Instruments.uni-trend.com

Programming Manual

UT3510+ and UT3515-Sx Series Benchtop Micro Ohm Meter

V1.1

April 2025

Warranty and Statement

Copyright

Brand Mark 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

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that builds on existing standards such as IEEE 488.1 and IEEE 488.2. It adheres to the floating-point rules defined by the IEEE 754 standard, uses ISO 646 7-bit encoding notation (equivalent to ASCII programming), and incorporates many other standards.

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

1.1 Introduction

This chapter introduces the RS232C command of UT3510+ and UT3515-Sx series in detail. These commands are all compliant with the SCPI.

Each command description contains the following contents:

Command name: The name corresponding to a SCPI command.

Command Syntax: The command format, including all required and optional parameters.

Query Syntax: The query format, including all required and optional parameters.

Query Respond: The returned data format of UT3510+.

1.2 Notation Conventions and Definitions

Notations and definitions for RS232C commands.

< >: The content in angle brackets indicates the parameter of the command.

[ ]: The content in square brackets indicates an optional command that can be omitted.

{ }: Curly brackets containing several optional parameters indicate that only one parameter can be selected.

The following notations will be used in the command.

<NL>: **Line separator** (decimal 10).

Space: Single ASCII character (decimal 0-9, 11-32). For example, carriage return (decimal 13) or space (decimal 32).

1.3 Command Structure

UT3510+ series commands are divided into two types: common commands and SCPI commands.

Common command: Defined by IEEE (Institute of Electrical and Electronic Engineers), suitable for all instruments.

SCPI command: Use a tree-like structure with three levels. The highest level is called the subsystem command. Lower- level commands within a subsystem are only valid when the subsystem command is selected. A colon ":" is used to separate high-level command from the low-level command.

Basic Rules of Tree-like Structure Commands:

- Case insensitive

  For example, LIMIT:NOMINAL <value> = limit:nominal <value> = LiMiT:NoMiNaL <value>

- No spaces around colon (␣ indicates a space)

  For example, Incorrect: LIMIT␣:␣NOMINAL <value>

  Correct: LIMIT:NOMINAL <value>

- Abbreviated command or full spelled command

  For example, LIMIT:NOMINAL <value> = LIM:NOM <value>

- Add a question mark "?" after a command to form a query command.

  For example, LIMIT:NOMINAL_C ?

Multiple Commands in One Line:

- Use a semicolon ";" to separate multiple commands at the same level within one subsystem command.

  For example, LIMIT:NOMINAL <value>; BIN <n> <low limit>,<high limit>

- A colon ":" after a semicolon ";" indicating that the following command restarts from the top level of the command tree.

  For example, LIMIT:NOMINAL <value>;:LIMIT:BIN <n> <low limit>,<high limit>

1.4 Command Abbreviations

Every command and character parameter has at least two spelled forms: an abbreviated from and a full spelled from. In some cases, the two forms are totally different. Please follow these rules for abbreviation.

- If the word length is four letters or shorter, the abbreviated form and full spelled are the same.
- If the word length is longer than four letters and the fourth letter is a vowel, the abbreviated form is the first three letters.

  For example, LIMIT abbreviates to LIM.

  If the word length are longer than four letters and the fourth letter is a consonant, the abbreviated form is the first four letters.

  For example, RANGE abbreviates to RANG.

  FREQUENCY abbreviates to FREQ.

- If a phrase needs to be abbreviated, the full spelled form consists the capital letter of the first word and the entire last word. The abbreviated form is derived from the full spelled form.
  For example, a phrase "Source RESistor",
  Full spelled: SRESISTOR
  Abbreviated form: SRES

1.5 Header and Parameter

UT3510+ series control command consists of a command header and parameters. The header can be in full spelled or abbreviated form. The full spelled is used for easy understanding, while the abbreviated form is used to improve the input efficiency. The parameter can be of two types as follows.

- Character Data and String Data
  Character data consists of ASCII characters. The abbreviated form is the same as the abbreviated form of command header.
  String data consists of ASCII characters enclosed in double quotes ("").

- Numeric Data
  Numeric data can be integer (NR1), fixed point (NR1), or floating point (NR3). The range of numeric data is ±9.9E37.
  Examples for NR1:
  123
  +123
  -123
  Examples for NR2:
  12.3
  +1.234
  -123.4
  Examples for NR3:
  12.3E+5
  123.4E-56

2. SCPI Reference Command

2.1 DISPlay Command

DISPlay subsystem is used to switch between different display pages or to display a string of text on the page prompt bar.

DISPlay Subsystem Tree

DISPlay	:PAGE	{TEST,SETUP(MSET),COMParator,FILE,SYSTem,SYSTEMINFO(SINF)}
	:LINE	<string>

DISPlay:PAGE command is used to set the display mode.

DISPlay:PAGE? command is used to query and return the display mode of measured results.

Command Syntax	DISPlay:PAGE {TEST,SETUP(MSET),COMParator,CORRection,FILE,SYStem,SYSTEMINFO(SINF)}
Parameter	TEST: Test page SETUP(MSET): Setup page COMParator: Comparator page FILE: File management page SYSTem: System configuration page SYSTEMINFO(SINF): System information page
For Example	DISPlay:PAGE TEST<NL> //Switch to the test page
Query Syntax	DISPlay:PAGE?
Query Respond	{test,comp,syst,file,sinf,mset}<NL>

2.2 FUNCtion Subsystem Command

Parameters set using the FUNCtion subsystem will not be saved by the instrument and will need to be reset the next time the instrument is switched on. Additionally, the parameter settings should be consistent with the current test mode.

FUNCtion Subsystem Tree

FUNCtion	:RANG	{Range number,max,min}
	:RATE	{SLOW,MEDium,FAST,HIGH }
	:MODE	{AUTO,HOLD,NOMinal}

	:IMP	{R,RT,T,LPR,LPRT}		
	:LPR	:RANG		{Range number,max,min}
	:LPR	:RANG	:MODE	{AUTO,HOLD,NOMinal}

FUNCtion:RANGe Command

FUNC:RANG command is used to set the range mode and range number.

Command Syntax	FUNCtion:RANGe {<Range number>,min,max}
Parameter	<Range number>: 0-8 (UT3516+) or 0-6 (UT3513+) min: Minimum range max: Maximum range
For Example	FUNCtion:RANGe 5<NL> // Switch to range 5 (2 k)
Query Syntax	FUNCtion:RANG?
Query Respond	{<Rnge number>: 0-8 (UT3516+) or 0-6 (UT3513+)}<NL>

FUNCtion:RANGe:MODE Command

FUNCtion:RANGe:MODE command is used to switch the range mode.

FUNCtion:RANGe:MODE? command is used to query and return the range state.

