UNI-T®

Instruments.uni-trend.com

Programming Manual

UT5583 Insulation Resistance Tester

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.

Product Certification

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

1. SCPI Introduction

This chapter contains the following contents.

- Command Parser —— learn about a certain rule of command parser
- Command Syntax —— write rule of command line
- Query Syntax —— write rule of query command
- Query Response—— format of query respond
- Command Reference

This chapter provides all SCPI commands used by the instrument, so user can totally control all functions of the instrument through these command.

1.1 Parse Command String

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

For Example Valid command string:

AAA:BBB CCC;DDD EEE;:FFF

The instrument command parser is responsible for all command parsing and execution, and you must understand its parsing rules before writing a program.

1.1.1 Command Parse Rule

1. Command parser only parses and responds to ASCII data.
2. Command parser starts to anlysis the command after receive the end mark. The instrument accepts the following contents as the end mark.在

- CR
- CR+LF
- LF

3. The command parser will terminate the parsing immediately after parsing an error, and the current command will be invalidated.
4. The command parser is case-insensitive for parsing command strings.
5. The command parser supports abbreviated form of command and the detailed see the following section.
6. Add ADDR□local address::□ in front in RS485 mode of SCPI protocol, the local address can set 1-32. It is convenient for communication with multimachine via SCPI.
   Such as ADDR□1::□IDN? □ represents a blank.

7. The data send by the instrument, its end mark is default to 0x0A (LF).
8. Multi-command can send via semicolon [;] .

1.1.2 Symbol Stipulation and Definition

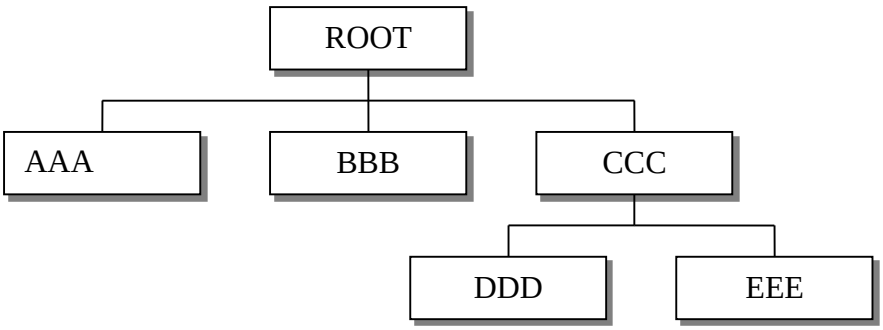
This chapter uses some symbols that are not part of the command tree, but only for a better understanding of the command string.

Mark	Description
<.....>	The text in angle brackets indicates the parameter of the command, such as <float> represnts float-point parameter. <integer> represnts integer parameter.
[.....]	The text in square brackets indicates the optional command.
{.....}	When the curly brackets contain several parameter items, it means that only one item can be selected from them.
Capital Letter	Abbreviated form of a command.
☐	Space mark represents a blank, it's only for reading.

1.1.3 Command Tree Structure

SCPI commands have a tree-like structure with three level (note: the command parser of this instrument can parse any level), where the highest level is called the subsystem command. Its subordinate can only be valid when subsystem command is selected. SCPI uses a colon (:) to separate high level commands from low level commands.

Figure 1-1 Command Tree Structure



For Example

ROOT:CCC:DDD ppp

ROOT Subsystem command

CCC Second level

DDD Third level

ppp Parameter

1.2 Command and Parameter

A command tree is consist of **command and [parameter]**, use a blank to separate (ASCII: 20H).

For Example AAA:BBB□1.234

Command [Parameter]

1.2.1 Command

Command word can be in long command format or in abbreviated form. Long format facilitates engineers to better understand the meaning of the command string; abbreviated form is suitable for writing.

1.2.2 Parameter

1. Single character command, no parameter.
   Such as AAA:BBB
2. Parameter can be string format and its abbreviated form is also follow the last section command "abbreviated rule".
   For example AAA:BBB 1.23
3. Parameter can be numerical value format.

<integer>	Integer 123, +123, -123
<float>	Arbitrary float-point number: Fixed float-point number: 1.23, -1.23 Float-point number with scientific notation: 1.23E+4, +1.23e-4

1.2.3 Separator

The instrument command parser can only receive allowable separator. Other separator will cause error "Invalid separator".

The allowable separator is as follows.

;	**Semicolon** is for separating two commands. *Such as* AAA:BBB 100.0;CCC:DDD
:	**Colon** is for separating command tree or restart the command tree. *Such as* AAA:BBB:CCC 123.4;:DDD:EEE 567.8
?	**Question mark** is for querying. *Such as* AAA?
□	**Blank** is for separating the parameter. *Such as* AAA:BBB□1.234

1.3 Command Reference

All commands is explained by the subsystem command order.

- DISPlay Display subsystem
- FUNCtion Function subsystem
- VOLTage Voltage subsystem
- TIMEr Time subsystem
- COMParator Comparator subsystem
- SYSTem System subsystem
- STATe State switching subsystem
- TRIGger Trigger subsystem
- FETCh? Fetch result subsystem
- CORRection Calibration subsystem
- FILE File subsystem

Common Command

- *IDN? Query subsystem of instrument information

1.4 DISPlay Subsystem

DISPlay subsystem is used to switch different display page.

Table 1-1 DISPlay Subsystem Tree

DISPlay	:PAGE	{ MEAS, MSET, COMP, FILE, SYST, SINF}

1.4.1 DISPlay:PAGE

DISP:PAGE is used to switch different display page.

Command Syntax	DISPlay:PAGE <Page name>
Parameter	<Page name> includes MEAS Measurement display page MSET Setup page COMP Comparator page FILE File page SYST System configuration page SINF System information page
For Example	Send > disp:page mset // Switch to the setup page
Query Syntax	DISPlay:PAGE?
Query Response	<Page name> includes MEAS Measurement display page

	MSET Setup page COMP Comparator page FILE File page SYST System configuration page SINF System information page
For Example	Send > disp:page? Return > MEAS

1.5 FUNCtion Subsystem

FUNCtion subsystem is used to set the test parameter of the instrument.

Note:

The parameter set by FUNCtion subsystem will not be automatically saved in the file. After the parameter is set, use FILE subsystem to save the parameter to the instrument's file.

