UNI-T

UT5300X+ and UT5320R-SxA Series

Hipot Tester

Programming Manual

(SCPI&MODBUS RTU)-REV.1.0

Feb, 2023

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.

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.
- ★ The information provided in this manual is subject to change without 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.
- ★ No part of this manual shall be photocopied, reproduced or adapted without the prior written permission of **UNI-T**.

1. SCPI Command Reference

This chapter includes contents

- Command parser — know about rule of command parser
- Command syntax — writing rule of command line
- Query syntax — query writing rule of command line
- Query Response — query the format of respond
- Command reference

This section provides all SCPI command of the instrument, these command can control instrument function totally.

1.1 Command String Analysis

The host computer can send a string command to the instrument, the instrument command parser start to analyse when capture end mark.

For Example Legal command String

AAA:BBB CCC;DDD EEE;:FFF

The instrument command parser is responsible for all the command analyzation and execution. You should know this command rule before complie programming.

1.1.1 Command Parser Rule

1. Command parser is only analyse and respond ASCII code data.
2. **Command parser start to analyse when capture end mark. This instrument is only receive three types as the following as the end mark.**
   - **CR**
   - **CR+LF**
   - **LF**
3. Command parser terminate analyse immediately if the command is error and the command turn to void.
4. Command parser is not case-insensitive.
5. Command parser supports command abbreviation, abbreviation format see the following section.
6. RS485 mode, add ADDR□loacl address::□ in front of SCPI protocol, loacl address can be set to 1-32.
   In order to communication with multiple device via SCPI protocol
   For Example ADDR□1::□IDN? □ represents a blank
7. **The end of data sent by the instrument defaults to 0x0A**

0x0A (LF) .

8 . **Send multi-instructions via semicolon ; .**

1.1.2 Symbol Stipulation and Definition

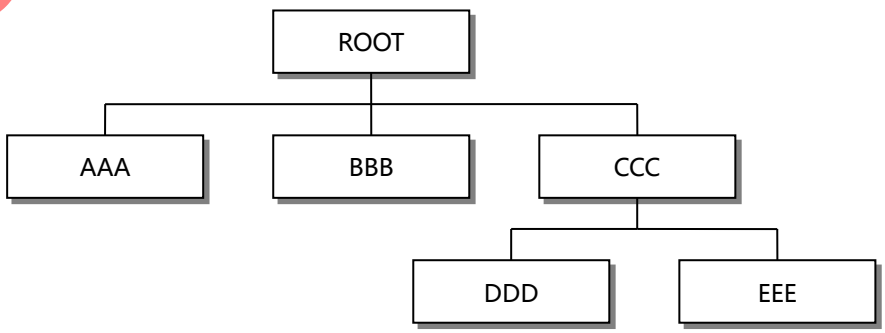
This section has some symbols that they are not the part of command tree, but for better understanding of command string.

Mark	Description
<.....>	Word in angle brackets presents parameter of command, *Such as* <float> represents floating-point parameter <integer> represents integer parameter
[.....]	Word in square bracketsp presents optional command.
{.....}	Braces contains several parameter items, only one of them can be selected.
Capital letter	Abbreviation format of command.
☐	Blank character, it represents a bland, only for reading.

1.1.3 Command Tree Structure

SCPI command with tree structure can divide into three stage. (Notes: this instrument command parser can down to analyse arbitrary layer.) The highest level is called subsystem commands. The command on the lower degree only valid if the subsystem command is selected. SCPI use colon mark(:) to separate the higher level and lower level commands.

Figure 0-1 Command Tree Structure



For Example

ROOT:CCC:DDD ppp

ROOT Sub-system command

CCC Second stage

DDD Third stage

ppp Parameter

1.2 Command and Parameter

A command tree consists of **Command and[Parameter]** , use a blank (ASCII: 20H) to separate.

For Example

AAA:BBB□1.234

Command [Parameter]

1.2.1 Command

Command string can be long string form or abbreviation form, long string form is for engineer to understand string meaning;
Abbreviation form is for write.

1.2.2 Parameter

1. Single command word command, no parameter.
   For Example AAA:BBB
2. Parameter can be character string form, abbreviation form should obey "command abbreviation rule" at last section.
   For Example AAA:BBB□1.23
3. Parameter can be numeric value.

<Integer>	Integer 123, +123, -123
<Float>	Floating point number of arbitary form: <Fixfloat>: fixed floating point number: 1.23, -1.23 <Sciloat>: floating point number presents by scientific notation: 1.23E+4, +1.23e-4 <Mpfloat>: floating point number presents by multiplying power: 1.23k, 1.23MA, 1.23G, 1.23u

Table 0-1 Abbreviation of Multiplying Power

Numeric Value	Multiplying power
1E18 (EXA)	EX
1E15 (PETA)	PE
1E12 (TERA)	T
1E9 (GIGA)	G
1E6 (MEGA)	MA
1E3 (KILO)	K
1E-3 (MILLI)	M

1E-6 (MICRO)	U
1E-9 (NANO)	N
1E-12 (PICO)	P
1E-15 (PEMTO)	F
1E-18 (ATTO)	A

Because SCPI is not case-insensitive, so the the written is different from standard name.

For Example

"1M" represents one milli, not 1 mega.

"1MA" represents 1 mega.

1.2.3 Separator

Command parser only receive allowed separator and other separators will make "Invalid separator" error, these separators are:

;	Semicolon mark, for separate two commands. *For Example* AAA:BBB 100.0;CCC:DDD
:	Colon mark, for separate command tree or restart command tree. *For Example* AAA:BBB:CCC 123.4;:DDD:EEE 567.8
?	Question mark, for query. *For Example* AAA?
□	Space mark, for separate parameter. *For Example* AAA:BBB□1.234

1.3 Command Reference

All command is followed subsystem command order to explain, sub-system as follows,

- DISPlay Display sub-system
- FUNCtion Function sub-system
- SYSTem System sub-system
- FILE File sub-system
- TEST Start test sub-system
- RESET Stop test sub-system
- FETCh? Fetch result sub-system

Common command

- IDN? Instrument information query sub-system
- SN? Instrument information serial number query sub-system

1.4 DISPlay Sub-system

DISPlay sub-system is used to switch different display page.

Figure 0-2 DISPlay Sub-system Tree

DISPlay	:PAGE	{TEST, MSET, FILE, SYST1, SYST2, SINF}

1.4.1 DISPlay:PAGE

DISP:PAGE is used to switch to the specific page.

Command Synatx	DISPlay:PAGE <Page Name >
Parameter	< Page Name > includes TEST Test display page MSET Measurement setup page FILE File page SYST1 System configuration page 1 SYST2 System configuration page 2, it's mainly about communications SINF System information page
For Example	Send > disp:page mset // Switch to setup page
Query Syntax	DISPlay:PAGE?
Query Response	<Page Name> TEST Test display page MSET Measurement setup page FILE File page

	SYST1	System configuration page 1
	SYST2	System configuration page 2, it's mainly about communications
	SINF	System information page
For Example	Send > disp:page? Return > TEST	

1.5 FUNCtion Sub-system

FUNCtion sub-system command is used to set test parameter of the instrument.

Notice:

The parameter set by FUNCtion sub-system will not automatically stored in the file.

After the parameter is set, it need to load FILE subsystem to save the parameter into the internal file.

FUNCtion	:STEP	:NEW		Creat a new empty test project to compile a new test project.
		:INS		Add a new test project after the current one.
		:DEL		Delete the current test project In the existing test scenario.
	:TYPE	<integer (1~20) >,{AC,DC,IR,CK}		Test mode setting for selecting step
	:AC	:VOLT	<integer>,<integer>	AC withstand output voltage setting
		:TTIM	<integer>,<float>	Test time setting of AC withstand voltage
		:RTIM	<integer>,<float>	Rising time setting of AC withstand voltage
		:FTIM	<integer>,<float>	Falling time setting of AC withstand voltage
		:UPPC	<integer>,<float>	Upper limit of current setting of AC withstand voltage
		:LOWC	<integer>,<float>	Lower limit of current setting of AC withstand voltage
		:ARC	<integer>,<integer>	Electric arc level setting

				of AC withstand voltage
		:FREQ	<integer>,{50,60}	Output frequency setting of AC withstand voltage
		:RANGe	<integer>,{AUTO,FIXED}	Current range setting of AC withstand voltage
		:OFFSet	<integer>, {OFF,GET}	Compensation to zero setting of AC withstand voltage
		:CH1	<integer>, {HIGH,LOW,OPEN}	Channel 1 setting of AC withstand voltage
		:CH2	<integer>, {HIGH,LOW,OPEN}	Channel 2 setting of AC withstand voltage
		:CH3	<integer>, {HIGH,LOW,OPEN}	Channel 3 setting of AC withstand voltage
		:CH4	<integer>, {HIGH,LOW,OPEN}	Channel 4 setting of AC withstand voltage
		:CH5	<integer>, {HIGH,LOW,OPEN}	Channel 5 setting of AC withstand voltage
		:CH6	<integer>, {HIGH,LOW,OPEN}	Channel 6 setting of AC withstand voltage
		:CH7	<integer>, {HIGH,LOW,OPEN}	Channel 7 setting of AC withstand voltage
		:CH8	<integer>, {HIGH,LOW,OPEN}	Channel 8 setting of AC withstand voltage
	:DC	:VOLT	<integer>,<integer>	DC withstand output voltage setting
		:TTIM	<integer>,<float>	Test time setting of DC withstand voltage
		:RTIM	<integer>,<float>	Rising time setting of DC withstand voltage
		:FTIM	<integer>,<float>	Falling time setting of DC withstand voltage
		:UPPC	<integer>,<float>	Upper limit of current setting of DC withstand voltage
		:LOWC	<integer>,<float>	Lower limit of current setting of DC withstand voltage
		:ARC	<integer>,<integer>	Electric arc level setting of DC withstand voltage
		:RANGe	<integer>,{AUTO,FIXED}	Current range setting of DC withstand voltage
		:OFFSet	<integer>, {OFF,GET}	Compensation to zero setting of DC withstand

