOUTPUT: NORMAL: RATE1 100,-100

: AOUTPUT: NORMAL: RATE1?

->: AOUTPUT: NORMAL: RATE1 100.0E+00,-100.0E+00

Explanations

- When the D/A output setting sets to the manual mode, (:AOUTput[:NORMal]:MODE<x> MANual) sets to output the rated output o +5V， and then sets to output -5V.
- When the D/A output setting sets to the compare mode, (: AOUTput[:NORMal]:MODE<x> COMPare) sets the upper limit and then sets the lower limit.
- When the D/A output setting sets to the fixed manual, (:AOUTput[:NORMal]:MODE<x> FIXed)， there is no need to set a rate value (these value values do not affect the output).

3.3 Communication Command

This command group is relate to communication. The function is corresponding to this group of commands cannot be executed by the sub-key of the power meter.

3.3.1 :COMMunicate?

1. Function
   Query all the communication settings.
2. Syntax
   : COMMunicate?

3.3.2 :COMMunicate:HEADer

1. Function
   Set or query whether command header is returned in the query response.
2. Syntax
   : COMMunicate: HEADer {<Boolean>}
   : COMMunicate: HEADer?
3. Example
   : COMMUNICATE: HEADER ON
   : COMMUNICATE: HEADER? ->: COMMUNICATE: HEADER 1
4. Explanations
   Response message with returned command header ": INPUT: VOLTAGE: RANGE 150.0E+00"
   Response message without command header "150.0E+00"

3.3.3 :COMMunicate:LOCKout

1. Function
   Set or delete the local key lock settings.
2. Syntax
   : COMMunicate: LOCKout {<Boolean>}
   : COMMunicate: LOCKout?
3. Example
   : COMMUNICATE: LOCKOUT ON
   : COMMUNICATE: LOCKOUT? ->: COMMUNICATE: LOCKOUT 1

3.4 Harmonic Command

The group commands and harmonics are related.

3.4.1 :HARMonics?

1. Function
   Query all the harmonic measurement settings.
2. Syntax
   : HARMonics?

3.4.2 :HARMonics: MODE {NORMal|IEC}

1. Function
   Select the harmonic measurement mode to Normal or IEC.
2. Syntax
   : HARMonics: MODE {NORMal|IEC}
   NORMal (normal mode), IEC mode
3. Example
   : HARMonics: MODE IEC

3.4.3 :HARMonics: MODE?

1. Function
   Query the harmonic measurement mode.
2. Syntax
   :HARMonics: MODE?
3. Example
   : HARMonics: MODE? ->: HARMonics: MODE NORMAL

3.4.4 :HARMonics:PLLSource

1. Function
   Set or query the PLL source.
2. Syntax
   : HARMonics: PLLSource {U|I}
   : HARMonics: PLLSource?
3. Example
   : HARMONICS: PLLSOURCE U1
   : HARMONICS: PLLSOURCE? ->: HARMONICS: PLLSOURCE U1

3.4.5 :HARMonics:ORDer

1. Function
   Set or query the maximum and minimum harmonic order of the harmonic anaylsis.
2. Syntax
   : HARMonics: ORDer {<NRf>, <NRf>}
   : HARMonics: ORDer?
   The first <NRf> = 1 (the minimum harmonic order of harmonic analysis, fix to 1)
   The second <NRf> = 1~50 (the maximum harmonic order of harmonic analysis)
3. Example
   : HARMONICS: ORDER 1, 50
   : HARMONICS: ORDER? ->:HARMONICS1: ORDER 1, 50

3.4.6 :HARMonics:THD

1. Function
   Set or query THD calculation formula.
2. Syntax
   : HARMonics: THD {TOTal|FUNDamental}
   : HARMonics: THD?
3. Example
   : HARMONICS: THD FUNDAMENTAL
   : HARMONICS: THD? ->: HARMONICS: THD FUNDAMENTAL

3.5 Data Hold Command

This command group is used to execute and output the data hold function.

3.5.1 :HOLD

1. Function
   Set or query the data hold state (on or off).
2. Syntax
   : HOLD {<Boolean>}
   : HOLD?
3. Example
   : HOLD OFF
   : HOLD? ->: HOLD 0

3.6 Input Command

This command group is used to execute the function that related to the measurement conditions of the input unit.

3.6.1 :INPut?

1. Function
   Query all the input settings.
2. Syntax
   : INPut?

3.6.2 [:INPut]:CFACtor

1. Function
   Set or query the crest factor.
2. Syntax
   [: INPut]: CFACtor {<NRf>}
   [: INPut]: CFACtor?
3. Example
   : INPUT: CFACTOR 3
   : INPUT: CFACTOR? ->: INPUT: CFACTOR 3

3.6.3 [:INPut]:WIRing

Set or query the wiring.

1. Syntax
   [: INPut]: WIRing {(P1W2|P1W3|P3W3|P3W4)}
   [: INPut]: WIRing?
   P1W2 = single phase two wires [1P2W]
   P1W3 = single phase three wires [1P3W]
   P3W3 = three-phase three wires [3P3W]
   P3W4 = three-phase four wires [3P4W]
   Note: for the UTE310 power meter, it can only select P1W2.
2. Example
   : INPUT: WIRING P1W2
   : INPUT: WIRING? ->: INPUT: WIRING P1W2

3.6.4 [:INPut]:MODE

1. Function
   Set or query the measurement mode of voltage and current.
2. Syntax
   [:INPut]:MODE {RMS|VMEan|DC}
   [:INPut]:MODE?
3. Example
   :INPUT:MODE RMS
   :INPUT:MODE? ->:INPUT:MODE RMS

3.6.5 [:INPut]:VOLTage?

1. Function
   Query all the voltage settings.
2. Syntax
   [:INPut]:VOLTage?

3.6.6 [:INPut]:VOLTage:RANGe

1. Function
   Set or query the voltage range of all the input units.
2. Syntax
   [:INPut]:VOLTage:RANGe {<Voltage>}
   [:INPut]:VOLTage:RANGe?
   When the crest factor is 3,
   <Voltage>=15, 30, 60,150, 300, 600V
   When the crest factor is 6 or 6A,
   <Voltage> = 7.5, 15, 30, 75, 150, 300V
3. Example
   :INPUT:VOLTAGE:RANGE 600V
   :INPUT:VOLTAGE:RANGE? ->:INPUT:VOLTAGE:RANGE 600.0E+00

3.6.7 [:INPut]:VOLTage:AUTO

1. Function
   Set or query the state (on or off) of auto voltage range
2. Syntax
   [:INPut]:VOLTage:AUTO {<Boolean>}
   [:INPut]:VOLTage:AUTO?

3. Example
   : INPUT: VOLTAGE: AUTO ON
   : INPUT: VOLTAGE: AUTO? ->: INPUT: VOLTAGE: AUTO 1

3.6.8 [:INPut]:VOLTage:CONFig

1. Function
   Set or query the valid jump range of automatic range.
2. Syntax
   [: INPut]: VOLTage: CONFig {ALL|<Voltage> [, <Voltage>] [, <Voltage>]...}
   [: INPut]: VOLTage: CONFig?
   ALL= all the ranges are valid
   <Voltage> range refer to the command (: INPut: VOLTage: RANGe).
3. Example
   : INPUT: VOLTAGE: CONFIG ALL
   : INPUT: VOLTAGE: CONFIG? ->: INPUT: VOLTAGE: CONFIG ALL
   : INPUT: VOLTAGE: CONFIG 600, 150, 15
   : INPUT: VOLTAGE: CONFIG? ->: INPUT: VOLTAGE: CONFIG600.0E+00,
   150.0E+00, 15.0E+00

4. The user can give the enable voltage range in the parameter. To enable all ranges, configure the parameter to "ALL".

3.6.9 [:INPut]:VOLTage:POJump

1. Function
   Set or query target range when the voltage peak is over the range.
2. Syntax
   [: INPut]: VOLTage: POJump {OFF|<Voltage>}
   [: INPut]: VOLTage: POJump?
   OFF = not jump to the target voltage range
   <Voltage> range refer to the command (: INPut: VOLTage: RANGe)
3. Example
   : INPUT: VOLTAGE: POJUMP 600V
   : INPUT: VOLTAGE: POJUMP? ->: INPUT: VOLTAGE: POJUMP 600.0E+00

3.6.10 [:INPut]:CURRent?

1. Function
   Query all the electrical parameter of current measurement.
2. Syntax

[:INPut]:CURRent?