Table 1-2 FUNCtion Subsystem Tree

FUNCtion	:RANGe	{Range number, MAX, MIN}		Range number setting
		MODE	{AUTO,HOLD,NOMinal}	Range mode setting
	:SPEED	{SLOW,MED,FAST}		Speed setting
	:CONTCHECK（CC）	{ON,OFF,0,1}		Contact inspection setting
	:DM	{R,RI}		Display mode setting
	:DD	<integer(4~5)>		Display digit setting

The parameter set by FUNCtion subsystem will not be saved in the system. It should reset when next boot up.

1.5.1 FUNCtion:RANGe

FUNC:RANG is used to set the range number, if the current range number is automatic, it will turn to lock.

Command Syntax	FUNCtion:RANGe {<integer(1~6)>,MIN,MAX}
Parameter	<Range number> 1~6 MIN (the minimum range)(=1)

	MAX (the maximum range) (=6)
For Example	Send > FUNC:RANG 2 // Switch to Range 2 Send > FUNC:RANG MAX // Switch to the maximum range (6)
Query Syntax	FUNCtion:RANGe?
Query Response	Range number 1~6
For Example	Send > FUNC:RANG? Return > 2

1.5.2 FUNCtion:RANGe :MODE

FUNC:RANG:MODE is used to switch the range mode.

Command Syntax	FUNCtion:RANGe:MODE {AUTO,HOLD,NOMinal}
For Example	Send > FUNC:RANG:MODE NOM // Switch to nominal mode
Query Syntax	FUNCtion:RANGe:MODE?
Query Response	{AUTO,HOLD,NOM} // Return to capital letter

1.5.3 FUNCtion:SPEED

FUNC:SPEED is used to set the test speed.

Command Syntax	FUNCtion:SPEED {SLOW,MED,FAST}
For Example	Send > FUNC:SPEED MED // Set to middle speed
Query Syntax	FUNCtion:SPEED?
Query Response	{SLOW,MED,FAST} // Return to capital letter

1.5.4 FUNCtion:CONTCHECK (CC)

FUNC: CONTCHECK or FUNC: CC is used to set the switch of contact inspection.

Command Syntax	FUNCtion:CONTCHECK {OFF,ON,0,1} FUNCtion:CC { OFF,ON,0,1}
For Example	Send > FUNC:CC ON // Turn on contact inspection
Query Syntax	FUNCtion:CONTCHECK? FUNCtion:CC?
Query Response	{ OFF,ON }

1.5.5 FUNC:DM

FUNC:DM is used to set the display mode.

Command Syntax	FUNCtion:DM { R,RI}
For Example	Send > FUNC:DM RI // Set to dual display of insulation resistance, leakage current Send > FUNC:DM R

	// Set the display mode to insulation resistance
Query Syntax	FUNCtion:DM?
Query Response	{ R,RI }

1.5.6 FUNC:DD

FUNC:DD is used to set the display digit.

Command Syntax	FUNCtion:DD <integer(4~5)>

For Example	Send > FUNC:DD 4
	// Set the display digit of insulation resistance and leakage current to 4
	Send > FUNC:DD 5
	// Set the display digit of insulation resistance and leakage current to 5

Query Syntax	FUNCtion:DD?
Query Response	<integer(4~5)>

1.6 VOLTage Subsystem

Voltage subsystem is used to set the test voltage.

Command Syntax	VOLTage <float>
Parameter	<float> 1~1000
For Example	Send > VOLT 100 // Set to 100V Send > VOLT 6.3 // Set to 6.3V
Query Syntax	VOLTage?
Query Response	<FixFloat> total digit is 6, decimal point is 1, insufficient digit will make up by prefix blank.
For Example	Send > VOLT? Return > ☐100.2 //☐ represents a blank Return > ☐☐☐6.3 //☐ represents a blank

Notes:

1. The parameter set by VOLTage subsystem will not be automatically saved in the file. After the parameter is set, use FILE subsystem to save the parameter to the instrument's file.
2. The voltage value can only be set in the stop state.
3. Returned voltage value is float-point 6 digits, decimal point is 1, insufficient digit will make up by prefix blank.

1.7 TIMEr Subsystem

TIMEr subsystem is used to set the time of charge, test, discharge and trigger delay.

Note:
The parameter set by TIMEr subsystem will not be automatically saved in the file. After the parameter is set, use FILE subsystem to save the parameter to the instrument's file.

Table 1-3 TIMEr Subsystem Tree

TIMEr	:CHARge	<float>	Charge time setting, 0 represents OFF.
	:TEST	<float>	Test time setting, 0 represents continuous.
	:DISCHarge	<float>	Discharge time setting, 0 represents OFF.
	:TRIGdelay	<float>	Trigger delay time setting, 0 represents OFF.

1.7.1 TIMEr:CHARge

TIMEr:CHARge is used to set charge time.

Command Syntax	TIMEr:CHARge <float>
Parameter	<float> float-point number, 0.1~999.9 // Set to 0, it represents the timer is OFF (the unit is s.)
For Example	Send > TIME:CHAR 1.5 // Charge timer sets to 1.5s Send > TIME:CHAR 0 // Charge timer sets to OFF
Query Syntax	TIMEr:CHARge?
Query Response	<FixFloat> total digit is 5, decimal point is 1, insufficient digit will make up by prefix blank.
For Example	Send > TIME:CHAR? Return > □50.0 // Charge timer sets to 50s Return > □□0.0 // Charge timer sets to OFF

1.7.2 TIMEr:TEST

TIMEr:TEST is used to set test time.

Command Syntax	TIMEr:TEST <float>
Parameter	<float> float-point number, 0.1~999.9 // Set to 0, it represents continuous test (the unit is s.)

For Example	Send > TIME:TEST 1.5 // Test time sets to 1.5s Send > TIME:TEST 0 // Test time sets to continuous
Query Syntax	TIMEr:TEST?
Query Response	<FixFloat> total digit is 5, decimal point is 1, insufficient digit will make up by prefix blank.
For Example	Send > TIME:TEST? Return > ☐50.0 // Test time sets to 50s Return > ☐☐0.0 // Test time sets to continuous

1.7.3 TIMEr:DISCHarge

TIMEr:DISCHarge is used to set discharge time.