				voltage
		:RAMP	<integer>,{OFF,ON}	Rising judgement setting of DC withstand voltage
		:WAIT	<integer>,<float>	Waiting judgement setting of DC withstand voltage
		:CHAR	<integer>,<float>	Lower limit of charge setting of DC withstand voltage
		:CH1	<integer>, {HIGH,LOW,OPEN}	Channel 1 setting of DC withstand voltage
		:CH2	<integer>, {HIGH,LOW,OPEN}	Channel 2 setting of DC withstand voltage
		:CH3	<integer>, {HIGH,LOW,OPEN}	Channel 3 setting of DC withstand voltage
		:CH4	<integer>, {HIGH,LOW,OPEN}	Channel 4 setting of DC withstand voltage
		:CH5	<integer>, {HIGH,LOW,OPEN}	Channel 5 setting of DC withstand voltage
		:CH6	<integer>, {HIGH,LOW,OPEN}	Channel 6 setting of DC withstand voltage
		:CH7	<integer>, {HIGH,LOW,OPEN}	Channel 7 setting of DC withstand voltage
		:CH8	<integer>, {HIGH,LOW,OPEN}	Channel 8 setting of DC withstand voltage
	:IR	:VOLT	<integer>,<integer>	Output voltage setting of insulated resistance
		:TTIM	<integer>,<float>	Delay judgement time setting of insulated resistance
		:RTIM	<integer>,<float>	Rising time setting of insulated resistance
		:FTIM	<integer>,<float>	Falling time setting of insulated resistance
		:UPPC	<integer>,<float>	Upper limit of resistance setting of insulated resistance
		:LOWC	<integer>,<float>	Lower limit of resistance setting of insulated resistance
		:RANGe	<integer>,{AUTO,FIXED}	Resistance range of insulated resistance
		:CHAR	<integer>,<float>	Lower limit of charge setting of insulated

				resistance
		:CH1	<integer>, {HIGH,LOW,OPEN}	Channel 1 setting of insulated resistance
		:CH2	<integer>, {HIGH,LOW,OPEN}	Channel 2 setting of insulated resistance
		:CH3	<integer>, {HIGH,LOW,OPEN}	Channel 3 setting of insulated resistance
		:CH4	<integer>, {HIGH,LOW,OPEN}	Channel 4 setting of insulated resistance
		:CH5	<integer>, {HIGH,LOW,OPEN}	Channel 5 setting of insulated resistance
		:CH6	<integer>, {HIGH,LOW,OPEN}	Channel 6 setting of insulated resistance
		:CH7	<integer>, {HIGH,LOW,OPEN}	Channel 7 setting of insulated resistance
		:CH8	<integer>, {HIGH,LOW,OPEN}	Channel 8 setting of insulated resistance
	:CK	:VOLT	<integer>,<integer>	Output voltage setting of contact inspection
		:LOWC	<integer>,<float>	Lower limit of current setting of contact inspection
		:CH1	<integer>, {OFF,ON}	Channel 1 setting of contact inspection
		:CH2	<integer>, { OFF,ON}	Channel 2 setting of contact inspection
		:CH3	<integer>, { OFF,ON}	Channel 3 setting of contact inspection
		:CH4	<integer>, { OFF,ON}	Channel 4 setting of contact inspection
		:CH5	<integer>, { OFF,ON}	Channel 5 setting of contact inspection
		:CH6	<integer>, { OFF,ON}	Channel 6 setting of contact inspection
		:CH7	<integer>, { OFF,ON}	Channel 7 setting of contact inspection
		:CH8	<integer>, { OFF,ON}	Channel 8 setting of contact inspection
	:SOUR?			Query all measurement setting data for the current test step

1.5.1 FUNCtion:STEP? (Query Test Step)

FUNC:STEP? command is used to query the current test step and the total test steps.

Query Syntax	FUNC:STEP?
Query Response	<2 digits integer >/<2 digits integer > The current test step / The total test steps
For Example	Send > FUNC:STEP? Return > 02/05 // The current test step is the step 2, the total test steps are 5 steps.

1.5.2 FUNCtion:STEP:NEW (New Test Project)

FUNCtion:STEP:NEW command is used to create a new empty test project for compiling a new test project.

Command Synatx	FUNCtion:STEP:NEW
For Example	Send > FUNC:STEP:NEW // Create a new empty test project.

1.5.3 FUNCtion:STEP (Editing the instruction for step n of the current test scenario)

FUNCtion:STEP command is used to edit the instruction for step n of the current test scenario (n is smaller than the total steps.)

Command Synatx	FUNCtion:STEP <integer (1~20) >
For Example	Send > FUNC:STEP 1 // Editing the first step for the current test scenario

1.5.4 FUNCtion:STEP:INS (Add Test Project)

FUNCtion:STEP:INS command is used to add a new test project after the current test scenario.

Command Synatx	FUNCtion:STEP:INS
For Example	Send > FUNC:STEP:INS // Add a new test project

1.5.5 FUNCtion:STEP:DEL (Delete Test Project)

FUNCtion:STEP:DEL command is used to delete the current test project within the existing test scenario.

Command Synatx	FUNCtion:STEP:DEL
For Example	Send > FUNC:STEP:DEL // Delete the current test project

1.5.6 FUNCtion:TYPE (Test Mode)

FUNCtion:TYPE command is used to set the test mode for the selected step and all parameter sets to the default.

Command Synatx	FUNCtion:TYPE <integer (1~20) >,{AC,DC,IR,CK}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {AC,DC,IR,CK} **CK only UT5320R-S4/ UT5320R-S8** AC represents AC withstand voltage; DC represents DC withstand voltage; IR represents insulated resistance; CK represents contact checking.
For Example	Send > FUNC:TYPE 1,IR // Set the test mode of step 1 to insulation and initialize the measurement settings parameter.
Query Syntax	FUNCtion:TYPE? <integer (1~20) >
Query Response	{AC,DC,IR,CK}
For Example	Send > FUNC:TYPE? 1 // Query the test mode of step 1. Return > IR

1.5.7 FUNCtion:AC:VOLT (AC Withstand Output Voltage)

FUNCtion:AC:VOLT command is used to set output voltage of AC withstand.

Command Synatx	FUNCtion:AC:VOLT <integer (1~20) >,<integer (50~5000) >
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <integer (50~5000) > Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:VOLT 2,1000 // Set step 2 output voltage of AC withstand to 1000 V.
Query Syntax	FUNCtion:AC:VOLT? <integer (1~20) >
Query Response	<integer (50~5000) >
For Example	Send > FUNC:AC:VOLT? 2 // Query step 2 output voltage value of AC withstand. Return > 1000 // 1000 V

1.5.8 FUNCtion:AC:TTIM (Test Time of AC Withstand Voltage)

FUNCtion:AC:TTIM command is used to set test time of AC withstand voltage.

Command Synatx	FUNCtion:AC:TTIM <integer (1~20) >,<float>

Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~999.9 (0 represents continuous test) Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:TTIM 1,60 // Set step 1, test time of AC withstand voltage to 60s.
Query Syntax	FUNCtion:AC:TTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:AC:TTIM? 1 // Query step 1, test time of AC withstand voltage. Return > 60.0 // 60s

1.5.9 FUNCtion:AC:RTIM (Rising Time of AC Withstand Voltage)

FUNCtion:AC:RTIM command is used to set rising time of AC withstand voltage.

Command Synatx	FUNCtion:AC:RTIM <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0.1~999.9 Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:RTIM 1,1.5 // Set step 1, rising time of AC withstand voltage to 1.5s.
Query Syntax	FUNCtion:AC:RTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:AC:RTIM? 1 // Query step 1, rising time of AC withstand voltage. Return > 1.5 // 1.5s

1.5.10 FUNCtion:AC:FTIM (Falling Time of AC Withstand Voltage)

FUNCtion:AC:FTIM command is used to set falling time of AC withstand voltage.

Command Synatx	FUNCtion:AC:FTIM <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~999.9 (0 represents OFF) Notice: Test mode of this step must be AC, otherwise this instruction is invalid.

For Example	Send > FUNC:AC:FTIM 1,1.0 // Set step 1, falling time of AC withstand voltage to 1.0s.
Query Syntax	FUNCtion:AC:FTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:AC:FTIM? 1 // Query step 1, rising time of AC withstand voltage. Return > 1.0 // 1.0s

1.5.11 FUNCtion:AC:UPPC (Upper Limit of Current of AC Withstand Voltage)

FUNCtion:AC:UPPC command is used to set the upper limit of current of AC withstand voltage.

Command Synatx	FUNCtion:AC:UPPC <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, UT5310 series data range: 0.001~10.00, data unit: mA UT5320 series data range: 0.001~20.00, data unit: mA Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:UPPC 1,5.0 // Set step 1, the upper limit of current of AC withstand voltage to 5mA.
Query Syntax	FUNCtion:AC:UPPC? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:AC:UPPC? 1 // Query step 1, the upper limit of current of AC withstand voltage. Return > 5.000 // 5mA

1.5.12 FUNCtion:AC:LOWC (Lower Limit of Current of AC Withstand Voltage)

FUNCtion:AC:LOWC command is used to set the lower limit of current of AC withstand voltage.

Command Synatx	FUNCtion:AC:LOWC <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, UT5310 series data range: 0~10.00, data unit: mA (0 represents OFF) UT5320 series data range: 0~20.00, data unit: mA (0 represents OFF) Notice: Test mode of this step must be AC, otherwise this instruction is invalid.

For Example	Send > FUNC:AC:LOWC 1,0 // Set step 1, the lower limit of current of AC withstand voltage to OFF.
Query Syntax	FUNCtion:AC:LOWC? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:AC:LOWC? 1 // Query step 1, the lower limit of current of AC withstand voltage. Return > 0.000 // Turn off the lower limit of current.

1.5.13 FUNCtion:AC:ARC (Electric Arc level of AC Withstand Voltage)

FUNCtion:AC:ARC command is used to set electric arc of AC withstand voltage.

Command Synatx	FUNCtion:AC:ARC <integer (1~20) >,<integer (0~9) >
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <integer (0~9) >, 0 represents turn off electric arc detection; 1~9 represents electric arc level; Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:ARC 1,5 // Set step 1, electric arc level of AC withstand voltage to 5.
Query Syntax	FUNCtion:AC:ARC? <integer (1~20) >
Query Response	<integer (0~9) >
For Example	Send > FUNC:AC:ARC? 1 // Query step 1, electric arc level of AC withstand voltage. Return > 5 // 5 level

1.5.14 FUNCtion:AC:FREQ (AC Withstand Voltage Output Frequency)

FUNCtion:AC:FREQ command is used to set output frequency of AC withstand voltage.