Command Syntax	FUNCtion:RANGe:MODE{AUTO,HOLD(MANual),NOMinal}
Parameter	AUTO: Automatic range HOLD (MANual): Manual range NOMinal: Nominal range
For Example	FUNCtion:RANGe:MODE AUTO<NL> // Switch to the automatic range
Query Syntax	FUNCtion:RANGe:MODE?
Query Respond	{AUTO,HOLD,NOMinal}<NL>

FUNCtion:RATE Command

FUNCtion:RATE or FUNC:SPEED command are used to set the test speed. FUNCtion:RATE? command is used to query and return the test speed.	
Command Syntax	FUNCtion:RATEE{SLOW,MEDium,FAST,HIGH}
Parameter	SLOW: Slow speed
	MEDium: Medium speed

	FAST: Fast speed
	HIGH: High speed
For Example	FUNCtion:RATE SLOW<NL> // Switch to slow speed
Query Syntax	FUNCtion:RATE?
Query Respond	{SLOW,MEDium,FAST,HIGH}<NL>

FUNCtion:IMP Command

FUNCtion:IMP command is used to set the test mode.

FUNCtion:IMP? command is used to query and return the test mode.

Command Syntax	FUNCtion:IMP{R,RT,T,LPR,LPRT}
Parameter	R: R mode RT: RT mode T: T mode LPR: LPR mode LPRT : LPRT mode
For Example	FUNCtion:IMP T<NL> // Switch to the T mode
Query Syntax	FUNCtion:IMP?
Query Respond	{R,RT,T,LPR,LPRT}<NL>

FUNCtion:LPR:RANGe Command

FUNCtion:LPR:RANGe command is used to set the range number for LPR mode. FUNCtion:LPR:RANGe? command is used to query and return the range number of LPR mode.

Command Syntax	FUNCtion:LPR:RANGe{<Range number>,min,max}
Parameter	<Range number>: 0-3 min: Minimum range max: Maximum range
For Example	FUNCtion:LPR:RANGe 1<NL> // Switch to the range 1 of LPR mode
Query Syntax	FUNCtion:LPR:RANGe?
Query Respond	{0,1,2,3}<NL>

FUNCtion:LPR:RANGe:MODE Command

FUNC:LPR:RANG:MODE command is used to set the range mode for LPR mode.

FUNC:LPR:RANG:MODE? command is used to query and return the range mode of LPR mode.

Command Syntax	FUNCtion:LPR:RANGe:MODE{AUTO,HOLD(MANual),NOMinal}

Parameter	AUTO: Automatic range HOLD (MANual): Manual range NOMinal: Nominal range
For Example	FUNCtion:LPR:RANGe:MODE NOMinal<NL> // Switch to the range mode of LPR mode to nominal range
Query Syntax	FUNCtion:LPR:RANGe:MODE?
Query Respond	{AUTO,HOLD,NOM}<NL>

FUNCtion:SCAN Command

FUNC:SCAN command is used to set the mode to either scan mode or single-road mode in multi-channel channels.

Command Syntax	FUNCtion:SCAN {0,<n>}
Parameter	ON: Enable the scan mode 0: Enable the scan mode <n>: 1-30 (UT3515-S30), 1-20 (UT3515-S20), or 1-10 (UT3515-S10)
For Example	FUNCtion:SCAN 5<NL> // Enable the single-road for 5CH FUNCtion:SCAN ON<NL> // Scan mode
Query Syntax	FUNCtion:SCAN?
Query Respond	SCAN<NL>, SINGLE<NL>

FUNCtion:CH<n> Command

FUNCtion:CH<n> command is used to set the current channel switch.
FUNCtion:CH<n>? command is used to query and return the switch state of the current channel.

Command Syntax	FUNCtion:CH<n> <state>
Parameter	<n> indicates the channel number to be set. <state> indicates the channel switch state, which can be either OPEN or CLOSE.
For Example	FUNCtion:CH1,CLOSE // Set the CH1 switch state to close
Query Syntax	FUNCtion:CH? <1-30>
Query Respond	<CHn>,<{OPEN,CLOSE}><NL>

FUNCtion:CH:MULTI <START_CH>,<STOP_CH>,<OPEN,CLOSE> Command

FUNCtion:CH:MULTI <START_CH>,<STOP_CH>,<OPEN,CLOSE> command is used to set the switch state for channels from START_CH to STOP_CH, including both the start and stop channels.

Command Syntax	FUNCtion:CH:MULTI <START_CH>,<STOP_CH>,<OPEN,CLOSE>
Parameter	<START_CH> indicates the starting channel number to be set. <STOP_CH> indicates the stopping channel number to be set. <OPEN,CLOSE> indicates the switch state, which can be either OPEN or CLOSE.
For Example	FUNCtion:CH:MULTI 1,30,OPEN,CLOSE<NL> // Set the switch state of CH1 to CH30 to CLOSE, including both CH1 and CH30.

2.3 COMParator Subsystem Command

COMParator subsystem command is used to set the comparator parameters.

COMParator Subsystem Tree

COMParator	:STATe	{OFF,1-BIN,2-BIN,3-BIN,4-BIN,5-BIN,6-BIN}
	:BEEP	{OFF,PASS(OK),FAIL(NG)}
	:MODE	{ABS,PER,SEQ}
	:NOMinal	<float>
	:BIN	<1-6>,<float lower>,<float upper>

COMParator:STATe Command

COMParator:STATe command is used to disable the comparator or set the BIN.

COMParator:STATe? command is used to query and return the state of the comparator and BIN.

Command Syntax	COMParator:STATe{0,1,2,3,4,5,6}
Parameter	0: Disable the comparator 1: Enable the comparator and set BIN1 for sorting. 2: Enable the comparator and set BIN2 for sorting. 3: Enable the comparator and set BIN3 for sorting. 4: Enable the comparator and set BIN4 for sorting. 5: Enable the comparator and set BIN5 for sorting. 6: Enable the comparator and set BIN6 for sorting.
For Example	COMParator:STATe 1<NL> // Switch to BIN1
Query Syntax	COMParator:STATe?
Query Respond	{0,1,2,3,4,5,6}<NL>

COMParator:BEEP Command

COMParator:BEEP command is used to set the beep function.

COMParator:BEEP? command is used to query and return the state of the beep function.

Command Syntax	COMParator:BEEP {OFF,OK,NG}
Parameter	OFF: Disable the beep function OK: The beep will alert when the test result is qualified. NG: The beep will alert when the test result is unqualified.
For Example	COMParator:BEEP OFF<NL> // Disable the beep function
Query Syntax	COMParator:BEEP?
Query Respond	{OFF,OK,NG}<NL>

COMParator:MODE Command

COMParator:MODE command is used to set the comparison mode.

COMParator:MODE? command is used to query and return the comparison mode.

Command Syntax	COMParator:MODE {ABS,PER,SEQ}
Parameter	ABS: Absolute deviation mode PER: Percentage deviation mode SEQ: Sequence mode
For Example	COMParator:MODE ABS<NL> // Set the comparison mode to ABS.
Query Syntax	COMParator:MODE?
Query Respond	{ABS,PER,SEQ}<NL>

COMParator:NOMinal Command

COMParator:NOMinal command is used to set the nominal value. The comparator uses the nominal to calculate the absolute deviation and percentage deviation.

COMParator:NOMinal? command is used to query and return the nominal value.