Command Syntax	TIMEr:DISCHarge <float>
Parameter	<float> float-point number, 0.1~999.9 // Set to 0, it represents the timer is OFF (the unit is s.)
For Example	Send > TIME:DISCHarge 1.5 // Discharge time sets to 1.5s Send > TIME:DISCHarge 0 // Discharge time sets to continuous
Query Syntax	TIMEr:DISCHarge?
Query Response	<FixFloat> total digit is 5, decimal point is 1, insufficient digit will make up by prefix blank.
For Example	Send > TIME:DISCHarge? Return > ☐50.0 // Discharge time sets to 50s Return > ☐☐0.0 // Discharge time sets to continuous

1.7.4 TIMEr:TRIGdelay

TIMEr:TRIGdelay is used to set trigger delay.

Command Syntax	TIMEr:TRIGdelay <integer>
Parameter	<integer> 0~9999 // Set to 0, it represents the timer is OFF (the unit is ms.)
For Example	Send > TIME:TRIGdelay 10 // Trigger delay sets to 10ms Send > TIME:TRIGdelay 0 // Trigger delay sets to OFF
Query Syntax	TIMEr:TRIGdelay?
Query Response	<integer> positive integer 4 digits, insufficient digit will make up by prefix blank.

For Example	Send > TIME:TRIGdelay? Return > ☐☐10 // Trigger delay sets to 10ms Return > ☐☐☐0 // Trigger delay sets to OFF

1.8 COMParator Subsystem

COMP subsystem is used to set the parameter of comparator.

Notes:

- The parameter set by COMParator subsystem will not be automatically saved in the file. After the parameter is set, use FILE subsystem to save the parameter to the instrument's file.
- Comparator mode set to **[Single]**, the instrument will automatically set **[Test Time]** to **[Continuous]**. There is no need to set the **[Test Time]** through instructions.
- Comparator mode set to **[Perod]**, it need to set the specified **[Test Time]** by through the instructions.

Table 1-4 COMParator Subsystem Tree

COMParator	[:STATe]	{OFF,ON,0,1}	Comparator state
	:MODE	{SINGLE, PERIOD}	Comparator mode
	:BEEP	{OFF,PASS,FAIL}	Beeper setting
	:LOWer	<float>	Lower limit
	:UPper	<float>	Upper limit
	:LMT	<float>,<float>	Lower limit, upper limit

1.8.1 COMParator[:STATe]

COMParator[:STATe] is used to turn on/off the comparator.

Command Syntax	COMParator[:STATe] {OFF,ON,0,1}
For Example	Send > COMP:STAT ON // Turn on the comparator Send > COMP OFF // Turn off the comparator
Query Syntax	COMP:STAT?
Query Response	{OFF,ON}

1.8.2 COMParator:MODE

COMParator:MODE is used to set the comparator mode.

Command Syntax	COMParator:MODE {SINGLE, PERIOD}
For Example	Send > COMP:MODE PERIOD // Set to period

Query Syntax	COMP:MODE?
Query Response	{SINGLE, PERIOD}

1.8.3 COMParator:BEEP

COMParator:BEEP is used to set the comparator beep.

Command Syntax	COMParator:BEEP {OFF,PASS,FAIL}
For Example	Send > COMP:BEEP PASS // The beep is qualified.
Query Syntax	COMP:BEEP?
Query Response	{OFF,PASS,FAIL}

1.8.4 COMParator:LOWer

COMParator:LOWer is used to set the lower limit of comparator.

Command Syntax	COMParator:LOWer <float>
For Example	Send > COMP:LOW 10e6 // The lower limit sets to 10MΩ
Query Syntax	COMP:LOW?
Query Response	<Scifloat scientific notation>
For Example	Send > COMP:LOW? Return > 1.0000e+06 // =1MΩ

1.8.5 COMParator:UPper

COMParator:UPper is used to set the upper limit of comparator. The upper limit sets to 1E20, which represents infinitely great.

Command Syntax	COMParator:UPper <float>
For Example	Send > COMP:UP 100E6 // The upper limit sets to 100MΩ Send > COMP:UP 1E20 // The upper limit sets to 1E20 (infinitely great), that is not compare with the upper limit
Query Syntax	COMP:UP?
Query Response	<Scifloat scientific notation>
For Example	Send > COMP:UP? Return > 1.0000e+10 // =10GΩ Return > 1.0000e+20 // E20 (infinitely great), that is not compare with the upper limit

1.8.6 COMParator:LMT

COMParator:LMT is used to set the upper and lower limit of comparator.

Command Syntax	COMParator:LMT <float>,<float>

For Example	Send > COMP:LMT 10E6,100E6
	// Lower limit =10MΩ, upper limit =100MΩ
	Send > COMP:LMT 1E9,1E20
	// Lower limit =1GΩ, upper limit sets to OFF

Query Syntax	COMP:LMT?
Query Response	<Scifloat scientific notation>
For Example	Send > COMP:LMT? Return > 1.0000e+06 ,1.0000e+20 // Lower limit =1MΩ, upper limit sets to OFF

1.9 SYSTem Subsystem

SYSTem subsystem is used to set the parameter of system. These instructions are related to <System Configuration>.

Note:

The parameter set by SYSTem subsystem will not be automatically saved in the file. After the parameter is set, use FILE subsystem to save the parameter to the instrument's file.

Table 1-5 SYSTem Subsystem Tree

SYSTem	:LANGuage	{ENGLISH,CHINESE,EN,CN}	System language setting
	:TIME	<YEAR>,<MONTH>,<DAY>,<HOUR>,<MINUTE>,<SECOND>	Built-in time setting
	:VOLume	{LOW,MED,HIGH}	Volume setting
	:KEYSound	{OFF(0),ON(1)}	Key sound setting
	:LIGHT	{L10,L30,L50,L70,L90,L100}	Backlight brightness setting
	:RESult	{FETCH,AUTO}	Result send setting

	:FILTER	{50Hz,60Hz}	Power frequency filter setting
	:DEFault		Factory setting

1.9.1 SYSTem:LANGuage

Instrument's language setting.

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

1.9.2 SYSTem:TIME

SYSTem:TIME is used to set system time

Command Syntax	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

1.9.3 SYSTem:VOLume

SYSTem:VOLume is used to set the volume.

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

1.9.4 SYSTem:KEYSound

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

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

1.9.5 YSTem:LIGHT

SYSTem:LIGHT is used to set the backlight brightness of LCD.