Command Synatx	FUNCtion:AC:FREQ <integer (1~20) >,{50,60}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {50,60} Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:FREQ 1,60 // Set step 1, output frequency of AC withstand voltage to 60 Hz.
Query Syntax	FUNCtion:AC:FREQ? <integer (1~20) >

Query Response	{50,60}
For Example	Send > FUNC:AC:FREQ? 1 // Quert step 1, output frequency value of AC withstand voltage. Return > 60 // 60 Hz

1.5.15 FUNCtion:AC:RANGe (Current Range of AC Withstand Voltage)

FUNCtion:AC:RANGe command is used to set the current range of AC withstand voltage.

Command Synatx	FUNCtion:AC:RANGe <integer (1~20) >,{AUTO,FIXED}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {AUTO,FIXED}, AUTO: automatic range; FIXED: fixed range; Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:RANG 1,FIXED // Set step 1, the current range of AC withstand voltage to FIXED.
Query Syntax	FUNCtion:AC:RANGe? <integer (1~20) >
Query Response	{AUTO,FIXED}
For Example	Send > FUNC:AC:RANG? 1 // Query step 1, the current range of AC withstand voltage. Return > FIXED // FIXED range

1.5.16 FUNCtion:AC:OFFSet (Compersention to Zero of AC Withstand Voltage)

FUNCtion:AC:OFFSet command is used to set compersention to zero of AC withstand voltage.

Command Synatx	FUNCtion:AC:OFFSet <integer (1~20) >,{OFF,GET}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {OFF,GET}, OFF: Turn off compersention to zero; GET: compersention to zero auto test one time, DUT must be move off from the test wire at this moment; And the instrument should in <Test Display > or <Measurement Setup>, GET instruction can only be valid; Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:OFFS 1,OFF // Set step 1, compersention to zero of AC withstand voltage to FIXED. Send > FUNC:AC:OFFS 1,GET // Automatically compersention to

	zero of AC withstand voltage in step 1, // Test is decided by the size of rising time.
Query Syntax	FUNCtion:AC:OFFSet? <integer (1~20) >
Query Response	<float> , data unit: mA
For Example	Send > FUNC:AC:OFFS? 1 // Query step 1, the setting value of compersention to zero of AC withstand voltage. Return > 0.004 // 0.004mA

1.5.17 FUNCtion:AC:CH1 (Channel 1 Setting of AC Withstand Voltage)

FUNCtion:AC:CH1 command is used to set channel 1 parameter of AC withstand voltage. (**Only UT5320R-S4/S8)**

Command Synatx	FUNCtion:AC:CH1 <integer (1~20) >,{HIGH,LOW,OPEN}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {HIGH,LOW,OPEN}, HIGH: high-end; LOW: low-end; OPEN: open-circuit; Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:AC:CH1 1,LOW // Set step 1, channel 1 of AC withstand voltage to low-end.
Query Syntax	FUNCtion:AC:CH1? <integer (1~20) >
Query Response	{HIGH,LOW,OPEN}
For Example	Send > FUNC:AC:CH1? 1 // Query step 1, channel 1 parameter setting of AC withstand voltage. Return > OPEN // Open-circuit

Other channel setting is the same as above table, for example

Command Synatx	FUNCtion:AC:CH2 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 2 parameter of AC withstand voltage.
	FUNCtion:AC:CH5 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 5 parameter of AC withstand voltage.
	FUNCtion:AC:CH8 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 8 parameter of AC withstand voltage.

Query Syntax	FUNCtion:AC:CH4? <integer (1~20) > // This command is used to query Channel 4 parameter of AC withstand voltage. FUNCtion:AC:CH6? <integer (1~20) > // This command is used to query Channel 6 parameter of AC withstand voltage. FUNCtion:AC:CH7? <integer (1~20) > // This command is used to query Channel 7 parameter of AC withstand voltage.

1.5.18 FUNCtion:DC:VOLT (DC Withstand Output Voltage)

FUNCtion:DC:VOLT command is used to set output voltage of DC withstand.

Command Synatx	FUNCtion:DC:VOLT <integer (1~20) >,<integer (50~6000) >
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <integer (50~6000) > Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:VOLT 2,1000 // Set step 2, output voltage of DC withstand to 1000 V.
Query Syntax	FUNCtion:DC:VOLT? <integer (1~20) >
Query Response	<integer (50~6000) >
For Example	Send > FUNC:DC:VOLT? 2 // Query step 2, output voltage value of DC withstand. Return > 1000 // 1000V

1.5.19 FUNCtion:DC:TTIM (Test Time of DC Withstand Voltage)

FUNCtion:DC:TTIM command is used to set test time of DC withstand voltage.

Command Synatx	FUNCtion:DC:TTIM <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~999.9 (0 represents continuous test) Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:TTIM 1,60 // Set step 1, test time of DC withstand voltage to 60s.
Query Syntax	FUNCtion:DC:TTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:DC:TTIM? 1 // Query step 1, test time of DC withstand voltage. Return > 60.0 // 60s

1.5.20 FUNCtion:DC:RTIM (Rising Time of DC Withstand Voltage)

FUNCtion:DC:RTIM command is used to set rising time of DC withstand voltage.

Command Synatx	FUNCtion:DC:RTIM <integer (1~20) >,<float>
Parameter	Parameter1：<integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2：<float>，data range：0.1~999.9 Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:RTIM 1,1.5 // Set step 1, rising time of DC withstand voltage to 1.5s.
Query Syntax	FUNCtion:DC:RTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:DC:RTIM? 1 // Query step 1, rising time of DC withstand voltage. Return > 1.5 // 1.5s

1.5.21 FUNCtion:DC:FTIM (Falling Time of DC Withstand Voltage)

FUNCtion:DC:FTIM command is used to set falling time of DC withstand voltage.

Command Synatx	FUNCtion:DC:FTIM <integer (1~20) >,<float>
Parameter	Parameter1：<integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2：<float>，data range：0~999.9（0 represents OFF） Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:FTIM 1,1.0 // Set step 1, falling time of DC withstand voltage to 1.0s.
Query Syntax	FUNCtion:DC:FTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:DC:FTIM? 1 // Query step 1, rising time of DC withstand voltage. Return > 1.0 // 1.0s

1.5.22 FUNCtion:DC:UPPC （Upper Limit of Current of DC Withstand Voltage)

FUNCtion:DC:UPPC command is used to set the upper limit of current of DC withstand voltage.

Command Synatx	FUNCtion:DC:UPPC <integer (1~20) >,<float>
Parameter	Parameter1：<integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2：<float>，

	UT5310 series data range: 0.001~5.00, data unit: mA UT5320 series data range: 0.001~10.00, data unit: mA Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:UPPC 1,5.0 // Set step 1, the upper limit of current of DC withstand voltage to 5mA.
Query Syntax	FUNCtion:DC:UPPC? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:DC:UPPC? 1 // Query step 1, the upper limit of current of DC withstand voltage. Return > 5.000 // 5mA

1.5.23 FUNCtion:DC:LOWC (Lower Limit of Current of DC Withstand Voltage)

FUNCtion:DC:LOWC command is used to set the lower limit of current of DC withstand voltage.

Command Synatx	FUNCtion:DC:LOWC <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, UT5310 series data range: 0~5.00, data unit: mA (0 represents OFF) UT5320 series data range: 0~10.00, data unit: mA (0 represents OFF) Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:LOWC 1,0 // Set step 1, the lower limit of current of DC withstand voltage to OFF.
Query Syntax	FUNCtion:DC:LOWC? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:DC:LOWC? 1 // Query step 1, the lower limit of current of DC withstand voltage. Return > 0.000 // Turn off the lower limit of current.

1.5.24 FUNCtion:DC:ARC (Electric Arc level of DC Withstand Voltage)

FUNCtion:DC:ARC command is used to set electric arc of DC withstand voltage.

Command Synatx	FUNCtion:DC:ARC <integer (1~20) >,<integer (0~9) >
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot

	greater than the total steps! Parameter2: <integer (0~9) >, 0 represents turn off electric arc detection; 1~9 represents electric arc level; Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:ARC 1,5 // Set step 1, electric arc level of DC withstand voltage to 5.
Query Syntax	FUNCtion:DC:ARC? <integer (1~20) >
Query Response	<integer (0~9) >
For Example	Send > FUNC:DC:ARC? 1 // Query step 1, electric arc level of DC withstand voltage. Return > 5 // 5 level

1.5.25 FUNCtion:DC:RANGe (Current Range of DC Withstand Voltage)

FUNCtion:DC:RANGe command is used to set the current range of DC withstand voltage.

Command Synatx	FUNCtion:DC:RANGe <integer (1~20) >,{AUTO,FIXED}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {AUTO,FIXED}, AUTO: automatic range; FIXED: fixed range; Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:RANG 1,FIXED // Set step 1, the current range of DC withstand voltage to FIXED.
Query Syntax	FUNCtion:DC:RANGe? <integer (1~20) >
Query Response	{AUTO,FIXED}
For Example	Send > FUNC:DC:RANG? 1 // Query step 1, the current range of DC withstand voltage. Return > FIXED // FIXED range

1.5.26 FUNCtion:DC:OFFSet (Compersention to Zero of DC Withstand Voltage)

FUNCtion:DC:OFFSet command is used to set compersention to zero of DC withstand voltage.

Command Synatx	FUNCtion:DC:OFFSet <integer (1~20) >,{OFF,GET}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {OFF,GET},

	OFF: Turn off compersention to zero; GET: compersention to zero auto test one time, DUT must be move off from the test wire at this moment; And the instrument should in <Test Display > or <Measurement Setup>, GET instruction can only be valid; Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:OFFS 1,OFF // Set step 1, compersention to zero of DC withstand voltage to FIXED. Send > FUNC:DC:OFFS 1,GET // Automatically compersention to zero of AC withstand voltage in step 1, // Test is decided by the size of rising time.
Query Syntax	FUNCtion:DC:OFFSet? <integer (1~20) >
Query Response	<float> , data unit: uA
For Example	Send > FUNC:DC:OFFS? 1 // Query step 1, the setting value of compersention to zero of DC withstand voltage. Return > 53.5 // 53.5uA

1.5.27 FUNCtion:DC:RAMP (Rising Judgement of DC Withstand Voltage)

FUNCtion:DC:RAMP command is used to set rising judgement of DC withstand voltage.