Command Syntax	COMParator:NOMinal <float>
Parameter	<float> represents the nominal value in NR1, NR2, or NR3 formats.
For Example	COMParator:NOMinal 1.2000<NL> // Set the nominal value to 1.2 Ω COMParator:NOMinal 1E3<NL> // Set the nominal value to 1 k COMParator:NOMinal 1000<NL> // Set the nominal value to 1 k
Query Syntax	COMParator:NOMinal?
Query Respond	<NR3><NL>

COMParator:BIN<n> Command

COMParator:BIN<n> command is used to set the upper and lower limit for the BIN<n>.

COMParator:BIN<n>? command is used to query and return the upper and lower limit of the current BIN.

Command Syntax	COMParator:BIN<n> <low limit>,<high limit>
Parameter	<n> represents 1 to 6 (NR1), BIN number. <low limit> represents the nominal value in NR1, NR2, or NR3 formats. <high limit> represents the nominal value in NR1, NR2, or NR3 formats.
For Example	COMP:BIN 1,-10,+10<NL> // In percentage sorting mode: the lower limit of BIN1 is -10%， the upper limit of BIN1 is 10%.
Query Syntax	COMParator:BIN? <1-6>
Query Respond	<NR3>,<NR3><NL>

2.4 TRIGger Subsystem Command

TRIGger command is used to set the trigger source and generate one trigger.

TRIGger Subsystem Tree

TRIGger	:IMMediate	
	:SOURce	{INT,EXT}
	:DELAy	<float>
TRG		

TRIGger:IMMediate Command

TRIGger:IMMediate command is used to set the trigger source to EXT. It will generate one trigger and return the trigger test results.

Command Syntax	TRIGger:IMMediate
For Example	TRIGger:IMMediate<NL> // The instrument test once and returns the trigger test data.
Query Respond	<NR3>,BIN<n><NL>

TRIGger:SOURce Command

TRIGger:SOURce command is used to set the trigger source.

TRIGger:SOURce? command is used to query and return the trigger source.

Command Syntax	TRIGger:SOURce{INT,EXT}
Parameter	INT: Internal trigger EXT: External trigger
For Example	TRIGger:SOURce INT<NL> // Set the trigger source to internal trigger
Query Syntax	TRIGger:SOURce?
Query Respond	{INT,EXT}<NL>

TRIGger:DELAy Command

TRIGger:DELAy command is used to set the trigger delay.

TRIGger:DELAy? command is used to query and return the trigger delay.

Command Syntax	TRIGger:DELAy{0,<float>}
Parameter	0: 0s <float>: 0.1-10.0s
For Example	TRIG:DELA 0.1<NL> // Set the trigger delay to 0.1s.

Query Syntax	TRIG:DELA?
Query Respond	{0,<float>}<NL>

TRG Command

When the trigger source is EXT, the instrument will generate one trigger and return the trigger test data.

Command Syntax	TRG
For Example	TRG<NL> // The instrument will generate one trigger and return the trigger test data.
Query Respond	<NR3>,BIN<n><NL>

2.5 FETCh? Query Command

FETCh? command is used to query and return the latest test results.

Query Syntax	FETCh?
Query Respond	<scifloat>,{BIN0-BIN3}<NL>
For Example	FETCh?<NL> // Query the latest test results.
Query Respond	<NR3>,BIN<n><NL>

2.6 SYSTem Subsystem Command

SYSTem subsystem command is used to set system-related parameters. Parameters set using the SYSTem subsystem command will not be saved by the instrument.

SYSTem Subsytem

SYSTem	:LANGuage	{ENGLISH,CHINESE,EN,CN}
	:BEEP	{ON(1),OFF(0)}
	:SETZero	{ON(1),OFF(0)}
	:RESET	{ON(1)}
	:MODE	{SCAN(1),SINGLE(0)}
	:UPLOAD	{FETCH(0),AUTO(1),AUTOCH(2)}

SYSTem:LANGuage Command

SYSTem:LANGuage command is used to set the system language. SYSTem:LANGuage? command is used to query and return the system language.

Command Syntax	SYSTem:LANGuage {ENGLISH,CHINESE,EN,CN}
Parameter	ENGLISH: Set the system language to English CHINESE: Set the system language to simplified Chinese EN: Set the system language to English CN: Set the system language to simplified Chinese
For Example	SYSTem:LANGuage ENGLISH<NL> // Set the system language to English
Query Syntax	SYST:LANG?
Query Respond	{ENGLISH,CHINESE}<NL>

SYSTem:BEEPer Command

SYSTem:BEEPer command is used to set the key sound.

SYSTem:BEEPer? command is used to query and return the state of the key sound.

Command Syntax	SYSTem:BEEPer {OFF,ON,0,1}
Parameter	OFF （0） : Disable the key sound ON （1） : Enable the key sound
For Example	SYSTem:BEEPer OFF<NL> // Disable the key sound
Query Syntax	SYST:BEEP?
Query Respond	{OFF,ON}<NL>

SYSTem:SETZero Command

SYSTem:SETZero command is used to set 0 ADJ (zero clearing function).

SYSTem:SETZero? command is used to query and return the state of 0 ADJ (zero clearing function).

Command Syntax	SYSTem:SETZero {OFF,ON,0,1}
Parameter	OFF （0） : Disable 0 ADJ (zero clearing function) ON （1） : Enable 0 ADJ (zero clearing function)
For Example	SYSTem:SETZero OFF<NL> // Disable 0 ADJ (zero clearing function)
Query Syntax	SYST: SETZ?
Query Respond	{OFF,ON}<NL>

SYSTem:RESET Command

SYSTem:RESET command is used to restore to the factory settings.

Command Syntax	SYSTem:RESET { ON,1}
Parameter	ON （1） : Start to restore to the factory settings
For Example	SYSTem:RESET ON<NL> // Set the instrument to restore to the factory settings

SYSTem:MODE Command

SYSTem:MODE command is used to set the system mode either scan mode or single mode.

SYSTem:MODE? command is used to query and return the system mode.

Command Syntax	SYSTem:MODE {SINGLE,SCAN,0,1}
Parameter	SINGLE (0) indicates that the system is set to single mode. SCAN (1) indicates that the system is set to scan mode.
For Example	SYSTem:MODE SINGLE<NL> // Set the system mode to single mode.
Query Syntax	SYST:MODE?
Query Respond	{SINGLE,SCAN}<NL>

SYSTem:UPLOAD Command

SYSTem:UPLOAD command is used to set upload mode.

SYSTem:UPLOAD? command is used to query and return the upload mode.

Command Syntax	SYSTem:UPLOAD{FETCH,AUTO,AUTOCH,0,1,2}
Parameter	FETCH (0) : Indicates querying and uploading using the FETCH command. AUTO (1): Sets the mode to automatic upload. In multi-channel scanning mode, data will be uploaded all at once after a full polling cycle for all channels is completed. AUTOCH (2): In multi-channel mode, data will be uploaded after each test is completed. Return example: "CH1, +4.5680e-03, PASS".
For Example	SYSTem:MODE SINGLE<NL> // Set the system mode to single mode.
Query Syntax	SYST:UPLOAD?
Query Respond	{FETCH,AUTO,AUTOCH}<NL>

2.7 CORRect Subsystem

CORR subsystem is used to complete a short-circuit correction.