3.6.11 [:INPut]:CURRent:RANGe

1. Function
   Set or query the current range.
2. Syntax
   [:INPut]:CURRent:RANGe {<Current>|(EXTernal,<Voltage>)}
   [:INPut]:CURRent:RANGe?
   （1） For the direct current input
   When the crest factor sets to 3,
   <Current> = 5, 10, 20, 50, 100, 200, 500(mA), 1, 2, 5, 10, 20(A)
   When the crest factor sets to 6 or 6A,
   <Current> =2.5, 5, 10, 25, 50, 100, 250(mA), 0.5, 1, 2.5, 5, 10(A)
   （2） For the external current sensor input=
   When the crest factor sets to 3,
   < Voltage > = 2.5, 5, 10(V)(/EX1)
   < Voltage > = 50, 100, 200, 500(mV), 1, 2(V)(/EX2)
   When the crest factor sets to 6 or 6A,
   <Voltage> = 1.25, 2.5, 5(V)(/EX1)
   <Voltage> = 25, 50, 100, 250(mV), 0.5, 1(V)(/EX2)
3. Example
   : INPUT: CURRENT: RANGE 20A
   : INPUT: CURRENT: RANGE? ->: INPUT: CURRENT: RANGE 20.0E+00
   : INPUT: CURRENT: RANGE EXTERNAL, 10V
   : INPUT: CURRENT: RANGE? ->: INPUT: CURRENT: RANGE EXTERNAL, 10.0E+00

3.6.12 [:INPut]:CURRent:AUTO

1. Function
   Set or query the state (on/off) of automatic range.
2. Syntax
   [:INPut]:CURRent:AUTO {<Boolean>}
   [:INPut]:CURRent:AUTO?
3. Example
   : INPUT: CURRENT: AUTO ON

3.6.13 [:INPut]:CURRent:CONFig

1. Function
   Set or query the valid jump range of automatic current range.

2. Syntax
   [: INPut]: CURRent: CONFig {ALL|<Current> [, <Current>] [, <Current>]...}
   [: INPut]: CURRent: CONFig?
   ALL = all the ranges are valid
   <Current> = refer to the command (: INPut: CURRent: RANGe)
3. Example
   : INPUT: CURRENT: CONFIG ALL
   : INPUT: CURRENT: CONFIG? ->: INPUT: CURRENT: CONFIG ALL
   : INPUT: CURRENT: CONFIG 20, 5, 1
   : INPUT: CURRENT: CONFIG? ->: INPUT: CURRENT: CONFIG 20.0E+00, 5.0E+00, 1.0E+00
4. Explanation
   The user can give the enable current range in the parameter. To enable all ranges, configure the parameter to "ALL".

3.6.14 [:INPut]:CURRent: POJump

1. Function
   Set the target range when the current peak is over the range.
2. Syntax
   [: INPut]: CURRent: POJump {OFF|<Current>}
   [: INPut]: CURRent: POJump?
   OFF= not jump to the target current range
   <Current> = refer to the command (: INPut: CURRent: RANGe)
3. Example
   : INPUT: CURRENT: POJUMP 20A
   : INPUT: CURRENT: POJUMP? ->: INPUT: CURRENT: POJUMP 20.0E+00

3.6.15 [:INPut]:CURRent:EXTSensor:CONFig

1. Function
   Set or query the jumping current range when the external current sensor is over the range.
2. Syntax
   [: INPut]: CURRent: EXTSensor: CONFig {ALL|<Voltage> [, <Voltage>] [, <Voltage>]...}
   [: INPut]: CURRent: EXTSensor: CONFig?
   ALL = all the ranges are valid
   <Voltage> = refer to the command (: INPut: CURRent: RANGe)
3. Example
   : INPUT: CURRENT: EXTSENSOR: CONFIG ALL
   : INPUT: CURRENT: EXTSENSOR: CONFIG? ->: INPUT: CURRENT: EXTSENSOR: CONFIG ALL
   : INPUT: CURRENT: EXTSENSOR: CONFIG 2, 0.5, 0.1
   : INPUT: CURRENT: EXTSENSOR: CONFIG? ->: INPUT: CURRENT: EXTSENSOR: CONFIG 2.00E+00,500.0E-03,100.0E-03

3.6.16 [:INPut]:CURRent:EXTSensor:POJump

1. Function
   Set or query the jumping current range when the external current sensor is over the range.
2. Syntax
   [:INPut]:CURRent:EXTSensor:POJump {OFF|<Voltage>}
   [:INPut]:CURRent:EXTSensor:POJump?
   OFF= not execute the range jumping
   <Voltage> = the range jumping refer to the command (:INPut:CURRent:RANGe)
3. Example
   :INPUT:CURRENT:EXTSENSOR:POJUMP 2V
   :INPUT:CURRENT:EXTSENSOR:POJUMP? ->:INPUT:CURRENT:EXTSENSOR:POJUMP 2.00E+00

3.6.17 [:INPut]:CURRent:SRATio?

1. Function
   Query the conversion ratio of the external current sensor of all input unit.
2. Syntax
   [:INPut]:CURRent:SRATio?

3.6.18 [:INPut]:CURRent:SRATio[:ALL]

1. Function
   Set the conversion ratio of the external current sensor of all input unit.
2. Syntax
   [:INPut]:CURRent:SRATio[:ALL]{<NRf>}
   <NRf> = 0.001~ 9999.
3. Example
   :INPUT:CURRENT:SRATIO:ALL 10

3.6.19 [:INPut]:RCONfig

1. Function
   Set or query the state (on/off) of range configuration function.
2. Syntax
   [:INPut]:RCONfig {<Boolean>}
   [:INPut]:RCONfig?
3. Example
   :INPUT:RCONFIG OFF
   :INPUT:RCONFIG? ->:INPUT:RCONFIG 0

3.6.20 [:INPut]: SCALing?

1. Function
   Query all the scaling settings.
2. Syntax
   [: INPut]: SCALing?

3.6.21 [:INPut]: SCALing[:STATe]

1. Function
   Set or query the state (on/off) of conversion ratio function.
2. Syntax
   [: INPut]: SCALing [: STATe] {<Boolean>}
   [: INPut]: SCALing: STATe?
3. Example
   : INPUT: SCALING: STATE OFF
   : INPUT: SCALING: STATE? ->: INPUT: SCALING: STATE 0

3.6.22 [:INPut]:SCALing:{VT|CT|SFACtor}?

1. Function
   Query the VT conversion ratio, CT conversion ratio and power factor.
2. Syntax
   [: INPut]: SCALing :{ VT|CT|SFACtor}?

3.6.23 [:INPut] : SCALing : {VT|CT|SFACtor}[:ALL]

1. Function
   The VT conversion ratio, CT conversion ratio, and power factor of all input units are set globally.
2. Syntax
   [: INPut]: SCALing :{ VT|CT|SFACtor} [: ALL] {<NRf>}
   <NRf>= 0.001 ~ 9999.
3. Example
   : INPUT: SCALING: VT: ALL 1

3.6.24 [:INPut] :SYNChronize

1. Function

Set or query the synchronization source.

2. Syntax
   [: INPut]: SYNChronize {VOLTage|CURRent|OFF}
3. Example
   : INPUT: SYNCHRONIZE VOLTAGE
   : INPUT: SYNCHRONIZE? ->: INPUT: SYNCHRONIZE VOLTAGE

3.6.25 [:INPut] : FILTer?

1. Function
   Query all input filter settings.
2. Syntax
   [: INPut]: FILTer?

3.6.26 [:INPut]:FILTer:LINE

1. Function
   Set or query the line filter.
2. Syntax
   [: INPut]: FILTer: LINE {<Boolean>}
   [: INPut]: FILTer: LINE?
3. Example
   : INPUT: FILTER: LINE OFF
   : INPUT: FILTER: LINE? ->: INPUT: FILTER: LINE 0

3.6.27 [:INPut]:FILTer: FREQuency

1. Function
   Set or query the frequency filter.
2. Syntax
   [: INPut]: FILTer: FREQuency {<Boolean>}
   [: INPut]: FILTer: FREQuency?
3. Example
   : INPUT: FILTER: FREQUENCY OFF
   : INPUT: FILTER: FREQUENCY? ->: INPUT: FILTER: FREQUENCY 0

3.6.28 [:INPut]:POVer?

1. Function
   Query the peak overrange.
2. Syntax
   [: INPut]: POVer?

3. Example
   : INPUT: POVER? -> 0
4. Explanations
   The peak overrange message for each input unit is shown below. The response message returns the decimal number of the sum of the bit values, for example, 1 in the response message indicates that a peak overrange has occurred in U1.

7	6	5	4	3	2	1	0
						I1	U1

3.6.29 [:INPut]: CRANge?