Command Synatx	FUNCtion:DC:RAMP <integer (1~20) >,{OFF,ON}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {OFF,ON}, OFF: turn off rising judgement, ON: turn on rising judgement; Notice: Test mode of this step must be AC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:RAMP 1,ON // Turn on rising judgement of DC withstand voltage in step 1.
Query Syntax	FUNCtion:DC:RAMP? <integer (1~20) >
Query Response	{OFF,ON}
For Example	Send > FUNC:DC:RAMP? 1 // Query step 1, rising judgement of DC withstand voltage. Return > ON // Turn on rising judgement.

1.5.28 FUNCtion:DC:WAIT (Waiting Judgement of DC Withstand Voltage)

FUNCtion:DC:WAIT command is used to set waiting judgement of DC

withstand voltage.

Command Synatx	FUNCtion:DC:WAIT <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~999.9, data unit: s; Rising time < the setting value of waiting judgement < (rising time+test time), otherwise it is invalid; Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:WAIT 1,10 // Set step 1, waiting judgement of DC withstand voltage to 10s.
Query Syntax	FUNCtion:DC:WAIT? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:DC:WAIT? 1 // Query step 1, waiting judgement of DC withstand voltage to 10s. Return > 10.0 // 10s

1.5.29 FUNCtion:DC:CHAR (Lower Limit of Charge of DC Withstand Voltage)

FUNCtion:DC:CHAR command is used to set the lower limit of charge of DC withstand voltage.

Command Synatx	FUNCtion:DC:CHAR <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~350.0 (0 represents OFF), data unit: uA; Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:CHAR 1,35 // Set step 1, the lower limit of charge of DC withstand voltage to 35uA.
Query Syntax	FUNCtion:DC:CHAR? <integer (1~20) >
Query Response	<float>, data unit: uA
For Example	Send > FUNC:DC:CHAR? 1 // Query step 1, the lower limit of charge of DC withstand voltage to 35uA. Return > 35.0 // 35uA

1.5.30 FUNCtion:DC:CH1 (Channel 1 Setting of DC Withstand Voltage)

FUNCtion:DC:CH1 command is used to set channel 1 parameter of DC withstand voltage. (**Only UT5320R-S4/S8**)

Command Synatx	FUNCtion:DC:CH1 <integer (1~20) >,{HIGH,LOW,OPEN}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps!

	Parameter2：{HIGH,LOW,OPEN}, HIGH：high-end ； LOW：low-end ； OPEN：open circuit； Notice: Test mode of this step must be DC, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:CH1 1,LOW // Set step 1, channel 1 of DC withstand voltage to low-end.
Query Syntax	FUNCtion:DC:CH1? <integer (1~20) >
Query Response	{HIGH,LOW,OPEN}
For Example	Send > FUNC:DC:CH1? 1 // Query step 1, channel 1 setting parameter of DC withstand voltage to low-end. Return > OPEN // Open circuit

Other channel setting is the same as above table, for example

Command Synatx	FUNCtion:DC:CH2 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 2 parameter of DC withstand voltage.
	FUNCtion:DC:CH5 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 5 parameter of DC withstand voltage.
	FUNCtion:DC:CH8 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 8 parameter of DC withstand voltage.

Query Syntax	FUNCtion:DC:CH4? <integer (1~20) > // This command is used to query Channel 4 parameter of DC withstand voltage. FUNCtion:DC:CH6? <integer (1~20) > // This command is used to query Channel 6 parameter of DC withstand voltage. FUNCtion:DC:CH7? <integer (1~20) > // This command is used to query Channel 7 parameter of DC withstand voltage.

1.5.31 FUNCtion:IR:VOLT (Insulated Resistance Output Voltage)

FUNCtion:IR:VOLT command is used to set output voltage of insulated resistance.

Command Synatx	FUNCtion:IR:VOLT <integer (1~20) >,<integer (50~2500) >
Parameter	Parameter1： <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2： <integer (50~2500) > Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:VOLT 2,1000 // Set step 2, output voltage of insulated resistance to 1000 V.
Query Syntax	FUNCtion:IR:VOLT? <integer (1~20) >
Query Response	<integer (50~2500) >

For Example	Send > FUNC:IR:VOLT? 2 // Query step 2, output voltage setting value of insulated resistance. Return > 1000 // 1000 V

1.5.32 FUNCtion:IR:TTIM (Test Time of Insulated Resistance)

FUNCtion:IR:TTIM command is used to set test time of insulated resistance.

Command Synatx	FUNCtion:IR:TTIM <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~999.9 (0 represents continuous test) Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:TTIM 1,60 // Set step 1, test time of insulated resistance to 60s.
Query Syntax	FUNCtion:IR:TTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:IR:TTIM? 1 // Query step 1, test time of insulated resistance. Return > 60.0 // 60s

1.5.33 FUNCtion:IR:RTIM (Rising Time of Insulated Resistance)

FUNCtion:IR:RTIM command is used to set rising time of insulated resistance.

Command Synatx	FUNCtion:IR:RTIM <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0.1~999.9 Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:DC:RTIM 1,1.5 // Set step 1, rising time of insulated resistance to1.5s.
Query Syntax	FUNCtion:DC:RTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:DC:RTIM? 1 // Query step 1, rising time of insulated resistance. Return > 1.5 // 1.5s

1.5.34 FUNCtion:IR:FTIM (Falling Time of Insulated Resistance)

FUNCtion:IR:FTIM command is used to set falling time of insulated resistance.

Command Synatx	FUNCtion:IR:FTIM <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~999.9 (0 represents continuous test) Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:FTIM 1,1.0 // Set step 1, falling time of insulated resistance to 1.0s.
Query Syntax	FUNCtion:IR:FTIM? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:IR:FTIM? 1 // Query step 1, falling time of insulated resistance to 1.0s. Return > 1.0 // 1.0s

1.5.35 FUNCtion:IR:UPPC (Upper Limit of Resistance of Insulated Resistance)

FUNCtion:IR:UPPC command is used to set the upper limit of resistance of insulated resistance.

Command Synatx	FUNCtion:IR:UPPC <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~1E4 (0 represents OFF), data unit: MΩ Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:UPPC 1,1E3 // Set step 1, the upper limit of resistance of insulated resistance to 1 GΩ.
Query Syntax	FUNCtion:IR:UPPC? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:IR:UPPC? 1 // Query step 1, the upper limit of resistance of insulated resistance. Return > 1000.0 // 1000 MΩ

1.5.36 FUNCtion:IR:LOWC (Lower Limit of Resistance of Insulated Resistance)

FUNCtion:IR:LOWC command is used to set the lower limit of resistance of

insulated resistance.

Command Synatx	FUNCtion:IR:LOWC <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0.1~1E4, data unit: MΩ Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:LOWC 1,1E2 // Set step 1, the lower limit of resistance of insulated resistance to 100 MΩ.
Query Syntax	FUNCtion:IR:LOWC? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:IR:LOWC? 1 // Query step 1, the lower limit of resistance of insulated resistance. Return > 100.0 // 1000 MΩ

1.5.37 FUNCtion:IR:RANGe (Current Range of Insulated Resistance)

FUNCtion:IR:RANGe command is used to set the current range of insulated resistance.

Command Synatx	FUNCtion:IR:RANGe <integer (1~20) >,{AUTO,FIXED}
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: {AUTO,FIXED}, AUTO: automatic range; FIXED: fixed range; Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:RANG 1,FIXED // Set step 1, the current range of insulated resistance to FIXED.
Query Syntax	FUNCtion:IR:RANGe? <integer (1~20) >
Query Response	{AUTO,FIXED}
For Example	Send > FUNC:IR:RANG? 1 //Query step 1, the current range of insulated resistance Return > FIXED // FIXED range

1.5.38 FUNCtion:IR:CHAR (Lower Limit of Charge of Insulated Resistance)

FUNCtion:IR:CHAR command is used to set the lower limit of charge of insulated resistance.

Command Synatx	FUNCtion:IR:CHAR <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0~350.0 (0 represents OFF) , data

	unit：uA； Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:CHAR 1,35 // Set step 1, the lower limit of charge of insulated resistance to 35uA.
Query Syntax	FUNCtion:IR:CHAR? <integer (1~20) >
Query Response	<float>, data unit：uA
For Example	Send > FUNC:IR:CHAR? 1 // Query step 1, the lower limit of charge of insulated resistance. Return > 35.0 // 35uA

1.5.39 FUNCtion:IR:CH1 (Channel 1 Setting of Insulated Resistance)

FUNCtion:IR:CH1 command is used to set channel 1 parameter of insulated resistance. （**Only UT5320R-S4/S8**）

Command Synatx	FUNCtion:IR:CH1 <integer (1~20) >,{HIGH,LOW,OPEN}
Parameter	Parameter1：<integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2：{HIGH,LOW,OPEN}, HIGH：high-end ； LOW：low-end ； OPEN：open circuit； Notice: Test mode of this step must be IR, otherwise this instruction is invalid.
For Example	Send > FUNC:IR:CH1 1,LOW // Set step 1, channel 1 of insulated resistance to low-end.
Query Syntax	FUNCtion:IR:CH1? <integer (1~20) >
Query Response	{HIGH,LOW,OPEN}
For Example	Send > FUNC:IR:CH1? 1 // Query step 1, channel 1 of insulated resistance. Return > OPEN // Open circuit

Other channel setting is the same as above table, for example

Command Synatx	FUNCtion:IR:CH2 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 2 parameter of insulated resistance.
	FUNCtion:IR:CH5 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 5 parameter of insulated resistance.
	FUNCtion:IR:CH8 <integer (1~20) >,{HIGH,LOW,OPEN} // This command is used to set Channel 8 parameter of insulated resistance.

Query Syntax	FUNCtion:IR:CH4? <integer (1~20) > // This command is query to set Channel 4 parameter of insulated resistance. FUNCtion:IR:CH6? <integer (1~20) > // This command is query to set Channel 6 parameter of insulated resistance.