CORRect Subsystem

CORRect	:SHORt	
	:SHOR:SKIP	
	:SHOR:START	
	:SHOR:CLOSE	
	:SHOR:SELECT	<START CH>,<STOP CH>

CORRect:SHORt Command

CORR:SHOR command is used to complete a short-circuit correction. The test terminal should be short-circuited before sending this command.

Command Syntax	CORRect:SHORt
For Example	CORRect:SHORt<NL> // Short-circuit correction
Query Respond	Please Open The Set-Zero First<NL> // This prompt indicates that short-circuit zeroing requires the 0AJD mode to be enabled first. Tese Mode Error<NL> // This error message indicates that the current test mode is incorrect. Only R mode and LPR mode support short-circuit zeroing. Clear Zero Start<NL> // This prompt confirms that short-circuit zeroing has started.
	{PASS，FAIL}<NL> //This prompt indicates the result of the short-circuit zeroing process. If the zeroing is successful, it will display "PASS"; if it fails, it will display "FAIL".

Multi-channel Mode

Command Syntax	CORRect:SHORt
For Example	CORRect:SHORt<NL> // Start short-circuit correction
Query Respond	Please Open 0ADJ First<NL> // This prompt indicates that short-circuit zeroing requires the 0AJD mode to be enabled first. Clearing zero has already started<NL> // This prompt confirms that short-circuit zeroing has started. Clear Zero Start.Now it's CH01.Please choose an action<NL> // This prompt indicates that short-circuit zeroing has started and displays the current channel (CH01). Please proceed to the next command.

CORRect:SHORt:SKIP Command

CORR:SHOR:SKIP command is used to skip the current channel during the multi-channel short-circuit zeroing process.

Multi-channel Mode

Command Syntax	CORRect:SHORt:SKIP
For Example	CORRect:SHORt:SKIP<NL> // This prompt indicates that the current channel is skipped

	during the short-circuit zeroing process.
Query Respond	Please Open The Set-Zero First<NL> // This prompt indicates that short-circuit zeroing process has not been enabled; the CORRect:SHORt command should be sent first. Set-Zero Already Closed<NL> // This prompt indicates that the short-circuit zeroing process has ended. Skip OK.Now it's CH02.Please choose an action<NL> // This prompt indicates that skipping the current channel was successful and displays the current channel. Please proceed to the next command.

CORRect:SHORt:START Command

CORR:SHOR:START command is used to initiate short-circuit zeroing for the current channel during the multi-channel zero-clearing process.

Multi-channel Mode

Command Syntax	CORRect:SHORt:START
For Example	CORRect:SHORt:START<NL> // This prompt indicates that short-circuit zeroing process for the current channel has started.
Query Respond	Please Open The Set-Zero First<NL> // This prompt indicates that short-circuit zeroing process has not been enabled; the CORRect:SHORt command should be sent first. Set-Zero Failed.Set-Zero Already Closed.<NL> // This prompt indicates that short-circuit zeroing process has failed and that short-circuit zeroing is disabled. Now it's CH%02d.Please choose an action<NL> // This prompt indicates that short-circuit zeroing process for the current channel was successful and displays the current channel. Please proceed to the next command.

CORRect:SHORt:CLOSE Command

CORR:SHOR:CLOSE command is used to disable short-circuit zeroing for multi-channel.

Multi-channel Mode

Command Syntax	CORRect:SHORt:CLOSE
For Example	CORRect:SHORt:CLOSE<NL> // Disable short-circuit correction
Query Respond	Please Open The Set-Zero First<NL> // This prompt indicates that short-circuit zeroing process has not been enabled; the CORRect:SHORt command should be sent first. Set-Zero Already Closed.<NL> // This prompt indicates that the short-circuit zeroing process has ended.

CORRect:SHORt:SELECT <START CH>,<STOP CH> Command

CORRect:SHORt:SELECT command is used to set short-circuit zeroing process for channels from START CH to STOP CH, including both the start and stop channels.

Command Syntax	CORRect:SHORt:SELECT <START CH>,<STOP CH>
Parameter	<START CH> indicates the starting channel for short-circuit zeroing, and the current channel also participates in the short-circuit zeroing process. <STOP CH> indicates the stopping channel for short-circuit zeroing, and the current channel also participates in the short-circuit zeroing process.
For Example	CORRect:SHORt:SELECT 1,10<NL> // Set short-circuit zeroing for multi-channel from CH1 to CH10 to CLOSE, including both CH1 and CH10.

Query Respond	Please Open OADJ First<NL> // This prompt indicates that short-circuit zeroing requires the OAJD mode to be enabled first. Clear Zero Start<NL> // This prompt confirms that short-circuit zeroing has started. Set-Zero in channel2 failed.Set-Zero Already Closed.<NL> // This prompt indicates that short-circuit zeroing process for CH2 has failed and that short-circuit zeroing for multi-channel is disabled. Clear OK.Set-Zero Already Closed.<NL> // This prompt confirms that short-circuit zeroing was successful and disables short-circuit zeroing for multi-channel.

2.8 *IDN? Query Command

FETCh? command is use query and return the instrument's version number.

Query Syntax	IDN?
Query Respond	< Manufacturer>,<MODEL>,<Revision><NL>
For Example	*IDN?<NL> // Query the latest test result.
Query Respond	UNI-T,UT3516+,CRM1224170004,REV V3.37<NL>

2.9 ERRor? Query Command

ERRor? command is use query and return the latest error message.

Query Syntax	ERRor?
Query Respond	Error string
For Example	ERR?<NL> // Query the latest error message.
Query Respond	No error.<NL><NL>

Error codes are shown in the following table.

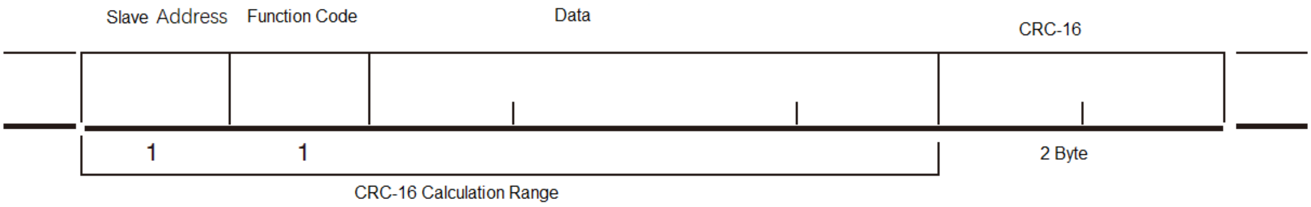
Error Code	Description
*E00	No error
*E01	Bad command
*E02	Parameter error
*E03	Missing parameter
*E04	buffer overrun
*E05	Syntax error
*E06	Invalid separator
*E07	Invalid multiplier
*E08	Numeric data error
*E09	Value too long
*E10	Invalid command
*E11	Unknown error

3. Modbus (RTU) Communication Protocol

3.1 Data Format

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

Command Frame



Command Frame Description

	A mute interval time of at least 3.5 characters is required.
Slave-station Address	1 byte Modbus supports slave station address from 00 to 0x63. Address 00 is specified for uniform broadcasting.
Function Code	1 byte 0x03: read multiple registers 0x04: =03H，not use 0x06: write a single register, which can be replaced by 10H 0x08: echo test (only for debugging) 0x10: write multiple registers
Data	The specified register address, quantity, and content
CRC-16	2 bytes, LSB (Least significant bit) Cyclic Redundancy Check Calculate the CRC-16 check code for all data from the slave station address to the last data.