1. Function
   Query the state of overrange.
2. Syntax
   [: INPut]: CRANge?
3. Example
   : INPUT: CRANGE? -> 0
4. Explanations
   Check the range state, as shown in the following table. The response message returns the decimal number of the sum of the bit values, for example, 64 in the response message indicates that the current is overranged.

7	6	5	4	3	2	1	0
AP	AO	AH	AL	VP	VO	VH	VL

The explanations for each state in the above table is as follows.

- VL: the voltage equals to or lower than the automatic range
- VH: the voltage equals to or higher than the automatic range
- VO: the voltage is overrange
- VP: the voltage is overrange
- AL: the current equals to or lower than the automatic range
- AH: the current equals to or higher than the automatic range
- AO: the current is overrange
- AP: the current is overrange

3.7 Integration Command

3.7.1 :INTEGrate?

1. Function
   Query all the integration settings.

2. Syntax
   : INTEGrate?

3.7.2 :INTEGrate:MODE

1. Function
   Set or query the integration mode.
2. Syntax
   : INTEGrate: MODE {NORMal|CONTinuous}
   : INTEGrate: MODE?
   NORMal = standard integration mode
   CONTinuous = continuous integration mode
3. Example
   : INTEGRATE: MODE NORMAL
   : INTEGRATE: MODE? ->: INTEGRATE: MODE NORMAL

3.7.3 :INTEGrate: TIMer

1. Function
   Set or query the timer of integration.
2. Syntax
   : INTEGrate: TIMer {<NRf>, <NRf>, <NRf>}
   : INTEGrate: TIMer?
   {<NRf>, <NRf>, <NRf>}= 0, 0, 0 ~ 10000, 0, 0
   The first <NRf>= 0~10000 (hour)
   The second <NRf>= 0~59 (minute)
   The third <NRf>= 0~59 (second)
3. Example
   : INTEGRATE: TIMER 1, 0, 0
   : INTEGRATE: TIMER? ->: INTEGRATE: TIMER 1, 0, 0

3.7.4 :INTEGrate: STARt

1. Function
   Start the integration.
2. Syntax
   : INTEGrate: STARt
3. Example
   : INTEGRATE: START

3.7.5 :INTEGrate: STOP

1. Function

Stop the integration.

2. Syntax
   : INTEGrate: STOP
3. Example
   : INTEGRATE: STOP

3.7.6 :INTEGrate: RESet

1. Function
   Reset the integration.
2. Syntax
   : INTEGrate: RESet
3. Example
   : INTEGRATE: RESET

3.7.7 :INTEGrate: STATe?

1. Function
   Query the state of the integration.
2. Syntax
   : INTEGrate: STATe?
3. Example
   : INTEGRATE: STATE? -> RESET
4. Explanations
   The response returned by the instruction is as follows
   RESet = Rest the integration
   STARt = Start the integration
   STOP = Stop the integration

3.8 Mathematic Operation Command

3.8.1 :MATH?

1. Function
   Set or query the mathematic operation.
2. Syntax
   : MATH {EFFiciency|CFU1|CFI1|ADD|SUB|MUL|DIV|DIVA|DIVB|AVW1}
3. Example
   : MATH CFU1
   : MATH? ->: MATH CFU1
4. Explanations

The explanation of the operation formula at the above
CFU: the voltage peak factor
CFI: the current peak factor
ADD: A＋B
SUB: A—B
MUL: A× B
DIV: A/B
DIVA: A/B^2
DIVB: A^2/B
AVW: integration average active power

3.9 Measurement Command

3.9.1 :MEASure?

1. Function
   Query the output setting of all measured and calculated data.
2. Syntax
   : MEASure?

3.9.2 :MEASure:AVERaging?

1. Function
   Query all the average settings.
2. Syntax
   : MEASure: AVERaging?

3.9.3 :MEASure:AVERaging[:STATe]

1. Function
   Set or query the state (on/off) of average function.
2. Syntax
   : MEASure: AVERaging [: STATe] {<Boolean>}
   : MEASure: AVERaging: STATe?
3. Example
   : MEASURE: AVERAGING: STATE ON
   : MEASURE: AVERAGING: STATE? ->: MEASURE: AVERAGING: STATE 1

3.9.4 :MEASure:AVERaging:TYPE

1. Function
   Set or query the average type.
2. Syntax
   : MEASure: AVERaging: TYPE {LINear|EXPonent}
   : MEASure: AVERaging: TYPE?
3. Example
   : MEASURE: AVERAGING: TYPE LINEAR
   : MEASURE: AVERAGING: TYPE? ->: MEASURE: AVERAGING: TYPE LINEAR

3.9.5 :MEASure:AVERaging: COUNt

1. Function
   Set or query the average factor.
2. Syntax
   : MEASure: AVERaging: COUNt {<NRf>}
   : MEASure: AVERaging: COUNt?
   <NRf>= 8, 16, 32, 64 (move average constant or exponential average factor)
3. Example
   : MEASure: AVERAGING: COUNT 8
   : MEASure: AVERAGING: COUNT? ->: MEASure: AVERAGING: COUNT 8
4. Explanation
   The average function can be enabled in all harmonic measurement, but only valid when the average mode is exponential average.

3.9.6 :MEASure: MHOLd

1. Function
   Set the state (on/off) of the maximum hold.
2. Syntax
   : MEASure: MHOLd {<Boolean>}
   : MEASure: MHOLd?
3. Example
   : MEASURE: MHOLD ON
   : MEASURE: MHOLD? ->: MEASURE: MHOLD 1

3.10 Numeric Command

3.10.1 :NUMeric?

1. Function
   Query the output setting of all numeric data.
2. Syntax
   : NUMeric?

3.10.2 :NUMeric:FORMat

1. Function
   This command can set or query the output format of data, the detail refer to “the format of numeric data”.
   （1） ASCii
   Physical quantity values are output as <NR3>, except for integration with time-TIME, which is output as <NR1>Format. Comma separates data items.
   （2） FLOat
   A header may precede individual blocks of numeric data, for example "#240" or "#3208". Physical quantities in IEEE standard single-precision floating-point (4-byte) format immediately following the header. Byte order within the data is MSB before LSB.
2. Syntax
   : NUMeric: FORMat {ASCii|FLOat}
   : NUMeric: FORMat?
3. Example
   : NUMERIC: FORMAT ASCII
   : NUMERIC: FORMAT? ->: NUMERIC: FORMAT ASCII

3.10.3 :NUMeric: NORMal?

1. Function
   Query the output setting of all numeric data.
2. Syntax
   : NUMeric: NORMal?
3. Explanation
   The command: NUMeric [: NORMal]: ITEM<x> outputs the number of data, it determined by the command: NUMeric [: NORMal] NUMber.

3.10.4 :NUMeric[:NORMal]:VALue?

1. Function
   Query the numeric data, the format of each numeric data, the detail refer to "the format of numeric data".
2. Syntax
   : NUMeric [: NORMal]: VALue? {<NRf>}
   <NRf> = 1~255
   If the parameter <NRF> is assigned, then only output the specified numeric data. If the parameter <NRF> is omitted, then output the numeric data from 1 to N in sequence. N is determined by the command: NUMeric [: NORMal]: NUMber.
3. Example
   When the parameter **<NRf>** is assigned,
   : NUMERIC: NORMAL: VALUE? 1-> 103.79E+00
   When the parameter **<NRf>** is omitted,
   : NUMERIC: NORMAL: VALUE? ->103.79E+00, 1.0143E+00,105.27E+00, ... 50.001E+00
   When the command: NUMeric: FORMat sets to **{FLOat}**
   : NUMERIC: NORMAL: VALUE? -> #N (N-bit byte)(data byte sequence)

3.10.5 :NUMeric[:NORMal]:NUMber

1. Function
   Set or query the output number of the numeric data by the command: NUMeric [: NORMal]: VALue?
2. Syntax
   : NUMeric [: NORMal]: NUMber {<NRf>|ALL}
   : NUMeric [: NORMal]: NUMber?
   <NRf> = 1~ 255(ALL)
3. Example
   : NUMERIC: NORMAL: NUMBER 10
   :NUMERIC:NORMAL:NUMBER? ->: NUMERIC: NORMAL: NUMBER 10

3.10.6 :NUMeric[:NORMal]: :ITEM<x>

1. Function
   Set or query the specified output numeric data.
2. Syntax
   : NUMeric [: NORMal]: ITEM<x> {NONE| <Function> [, <Element>] [, <Order>]}
   : NUMeric [: NORMal]: ITEM<x>?
   <x> = 1 ~ 255 (item number)
   NONE = no output

<Function> = {U|I|P|S|Q|...}
<Element> = {<NRf>} (UTE310<NRf> = 1)
<Order> = {TOTal|DC|<NRf>}
(<NRf> = 1~50)

3. Example
: NUMERIC: NORMAL: ITEM1 U, 1
: NUMERIC: NORMAL: ITEM1? ->: NUMERIC: NORMAL: ITEM1 U, 1
: NUMERIC: NORMAL: ITEM1 UK, 1, 1
: NUMERIC: NORMAL: ITEM1? ->: NUMERIC: NORMAL: ITEM1 UK, 1, 1
4. Explanations
- If the parameter <Element> is omitted, the parameter <Element> is 1 by default;
- If the parameter <Order> is omitted, the parameter <Order> is 1;
- <Function> refers to "Function Parameter Table";
- For the function that do not require the <Element> and <Order> parameters, the <Element> and <Order> parameters are not returned in the response;
- UTE310 does not execute the parameter <Order> = DC measurement.