	FUNCtion:IR:CH7? <integer (1~20) > // This command is query to set Channel 7 parameter of insulated resistance.

1.5.40 FUNCtion:CK:VOLT (Output Voltage of Contact Checking)

FUNCtion:CK:VOLT command is used to set output voltage of contact checking. (**Only UT5320R-S4/S8**)

Command Synatx	FUNCtion:CK:VOLT <integer (1~20) >,<integer (50~400) >
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <integer (50~400) > Notice: Test mode of this step must be CK, otherwise this instruction is invalid.
For Example	Send > FUNC:CK:VOLT 2,200 // Set step 2, output voltage of contact checking to 200 V.
Query Syntax	FUNCtion:CK:VOLT? <integer (1~20) >
Query Response	<integer (50~400) >
For Example	Send > FUNC:CK:VOLT? 2 // Query step 2, output voltage of contact checking. Return > 200 // 200 V

1.5.41 FUNCtion:CK:LOWC (Lower Limit of Current of Contact Checking)

FUNCtion:CK:LOWC command is used to set the lower limit of current of contact checking. (**Only UT5320R-S4/S8**)

Command Synatx	FUNCtion:CK:LOWC <integer (1~20) >,<float>
Parameter	Parameter1: <integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2: <float>, data range: 0.001~10.00, data unit: mA Notice: Test mode of this step must be CK, otherwise this instruction is invalid.
For Example	Send > FUNC:CK:LOWC 1,0.6 // Set step 1, the lower limit of current of contact checking to 0.6mA.
Query Syntax	FUNCtion:CK:LOWC? <integer (1~20) >
Query Response	<float>
For Example	Send > FUNC:CK:LOWC? 1 // Query step 1, the lower limit of current of contact checking. Return > 0.600 // 0.6mA

1.5.42 FUNCtion:CK:CH1 (Channel 1 Setting of Contact Checking)

FUNCtion:CK:CH1 command is used to set channel 1 parameter of contact checking（**Only UT5320R-S4/S8**）

Command Synatx	FUNCtion:CK:CH1 <integer (1~20) >,{OFF,ON}
Parameter	Parameter1：<integer (1~20) > is selected step, this parameter cannot greater than the total steps! Parameter2：{OFF,ON}，OFF：turn off contact checking；ON：turn on contact checking； Notice: Test mode of this step must be CK, otherwise this instruction is invalid.
For Example	Send > FUNC:CK:CH1 1,ON // Turn on channel 1 of contact checking in step 1.
Query Syntax	FUNCtion:CK:CH1? <integer (1~20) >
Query Response	{OFF,ON}
For Example	Send > FUNC:CK:CH1? 1 // Query step 1, channel 1 setting parameter of contact checking. Return > ON // Turn on contact checking.

Other channel setting is the same as above table, for example

Command Synatx	FUNCtion:CK:CH2 <integer (1~20) >,{OFF,ON} // This command is used to set Channel 2 parameter of contact checking.
	FUNCtion:CK:CH5 <integer (1~20) >,{OFF,ON} // This command is used to set Channel 5 parameter of contact checking.
	FUNCtion:CK:CH8 <integer (1~20) >,{OFF,ON} // This command is used to set Channel 8 parameter of contact checking.

Query Syntax	FUNCtion:CK:CH4? <integer (1~20) > // This command is used to query Channel 4 parameter of contact checking. FUNCtion:CK:CH6? <integer (1~20) > // This command is used to set Channel 6 parameter of contact checking. FUNCtion:CK:CH7? <integer (1~20) > // This command is used to set Channel 7 parameter of contact checking.

1.5.43 FUNCtion:SOUR? (Query All Test Settings of the Current Test Step)

FUNCtion:SOUR? command is used to query all test settings of the current test step, the format of return data varies with the model;

1. UT5310/UT5320 Series Format of Returned Test Data Setting

Test Mode	Returned Test Data Setting
AC Withstand Voltage（AC）	Send 13 data with comma-separated: Total steps, the current step, test mode, output voltage, the upper limit

	of current, the lower limit of current, test time, rising time, falling time, electric arc, output frequency, current range, compensation to zero (mA)
DC Withstand Voltage（DC）	Send 15 data with comma-separated: Total steps, the current step, test mode, output voltage, the upper limit of current, the lower limit of current, test time, rising time, falling time, electric arc, the lower limit of charge, current range, compensation to zero (uA), waiting judgement, rising judgement
Insulated Rsistance （IR）	Send 11 data with comma-separated: Total steps, the current step, test mode, output voltage, the upper limit of resistance, the lower limit of resistance, rising judgement, the lower limit of charge, resistance range

2. UT5320R-S4/ UT5320R-S8 Format of Returned Test Data

Test Mode	Returned Test Data Setting
AC Withstand Voltage （AC）	Send 14data with comma-separated: Total steps, the current step, test mode, output voltage, the upper limit of current, the lower limit of current, rising judgement, output frequency, current range, compensation to zero (mA), channel sweep setting
DC Withstand Voltage （DC）	Send 16 data with comma-separated: Total steps, the current step, test mode, output voltage, the upper limit of current, the lower limit of current, rising time, falling time, electric arc, the lower limit of charge, current range, compensation to zero (uA), waiting judgement, rising judgement, channel sweep setting
Insulated Rsistance （IR）	Send 12 data with comma-separated: Total steps, the current step, test mode, output voltage, the upper limit of resistance, the lower limit of resistance, delay judgement time, rising time, falling time, the lower limit of charge, resistance range, channel sweep setting
Contack Checking （CK）	Send 6 data with comma-separated: Total steps, the current step, test mode, output voltage, the lower limit of current, channel sweep setting

Description of Test Data Setting

Total steps	<integer (1~20) >
The current step	<integer (1~20) >
Test mode	<integer (0~3) >，0：AC，1：DC，2: IR，3：CK；
Output voltage	<integer>, unit is V;
The upper limit of current The lower limit of current The upper limit of	<float> When test mode is AC, DC, CK, the unit is mA; when test mode is IR, the unit is MΩ; 0.000 represents OFF

resistance The lower limit of resistance	
Test time Rising time Electric arc Delay judgement time Waiting judgement	<float> Unit is s.
Electric arc	<integer (0~9) >，0 represents OFF;
Output frequency（AC）	<integer (0~1) >，0 represents 50Hz, 1 represents 60Hz;
Current range Resistance range	<integer (0~1) >， 0 represents fixed range,1 represents auto range;
Compensation to zero	<float>，AC 时单位为 mA, DC 时单位为 uA, 0.000 represents OFF;
The lower limit of charge	<float>，unit is mA, 0.000 represents OFF;
Rising judgement（DC）	<integer (0~1) >，0 represents OFF, 1 represents ON;
Channel sweep setting (**UT5320R-S4**)	1.When test mode is CK: <integer (0~1) ><integer (0~1) ><integer (0~1) ><integer (0~1) >, 0 represents OFF，1 represents ON; CH1 CH2 CH3 CH4 eg: 1101 represents CH1, CH2, CH4 represents contact checking is enabled; CH3represents contact checking is disabled; 2. When test mode is AC, DC, IR: <integer (0~2) ><integer (0~2) ><integer (0~2) ><integer (0~2) >, 0 represents open circuit，1 represents high-end ，2 represents low-end; CH1 CH2 CH3 CH4 eg: 1200 CH1 represents high-end ，CH2 represents low-end，CH3, CH4 represents open-circuit;
Channel sweep setting （**UT5320R-S8**）	1.When test mode is: <integer(0~1)><integer (0~1) ><integer (0~1) ><integer (0~1) ><integer (0~1) ><integer (0~1) ><integer (0~1) ><integer (0~1) >, 0 represents OFF，1 represents ON; Total 8 digits. 2. When test mode is AC, DC, IR: <integer(0~2)><integer (0~2) ><integer (0~2) ><integer (0~2) ><integer (0~2) ><integer (0~2) ><integer (0~2) ><integer (0~2) >, 0 represents open-circuit，1 represents high-end，2

	represents low-end; Total 8 digits.

1.6 SYSTem Sub-system

SYSTem Sub-system command is used to set the parameter of system. Most of these instructions are related to the instrument < System Seup > page.

Figure 0-3 SYSTem1 Sub-system Tree It is related to the test scheme of the instrument.

SYSTem	:TRIGger	{LOCAL,PLC}	Start mode setting
	:VOLume	{LOW,MED,HIGH}	Volume setting
	:KEYSound	{OFF(0),ON(1)}	Key sound setting
	:PASSBeep	{LONG,SHORT,OFF}	Pass beep setting
	:FAILBeep	{LONG,SHORT,OFF}	Fail beep setting
	:DELAy	<float>	Delay test setting
	:STEP	<float>	Item interval setting
	:FAIL	{STOP,CONT,REST,NEXT}	Fail mode setting
	:DISP	{ALL,LAST,PF}	Diplay mode setting
	:SMOD	{NORMAL,REPEAT,STEP}	Step mode setting
	:RESEt	{OFF(0),ON(1)}	Reset setting
	:CTRL	{FILE,STEP}	Sorting mode setting
	:PASSHold	<float>	Pass hold setting
	:TURN	{OFF(0),ON(1)}	Adjustable mode setting

Notice:
The parameter set by SYSTem1 sub-system will not automatically stored in the file.
After the parameter is set, it need to load FILE subsystem to save the parameter into the internal file.

Figure 0-4 SYSTem2 Sub-system Tree It can automatic save into system' s storage and don' t need extra instruction.

SYSTem	: LANGuage	{ENGLISH,CHINESE,EN,CN}	System language setting
	: RESult	{FETCH,AUTO}	Result send setting
	: TIME	<YEAR>,<MONTH>,<DAY>,<HOUR>,<MINUTE>,<SECOND>	Time setting of the instrumen
	: DEFault		Restore to the

		factory setting

1.6.1 SYSTem:TRIGger (Start Mode)

SYSTem:TRIGger command is used to set the start mode for high voltage.