Command Syntax	SYSTem:LIGHT {L10,L30,L50,L70,L90,L100}
For Example	Send > SYST:LIGHT L90 // Brightness sets to 90%
Query Syntax	SYST:LIGHT?
Query Response	{L10,L30,L50,L70,L90,L100}

1.9.6 SYSTem:RESult

SYSTem:RESult is used to set data send method, automatic send or send via FETCH command.

Command Syntax	SYSTem:RESult {FETCH,AUTO}
Parameter	{FETCH,AUTO} FETCH: the data need use the command fetch? to return to the host computer, the instrument is passively sent. AUTO: After the completion of each test, the test results are automatically sent to the host computer, and the instrument actively sends data without the participation of the host computer.
For Example	Send > SYST:RES AUTO // Set to automatic send
Query Syntax	SYST:RES?
Query Response	{FETCH,AUTO}

1.9.7 SYSTem:FILTER

SYSTem:FILTER is used to set power frequency filter.

Command Syntax	SYSTem:FILTER {F50,F60}
For Example	Send > SYST:FILTER F60 // Power frequency filter sets to 60 Hz
Query Syntax	SYST:FILTER?
Query Response	{F50,F60}

1.9.8 SYSTem:DEFault

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

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

1.10 STATe Subsystem

STATe subsystem is used to switch the instrument state.

Table 1-6 STATe Subsystem Tree

STATe	:CHARage	Start test
	:DISCHarge	Stop test

1.10.1 STATe?

STATe? is used to query the test state.

Query Syntax	STATe?
Query Response	<integer(0~3)> 0: under the stop state 1: under the charging state 2: under the testing state 3: under the discharging state
For Example	Send > STATe? Return > 2 // Under the testing state

1.10.2 STATe:CHARage

STATe:CHARage is used to start the test.

Command Syntax 1	STATe:CHARage
Command Syntax 2	STARt // Command Syntax 2 function is equivalent to Command Syntax1
For Example	Send > STARt // Start the test, the stop state switches to charging or testing state, it only valid in [Test] page.

1.10.3 STATe:DISCHarge

STATe:DISCHarge is used to stop the test.

Command Syntax 1	STATe:DISCHarge
Command Syntax 2	STOP // Command Syntax 2 function is equivalent to Command Syntax 1
For Example	Send > STOP // Stop the test, the instrument will switch to the stop state and only valid in **[Test]** page.

1.11 TRIGger Subsystem

TRIGger subsystem is used to control trigger mode and perform bus trigger.

Table 1-7 TRIGer Subsystem Tree

TRIGger	:SOURce	{ INT,MAN,BUS,EXT}	Trigger mode
	:EDGE	{Rising,Falling}	Trigger edge
			When the comparator mode sets to **[Single]**, it will generate one time.

1.11.1 TRIGer:SOURce

TRIGger:SOURce is used to select trigger source.

Command Syntax	TRIGger:SOURce { INT,MAN,BUS,EXT}
Parameter	{ INT,MAN, BUS,EXT} INT: internal trigger MAN: manual trigger BUS: bus trigger EXT: external trigger
For Example	Send > TRIG:SOUR BUS // Select bus trigger
Query Syntax	TRIGger:SOURce?
Query Response	{ INT,MAN,BUS,EXT}

1.11.2 TRIGer:EDGE

TRIGger:EDGE is used to select the edge of external trigger.

Command Syntax	TRIGger:EDGE {Rising,Falling}
Parameter	Rising: rising edge trigger Falling: falling edge trigger
For Example	Send > TRIG:Edge Rising // Select rising edge trigger
Query Syntax	TRIGger:EDGE?
Query Response	{Rising,Falling}

1.11.3 TRIGer

TRIGger is used in single comparator mode, when the trigger source sets to BUS, it generates one time.

Command Syntax	TRIGger
For Example	Send > TRIG // Trigger one time

Notes:
1. It only valid in <Test> page.
2. Trigger mode must set to **[Bus]**.
3. In single comparator mode, the trigger can only be valid when the state is **[Testing]**.
4. In period comparator mode, use start command (STARt) to directly complete the test.
5. When **[Result Send]** sets to automatic in <System Configuration>, it will return the measured data after trigger one time.
6. When **[Result Send]** sets to FETCH in <System Configuration>, it will not return the measured data.
7. Return data format to see FETCH subsystem.

1.12 FETCH Subsystem

In <Test> page, send command FETCh? will return the current measured data.

Notes:
1. It only valid in <Test> page.
2. If user want to use this command, it should set **[Result Send]** to **[FETCH]** in **<System Configuration>**.
3. Space completion is added at the end to ensure that the return value is the same length.
4. In the stop state, it will always return the last measured data.

Query Syntax	FETCh?
Query Response	<SciFloat>,<SciFloat>,<SciFloat>,{OFF☐☐,PASS☐,UFAIL,LFAIL,OPEN☐}

	<SciFloat> represent float-point number, resistance value (the unit is Ω) <SciFloat> represent float-point number, current value (the unit is A) <SciFloat> represent float-point number, voltage value (the unit is V) {OFF□□,PASS□,UFAIL,LFAIL,OPEN□} , it represents comparative result OFF□□ represents the comparator is not open; PASS□ represents the result is qualified; UFAIL represents the result is over the upper limit; LFAIL represents the result is over the lower limit; OPEN□ represents open-circuit
For Example	Send > FETCh? Return > 9.9732e+07,1.0027e-06,□□99.9,OFF□□ Return > 9.9631e+07,5.0193e-06,□500.1,PASS□

1.13 CORRection Subsystem

CORRection subsystem is used to set calibration for the instrument. The test wire should be open-circuit befor calibration.

Command Syntax	CORRection
For Example	Send > CORR? Return > Open Clear Zero Starting... // During calibration, please wait... Return > PASS // Clear zero is completed
Limit	1. It can only used in the stop state. 2. The test wire should be open-circuit at first and do not contact to any objects, and then send CORRection command. 3. During the calibration, the command parser refuses to receive any commands.

1.14 FILE Subsystem

FILE subsystem is used to manage the file. It can save user-parameter to internal of the instrument or load the flash to the system.

Table 1-8 FILE Subsystem Tree

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

1.14.1 FILE:SAVE

FILE:SAVE can save the current settings to the specified file, the file number is 1~100.

Command Syntax	FILE:SAVE <integer (1~100)>
For Example	Send > FILE:SAVE 2 // Save to the 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 1

1.14.2 FILE:LOAD

FILE:LOAD can read the file data to the system, the file number is 1~100.