3.10.7 :NUMeric[:NORMal]:PRESet

1. Function
The output mode of preset numeric data.
2. Syntax
: NUMeric[:NORMal]:PRESet {<NRf>}
<NRf> = 1~4 (the default value is 2)
3. Example
: NUMERIC: NORMAL: PRESET 1
4. Data Mode Table

Table 3.1 Method 1

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4~255	NONE	

Table 3.2 Method 2

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	S	1
5	Q	1
6	LAMBda	1
7	PHI	1

8	FU	1
9	FI	1
10~255	NONE	

Table 3.3 Method 3

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	S	1
5	Q	1
6	LAMBda	1
7	PHI	1
8	FU	1
9	FI	1
10	UPPeak	1
11	UMPeak	1
12	IPPeak	1
13	IMPeak	1
14	PPPeak	1
15	PMPeak	1
16~255	NONE	

Table 3.4 Method 4

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	S	1
5	Q	1
6	LAMBda	1
7	PHI	1
8	FU	1
9	FI	1
10	UPPeak	1
11	UMPeak	1
12	IPPeak	1
13	IMPeak	1
14	TIME	1
15	WH	1
16	WHP	1
17	WHM	1
18	AH	1

19	AHP	1
20	AHM	1
21~255	NONE	

3.10.8 :NUMeric[:NORMal]:CLEar

1. Function
   Clear the output of numeric data
2. Syntax
   : NUMeric[:NORMal]:CLEar {ALL|<NRf>[,<NRf>]} ALL = clear all the items
   The first <NRf>= 1~ 255 (Number of the first cleared numeric data item.)
   The second <NRf>= 1~ 255 (The number of the last cleared numeric data item, if this parameter is omitted, it is cleared until the last numeric data item.)
3. Example
   : NUMERIC: NORMAL: CLEAR ALL

3.10.9 :NUMeric[:NORMal]:DELete

1. Function
   Delete the output of numeric data.
2. Syntax
   : NUMeric[:NORMal]:DELete {<NRf>[,<NRf>]}
   The first <NRf>= 1~255 (the number of the first numeric data that must be delete.)
   The second <NRf>= 1~255 (the number of the last numeric data that must be delete.)
3. Example
   : NUMERIC: NORMAL: DELETE 1 (Delete ITEM1 and move forward ITEM2 and the following numeric data).
   : NUMERIC: NORMAL: DELETE 1, 3 (Delete from ITEM1 to ITEM3 and move forward ITEM4 and the following numeric data).
4. Explanations
   - After deleting an item of output numeric data, subsequent numeric data items will move forward and fill empty spaces. In addition, subsequent numeric data items moved forward create empty spaces at the end of the queue, which are set to NONE.
   - If the second <NRf> is omitted, then only the output item specified by the first number is deleted.

3.10.10 :NUMeric[:NORMal]:HEADer?

1. Function
   Query the header of numeric data.
2. Syntax

: NUMeric [: NORMal]: HEADer? {<NRf>}
<NRf> = 1~255 (item number)
If <NRF> is set, then only output the data name (data header) of the specified numeric data.
If <NRF> is not set, then output the data name from the first numeric data to the numeric data that assigned by the command NUMeric [: NORMal]: NUMber.

3. Example
When **<NRF>** is set,
: NUMERIC: NORMAL: HEADER? 1 -> U-E1
Not set **<NRF>**
: NUMERIC: NORMAL: HEADER? -> U-E1, I-E1, P-E1

3.10.11 :NUMeric[:NORMal] : LIST?

1. Function
Query the output setting of all harmonic measurement numeric data table.
2. Syntax
: NUMeric: LIST?
3. Explanation
The number of numeric output by the command NUMeric: LIST: ITEM<x> is decided by the command NUMeric: LIST: NUMber.

3.10.12 :NUMeric:LIST:VALue?

1. Function
Query all numeric data data of harmoince measurements.
2. Syntax
: NUMeric: LIST: VALue? {<NRf>}
<NRf>= 1~32 (number)
3. Example
When **NRF** is set,
: NUMERIC: LIST: VALUE? 1
-> 103.58E+00, NAN, 103.53E+00, 0.09E+00, 2.07E+00, 0.04E+00,.., 0.01E+00,0.01E+00
When **NRF** is not set **(: NUMeric: LIST: NUMber** sets to **5)**
: NUMERIC: LIST: VALUE?
-> 103.58E+00, NAN, 103.53E+00, 0.09E+00, 2.07E+00, 0.04E+00,..., 0.00E+00, 0.00E+00
When: NUMeric: FORMat sets to **{FLOat}**
: NUMERIC: LIST: VALUE? -> #N (N-byte number)(data byte sequence)
4. Explanations
- A single numeric data table can consist of up to 52 numeric data, in the order of TOTal, DC, 1st harmonic ..., :NUMeric:LIST:ORDer;
- If <NRF> is assigned, then only the specified numeric data table can be output.
- If <NRF> is omitted, then output from 1 to the data that specified by the command in

sequence.

- UTE310 does not perform DC measurement.

3.10.13 :NUMeric:LIST: NUMber

1. Function
   Set or query the number of the numeric data transmitted by the command: NUMeric: LIST: VALue?
2. Syntax
   : NUMeric: LIST: NUMber {<NRf>|ALL}
   : NUMeric: LIST: NUMber?
   <NRf>= 1~32 (ALL)
   By default, <NRF> value sets to 1.
3. Example
   : NUMERIC: LIST: NUMBER 5
   : NUMERIC: LIST: NUMBER? ->: NUMERIC: LIST: NUMBER 5
4. Explanation
   If the parameter in the command: NUMeric: LIST: VALue? is omitted, then output from 1 to the specified data by this command.

3.10.14 :NUMeric:LIST: ORDer

1. Function
   Set or query the maximum data output of harmonic measurement numeric table.
2. Syntax
   : NUMeric: LIST: ORDer {<NRf>|ALL}
   : NUMeric: LIST: ORDer?
   <NRf> = 1~50(ALL)
3. Example
   : NUMERIC: LIST: ORDER 50
   : NUMERIC: LIST: ORDER? ->: NUMERIC: LIST: ORDER 50

3.10.15 :NUMeric:LIST: SELect

1. Function
   Set or query the data output of harmonic measurement numeric table.
2. Syntax
   : NUMeric: LIST: SELect {EVEN|ODD|ALL}
   : NUMeric: LIST: SELect?
3. Example
   : NUMERIC: LIST: SELECT ALL
   : NUMERIC: LIST: SELECT? ->: NUMERIC: LIST: SELECT ALL
4. Explanations

EVEN = output TOTal, DC, odd harmonic
ODD = output TOTal, DC, even harmonic
ALL = output all

3.10.16 :NUMeric:LIST: ITEM<x>

1. Function
   Query or set the data output of specified harmonic measurement numeric table
2. Syntax
   : NUMeric: LIST: ITEM<x> {NONE|<Function>, <Element>}
   : NUMeric: LIST: ITEM<x>?
   <x> = 1 ~ 32 (number)
   NONE = not output
   <Function> = {U|I|P|PHIU|PHII|UHDF|IHDF|PHDF}
   <Element> = {<NRf>}(<NRf> = 1~3) UTE310 only takes 1
3. Example
   : NUMERIC: LIST: ITEM1 U, 1
   : NUMERIC: LIST: ITEM1? ->: NUMERIC: LIST: ITEM1 U, 1

3.10.17 :NUMeric:LIST: PRESet

1. Function
   Preset the output mode of harmonic measurement numeric data table, the default output mode is 2.
2. Syntax
   : NUMeric: LIST: PRESet {<NRf>}
   <NRf>= 1~4 (the default value is 2)
3. Example
   : NUMERIC: LIST: PRESET 1
4. Explanation
   The output mode of the command: NUMeric: LIST: PRESet refers to Table 3.5, Table 3.6, Table 3.7 and Table 3.8.