Command Synatx	SYSTem:TRIGger {LOCAL,PLC}
Parameter	{LOCAL,PLC} LOCAL: START key on the fronet panel. PLC: Signal start on HANDLER interface.
For Example	Send > SYST:TRIG LOCAL // set to LOCAL.
Query Syntax	SYST:TRIG?
Query Response	{LOCAL,PLC}

1.6.2 SYSTem:VOLume

SYSTem:VOLume command is used to set the volume.

Command Synatx	SYSTem:VOLume {LOW,MED,HIGH}
Parameter	{LOW,MED,HIGH} LOW: low sound MED: middle sound HIGH: high sound
For Example	Send > SYST:VOL HIGH // Set the sound to high.
Query Syntax	SYST: VOL?
Query Response	{LOW,MED,HIGH}

1.6.3 SYSTem:KEYSound

SYSTem:KEYSound command is used to set the switch of key sound.

Command Synatx	SYSTem:KEYSound {OFF,ON,0,1}
For Example	Send > SYST:KEYS ON // Tunr on key sound.
Query Syntax	SYST:KEYS?
Query Response	{OFF,ON}

1.6.4 SYSTem:PASSBeep

SYSTem:PASSBeep command is used to set pass beep.

Command Synatx	SYSTem:PASSBeep {LONG,SHORT,OFF}
For Example	Send > SYST:PASSB OFF // Turn off pass beep.
Query Syntax	SYST:PASSB?
Query Response	{LONG,SHORT,OFF}

1.6.5 SYSTem:FAILBeep

SYSTem:FAILBeep command is used to set fail beep.

Command Synatx	SYSTem:FAILBeep {LONG,SHORT,OFF}
For Example	Send > SYST:FAILB SHORT // Set fail beep to dual short tone.
Query Syntax	SYST:FAILB?
Query Response	{LONG,SHORT,OFF}

1.6.6 SYSTem:DELAy

SYSTem:DELAy command is used to set delay test.

Command Synatx	SYSTem:DELAy <float>
Parameter	<float>, data range: 0~99.9 (0 represents OFF)
For Example	Send > SYST:DELA 2 // Set delay test to 2s.
Query Syntax	SYST:DELA?
Query Response	<float>

1.6.7 SYSTem:STEP (Item Interval)

SYSTem:STEP command is used to set item interval.

Command Synatx	SYSTem:STEP <float>
Parameter	<float>, data range: 0~99.9 (0 represents OFF)
For Example	Send > SYST:STEP 0.5 // Set item interval to 0.5s
Query Syntax	SYST:STEP?
Query Response	<float>

1.6.8 SYSTem:FAIL (Fail Mode)

SYSTem:FAIL command is used to set fail mode of the instrument.

Command Synatx	SYSTem:FAIL {STOP,CONT,REST,NEXT}
Parameter	{STOP,CONT,REST,NEXT} STOP CONT: continue REST NEXT: next step
For Example	Send > SYST:FAIL STOP // Fail stop mode.
Query Syntax	SYST:FAIL?
Query Response	{STOP,CONT,REST,NEXT}

1.6.9 SYSTem:DISP (Display Mode)

SYSTem:DISP command is used to set display mode of the instrument.

Command Synatx	SYSTem:DISP {ALL,LAST,PF}
Parameter	{ALL,LAST,PF} ALL LAST: last step PF: Pass/Fail
For Example	Send > SYST:DISP PF // Set display mode to pass/fail.
Query Syntax	SYST:DISP?
Query Response	{ALL,LAST,PF}

1.6.10 SYSTem:SMOD (Step Mode)

SYSTem:SMOD command is used to set step mode of the instrument.

Command Synatx	SYSTem:SMOD {NORMAL,REPEAT,STEP}
Parameter	{NORMAL,REPEAT,STEP} NORMAL REPEAT STEP: single step
For Example	Send > SYST:SMOD STEP // Set step mode to STEP.
Query Syntax	SYST:SMOD?
Query Response	{NORMAL,REPEAT,STEP}

1.6.11 SYSTem:RESEt

SYSTem:RESEt command is used to reset the switch.

Command Synatx	SYSTem:RESEt {OFF,ON,0,1}
For Example	Send > SYST:RESEt ON // Turn on RESET.
Query Syntax	SYST:RESE?
Query Response	{OFF,ON}

1.6.12 SYSTem:CTRL (Sorting Mode)

SYSTem:CTRL command is used to set sorting mode.

Command Synatx	SYSTem:CTRL {FILE,STEP}
Parameter	{FILE,STEP} FILE STEP: single step
For Example	Send > SYST:CTRL FILE // Set sorting mode to FILE.
Query Syntax	SYST:CTRL?
Query Response	{FILE,STEP}

1.6.13 SYSTem:PASSHold

SYSTem:PASSHold command is used to set pass hold.

Command Synatx	SYSTem:PASSHold <float>
Parameter	<float>, data range: 0~99.9 (0 reprenests key control)
For Example	Send > SYST:PASSH 0.5 // Set pass hold to 0.5s.
Query Syntax	SYST:PASSH?
Query Response	<float>

1.6.14 SYSTem:TURN (Adjustable Mode)

SYSTem:TURN command is used to set the switch of adjustable mode.

Command Synatx	SYSTem:TURN {OFF,ON,0,1}
For Example	Send > SYST:TURN ON // Turn on adjustable mode.
Query Syntax	SYST:TURN?
Query Response	{OFF,ON}

1.6.15 SYSTem:LANGuage

Lanauage setting of the instrument.

Command Synatx	SYSTem:LANGuage {ENGLISH,CHINESE,EN,CN}
For Example	Send > SYST:LANG EN // Set to English display.
Query Syntax	SYST:LANG?
Query Response	{ENGLISH,CHINESE}

1.6.16 SYSTem:RESult (Send Result)

SYSTem:RESult command can set the mode of data send: Auto send or send via FETCH instruction.

Command Synatx	SYSTem:RESult {FETCH,AUTO}
Parameter	{FETCH,AUTO} FETCH: data need send via fetch? instruction and then return to the host computer, the instrument passive send. AUTO: After the completion of each test, the test result is automatically sent to the host computer, and the instrument will auto send the data without the involvement of the upper computer.
For Example	Send > SYST:RES AUTO // Set to Auto Send.
Query Syntax	SYST:RES?
Query Response	{FETCH,AUTO}

1.6.17 SYSTem:TIME

Command Synatx	SYSTem:TIME <YEAR>,<MONTH>,<DAY>,<HOUR>,<MINUTE>,<SECOND>
For Example	Send > SYST:TIME 2022,1,17,11,15,20 // 2022-1-17 11:15:20
Query Syntax	SYSTem:TIME?
Query Response	<YEAR>-<MONTH>-<DAY> <HOUR>:<MINUTE>:<SECOND>
For Example	Send > SYST:TIME? Return > 2022-1-17 11:15:20

1.6.18 SYSTem:DEFault (Factory Setting)

This command can restore all the settings to factory setting. This command will not affect calibration data.

Command Synatx	SYSTem:DEFault
For Example	Send > SYST:DEF

1.7 FILE Sub-system

FILE Sub-system is used to manage file, it can save user's parameter into the internal storage or load the flash file into the system.

Figure 0-5 FILE Sub-system Tree

FILE	: SAVE	<integer (1~100) >	Save the specified file.
	: LOAD	<integer (1~100) >	Load the specified file.
	: DELete	<integer (1~100) >	Delete file.

1.7.1 FILE:SAVE

FILE:SAVE command is used to save the current settings into the specified file. The file number is 1~100.

Command Synatx	FILE:SAVE <integer (1~100) >
For Example	Send > FILE:SAVE 2 // Save into File 2.
Query Syntax	FILE? // Query the current load file number.
Query Response	<integer (1~20) >
For Example	Send > FILE? Return > 1 // The current load file is File 1.

1.7.2 FILE:LOAD (Load File)

FILE:LOAD command is used to load the file data into the system. The file number is 1~100.

Command Synatx	FILE:LOAD <integer (1~100) >
For Example	Send > FILE:LOAD 1 // Load File 1 data into the system.

1.7.3 FILE:DELete

FILE:DELete command is used to delete the specified file data. The file number is 1~100.

Command Synatx	FILE:DELete <integer (1~100) >
For Example	Send > FILE:DEL 5 // Delete File 5.

1.8 TEST (Start Test Sub-system)

TEST function is the same as START key.

Command Synatx 1	TEST
Command Synatx 2	FUNCtion:STARt // Function of Command Synatx is equal to Command Synatx 1.
For Example	Send > TEST // Start test.

1.9 RESET (Stop Test Sub-system)

RESET function is the same as STOP key.

Command Synatx 1	RESET
Command Synatx 2	FUNCtion:STOP // Function of Command Synatx is equal to Command Synatx 1.
For Example	Send > RESET // Stop test.

1.10 IDN? Sub-system

IDN? Sub-system command is used to return the version number of the instrument.

Query Syntax	IDN?
Query Response	<Manufacturer>,<MODEL>,<Function>,<Revision>
For Example	Send > IDN?

	Return > HAOYI,UT5310,HIPOT TESTER,REV A1.5

1.11 SN? Sub-system

SN? Sub-system command is used to return the serial number of the instrument.

Query Syntax	SN?
Query Response	<SN>
For Example	Send > SN? Return > H10032222110A001

1.12 FETCh? (Fetch Result Sub-system)

In <Measurement Display> page, Send FETCH? command will return the current test data.

Notice:
This instruction can only vaild in < Measurement Display > page!

Query Syntax	FETCh?	
Query Response	<Test Step>,< Test Mode>,< Test Voltage (kV)>, < Test Current (mA) or Test Resistance (MΩ)>, <Sorting Result >;	
Description of Response	<Test Step >	<integer (1~20) >
	< Test Mode >	{AC,DC,IR,CK} **CK only support UT5320R-S4/ UT5320R-S8!**
	< Test Voltage >	<float> , the default unit is kV . Return data will not be return.
	< Test Current or Test Resistance >	<float> , the default unit of AC/DC/CK is mA, the default unit of IR is MΩ. Return data will not with unit.
	< Sorting Result >	1. The item not including unfinished test step. Return data is: <Test Step >,< Test Mode>,<Voltage>, <Current or Resistance>; 2. The step test has finished. Return sorting results are {PASS,SHORT,ARC,GFI,VOLT ERR,HI-Limit,LO-Limit,Charge Lo, CK FAIL}

		PASS：Qualified SHORT：Short-circuit ARC：Electric-arc GFI：Grounded failed VOLT ERR：overvoltage HI-Limit：Over upper limit LO-Limit：Over lower limit Charge Lo：The lower limit of charge CK FAIL: bad contact
	Additional Notes	1. The data delimiter in the step is [,] 2. The data delimiter between different steps is [;] 3. The end mark of data defaults to 0x0A.