CRC-16 Calculation Method

1. Set the initial value of the CRC-16 register to 0xFFFF.
2. Perform an XOR operation on the CRC-16 register and the first byte of the message, and return the result to the CRC register.
3. Fill the MSB with zero and shift the CRC register to the right by 1 bit.
4. If the bit shifted from the LSB is "0", repeat step 3 (process the next shift bit). If the bit shifted from the LSB

is "1", perform an XOR operation on the CRC register and 0xA001, and return the result to the CRC register.

5. Repeat steps 3 and 4 until 8 bits have been processed.
6. If the information processing is not finished yet, perform an XOR operation on the CRC register and the next byte of the message, and return the result to the CRC register. Then repeat from step 3.
7. The result of the calculation (the value of the CRC register) is appended to the information from the lower byte.

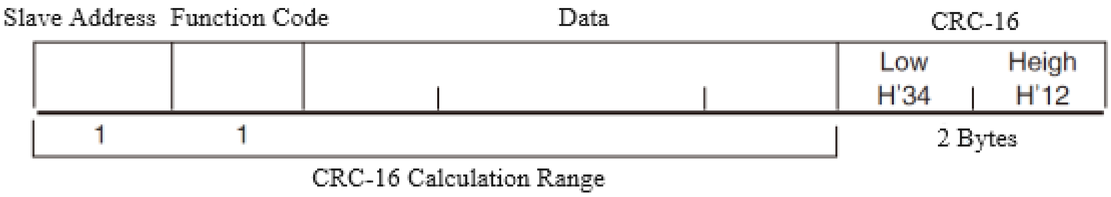
The following is a CRC calculation function in VB language.

```
      Function CRC16(data() As Byte) As Byte()

           im CRC16Lo As Byte, CRC16Hi As Byte            'CRC register

           im CL As Byte, CH As Byte                      'Polynomial code

           im SaveHi As Byte, SaveLo As Byte

            im i As Integer

            im flag As Integer

            RC16Lo = &HFF

            RC16Hi = &HFF

            L = &H1

            H = &HA0

            or i = 0 To UBound(data)

               CRC16Lo = CRC16Lo Xor data(i)       'Each data byte is XORed with the CRC.

                For flag = 0 To 7

                SaveHi = CRC16Hi

                SaveLo = CRC16Lo

                CRC16Hi = CRC16Hi \ 2                      'Shift the high bit to the right by one bit.

                CRC16Lo = CRC16Lo \ 2                       'Shift the low bit to the right by one bit.

                If ((SaveHi And &H1) = &H1) Then            'If the last bit of high byte is 1,

                    CRC16Lo = CRC16Lo Or &H80                'then shift the low bit to the right and fill with 1,

                End If                                      ' then shift the low bit to the right and fill with 1.

                If ((SaveLo And &H1) = &H1) Then             'If LSB is 1， perform XOR with the polynomial code.


              CRC16Hi = CRC16Hi Xor CH

              CRC16Lo = CRC16Lo Xor CL

           End If
```

```
 Next flag
ext i
im ReturnData(1) As Byte
eturnData(0) = CRC16Hi                      'CRC MSB
eturnData(1) = CRC16Lo                      'CRC LSB
RC16 = ReturnData
End Function
```

Calculated CRC-16 data should be appended to the end of the command frame.

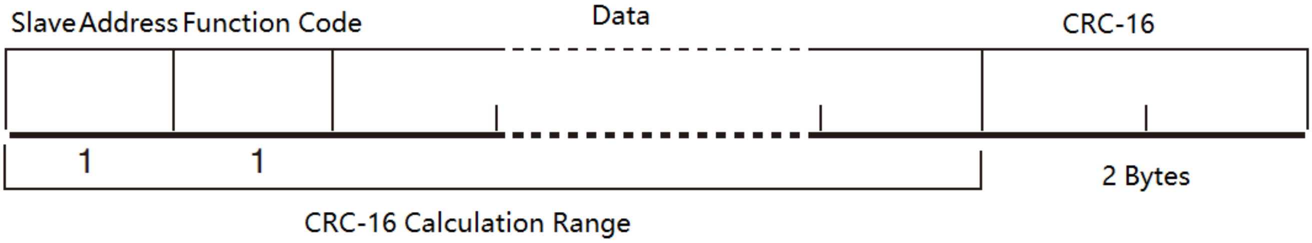
For example, 1234H.



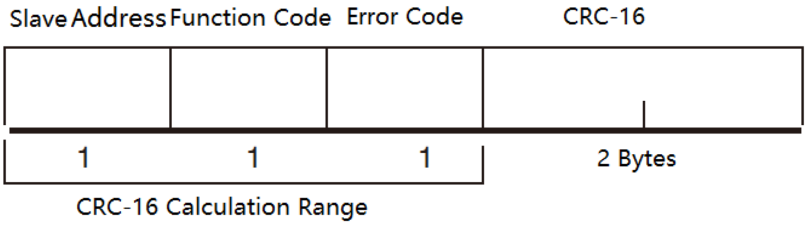
Response Frame

All other slave address instruments return a response frame, unless the command is broadcast from the 00H slave address.

Normal Response Frame



Exceptional Response Frame



Exceptional Response Frame Description

Slave-station Address	1 byte Original returned slave-station address
Function Code	1 byte The function code is a logical OR with BIT7 (0x80) of the command frame For example: 0x03 OR 0x80 = 0x83
Error Code	Exceptional 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 Calculate the CRC-16 check code for all data from the slave station address to the last byte of data.

No Response

The instrument does not handle or respond to the following cases, which may result in a communication time-out:

1. Slave station address error
2. Transmission error
3. CRC-16 error
4. Bit error.

   For example, the total bit count of function code 0x03 must be 8, and if the received bit count is less than or greater than 8 bits, it will cause an error.
5. The slave station address 0x00 represents a broadcast address, and the instrument will not respond.

Error Code Description

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	Error in the number of registers or bytes.	3
0x04	Execution error	Invalid data: written data is not within the allowed range.	4

3.2 Function Code

The instrument only supports certain function codes. Other function codes will result in an error response frame.

Function Code	Name	Description
0x03	Read multiple registers	Read data of multiple consecutive register
0x04	Same with 0x03	Replace by 0x03
0x08	Echo test	Original returned received data
0x10	Write multiple registers	Write multiple consecutive register

3.3 Register

The register quantity of the instrument is in 2-byte mode, requiring that 2 bytes be written each time. For example, the speed register is 0x3002, data is 2 bytes, and the numerical value must be written as 0x0001.