Table 3.5 Mode 1

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4~32	NONE	

Table 3.6 Mode 2

ITEM<x>	<Function>	<Element>

1	U	1
2	I	1
3	P	1
4	PHIU	1
5	PHII	1
6~32	NONE	

Table 3.7 Mode 3

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	UHDF	1
5	IHDF	1
6	PHDF	1
7~32	NONE	

Table 3.8 Mode 4

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	P	1
4	PHIU	1
5	PHII	1
6	UHDF	1
7	IHDF	1
8	PHDF	1
9~32	NONE	

3.10.18 :NUMeric:LIST: CLEar

1. Function
   Clear the data output of harmonic measurement numeric table.
2. Syntax
   : NUMeric: LIST: CLEar {ALL|<NRf> [, <NRf>]}
   ALL= clear all the data output
   The first <NRf>= 1~32 (the first data output to be cleared)
   The second <NRf>= 1~32 (the last data output to be cleared)
   Note: If the second <NRf> is omitted, then the clearing start from the first data output until all subsequent data output items are cleared.
3. Example
   : NUMERIC: LIST: CLEAR ALL

3.10.19 :NUMeric:LIST: DELete

1. Function
   Delete the data output of harmonic measurement numeric table.
2. Syntax
   : NUMeric: LIST: DELete {<NRf> [, <NRf>]}
   The first <NRf> = 1~32 (the first output to be deleted)
   The second <NRf> = 1~32 (the last output to be deleted)
   Note: If the second <NRf> is omitted, then only delete the first delete item assigned by the command.
   The following output item will move forward to fill the empty bit after the output is delete. In addition, subsequent numeric data items moved forward create empty spaces at the end of the queue, which are set to NONE.
3. Example
   : NUMERIC: LIST: DELETE 1 (delete ITEM1, and move forward ITEM2 and the following output items)
   : NUMERIC: LIST: DELETE 1, 3 (delete ITEM1~ ITEM3, and move forward ITEM4 and the following output items)

3.10.20 Function Parameter Table

1. Function Parameter Table
   Function in the command: NUMeric [: NORMal]: ITEM<x> {NONE|<Function> [, <Element>] [, <Order>]} is shown in Table 3.9。

Table 3.9 Function parameter table of numeric data command

<Function>	Function	<Order>
U	Voltage U	No
I	Current I	No
P	Active power P	No
S	Apparent power S	No
Q	Reactive power Q	No
LAMBda	Power factor λ	No
PHI	Phase difference Φ	No
FU	Voltage frequency fU	No
FI	Current frequency fI	No
UPPeak	Maximum voltage: U+pk	No
UMPeak	Minimum voltage: U-pk	No
IPPeak	Maximum current: I+pk	No
IMPeak	Minimum current: I-pk	No
PPPeak	Maximum power: P+pk	No
PMPeak	Minimum power: P-pk	No

TIME	Integration time	No
WH	Watt hour WP	No
WHP	Positive watt hour WP+	No
WHM	Negative watt hour WP-	No
AH	Ampere hour q	No
AHP	Positive ampere hour q+	No
AHM	Negative ampere hour q-	No
MATH	Computed value, such as efficiency	No
URANge	Voltage range	
IRANge	Current range	
URMS	True rms voltage Urms	
UMN	Rectified mean voltage calibrated to the rms value Umn	
UDC	Simple voltage average Udc	
URMN	Rectified mean voltage Urmn	
UAC	AC voltage component Uac	
IRMS	True rms current Irms	
IMN	Rectified mean current calibrated to the rms value Imn	
IDC	Simple current average Idc	
IRMN	Rectified mean current Irmn	
IAC	AC current component Iac	
UPeak	Voltage peak Upk	No
IPeak	Current peak Ipk	No
UK	Rms voltage of harmonic order k U(k)	Yes
IK	Rms current of harmonic order k I(k)	Yes
PK	Active power of harmonic order k P(k)	Yes
LAMBDAK	Power factor of harmonic order k λ(k)	Yes (k=1 only)
PHIK	Phase difference between the voltage and current of harmonic order k φ(k)	Yes (k=1 only)
PHIUk	Phase difference between harmonic voltage U(k) and the fundamental wave U(1) φU(k)	Yes (k=2 and higher)
PHIIk	Phase difference between harmonic current I(k) and the fundamental wave I(1) φI(k)	Yes (k=2 and higher)
UHDFk	Harmonic distortion factor of voltage Uhdf(k)	Yes
IHDFk	Harmonic distortion factor of current Ihdf(k)	Yes
PHDFk	Harmonic distortion factor of active power Phdf(k)	Yes
UTHD	Total harmonic distortion of voltage Uthd	No
ITHD	Total harmonic distortion of current Ithd	No

2. Function parameter table of numeric data output command

The parameter table of the command: NUMeric: LIST: ITEM<x> {NONE|<Function>, <Element>} is shown in Table 3.10

Table 3.10 Function parameter table of numeric data output command

<Function>	Function
U	Voltage U( )
I	Current I( )
P	Active power P( )
PHIU	Phase difference between harmonic voltage U(k) and the fundamental wave U(1) φU( )
PHII	Phase difference between harmonic current I(k) and the fundamental wave I(1) φI(k)
UHDF	Harmonic distortion factor of voltage Uhdf( )
IHDF	Harmonic distortion factor of current Ihdf( )
PHDF	Harmonic distortion factor of active power Phdf( )

3.11 Refresh Display Rate Command

3.11.1 :RATE

1. Function
   Set or query the data update rate.
2. Syntax
   : RATE {<Time>}
   : RATE?
   <Time>= 100, 250, 500 (ms), 1, 2, 5, 10, 20 (s)
3. Example
   : RATE 250MS
   : RATE? ->: RATE 250.0E-03

3.11.2 :RATE:AUTO?

1. Function
   Query the data update rate when it is automatic setting.
2. Syntax
   : RATE: AUTO?

3.11.3 :RATE:AUTO:TIMeout

1. Function
   Set or query the automatic delay of the data update rate

2. Syntax
   : RATE: AUTO: TIMeout {<NRf>}
   : RATE: AUTO: TIMeout?
   <NRf> = 1, 5, 10, 20(s)
3. Example
   : RATE: AUTO: TIMEOUT 1
   : RATE: AUTO: TIMEOUT? ->: RATE: AUTO: TIMEOUT 1

3.11.4 :RATE:AUTO:SYNChronize

1. Function
   Set or query the data update rate when it is the synchronization source in automatic mode.
2. Syntax
   : RATE: AUTO: SYNChronize {U<x>|I<x>}
   : RATE: AUTO: SYNChronize?
   <x> = 1(UTE310)
3. Example
   : RATE: AUTO: SYNCHRONIZE U1
   : RATE: AUTO: SYNCHRONIZE? ->: RATE: AUTO: SYNCHRONIZE U1

3.12 Set Related Commands

3.12.1 : STATus?

1. Function
   Query the setting of communication state.
2. Syntax
   : STATus?

3.12.2 : STATus:CONDition?

1. Function
   Query the value of condition register.
2. Syntax
   : STATus: CONDition?
3. Example
   : STATUS: CONDITION? -> 16

3.12.3 : STATus :EESE

1. Function
   Set or query the extension event enable register.
2. Syntax
   : STATus: EESE <Register>
   : STATus: EESE?
3. Example
   : STATUS: EESE #B0000000000000000
   : STATus: EESE? ->: STATUS: EESE 0

3.12.4 :STATus :EESR?

1. Function
   Query the content of the extension event register and delete.
2. Syntax
   : STATus: EESR?
3. Example
   : STATUS: EESR? -> 0

3.12.5 :STATus :ERRor?

1. Function
   Query the code and information of last error.
2. Syntax
   : STATus: ERRor?
3. Example
   : STATUS: ERROR? -> 113,"Underfined Header"
4. Explanation
   If no error is generated, then returns “no error”.

3.12.6 :STATus :FILTer<x>

1. Function
   Set or query the transmission filter.
2. Syntax
   : STATus: FILTer<x> {RISE|FALL|BOTH|NEVer}
   : STATus: FILTer<x>?
   Note: <x> = 1~16
3. Example
   : STATUS: FILTER1 RISE

: STATus: FILTER1? ->: STATUS: FILTER1 RISE

4. Explanation
   Set the relationship between the various bit changes in the condition register and the event-triggered setting. For example, if a bit is set to RISE, then an event occurs when the bit changes from 0 to 1.