For Example 1	Send > FETCH?
	Return >
	1,IR,0.103,100.272,PASS;2,AC,1.009,0.017,PASS;3,DC,2.009,0.0632,PASS;
	
	// Test step 1，Test mode：IR，Test voltage：0.103 kV, Insulated resistance：100.272MΩ ，Judgement：Pass；
	// Test step 2，Test mode：AC，Test voltage：1.009 kV, AC：0.017mA ，Judgement：Pass；
	// Test step 3，Test mode：DC，Test voltage：2.009kV, DC：0.0632mA ，Judgement：Pass；	

For Example 2	Send > FETCH?
	Return > 1,AC,0.062,0.007,PASS;2,AC,0,0;
	// Test step 1，Test mode：AC，Test voltage：0.062 kV, AC：0.007mA ，Judgement：Pass；
	// Test step 2 didn’t finish；	

2. Modbus（RTU）Communication Protocol

This chapter includes the following contents：

- Data format
- Function code
- Register
- Read out multiple registers
- Write multiple registers

2.1 Data Format

Following Modbus（RTU）communication protocol，the instrument respond to directive of upper computer and returns standard response frame.

2.1.1 Directive Frame

Figure 2-1 Modbus Directive Frame

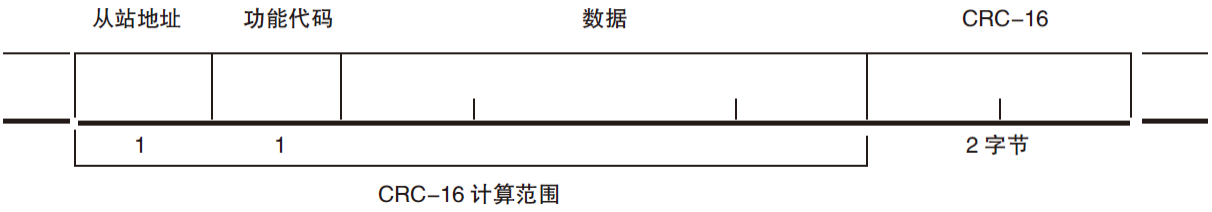


Table 2-1 Description of Directive Frame

	It needs mute interval time of 3.5 character at least
Slave station address	1byte Modbus support 00~0x63 slave station Unified broadcast time 00 If the instrument doesn’ t have optional RS485, the default slave station address is 0x01.
Function code	1byte 0x03：read out multiple registers 0x04：=03H，not use 0x06：write single register, it can replace by 10H 0x08: echo test（only used when debug） 0x10：write multiple registers
Data	the specified register address, quantity and content
CRC-16	2bytes，low order at the front Cyclic Redundancy Check Calculating all the data from slave station address to the last data, get CRC-16 check code.
	It needs mute interval time of 3.5 character at least.

2.1.2 Response Frame

Except 00H directive from slave address boardcast, other slave station address will returns response frame.

Figure 2-2 Normal Response Frame

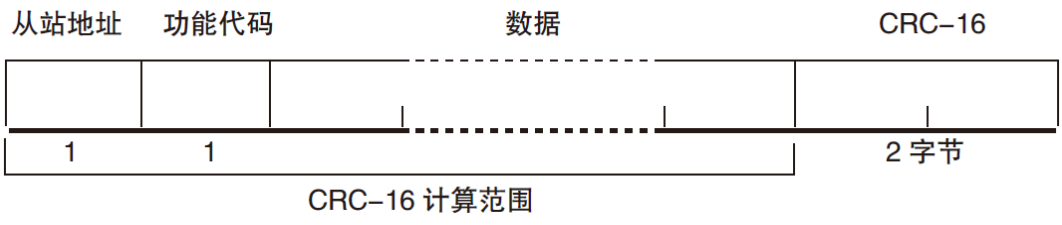


Figure 2-3 Anomaly Response Frame

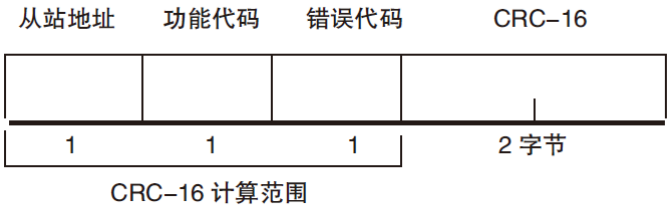


Table 2-2 Description of Anomaly Response

Slave station address	1byte Slave station address original returns
Function code	1byte Function code logic of program frame OR on BIT7 (0x80), For example: 0x03 OR 0x80 = 0x83
Error code	Exception code: 0x01 function code error (function code does not support) 0x02 register error (register does not exist) 0x03 data error 0x04 execution error
CRC-16	2bytes, low order in the front Cyclic Redundancy Check Calculating all the data from slave station address to the last data, get CRC-16 check code.

2.1.3 No Response

The instrument does not handle and response any case as follows, it may occurs communication time-out.

1. Slave station address error.
2. Transmission error.
3. CRC-16 error.
4. Bit error, For example: Function code 0x03 total bit must be 8 and received bit should less than or greater than 8 bytes.
5. It represents boardcast address when the slave station is 0x00. The instrument has no response.

2.1.4 Error Code

Table 2-3 Description of Error Code

Error Code	Name	Description	Priority
0x01	Function code error	Function code does not exist	1
0x02	Register error	Register does not exist	2

0x03	Data error	quantity of register or byte error	3
0x04	Execution error	Illegal data, written data is not in the allowed range	4

2.2 Function Code

The instrument can only support several function code, other function code doesn't support.

Table 2-4 Function Code

Function code	Name	Description
0x03	Read out multiple register	Read multiple serial registers data.
0x10	Write multiple registers	Write multiple serial registers data.

2.3 Register

The number of registers in the instrument is 2-byte mode, that is each time written 2 bytes, For example: speed register is 0x3002, data is 2 bytes, the numerical value must be written to 0x0001.

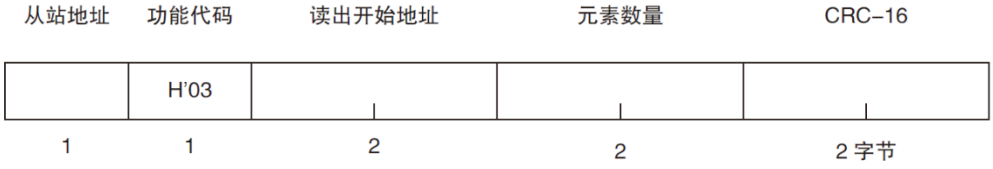
Data:

The instrument supports numerical value as follows,

1. One register, double bytes (16bits) integer, For example: 0x64 → 00 64
2. Two registers, four bytes (32bits) integer, For example: 0x12345678 → 12 34 56 78
3. Two registers, four bytes (32bits) float-point number with single precision, 3.14 → 40 48 F5 C3

2.4 Read Out Multiple Registers

Figure 2-4 Read Out Multiple Registers (0x03)

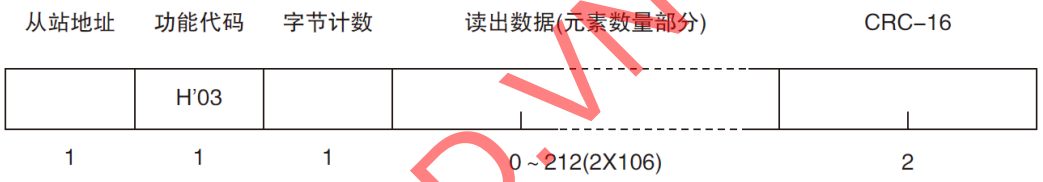


Read function code of multiple registers are 0x03.

Table 2-5 Description of Read Out Multiple Registers

Name	Name	Description
	Slave station address	If the RS485 address is not specified, the default value is 01.
0x03	Function code	
	Initial address	The initial address of register, refer to Modbus instruction set.
	Read number of register 0001~006A（106）	Read register quantity continuously, refer to Modbus instruction set. Make sure all the register address are exit, otherwise returns error fram.
CRC-16	Check code	

Figure 2-5 Read Out Response Frame of Multiple Registers（0x03）



Name	Name	Description
	Slave station address	Original returns
0x03 or 0x83	Function code	Normal：0x03 Error code：0x83
	The number of byte	=Register quantity x 2 For example：one register returns 02
	Data	Read data
CRC-16	Check code	

2.5 Written Multiple Registers

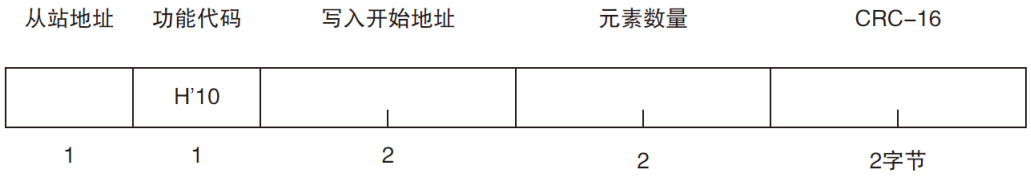
Figure 2-6 Written Multiple Registers（0x10）

Table 2-6 Written Multiple Registers

Name	Name	Description
	Slave station address	If the RS485 address is not specified, the default value is 01.
0x10	Function code	
	Initial address	The initial address of register, refer to Modbus

		instruction set.
	Written number of register 0001~0068（104）	Read register quantity continuously, refer to Modbus instruction set. Make sure all the register address are exit, otherwise returns error fram.
	Number of bytes	= Register quantity x 2
CRC-16	Check code	

Figure 2-7 Written Response Frame of Multiple Registers （0x03）



Name	Name	Description
	Slave station address	Original returns
0x10 or 0x90	Function code	Normal：0x10 Error code：0x90
	Initial address	
	The number of register	
	CRC-16 check code	

3. Modbus (RTU) Instruction Set

This chapter includes the following content：

- Register Address

3.1 Overview of Register

All register address of the instrument includes as the following table.