Command Syntax	FILE:LOAD <integer (1~100)>
For Example	Send > FILE:LOAD 1 // Read the file 1 data to the system

1.14.3 FILE:DELete

FILE:DELete can delete the specified file data, the file number is 1~100.

Command Syntax	FILE:DELete <integer (1~100)>
For Example	Send > FILE:DEL 5 // Delete file 5

1.14.4 SAV

SAV can save the settings to the current file.

Command Syntax	SAV
For Example	Send > SAV // Save to the current file

1.14.5 RCL

RCL can read the current file to the system.

Command Syntax	RCL
For Example	Send > RCL // Read the current file data to the system

1.15 *IDN? Subsystem

*IDN? subsystem is used to return the instrument's version number.

Query Syntax	*IDN?
Query Response	<Manufacturer>,<MODEL>,<SN>,<Revision>
For Example	Send > *IDN? Return > UNI-T,UT5583,CTLH322410001,REV A2.5

2. Modbus (RTU) Communication Protocol

This chapter contains the following contents.

- Data Format
- Function Code
- Register
- Read multiple registers
- Write multiple registers

2.1 Data Format

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

2.1.1 Instruction Frame

Figure 2-1 Modbus Instruction Frame

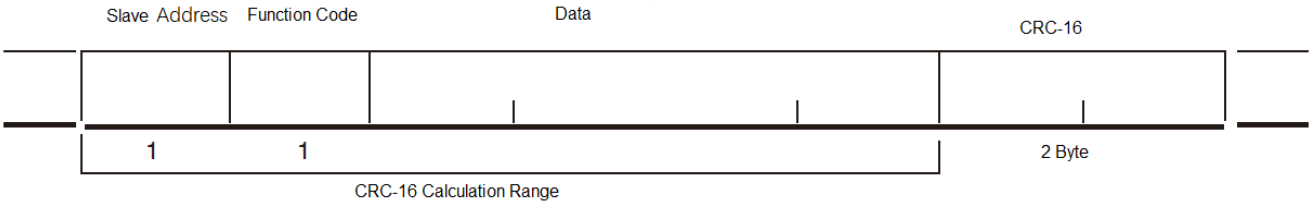


Table 2-1 Description of Instruction Frame

	It needs mute interval time of 3.5 character at least
Slave-station Address	1 byte Modbus supports 00~0x63 slave station 00 is assigned in unified broadcast If the instrument doesn't have optional RS485, the default slave station address is 0x01.

Function Code	1 byte 0x03: read multiple registers 0x04: =03H, not use 0x06: write a single register, which can replace by 10H, not use 0x10: write multiple registers
Data	The specified register address, quantity and content
CRC-16	2 bytes, LSB (least significant bit) Cyclic Redundancy Check Calculating all the data from slave station address to the last data, get CRC-16 check code

2.1.2 Response Frame

Except the instruction of 00H slave address boardcast, other slave station address will returns response frame.

Figure 2-2 Normal Response Frame

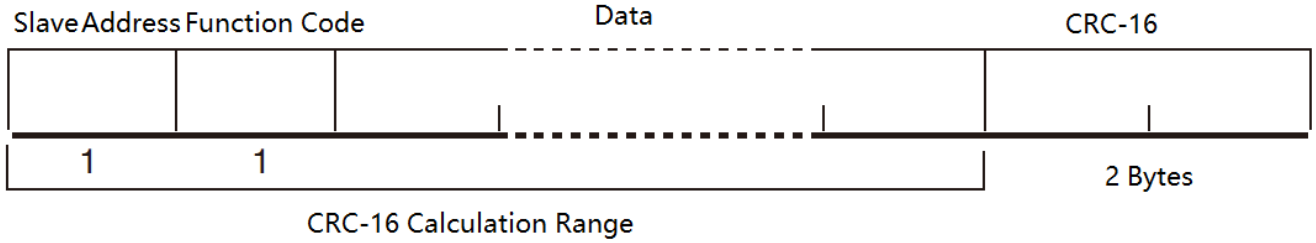


Figure 2-3 Normal Response Frame

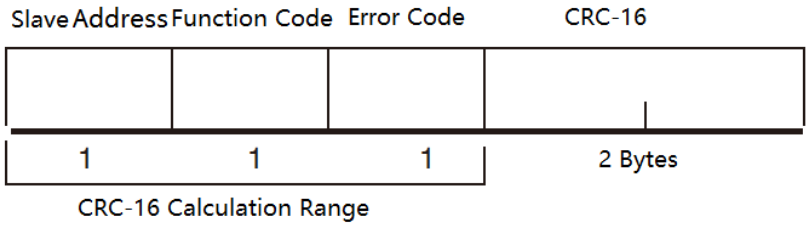


Table 2-2 Description of Abnormal Response Frame

Slave-station Address	1 byte Original returns slave station address
Function Code	1 byte Function code logical OR of instruction frame on BIT7 (0x80) For Example: 0x03 OR 0x80 = 0x83
Error 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	2 bytes, LSB (least significant bit) 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, total bit of function code 0x03 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 support	1
0x02	Register error	Register does not exist	2
0x03	Data error	Quantity of register or byte error	3
0x04	Execution error	Invalid data, write 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 multiple registers	Read multiple registers
0x10	Write multiple registers	Write multiple registers

2.3 Register

The register quantity of the instrument is 2-byte mode, it requires that it must wirte 2 bytes for each time, for example, speed register is 0x3002, data is 2 bytes, the numerical value must be written to 0x0001.

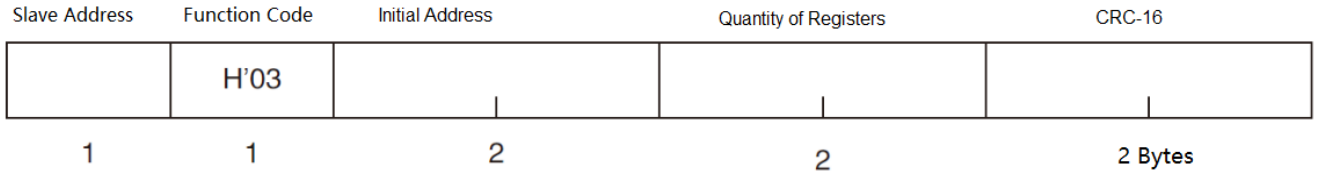
Data:

The instrument supports the following numerical value.