Data:

The instrument supports the following numerical values:

1. 1 register, double byte integer (16 bits)

   For example, 0x64 → 00 64

2. 2 registers, four bytes integer (32 bits)

   For example, 0x12345678 → 12 34 56 78

3. 2 registers, four bytes single float-point number (32 bits)

   For example, 3.14 → 40 48 F5 C3

3.4 Read Multiple Registers

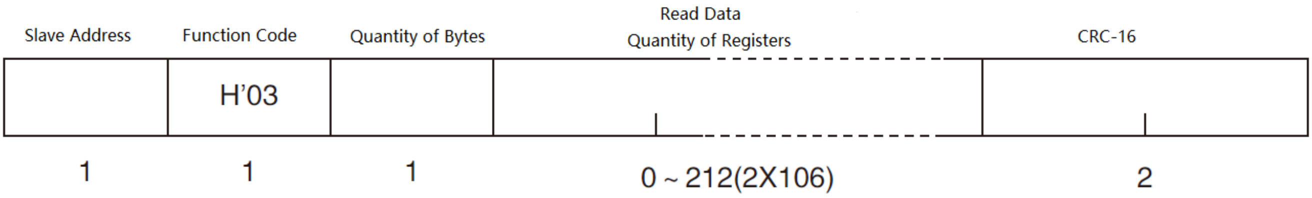
Read Multiple Registers (0x03)

Slave Address	Function Code	Initial Address	Quantity of Registers	CRC-16
	H’03			
1	1	2	2	2 Bytes

Name	Name	Description
	Slave station address	If no address is specified, the default address is 01.
0x03	Function code	
	Initial address	The initial address of the register refers to Modbus instruction set.

	Quantity of registers	The number of registers to read consecutively. Please refer to the Modbus command set to ensure that all these register addresses are exist; otherwise, an error frame will be returned.
CRC-16	Check code	

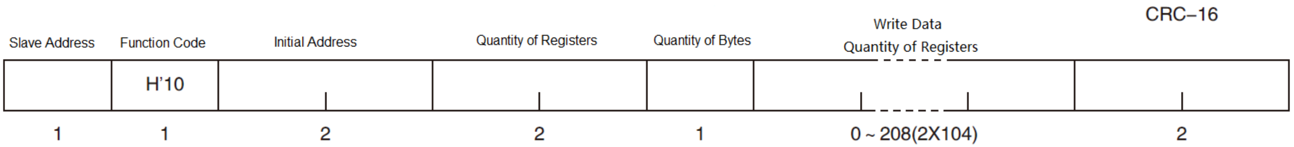
Read Multiple Registers （0x03） Response Frame



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

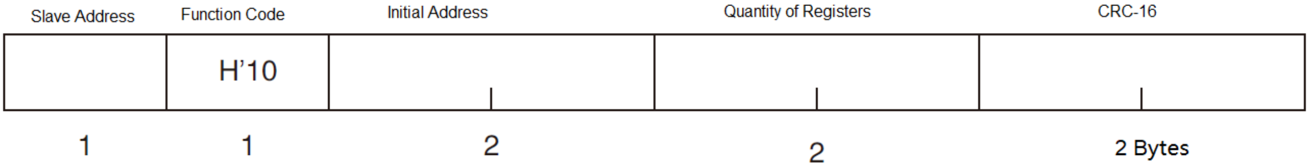
3.5 Write Multiple Registers

Write Multiple Registers (0x10)



Name	Name	Description
	Slave station address	The default address is 01
0x10	Function code	
	Initial address	The initial address of the register refers to Modbus instruction set.
	Quantity of write multiple registers 0001-0068 （104）	The number of registers to read consecutively. Please refer to the Modbus command set to ensure that all these register addresses are exist; otherwise, an error frame will be returned.
	Byte number	= quantity of register x2
CRC-16	Check code	

Write Multiple Registers (0x10) Response Frame



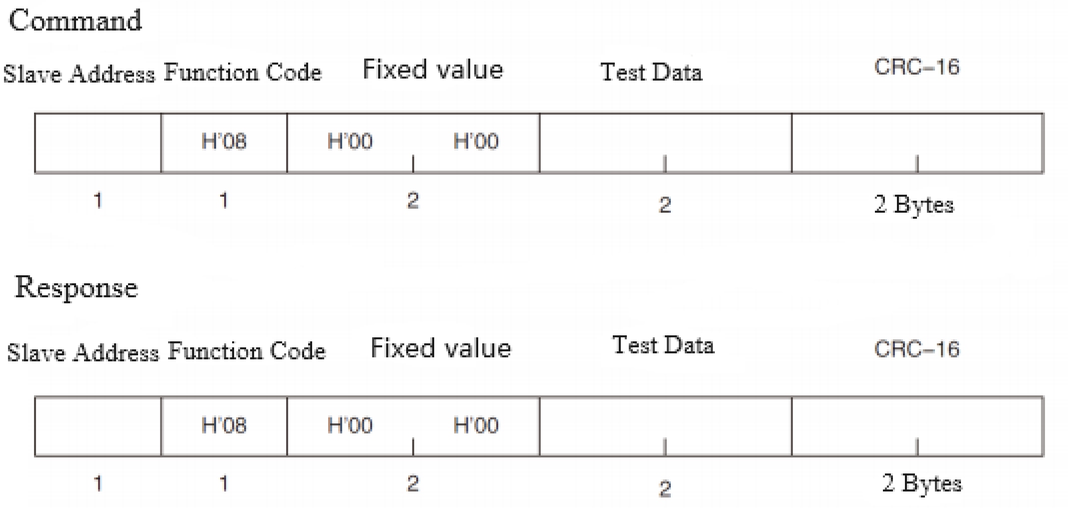
Name	Name	Description
	Slave station address	Original return
0x10	Function code	No exceptional: 0x10

or 0x90		Error code: 0x90
	Initial address	
	Quantity of register	
CRC-16	Check code	

3.6 Echo Test

The function code for the echo test is 0x08. It is used to debug Modbus.

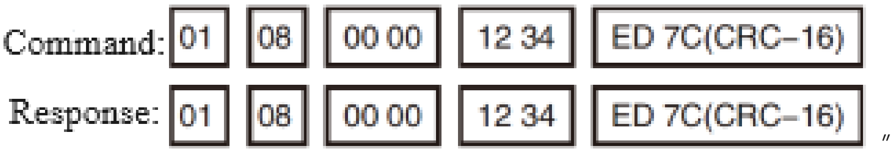
Write Multiple Registers (0x03) Response Frame



Echo Test (0x08)

Name	Name	Description
	Slave station address	Original return
0x08	Function code	
	Fixed value	00 00
	Test data	Arbitrary numerical value, such as 12 34
CRC-16	Check code	

For example, assume that the test data is 0x1234:



Command: 01 08 00 00 12 34 ED 7C(CRC-16)

Response: 01 08 00 00 12 34 ED 7C(CRC-16)

4. Modbus(RTU) Instruction Test

4.1 Register Overview

The following table lists all register addresses used by the instrument. Any address not in the table will return error code 0x02.