3.12.7 :STATus :QENable

1. Function
   Set or query whether messages other than errors are saved to the error queue.
   Note: Currently, the power meter does not support this command.
2. Syntax
   : STATus: QENable {<Boolean>}
   : STATus: QENable?
3. Example
   : STATUS: QENABLE ON
   : STATus: QENABLE? ->: STATus: QENABLE 1

3.12.8 :STATus :QMESsage

1. Function
   Set or query whether return the message when execute the command STATus: ERRor? Query
   Note: Currently, the power meter does not support this command.
2. Syntax
   : STATus: QMESsage {<Boolean>}
   : STATus: QMESsage?
3. Example
   : STATUS: QMESSAGE ON
   : STATus: QMESSAGE? ->: STATus: QMESSAGE 1

3.12.9 :STATus :SPOLl?

1. Function
   Execute the serial query function.
   Note: Currently, the power meter does not support this command.
2. Syntax
   : STATus: SPOLl?
3. Example
   : STATUS: SPOLL? ->: STATUS: SPOLL 0

3.13 Storage Command

3.13.1 :STORe?

1. Function
   Query all the storage settings.
2. Syntax
   : STORe?

3.13.2 :STORe[:STATe]

1. Function
   Set or query the storage state.
2. Syntax
   : STORe[: STATe]{<Boolean>}
   : STORe: STATe?
3. Example
   : STORE: STATE ON
   : STORE: STATE? ->: STORE: STATE 1

3.13.3 :STORe:INTerval

1. Function
   Set or query the storage interval.
2. Syntax
   : STORe: INTerval {<NRf>, <NRf>, <NRf>}
   : STORe: INTerval?
   {<NRf>, <NRf>, <NRf>}= 0, 0, 1~ 99, 59, 59
   The first parameter <NRf> = 0~99 (hour)
   The second parameter <NRf> = 0~59 (minute)
   The third parameter <NRf> = 1~59 (second)
3. Example
   : STORE: INTERVAL 0, 0, 1
   : STORE: INTERVAL? ->: STORE: INTERVAL 0, 0, 1

3.13.4 :STORe: ITEM

1. Function
   There are many measurement data, such as U, I, PF and so on. At the same time, in the

harmonic mode, many of the measurement data also have the harmonic number; therefore, the user only needs to store the data of interest, using this command can store the specified data or query stored data, and support up to 255 specified data storage.

2. Syntax
   : STORe: ITEM<x> {<Function>, ON|OFF [, order_1, order_2...]}
   : STORe: ITEM<x>? <Function>
3. Example
   : STORe: ITEM1 U, ON
   : STORe: ITEM1? U->1
4. Parameter Explanations
   Parameter 1 Function: the name of the measurement function, for example, the name of voltage is U. This parameter cannot be omitted.
   <Function>= {|U|I|P|S|Q|..., see Table 3.13.1}. , see Table 3.13.1 for details}.
   Parameter 2 ON|OFF: enable/disable the storage of the specified FUNC_NAME measurement, this parameter can not be omitted.
   Parameter 3 order_1, ... oder_n: Specify the number of harmonics. Only some of the measurement items will have values at different numbers of harmonics, so if the measurement item does not require a number of harmonics (see 3.13.1 for details), the order_n parameter is ignored; if the measurement item has to specify a number of harmonics and the order_n parameter does not exist, the command will be executed according to the most recent setting of the order_n parameter. In addition, parameter 2 does not affect the configuration of this parameter.
   <Order>= {TOTal|DC|<NRf>} (<NRf> = 1 ~ 50).

Table 3.13.1 Measurement Item and Order Parameter

Measurement Item	SCPI	Whether need order	Name of required data
Voltage	U	No	U
Current	I	No	I
Active power	P	No	P
Reactive power	S	No	S
Apparent power	Q	No	Q
Cont. Power factor	PF	No	LAMBda
Voltage and current angle	Angle	No	PHI
Voltage positive peak	Upeak(+)	No	UPpeak
Voltage negative peak	Upeak(-)	No	UMpeak
Current positive peak	Ipeak(+)	No	IPpeak
Current negative peak	Ipeak(-)	No	IMpeak
Power positive peak	Ppeak(+)	No	PPpeak
Power negative peak	Ppeak(-)	No	PMpeak
Voltage frequency	Ufreq	No	FU
Current frequency	Ifreq	No	FI
Integration time	IntegTime	No	TIME
Watt hour	Wh	No	WH

Positive watt hour	Wh(+)	No	WHP
Negative watt hour	Wh(-)	No	WHM
Ampere-hour	Ah	No	AH
Positive ampere-hour	Ah(+)	No	AHP
Negative ampere-hour	Ah(-)	No	AHM
Mathematical operation	Math	No	MATH
1 harmonic power factor	PFk(1)	No	LAMBdak
Harmonic order	Order	No	ORDer
Total harmonic voltage	U(Total)	No	U*
Total harmonic current	I(Total)	No	I *
Total harmonic power	P(Total)	No	P *
Harmonic voltage of K-order	Uk	Yes	UK
Harmonic current of K-order	Ik	Yes	IK
Harmonic power of K-order	Pk	Yes	PK
Total harmonic voltage distortion	Uthd	No	UTHD
Total harmonic current distortion	Ithd	No	ITHD
Harmonic voltage contains rate of K-order	Uhdfk	Yes	UHDFK
Harmonic current contains rate of K-order	Ihdfk	Yes	IHDFK
Harmonic power contains rate of K-order	Phdfk	Yes	PHDFK
Harmonic voltage phase angle of K-order	Udegreek	Yes	PHIUK
Harmonic current phase angle of K-order K	Idegreek	Yes	PHIIK
Voltage frequency in harmonic mode	UHfreq	No	FU*
Current frequency in harmonic mode	UHfreq	No	FI

3.14 System Command

3.14.1 :SYSTem?

1. Function
   Query all the system settings.
2. Syntax
   :SYSTem?

3.14.2 SYSTem: BRIGhtness

1. Function
   Query and set the brightness of LCD backlight.
2. Syntax
   : SYSTem: BRIGhness {<NRf>}

: SYSTem: BRIGhness?
<NRf> = 10, 20, 30,40,50,60,70,80,90,100

3. Example
: SYSTEM: BRIGHTNESS 100
: SYSTEM: BRIGHTNESS?
->: SYSTEM: BRIGHTNESS 100

3.14.3 SYSTem: BRIGhtness

1. Function
Set or return the beeper state when single click it
2. Syntax
: SYSTem: KEY: BEEPer {<Boolean>|OFF|ON}
: SYSTem: KEY: BEEPer?
3. Example
: SYSTEM: KEY: BEEPER OFF
: SYSTEM: KEY: BEEPER?
->: SYSTEM: KEY: BEEPER 0

3.14.4 SYSTem: KLOCK

1. Function
Set or return the switch state.
2. Syntax
: SYSTem: KLOCk {<Boolean>|OFF|ON}
: SYSTem: KLOCk?
3. Example
: SYSTEM: KLOCK OFF
: SYSTEM: KLOCK?
->: SYSTEM: KLOCK 0

3.14.5 :SYSTem:SERial?

1. Function
Query the serial number.
2. Syntax
: SYSTem: SERial?
3. Example
: SYSTEM: SERIAL? ->: SYSTEM: SERIAL "APA8888888888"

3.14.6 :SYSTem: VERsion [:FIRMware]?

1. Function
   Query the firmware version.
2. Syntax
   : SYSTem: VERsion [: FIRMware]?
3. Example
   : SYSTEM: VERSION: FIRMWARE? -> "V1.01.0003, V1.01.0002, V1.01.0003"

3.14.7 :SYSTem: SUFFix ?

1. Function
   Return the standard part code of the power meter.
2. Syntax
   : SYSTem: SUFFix?
3. Example
   : SYSTEM: SUFFIX? ->: SYSTEM: SUFFIX "-C1-D/C7/EX1/G5/DA4"

3.14.8 :SYSTem:MODel?