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

2.4 Read Multiple Registers

Figure 2-1 Read Multiple Regusters (0x03)



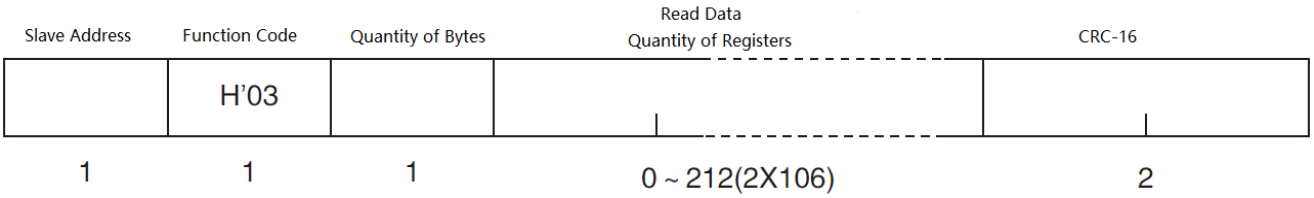
Function code of read multiple registers is 0x03.

Table 2-5 Read Multiple Regusters

Name	Name	Description
	Slave station address	If the RS485 address is not specified, the defaultis 01.

0x03	Function code	
	Initial address	The initial address of register refer to Modbus instruction set.
	Quantity of read mutiple registers 0001~006A (106)	Continuously read quantity of register refer to Modbus instruction set. To make sure all register address are exit, otherwise it returns error frame.
CRC-16	Check code	

Figure 2-2 Read Multiple Register (0x03) Response Frame



Name	Name	Description
	Slave station address	Original return
0x03 or 0x83	Function code	No abnormalities: 0x03 Error code: 0x83
	Byte number	= quantity of register x2 For Example, 1 register returns 02
	Data	Read data
CRC-16	Check code	

2.5 Write Multiple Registers

Figure 2-3 Write Multiple Registers (0x10)

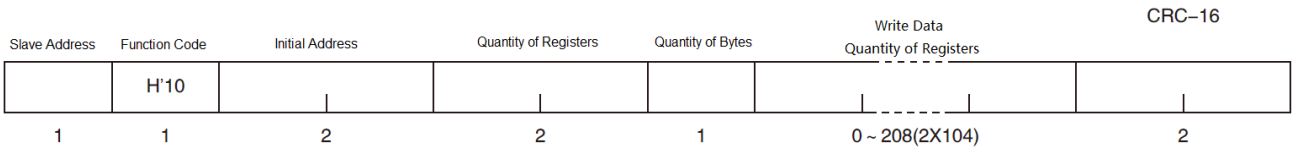
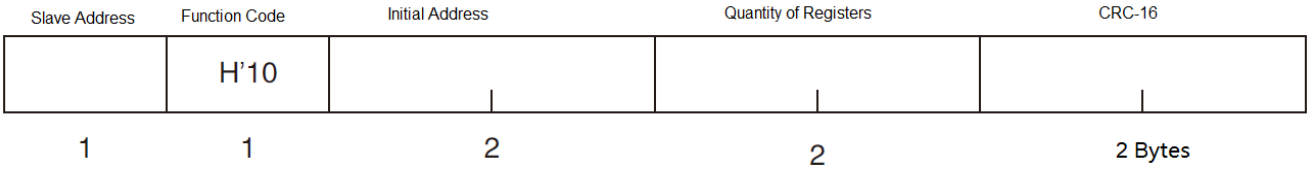


Table 2-1 Write Multiple Registers

Name	Name	Description
	Slave station address	If the RS485 address is not specified, the defaultis 01.
0x10	Function code	

	Initial address	The initial address of register refer to Modbus instruction set.
	Quantity of write multiple registers 0001~0068 (104)	Continuously read quantity of register refer to Modbus instruction set. To make sure all register address are exit, otherwise it returns error frame.
	Byte number	= quantity of register x2
CRC-16	Check code	

Figure 2-4 Write Multiple Registers (0x10) Response Frame



Name	Name	Description
	Slave station address	Original return
0x10 or 0x90	Function code	No abnormalities: 0x10 Error code: 0x90
	Initial address	
	Quantity of write multiple registers 0001~0068 (104)	
	CRC-16 Check code	

3. Modbus (RTU) Instruction Set

This chapter contains the following content.

- Register Address

3.1 Register Overview

All register address of the instrument refer to Table 3-1 register overview.

Notes:

1. Unless otherwise specified, numerical value of the instruction and response frame are all hexadecimal data.

2. Floating point number on-line conversion,
plese refer to website http://www.binaryconvert.com/convert_float.html

Table 3-1 Register Overview

Register		Byte Number	Name	Numerical Value	Description
Address	Quantity				
2000	2	4	Read the measured resistance value	4 bytes float-point number Byte sequence AABBCCDD	Read-only register, the data unit is Ω
2002	2	4	Read the measured current value	4 bytes float-point number Byte sequence AABBCCDD	Read-only register, the data unit is A
2004	2	4	Read the measured voltage value	4 bytes float-point number Byte sequence AABBCCDD	Read-only register, the data unit is V
2006	1	2	Read the comparator	2 bytes integer 0000: not compare 0001: qualified 0002: over the upper limit 0003: over the lower limit 0004: open-circuit	Read-only
2100	7	14	Trigger one time and read the measured results Resistance (4 bytes) Current (4 bytes) Voltage (4 bytes) Comparator (2 bytes)	Resistance: 4 bytes float-point number Current: 4 bytes float-point number Voltage: 4 bytes float-point number Comparator: 2 bytes integer	Read-only
2200	1	2	Range number	0001~0006	Read and write register, 2 bytes integer
2201	1	2	Range mode	0000: automatic range	Read and write register,