Register Address	Name	Numerical Value	Description
0200	Read the measured results	4-byte floating-point number	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0202	Read the comparator results of the channel	4-byte integer	Read-only register, data occupies 2 registers.
0204	Read the measured results	4-byte floating-point number	Read-only register, data occupies 2 registers, 4-byte integer. Byte order CCDD AABB, MSB.
0206	Triggers once and read the measured results AABB CCDD	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. It will automatically switch to the measurement page upon receiving the command, and the trigger mode will be set to external trigger.
0208	Triggers once and read the measured results CCDD AABB	4-byte floating-point number Byte order: MSB CCDD AABB	Read-only register, data occupies 2 registers, 4-byte integer. It will automatically switch to the measurement page upon receiving the command, and the trigger mode will be set to external trigger.
020A	Range number (R mode)	00 00 00 00-00 00 00 08	Read and write register, 4-byte integer.
020C	Automatic Range (R mode)	00 00 00 00: Auto range 00 00 00 01: Manual range 00 00 00 02: Nominal range	Read and write register, 4-byte integer.
020E	Range number (LPR mode)	00 00 00 01-00 00 00 04	Read and write register, 4-byte integer.
0210	Automatic Range (LPR mode)	00 00 00 00: Auto range 00 00 00 01: Manual range 00 00 00 02: Nominal range	Read and write register, 4-byte integer.
0212	Test mode	00 00 00 00: R 00 00 00 01: RT 00 00 00 02: T 00 00 00 03: LPR 00 00 00 04: LPRT	Read and write register, 4-byte integer.
0214	Test speed	00 00 00 00: Slow speed 00 00 00 01: Medium speed 00 00 00 02: Fast speed 00 00 00 03: High speed	Read and write register, 4-byte integer.
0216	System language	00 00 00 00: English	Read and write register, 4-byte

		00 00 00 01: Simplified Chinese	integer.
0218	Beeper	00 00 00 00: OFF 00 00 00 01: Qualified 00 00 00 02: Unqualified	Read and write register, 4-byte integer.
021A	Trigger setting	00 00 00 00: Internal trigger 00 00 00 01: External trigger	Read and write register, 4-byte integer.
021C	Trigger delay	0: Close trigger delay 4-byte floating-point number （0.1-9.9s）	Read and write register, 4-byte integer.
021E	Comparator BIN	00 00 00 00: Close comparator 00 00 00 01: 1-BIN 00 00 00 02: 2-BIN 00 00 00 03: 3-BIN 00 00 00 04: 4-BIN 00 00 00 05: 5-BIN 00 00 00 06: 6-BIN	Read and write register, 4-byte integer.
0220	Comparator mode	00 00 00 00: SEQ 00 00 00 01: ABS 00 00 00 02: PER	Read and write register, 4-byte integer.
0222	Nominal value	4-byte floating-point number	Read and write register, data occupies 2 registers.
0224	Lower limit of BIN1	4-byte floating-point number	Read and write register, data occupies 2 registers.
0226	Upper limit of BIN1	4-byte floating-point number	Read and write register, data occupies 2 registers.
0228	Lower limit of BIN2	4-byte floating-point number	Read and write register, data occupies 2 registers.
022A	Upper limit of BIN2	4-byte floating-point number	Read and write register, data occupies 2 registers.
022C	Lower limit of BIN3	4-byte floating-point number	Read and write register, data occupies 2 registers.
022E	Upper limit of BIN3	4-byte floating-point number	Read and write register, data occupies 2 registers.
0230	Lower limit of BIN4	4-byte floating-point number	Read and write register, data occupies 2 registers.
0232	Upper limit of BIN4	4-byte floating-point number	Read and write register, data occupies 2 registers.
0234	Lower limit of BIN5	4-byte floating-point number	Read and write register, data occupies 2 registers.
0236	Upper limit of BIN5	4-byte floating-point number	Read and write register, data occupies 2 registers.
0238	Lower limit of BIN6	4-byte floating-point number	Read and write register, data occupies 2 registers.
023A	Upper limit of BIN6	4-byte floating-point number	Read and write register, data occupies 2 registers.
023C	Execute the zero-clearing register to read its state	Read: 00 00 00 00: Zero-clearing is successful. 00 00 00 01: Zero-clearing is failed. 00 00 00 02 0 ADJ is not enabled.	Read-only register, data occupies 2 registers.

023E	0 ADJ	00 00 00 00: Close 00 00 00 01: Open	Read and write register, data occupies 2 registers.
0250	Read the measured results of CH1	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0252	Read the measured results of CH2	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0254	Read the measured results of CH3	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0256	Read the measured results of CH4	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0258	Read the measured results of CH5	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
025A	Read the measured results of CH6	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
025C	Read the measured results of CH7	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
025E	Read the measured results of CH8	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0260	Read the measured results of CH9	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0262	Read the measured results of CH10	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0264	Read the measured results of CH11	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0266	Read the measured results of CH12	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0268	Read the measured results of CH13	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
026A	Read the measured results of CH14	4-byte floating-point number Byte order: LSB	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.

		AABB CCDD	
026C	Read the measured results of CH15	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
026E	Read the measured results of CH16	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0270	Read the measured results of CH17	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0272	Read the measured results of CH18	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0274	Read the measured results of CH19	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0276	Read the measured results of CH20	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0278	Read the measured results of CH21	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
027A	Read the measured results of CH22	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
027C	Read the measured results of CH23	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
027E	Read the measured results of CH24	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0280	Read the measured results of CH25	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0282	Read the measured results of CH26	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0284	Read the measured results of CH27	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0286	Read the measured results of CH28	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0288	Read the measured results of	4-byte floating-point	Read-only register, data occupies 2

	CH29	number Byte order: LSB AABB CCDD	registers, 4-byte integer. Byte order AABB CCDD, LSB.
028A	Read the measured results of CH30	4-byte floating-point number Byte order: LSB AABB CCDD	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
028C	Trigger once and return 00 00 00 01	00 00 00 01: Trigger is completed	Read-only register, data occupies 2 registers, 4-byte integer. Byte order AABB CCDD, LSB.
0290	Return multi-channel comparator results	Comparators occupies 2 bits in each channel, from LSB to MSB. Comparator results 00: OFF 01: PASS 10: LOW 11: HIGH	Read-only register, data occupies 4 registers, 8-byte integer. From LSB to MSB, each 2-bit segment represents a comparator result for one channel. The lowest bit corresponds to the higher-numbered channel. For example, in the S30 model, the lowest bit represents CH30. Extra bits in the higher positions are replaced with 0.
02A0	Multi-channel comparator Lower limit of CH1	4-byte floating-point number	Read and write register, data occupies 2 registers.
02A2	Multi-channel comparator Upper limit of CH1	4-byte floating-point number	Read and write register, data occupies 2 registers.
02A4	Multi-channel comparator Lower limit of CH2	4-byte floating-point number	Read and write register, data occupies 2 registers.
02A6	Multi-channel comparator Upper limit of CH2	4-byte floating-point number	Read and write register, data occupies 2 registers.
02A8	Multi-channel comparator Lower limit of CH3	4-byte floating-point number	Read and write register, data occupies 2 registers.
02AA	Multi-channel comparator Upper limit of CH3	4-byte floating-point number	Read and write register, data occupies 2 registers.
02AC	Multi-channel comparator Lower limit of CH4	4-byte floating-point number	Read and write register, data occupies 2 registers.
02AE	Multi-channel comparator Upper limit of CH4	4-byte floating-point number	Read and write register, data occupies 2 registers.
02B0	Multi-channel comparator Lower limit of CH5	4-byte floating-point number	Read and write register, data occupies 2 registers.
02B2	Multi-channel comparator Upper limit of CH5	4-byte floating-point number	Read and write register, data occupies 2 registers.
02B4	Multi-channel comparator Lower limit of CH6	4-byte floating-point number	Read and write register, data occupies 2 registers.
02B6	Multi-channel comparator Upper limit of CH6	4-byte floating-point number	Read and write register, data occupies 2 registers.
02B8	Multi-channel comparator Lower limit of CH7	4-byte floating-point number	Read and write register, data occupies 2 registers.
02BA	Multi-channel comparator Upper limit of CH7	4-byte floating-point number	Read and write register, data occupies 2 registers.
02BC	Multi-channel comparator Lower limit of CH8	4-byte floating-point number	Read and write register, data occupies 2 registers.
02BE	Multi-channel comparator Upper limit of CH8	4-byte floating-point number	Read and write register, data occupies 2 registers.
02C0	Multi-channel comparator Lower limit of CH9	4-byte floating-point number	Read and write register, data occupies 2 registers.
02C2	Multi-channel comparator Upper limit of CH9	4-byte floating-point number	Read and write register, data occupies 2 registers.