1. Function
   Query the model.
2. Syntax
   : SYSTem: MODel?
3. Example
   : SYSTEM: MODEL? ->: SYSTEM: MODEL "UTE310"

3.14.9 :SYSTem: TIMer

1. Function
   Set the system time.
2. Syntax
   : SYSTEM: TIMer {<NRf>, <NRf>, <NRf>}
   {<NRf>,<NRf>,<NRf>}= 0,0,0 ~ 23,0,0
   The first <NRf>= 0~23(hour)
   The second <NRf>= 0~59(minute)
   The third<NRf>= 0~59(second)
3. Example
   : INTEGRATE: TIMER 1, 0, 0

3.14.10 :SYSTem: DATe

1. Function
   Set the data for the system data.
2. Syntax
   : SYSTEM: DATe {<NRf>, <NRf>, <NRf>}
   {<NRf>,<NRf>,<NRf>}= 0,0,0 ~ 2999,0,0
   The first <NRf>= 0~2999(年)
   The second <NRf>= 1~12(月)
   The third<NRf>= 1~31(日)
3. Example
   : INTEGRATE: TIMER 2023, 8, 15

3.15 Communication Command

3.15.1 *CAL?

1. Function
   Exeute the zeroing.
2. Syntax
   *CAL?
3. Example
   *CAL? -> 0

3.15.2 *CLS

1. Function
   Clear the standard event register, extension event register and error queue.
2. Syntax
   *CLS
3. Explanation
   If the command *CLS just lies after the command message terminator, the output queue will also be cleared.

3.15.3 *ESE

1. Function
   Set or query the standard event enable register.
2. Syntax
   *ESE {<NRf>}

*ESE?
<NRf> = 0~255 (the default value is 0)

3. Example
   *ESE 251
   *ESE? -> 251
4. Explanations
   The command expresses the sum of each bit in decimal digit. For example, *ESE 251 can set the standard enable register to be 11111011. In this case, bit 2 of the standard enable register is disabled, which means that bit 5 (ESB) of the status, byte register will not be set to 1 even if a query error is generated. The command *ESE? does not clear the contents of the standard event enable register.

3.15.4 *ESR

1. Function
   Query and delete the standard register.
2. Syntax
   *ESR?
3. Example
   *ESR? -> 32
4. Explanations
   - The sum of the values of each bit is returned in decimal;
   - After sending an SRQ, you can see the type of generated event. For example, if it returns 32, then the table shows that the standard event register is set to 00100000. This also means that the command syntax was incorrect and an SRQ was generated;
   - Querying with *ESR? will clear the contents of the standard event register.

3.15.5 *IDN?

1. Function
   Query the device model.
2. Syntax
   *IDN?
3. Explanation
   Returned information <Manufacturer>, <Model>, <Serial Number>, and <Firmware Version>.

3.15.6 *OPC

1. Function
   When the specific overlap command is executed, bit 0 of the standard event register is set.
   Note: Currently, the power meter does not support this command.
2. Syntax
   *OPC

3. Explanation
   Currently, the power meter does not support the overlap command, which always returns 1.

3.15.7 *OPC?

1. Function
   The ASCII code 1 will be returned when the specific overlap command is completed.
   Note: Currently, the power meter does not support this command.
2. Syntax
   *OPC?
3. Example
   *OPC? -> 1
4. Explanation
   Currently the power meter does not support the overlap command, which always returns 1.

3.15.8 *RST

1. Function
   Initial setup.
2. Syntax
   *RST
3. Example
   *RST
4. Explanation
   All setup parameters except communication settings are restore to the factory settings.

3.15.9 *SRE

1. Function
   Set or query the value of service request enable register.
   Note: Currently, the power meter does not support this command.
2. Syntax
   *SRE {<NRf>}
   *SRE?
   <NRf>= 0~255 (The default value is 0, all bits are disabled at this moment.)
3. Example
   *SRE 239
   *SRE? -> 175 (Invalid setting due to bit 6 MSS.)
4. Explanations
   The parameter value is the decimal value of the sum of each bit. For example, *SRE 239 can set the standard enable register to be binary of 11101111, and bit 4 of the service request enable register is disabled, i.e., bit 4 (MAV) of the status byte register is not set even though

the output queue is not empty. In addition, using the *SRE? command query does not clear the contents of the service request enable register.

3.15.10 *STB?

1. Function
   Query the value of all status byte registers.
   Note: Currently, the power meter does not support this command.
2. Syntax
   *STB?
3. Example
   *STB? -> 4 (The parameter value is the decimal value of the sum of each bit.)
4. Explanations
   - Read register does not require a serial query operation, bit 6 is the MSS bit and not the RQS bit. For example, if returned value is 4, indicating that the status byte register is set to 00000100; this means that the error queue is not empty, i.e., an error has been generated;
   - Execution of the *STB? command does not clear the contents of the status byte register;
   - Read register does not require the execution of a sequential query, bit 6 is an MSS bit, but not an RQS bit.

3.15.11 *TRG?

1. Function
   Execute single measurement (the function equals to pressing the Single key.)
2. Syntax
   *TRG
3. Example
   *TRG

Chapter 4 Overview of Modbus/TCP programming

Modbus/TCP is one of the communication protocols, which used to communicate with PCs, PLCs (sequencers) via Ethernet or other networks by using the TCP/IP protocol. This communication protocol is used to read and write the internal register of the instrument and to exchange data with connected devices.

In the following, a host device such as a PC will be referred to as a client device.

The instrument can connect to an IEEE802.3 network (100BASE-TX/10BASE-T).The Modbus/TCP protocol typically uses port number 502 to perform communication.

4.1 Function and Specification of Modbus/TCP

Ethernet port specification of Modbus/TCP

Item	Specification
Port	1
Connector type	RJ-45
Electrical and mechanical specifications	IEEE802.3
Transmission system	Ethernet (100BASE-TX/10BASE-T)
Communication protocol	TCP/IP
Support service	DHCP, remote control (Modbus/TCP)
Port number	Modbus/TCP: 502/TCP protocol

The instrument operates as a Modbus server. The number of simultaneous connections is 1.

4.2 Connection

Connect a UTP (Unshielded Twisted Pair) or STP (Shielded Twisted Pair) cable connected to a hub or other network device to the Ethernet port on the rear panel of the instrument. According to the user's operation manual, set the Modbus communication method in the system settings for communication method.

4.3 Function Code Table

The instrument supports the following function codes.

Code	Function	Description
03	Read hold register	From 0001 to continuous read data 0010
04	Read input register	From 0001 to continuous read data 3008, which up to 125
06	Write hold register	It can only write one register from the range of 0001 to 0010

4.4 Register Function and Application

Measurement data, setup data, and other types of data for the instrument are assigned to an internal

register for Modbus/TCP. A client device can send the command to the instrument using Modbus/TCP communication; it can read and write the internal register for the instrument. This enables retrieval of measurement data, and control the instrument, e.g., to start integration.

4.5 Register Distribution Table

Code	Address	Description
Input register	0001 ~ 0012	Measurement data status
	0101 ~ 0194	Measurement data
	2001 ~ 2510	Communication output data :NUMeric[:NORMal]:ITEM<x>
Hold register	0001 ~ 0010	Control data, control the integration operation

4.6 Input Register Address Map Table

Reg No.	Ref No.	H No.		RegisterFunctionDescription	
0001	30001	0000		Data update counter	(uint16)
0002	30002	0001			
0003	30003	0002		Peak overrange status	(uint16)
0004	30004	0003		Check range status	(uint16)
0005	30005	0004	H	Voltage range	(float, upper 2 bytes)
0006	30006	0005	L		(float, lower 2 bytes)
0007	30007	0006	H	Current range	(float, upper 2 bytes)
0008	30008	0007	L		(float, lower 2 bytes)
0009	30009	0008	H	Operation results	(float, upper 2 bytes)
0010	30010	0009	L		(float, lower 2 bytes)
0011	30011	000A	H	PLL synchronous frequency	(float, upper 2 bytes)
0012	30012	000B	L		(float, lower 2 bytes)
Normal Measurement Data					
0101	30101	0064	H	Voltage	(float, upper 2 bytes)
0102	30102	0065	L		(float, lower 2 bytes)
0103	30103	0066	H	Current	(float, upper 2 bytes)
0104	30104	0067	L		(float, lower 2 bytes)
0105	30105	0068	H	Active power	(float, upper 2 bytes)
0106	30106	0069	L		(float, lower 2 bytes)
0107	30107	006A	H	Apparent power	(float, upper 2 bytes)
0108	30108	006B	L		(float, lower 2 bytes)
0109	30109	006C	H	Reactive power	(float, upper 2 bytes)
0110	30110	006D	L		(float, lower 2 bytes)
0111	30111	006E	H	Power factor	(float, upper 2 bytes)
0112	30112	006F	L		(float, lower 2 bytes)
0113	30113	0070	H	Phase	(float, upper 2 bytes)
0114	30114	0071	L		(float, lower 2 bytes)