				0001: manual range 0002: nominal range	2 bytes integer
2202	1	2	Test speed	0000: slow speed 0001: middle speed 0002: fast speed	Read and write register, 2 bytes integer
2203	2	4	Voltage	4 bytes float-point number	Read and write register
2205	1	2	Display mode	0000: resistance 0001: dual display of resistance and current	Read and write register, 2 bytes integer
2206	1	2	Display digit	0000: 5 digits 0001: 4 digits	Read and write register, 2 bytes integer
2207	1	2	Contact inspection	0000: off 0001: on	Read and write register, 2 bytes integer
2208	1	2	Trigger mode	0000: internal 0001: manual 0002: bus 0003: external	Read and write register, 2 bytes integer
2209	1	2	Trigger edge	0000: rising edge 0001: falling edge	Read and write register, 2 bytes integer
2210	2	4	Charge time	4 bytes float-point number	Read and write register
2212	2	4	Test time	4 bytes float-point number	Read and write register
2214	2	4	Discharge time	4 bytes float-point number	Read and write register
2216	1	2	Trigger delay	4 bytes integer	Read and write register
2300	1	2	Comparator mode	0000: single 0001: period	Read and write register, 2 bytes integer
2301	1	2	Comparator state	0000: off 0001: on	Read and write register, 2 bytes integer
2302	1	2	Comparator beep	0000: off 0001: qualified 0002: disqualified	Read and write register, 2 bytes integer
2303	2	4	Upper limit of comparator	4 bytes float-point number	Read and write register
2305	2	4	Lower limit of comparator	4 bytes float-point number	Read and write register

2400	1	2	Save the settings to the current file	Fixed value 0001	Write-only register, 2 bytes integer
2401	1	2	Read the current file data	Fixed value 0001	Write-only register, 2 bytes integer
2402	1	2	Save the settings to the specified file	0001~0064	Write-only register, 2 bytes integer (corresponding to decimalism 1~100)
2403	1	2	Read the specified file data	0001~0064	Write-only register, 2 bytes integer
2500	1	2	System lanague	0000: English 0001: simplified-Chinese	Read and write register, 2 bytes integer
2501	1	2	Volume	0000: low 0001: middle 0002: high	Read and write register, 2 bytes integer
2502	1	2	Power frequency	0000: 50Hz 0001: 60Hz	Read and write register, 2 bytes integer
2600	1	2	Lock key	0000: unlock 0001: lock	Write-only register, 2 bytes integer
2602	1	2	Query the test state	0000: in the stop state 0001: in the charging state 0002: in the testing state 0003: in the discharging state	Read-only register, 2 bytes integer
2604	1	2	Test state setting	0000: stop 0002: start	Write-only register, 2 bytes integer
2606	1	2	Trigger one time (= Handler Trig signal)	Fixed value 0002	Write-only register, 2 bytes integer
2608	1	2	Perform a open-circuit zero clearing	Fixed value 0002	Write-only register, 2 bytes integer

3.2 Fetch Test Data

3.2.1 Fetch Measured Result of Resistance

Register 2000~2001 is used to fetch the measured data of resistance.

■ Read

1	2	3	4	5	6	7	8
01	03	2000		0002		CFCB	
Slave station	Read	Register		Quantity of register		Check code	

■ Response

1	2	3	4	5	6	7	8	9
01	03	04	4C	BE	B7	31	3A	A3
01	03	Byte	Single float-point number				CRC-16	

B4~B6 is single float-point number, and byte sequence is AA BB CC DD.

Measured data: 4C BE B7 31 converts to float-point number 4CBEB731(float-point number) = 9.9989896E7(decimalism), the unit is Ω.

3.2.2 Fetch Measured Result of Current

Register 2002~2003 is used to fetch the measured data of current.

■ Read

1	2	3	4	5	6	7	8
01	03	2002		0002		6E0B	
Slave station	Read	Register		Quantity of register		Check code	

■ Response

1	2	3	4	5	6	7	8	9
01	03	04	35	86	46	9E	A7	DE
01	03	Byte	Single float-point number				CRC-16	

B4~B6 is single float-point number, and byte sequence is AA BB CC DD.

Measured data: 35 86 46 9E converts to float-point number 3586469E (float-point number) = 1.00043E-6 (decimalism), the unit is A.

3.2.3 Fetch Measured Result of Voltage

Register 2004~2005 is used to fetch the measured data of voltage.

■ Read

1	2	3	4	5	6	7	8
01	03	2004		0002		8E0A	
Slave station	Read	Register		Quantity of register		Check code	

■ Response

1	2	3	4	5	6	7	8	9
01	03	04	42	C8	02	BB	2E	A6
01	03	Byte	Single float-point number				CRC-16	

B4~B6 is single float-point number, and byte sequence is AA BB CC DD.

Measured data: 42 C8 02 BB converts to float-point number 42C802BB (float-point number) = 100.005 (decimalism), the unit is V.

3.2.4 Fetch Comparator Result

Register 2006 is used to fetch the comparator result the comparator results. Returned 2 bytes integer represents the comparator results.

0000: not compare

0001: qualified

0002: over the upper limit

0003: over the lower limit

0004: open-circuit

■ Read

1	2	3	4	5	6	7	8
01	03	2006		0001		6FCB	
Slave station	Read	Register		Quantity of register		Check code	

■ Response

1	2	3	4	5	8	9
01	03	02	00	01	79	84
01	03	Byte	Integer		CRC-16	

3.2.5 Trigger One Time and Return Measured Result

■ Read

1	2	3	4	5	6	7	8
01	03	2100		0007		0E34	
Slave station	Read	Register		Quantity of register		Check code	

■ Response

1	2	3	4-7	8-11	12-15	16-17	18	19
01	03	04	4C BE AD 12	35 86 44 61	42 C8 03 0B	00 01	4A	74
01	03	Byte	Resistance (float-point number)	Current (float-point number)	Voltage (float-point number)	Comparator (Integer)	CRC-16	

Byte sequence of all measured data is AA BB CC DD.

B4-B7 is resistance value, 4CBEAD12 (float-point number) = 9.9969168E7 (decimalism), the unit is Ω.

B8-B11 is leakage value, 35864461 (float-point number) = 1.0003E-6 (decimalism), the unit is A.

B12-B15 is voltage value, 42C8030B (float-point number) = 100.005 (decimalism), the unit is V.