Notices:

1. Unless otherwise specified，numerical value of the instruction and response fram are all hexadecimal data.
2. Register only includes fetch test result and start/stop test. If need to customize other insturctions, please contact UNI-T sales department.
3. Floating point number on-line conversion,

Plese refer to website http://www.binaryconvert.com/convert_float.html

Register Addree	Name	Numerical Value	Description
0100	Read test voltage of step 1.	4 bytes floating point number	Read-only register, data takes 2 registers.
0102	Read test current/resistance of step 1.	4 bytes floating point number	Read-only register, data takes 2 registers.
0104	Read sorting result of step 1.	2 bytes integer **0011: PASS** **0100: Short-cuircuit** **0101: Electric arc** **0110 : Grounded failed** **0111: Overvoltage** **1000 : Over upper limit** **1001 : Over lower limit** **1010: Lower limit of charge** **1001: Bad contact**	Read-only register, data takes 1 register.
0105	Read test voltage of step 2.	4 bytes floating point number	Read-only register, data takes 2 registers.
0107	Read test current/resistance of step 2.	4 bytes floating point number	Read-only register, data takes 2 registers.
0109	Read sorting result of step 2.	2 bytes integer, ditto	Read-only register, data takes 1 register.
010A	Read test voltage of step 3.	4 bytes floating point number	Read-only register, data takes 2 registers.
010C	Read test current/resistance of step 3.	4 bytes floating point number	Read-only register, data takes 2 registers.
010E	Read sorting result of step 3.	2 bytes integer, ditto	Read-only register, data takes 1 register.
010F	Read test voltage of step 4.	4 bytes floating point number	Read-only register, data takes 2 registers.
0111	Read test current/resistance of step 4.	4 bytes floating point number	Read-only register, data takes 2 registers.
0113	Read sorting result of step 4.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0114	Read test voltage of step 5.	4 bytes floating point number	Read-only register, data takes 2 registers.

0116	Read test current/resistance of step 5.	4 bytes floating point number	Read-only register, data takes 2 registers.
0118	Read sorting result of step 5.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0119	Read test voltage of step 6.	4 bytes floating point number	Read-only register, data takes 2 registers.
011B	Read test current/resistance of step 6.	4 bytes floating point number	Read-only register, data takes 2 registers.
011D	Read sorting result of step 6.	2 bytes integer, ditto	Read-only register, data takes 1 register.
011E	Read test voltage of step 7.	4 bytes floating point number	Read-only register, data takes 2 registers.
0120	Read test current/resistance of step 7.	4 bytes floating point number	Read-only register, data takes 2 registers.
0122	Read sorting result of step 7.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0123	Read test voltage of step 8.	4 bytes floating point number	Read-only register, data takes 2 registers.
0125	Read test current/resistance of step 8.	4 bytes floating point number	Read-only register, data takes 2 registers.
0127	Read sorting result of step 8.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0128	Read test voltage of step 9.	4 bytes floating point number	Read-only register, data takes 2 registers.
012A	Read test current/resistance of step 9.	4 bytes floating point number	Read-only register, data takes 2 registers.
012C	Read sorting result of step 9.	2 bytes integer, ditto	Read-only register, data takes 1 register.
013D	Read test voltage of step 10.	4 bytes floating point number	Read-only register, data takes 2 registers.
013F	Read test current/resistance of step 10.	4 bytes floating point number	Read-only register, data takes 2 registers.
0131	Read sorting result of step 10.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0132	Read test voltage of step 11.	4 bytes floating point number	Read-only register, data takes 2 registers.
0134	Read test current/resistance of	4 bytes floating point number	Read-only register, data takes 2 registers.

	step 11.		
0136	Read sorting result of step 11.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0137	Read test voltage of step 12.	4 bytes floating point number	Read-only register, data takes 2 registers.
0139	Read test current/resistance of step 12.	4 bytes floating point number	Read-only register, data takes 2 registers.
013B	Read sorting result of step 12.	2 bytes integer, ditto	Read-only register, data takes 1 register.
013C	Read test voltage of step 13.	4 bytes floating point number	Read-only register, data takes 2 registers.
013E	Read test current/resistance of step 13.	4 bytes floating point number	Read-only register, data takes 2 registers.
0140	Read sorting result of step 13.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0141	Read test voltage of step 14.	4 bytes floating point number	Read-only register, data takes 2 registers.
0143	Read test current/resistance of step 14.	4 bytes floating point number	Read-only register, data takes 2 registers.
0145	Read sorting result of step 14.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0146	Read test voltage of step 15.	4 bytes floating point number	Read-only register, data takes 2 registers.
0148	Read test current/resistance of step 15.	4 bytes floating point number	Read-only register, data takes 2 registers.
014A	Read sorting result of step 15.	2 bytes integer, ditto	Read-only register, data takes 1 register.
014B	Read test voltage of step 16.	4 bytes floating point number	Read-only register, data takes 2 registers.
014D	Read test current/resistance of step 16.	4 bytes floating point number	Read-only register, data takes 2 registers.
014F	Read sorting result of step 16.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0150	Read test voltage of step 17.	4 bytes floating point number	Read-only register, data takes 2 registers.
0152	Read test current/resistance of step 17.	4 bytes floating point number	Read-only register, data takes 2 registers.
0154	Read sorting result of	2 bytes integer,	Read-only register, data

	step 17.	ditto	takes 1 register.
0155	Read test voltage of step 18.	4 bytes floating point number	Read-only register, data takes 2 registers.
0157	Read test current/resistance of step 18.	4 bytes floating point number	Read-only register, data takes 2 registers.
0159	Read sorting result of step 18.	2 bytes integer, ditto	Read-only register, data takes 1 register.
015A	Read test voltage of step 19.	4 bytes floating point number	Read-only register, data takes 2 registers.
015C	Read test current/resistance of step 19.	4 bytes floating point number	Read-only register, data takes 2 registers.
015E	Read sorting result of step 19.	2 bytes integer, ditto	Read-only register, data takes 1 register.
015F	Read test voltage of step 20.	4 bytes floating point number	Read-only register, data takes 2 registers.
0161	Read test current/resistance of step 20.	4 bytes floating point number	Read-only register, data takes 2 registers.
0163	Read sorting result of step 20.	2 bytes integer, ditto	Read-only register, data takes 1 register.
0500	Start/Stop test	2 bytes integer 0000：Stop 0002：Start	Read-only register, data takes 1 register.

3.2 Fetch Test Result

Take read test result of step 1 and step 2 as the example.
In example, test mode of step 1 is AC, test mode of step 2 is IR.
The default unit of test voltage is kV; The default unit of test current is mA;
The default unit of test resistance is MΩ.

3.2.1 Fetch Test Result of Step 1

Register 0100~0101 is used to fetch test voltage data of step 1.
Send:

1	2	3	4	5	6	7	8
01	03	01	00	00	02	C5	F7
Slave station	Read	Register		Register quantity		Check code	

Response:

1	2	3	4	5	6	7	8	9
01	03	04	**3F**	**03**	**22**	**F1**	DF	03
01	03	Byte	Float-point number with single precision				CRC-16	

B4~B7 is float-point number with single precision, byte order is AA BB CC DD.

Test data：3F 03 22 F1 converts to float-point number：

0x3F0322F1 = 5.122E-1 (decimalism)，that is 0.512kV;

3.2.2 Fetch Test Result of Current/Resistance of Step 1

Register 0102~0103 is used to fetch test current/resistance data of step 1.

Send:

1	2	3	4	5	6	7	8
01	03	01	02	00	02	64	37
Slave station	Read	Register		Register quantity		Check code	

Response:

1	2	3	4	5	6	7	8	9
01	03	04	**3C**	**42**	**FD**	**FF**	56	A7
01	03	Byte	Float-point number with single precision				CRC-16	

B4~B7 is float-point number with single precision,

0x3C42FDFF = 1.190E-2 (decimalism)，that is 0.012mA;

3.2.3 Fetch Sorting Result of Step 1

Register 0104 is used to fetch sorting test result of step 1.

Send:

1	2	3	4	5	6	7	8
01	03	01	04	00	01	C4	37
Slave station	Read	Register		Register quantity		Check code	

Response:

1	2	3	4	5	8	9
01	03	02	**00**	**03**	F8	45
01	03	Byte	Float-point number with single precision		CRC-16	

B4~B5 is integer，return result shows0x0003, that means PASS;

3.2.4 Simultaneous Fetch Test Result of Step 1 and 2

Register 0100~0109 is used to simultaneous fetch test result of step 1 and

2.

Refer to this example, test data of multiple steps can be simultaneous fetched.

Send:

1	2	3	4	5	6	7	8
01	03	01	00	00	0A	C4	31
Slave station	Read	Register		Register quantity		Check code	

Response:

1	2	3	4-7	8-11	12-13	14-17	18-21	22-23	24	25
01	03	14	**3F0322F1**	**3C42FDFF**	**0003**	**3DD2C1D2**	**42C8F3CD**	**0003**	1B	0
01	03	Byte	Voltage of step 1	Current of step 1	Sorting of step 1	Voltage of step 2	Resistance of step 2	Sorting of step 2	CRC-16	

B4~B7: 0x3F0322F1 voltage of step 1 (float-point number) = 0.512kV

B8~B11: 0x3C42FDFF current of step 1 (float-point number) = 0.012mA

B12~B13: 0x0003 sorting of step 1 (integer) = PASS

B4~B7: 0x3DD2C1D2 voltage of step 2 (float-point number) = 0.103kV

B8~B11: 0x42C8F3CD resistance of step 2 (float-point number) = 100.5MΩ

B12~B13: 0x0003 sorting of step 2 (integer) = PASS

3.3 Start/Stop Test

Wirtten:

1	2	3	4	5	6	7	8	9	10	11
01	10	05	00	00	01	02	**00**	**02**	72	91
Station number	Wirte	Register		Register quantity		Byte	Data		CRC16	

B8~B9:

0000：Stop

0002：Start

Wirtten Return：

1	2	3	4	5	6	7	8
01	10	05	00	00	01	01	05
Station number	Wirte	Register		Register quantity		CRC16	