0115	30115	0072	H	Voltage frequency	(float, upper 2 bytes)
0116	30116	0073	L		(float, lower 2 bytes)
0117	30117	0074	H	Current frequency	(float, upper 2 bytes)
0118	30118	0075	L		(float, lower 2 bytes)
0119	30119	0076	H	Voltage positive peak	(float, upper 2 bytes)
0120	30120	0077	L		(float, lower 2 bytes)
0121	30121	0078	H	Voltage negative peak	(float, upper 2 bytes)
0122	30122	0079	L		(float, lower 2 bytes)
0123	30123	007A	H	Current positive peak	(float, upper 2 bytes)
0124	30124	007B	L		(float, lower 2 bytes)
0125	30125	007C	H	Current negative peak	(float, upper 2 bytes)
0126	30126	007D	L		(float, lower 2 bytes)
Reg No.	Ref No.	H No.		Register Function Description	
0127	30127	007E	H	Power positive peak	(float, upper 2 bytes)
0128	30128	007F	L		(float, lower 2 bytes)
0129	30129	0080	H	Power negative peak	(float, upper 2 bytes)
0130	30130	0081	L		(float, lower 2 bytes)
0131	30131	0082	H	Integration time	(float, upper 2 bytes)
0132	30132	0083	L		(float, lower 2 bytes)
0133	30133	0084	H	Power integration value	
0134	30134	0085	L		
0135	30135	0086	H	Positive power integration value	
0136	30136	0087	L		
0137	30137	0088	H	Negative power integration value	
0138	30138	0089	L		
0139	30139	008A	H	Current integration value	(float, upper 2 bytes)
0140	30140	008B	L		(float, lower 2 bytes)
0141	30141	008C	H	Positive current integration value	(float, upper 2 bytes)
0142	30142	008D	L		(float, lower 2 bytes)
0143	30143	008E	H	Negative current integration value	(float, upper 2 bytes)
0144	30144	008F	L		(float, lower 2 bytes)
0145	30145	0090	H	Voltage effective value	(float, upper 2 bytes)
0146	30146	0091	L		(float, lower 2 bytes)
0147	30147	0092	H	Calibrated average voltage value	(float, upper 2 bytes)
0148	30148	0093	L		(float, lower 2 bytes)
0149	30149	0094	H	DC voltage	(float, upper 2 bytes)
0150	30150	0095	L		(float, lower 2 bytes)
0151	30151	0096	H	Average voltage	(float, upper 2 bytes)
0152	30152	0097	L		(float, lower 2 bytes)
0153	30153	0098	H	AC voltage	(float, upper 2 bytes)
0154	30154	0099	L		(float, lower 2 bytes)
0155	30155	009A	H	Current effective value	(float, upper 2 bytes)
0156	30156	009B	L		(float, lower 2 bytes)

0157	30157	009C	H	Calibrated average current value	(float, upper 2 bytes)
0158	30158	009D	L		(float, lower 2 bytes)
0159	30159	009E	H	DC current	(float, upper 2 bytes)
0160	30160	009F	L		(float, lower 2 bytes)
0161	30161	00A0	H	Average current	(float, upper 2 bytes)
0162	30162	00A1	L		(float, lower 2 bytes)
0163	30163	00A2	H	AC current	(float, upper 2 bytes)
0164	30164	00A3	L		(float, lower 2 bytes)
0165	30165	00A4	H	Voltage peak factor	(float, upper 2 bytes)
0166	30166	00A5	L		(float, lower 2 bytes)
0167	30167	00A6	H	Current peak factor	(float, upper 2 bytes)
0168	30168	00A7	L		(float, lower 2 bytes)
Reg No.	Ref No.	H No.		Register Function Description	
0169	30169	00A8	H		(float, upper 2 bytes)
0170	30170	00A9	L		(float, lower 2 bytes)
Harmonic Measurement Data					
0171	30171	00AA	H	Total harmonic voltage	(float, upper 2 bytes)
0172	30172	00AB	L		(float, lower 2 bytes)
0173	30173	00AC	H	1 harmonic voltage (fundamental wave)	(float, upper 2 bytes)
0174	30174	00AD	L		(float, lower 2 bytes)
0175	30175	00AE	H	Total harmonic current	(float, upper 2 bytes)
0176	30176	00AF	L		(float, lower 2 bytes)
0177	30177	00B0	H	1 harmonic current (fundamental wave)	(float, upper 2 bytes)
0178	30178	00B1	L		(float, lower 2 bytes)
0179	30179	00B2	H	Total harmonic power	(float, upper 2 bytes)
0180	30180	00B3	L		(float, lower 2 bytes)
0181	30181	00B4	H	1 harmonic power (fundamental wave)	(float, upper 2 bytes)
0182	30182	00B5	L		(float, lower 2 bytes)
0183	30183	00B6	H	1-order power factor	(float, upper 2 bytes)
0184	30184	00B7	L		(float, lower 2 bytes)
0185	30185	00B8	H	Phase angle	(float, upper 2 bytes)
0186	30186	00B9	L		(float, lower 2 bytes)
0187	30187	00BA	H	Phase difference (3) between 1-order (fundamental wave) voltage and 3-order harmonic voltage	(float, upper 2 bytes)
0188	30188	00BB	L		(float, lower 2 bytes)
0189	30189	00BC	H	Phase difference (3) between 1-order (fundamental wave) current and 3-order harmonic current	(float, upper 2 bytes)
0190	30190	00BD	L		(float, lower 2 bytes)
0191	30191	00BE	H	Total harmonic voltage distortion	(float, upper 2 bytes)
0192	30192	00BF	L		(float, lower 2 bytes)
0193	30193	00C0	H	Total harmonic current distortion	

	(float, upper 2 bytes)				
0194					
Measurement Data Mapped to Communication Output Items (:NUMeric[:NORMal]:ITEM<X>)				Measurement data of ItemX	
0001+(X-1)*2			H		(float, upper 2 bytes)
0001+(X-1)*2+1	L		(float, lower 2 bytes)	Item1	
2001	32001	07D0	H		(float, upper 2 bytes)
2002	32002	07D1	L	Item2	(float, lower 2 bytes)
2003	32003	07D2	H		(float, upper 2 bytes)
2004	32004	07D3	L	Item3	(float, lower 2 bytes)
2005	32005	07D4	H		(float, upper 2 bytes)
2006	32006	07D5	L	Item4	(float, lower 2 bytes)
2007	32007	07D6	H		(float, upper 2 bytes)
2008	32008	07D7	L	Item5	(float, lower 2 bytes)
2009	32009	07D8			(float, upper 2 bytes)
2010	32010	07D9		Item6	(float, lower 2 bytes)
2011	32011	07DA			(float, upper 2 bytes)
2012	32012	07DB			
Reg No.	Ref No.	H No.		Register Function Description	
2013	32013	07DC	H	Item7	(float, upper 2 bytes)
2014	32014	07DD	L	Item8	(float, lower 2 bytes)
2015	32015	07DE	H		(float, upper 2 bytes)
2016	32016	07DF	L	Item9	(float, lower 2 bytes)
2017	32017	07E0	H		(float, upper 2 bytes)
2018	32018	07E1	L	Item10	(float, lower 2 bytes)
2019	32019	07E2	H		(float, upper 2 bytes)
2020	32020	07E3	L		(float, lower 2 bytes)
to				Item255	
2509	32509	09CC			(float, upper 2 bytes)

Peak Overrange Status (Input register: 0003)

The following way maps the peak overrange information for each element to bits. The bit corresponding to the input in which a peak out of range occurs is set to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														I	U

Check Range Status

The range status is mapped to the corresponding bit is shown in the following table.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								AP	AO	AH	AL	VP	VO	VH	VL

VL: Voltage is in reduced autoranging or lower conditions.
VH: Voltage exceeds the conditions of the auto-range rising scale.
VO: Voltage is overrange.
VP: Voltage peak is overrange.
AL: Current is in reduced autoranging or lower conditions.
AH: Current exceeds the conditions of the auto-range rising scale.
AO: Current is overrange.
AP: Current peak is overrange.

4.7 Hold Register Address Map Table

Reg No.	Ref No.	H No.	Register	Description	Effective Range	Default Value	Read/Write
Control Data							
0001	40001	0000	NUMeric:HOLD	Data hold	0: OFF		
1: ON	0	Read/Write					
0002	40002	0001					
0003	40003	0002	INTEG:START/STOP	Integration ON/OFF	0: Stop		
1: ON	0	Read/Write					
0004	40004	0003	INTEG:RESET	Reset integration	1 is to reset, other value is invalid	-	Write
0005	40005	0004					
0006	40006	0005					
0007	40007	0006					
0008	40008	0007					