B16-B17 is comparator value, 0x0001 = qualified

1. This command is only valid in the following conditions.
   - ✓ <Test> display page
   - ✓ Trigger mode sets to **[Bus]**
   - ✓ Single comparator mode: Test state command **[2604]** can switch the instrument to **[Test]** state when the instrument is in the **[Test]** state.
   - ✓ Period comparator mode: Open the test time and the instrument stays in **[Stop]** state.
2. After the command is sent, data can only be returned when a measurement is completed, so the response will be delay.
   - ● Single comparator mode: Delay time = trigger delay + sampling time (relate to test speed)
   - ● Period comparator mode: Delay time = trigger delay + charge time + test time + discharge time

3.3 Parameter Setting

3.3.1 Test Range [2200]

■ Write

1	2	3	4	5	6	7	8	9	10	11
01	10	2200		0001		02	00	01	65	92
Station number	Write	Register		Quantity of register		Byte	Data		CRC	

B8-B9: range number 1~6

Write Return

1	2	3	4	5	6	7	8
01	10	2200		0001		0B	B1
Slave station	Write	Register		Quantity of register		CRC	

■ Read

1	2	3	4	5	6	7	8
01	03	2200		0001		8E72	
Slave station	Read	Register		Quantity of register		Check code	

Response

1	2	3	4	5	6	7
01	03	02	00	05	78	47
01	03	Byte	Integer		CRC-16	

B4-B5: 0005 range number is 5.

Notes:

- Instruction of range mode, test speed, display mode, display digit, contact inspection, trigger mode, trigger edge, comparator mode, comparator state, comparator beep, system language, volume and power frequency are all read and write register, 2 bytes integer.
  Usage is the same with the test range. The specified parameter refers to Register Overview.
- When comparator mode sets to **[Single]**, the instrument will automatically set **[Test Time]** to **[Continuous]**, don't need to set **[Test Time]** by the instruction.
- When comparator mode sets to **[Period]**, it need use the instruction to set the specified **[Test Time]**.

3.3.2 Test Voltage [2203]

Notes:

Minimum of test voltage: 1V; maximum of test voltage: 1000V

It returns error code 90 if the test voltage excceds the range.

■ Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	2203		0002		04	43 FA 00 00				06	AE
Station number	Write	Register		Quantity of register		Byte	Data				CRC	

B8-B11: test voltage, float-point number format

43FA0000 (float-point number) = 500.0 V (decimalism)

Write Return

1	2	3	4	5	6	7	8
01	10	2203		0002		BB	B0
Slave station	Write	Register		Quantity of register		CRC	

■ Read

1	2	3	4	5	6	7	8
01	03	2203		0002		3E73	
Slave station	Read	Register		Quantity of register		Check code	

Response

1	2	3	4	5	6	7	8	9
01	03	04	43	FA	00	00	CF	86
01	03	Byte	float-point number				CRC-16	

43FA0000 (float-point number) = 500.0 V (decimalism)

3.3.3 Charge Time [2210]

Notes:

- Minimum of charge time, test time and discharge time: 0.1s, maximum: 999.9s, if time sets to 0, which means the timer is off.
- It returns error code 90 if the test voltage exceeds the range.
- Instruction of test time and discharge time is the same as the charge time, the details refers to Register Overview.
- If test time sets to 0, which means the instrument will be continuous test when the instrument enters the test state.

■ Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	2210		0002		04	41 20 00 00				67	F4
Station number	Write	Register		Quantity of register		Byte	Data				CRC	

B8-B11: charge time, float-point number format

41200000 (float-point number) = 10.0s (decimalism)

Write Return

1	2	3	4	5	6	7	8
01	10	2210		0002		4A	75
Slave station	Write	Register		Quantity of register		CRC	

■ Read

1	2	3	4	5	6	7	8
01	03	2210		0002		CFB6	
Slave station	Read	Register		Quantity of register		Check code	

Response

1	2	3	4	5	6	7	8	9
01	03	04	41	20	00	00	EF	C5
01	03	Byte	float-point number				CRC-16	

41200000 (float-point number) = 10.0s (decimalism)

3.3.4 Trigger Delay [2216]

Note:

Minimum of trigger delay:1ms, maximum: 9999ms, if trigger delay sets to 0, which means the timer is off.

■ Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	2210		0002		04	00 00 00 64				F3	C3
Station number	Write	Register		Quantity of register		Byte	Data				CRC	

B8-B11: 0x00000064 = 100 ms (decimalism)

Write Return

1	2	3	4	5	6	7	8
01	10	2216		0002		AA	74
Slave station	Write	Register		Quantity of register		CRC	

■ Read

1	2	3	4	5	6	7	8
01	03	2216		0002		2FB7	
Slave station	Read	Register		Quantity of register		Check code	

Response

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	64	FB	D8
01	03	Byte	Integer				CRC-16	

0x00000064 = 100 ms (decimalism)

4.4 System Function

4.4.1 Query Test State [2602]

Register 2602 is used to fetch the test state of the instrument. Returned 2 bytes integer represents the result of test state.

0000: in the stop state

0001: in the charging state

0002: in the testing state

0003: in the discharge state

■ Read

1	2	3	4	5	6	7	8
01	03	2602		0001		2E82	
Slave station	Read	Register		Quantity of register		Check code	

■ Response

1	2	3	4	5	8	9
01	03	02	00	02	39	85
01	03	Byte	Integer		CRC-16	

0002 = in the testing state

4.4.2 Test State Setting [2604]

Write-only register

0000: when the test is stop, the instrument will switch to the stop state, it only valid in **[Test]** page.

0002: when the test is start, the instrument will switch from the stop state to the charge or test state, it only valid in **[Test]** page.

■ Write

1	2	3	4	5	6	7	8	9	10	11
01	10	2604		0001		02	00	02	61	D7
Station number	Write	Register		Quantity of register		Byte	Data		CRC	

Write Return

1	2	3	4	5	6	7	8
01	10	2604		0001		4B	40
Slave station	Write	Register		Quantity of register		CRC	

4.4.3 Trigger One Time and Not Return Test Result [2606]

Write-only register

It only valid in <Test Display> page, and trigger mode should be Bus.

- Write

1	2	3	4	5	6	7	8	9	10	11
01	10	2606		0001		02	00	02	60	35
Station number	Write	Register		Quantity of register		Byte	Data		CRC	

B8-B9: 0002 fixed value

Write Return

1	2	3	4	5	6	7	8
01	10	2606		0001		EA	80
Slave station	Write	Register		Quantity of register		CRC	

Notes:

- Instruction of file operation, key lock and perform a open-circuit zero clearing are all write-only register, 2 bytes integer. Usage is the same as Trigger One Time, the specified parameter refers to Register Overview.
- Instruction [2606] generates one time and not return the test results. It need to cooperate with other fetch result instruction to get the test results.