02C4	Multi-channel comparator Lower limit of CH10	4-byte floating-point number	Read and write register, data occupies 2 registers.
02C6	Multi-channel comparator Upper limit of CH10	4-byte floating-point number	Read and write register, data occupies 2 registers.
02C8	Multi-channel comparator Lower limit of CH11	4-byte floating-point number	Read and write register, data occupies 2 registers.
02CA	Multi-channel comparator Upper limit of CH11	4-byte floating-point number	Read and write register, data occupies 2 registers.
02CC	Multi-channel comparator Lower limit of CH12	4-byte floating-point number	Read and write register, data occupies 2 registers.
02CE	Multi-channel comparator Upper limit of CH12	4-byte floating-point number	Read and write register, data occupies 2 registers.
02D0	Multi-channel comparator Lower limit of CH13	4-byte floating-point number	Read and write register, data occupies 2 registers.
02D2	Multi-channel comparator Upper limit of CH13	4-byte floating-point number	Read and write register, data occupies 2 registers.
02D4	Multi-channel comparator Lower limit of CH14	4-byte floating-point number	Read and write register, data occupies 2 registers.
02D6	Multi-channel comparator Upper limit of CH14	4-byte floating-point number	Read and write register, data occupies 2 registers.
02D8	Multi-channel comparator Lower limit of CH15	4-byte floating-point number	Read and write register, data occupies 2 registers.
02DA	Multi-channel comparator Upper limit of CH15	4-byte floating-point number	Read and write register, data occupies 2 registers.
02DC	Multi-channel comparator Lower limit of CH16	4-byte floating-point number	Read and write register, data occupies 2 registers.
02DE	Multi-channel comparator Upper limit of CH16	4-byte floating-point number	Read and write register, data occupies 2 registers.
02E0	Multi-channel comparator Lower limit of CH17	4-byte floating-point number	Read and write register, data occupies 2 registers.
02E2	Multi-channel comparator Upper limit of CH17	4-byte floating-point number	Read and write register, data occupies 2 registers.
02E4	Multi-channel comparator Lower limit of CH18	4-byte floating-point number	Read and write register, data occupies 2 registers.
02E6	Multi-channel comparator Upper limit of CH18	4-byte floating-point number	Read and write register, data occupies 2 registers.
02E8	Multi-channel comparator Lower limit of CH19	4-byte floating-point number	Read and write register, data occupies 2 registers.
02EA	Multi-channel comparator Upper limit of CH19	4-byte floating-point number	Read and write register, data occupies 2 registers.
02EC	Multi-channel comparator Lower limit of CH20	4-byte floating-point number	Read and write register, data occupies 2 registers.
02EE	Multi-channel comparator Upper limit of CH20	4-byte floating-point number	Read and write register, data occupies 2 registers.
02F0	Multi-channel comparator Lower limit of CH21	4-byte floating-point number	Read and write register, data occupies 2 registers.
02F2	Multi-channel comparator Upper limit of CH21	4-byte floating-point number	Read and write register, data occupies 2 registers.
02F4	Multi-channel comparator Lower limit of CH22	4-byte floating-point number	Read and write register, data occupies 2 registers.
02F6	Multi-channel comparator Upper limit of CH22	4-byte floating-point number	Read and write register, data occupies 2 registers.
02F8	Multi-channel comparator Lower limit of CH23	4-byte floating-point number	Read and write register, data occupies 2 registers.
02FA	Multi-channel comparator Upper limit of CH23	4-byte floating-point number	Read and write register, data occupies 2 registers.
02FC	Multi-channel comparator	4-byte floating-point	Read and write register, data

	Lower limit of CH24	number	occupies 2 registers.
02FE	Multi-channel comparator Upper limit of CH24	4-byte floating-point number	Read and write register, data occupies 2 registers.
0300	Multi-channel comparator Lower limit of CH25	4-byte floating-point number	Read and write register, data occupies 2 registers.
0302	Multi-channel comparator Upper limit of CH25	4-byte floating-point number	Read and write register, data occupies 2 registers.
0304	Multi-channel comparator Lower limit of CH26	4-byte floating-point number	Read and write register, data occupies 2 registers.
0306	Multi-channel comparator Upper limit of CH26	4-byte floating-point number	Read and write register, data occupies 2 registers.
0308	Multi-channel comparator Lower limit of CH27	4-byte floating-point number	Read and write register, data occupies 2 registers.
030A	Multi-channel comparator Upper limit of CH27	4-byte floating-point number	Read and write register, data occupies 2 registers.
030C	Multi-channel comparator Lower limit of CH28	4-byte floating-point number	Read and write register, data occupies 2 registers.
030E	Multi-channel comparator Upper limit of CH28	4-byte floating-point number	Read and write register, data occupies 2 registers.
0310	Multi-channel comparator Lower limit of CH29	4-byte floating-point number	Read and write register, data occupies 2 registers.
0312	Multi-channel comparator Upper limit of CH29	4-byte floating-point number	Read and write register, data occupies 2 registers.
0314	Multi-channel comparator Lower limit of CH30	4-byte floating-point number	Read and write register, data occupies 2 registers.
0316	Multi-channel comparator Upper limit of CH30	4-byte floating-point number	Read and write register, data occupies 2 registers.
0320	Multi-channel switch CH1	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0321	Multi-channel switch CH2	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0322	Multi-channel switch CH3	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0323	Multi-channel switch CH4	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0324	Multi-channel switch CH5	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0325	Multi-channel switch CH6	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0326	Multi-channel switch CH7	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0327	Multi-channel switch CH8	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0328	Multi-channel switch CH9	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0329	Multi-channel switch CH10	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.

032A	Multi-channel switch CH11	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
032B	Multi-channel switch CH12	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
032C	Multi-channel switch CH13	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
032D	Multi-channel switch CH14	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
032E	Multi-channel switch CH15	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
032F	Multi-channel switch CH16	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0330	Multi-channel switch CH17	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0331	Multi-channel switch CH18	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0332	Multi-channel switch CH19	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0333	Multi-channel switch CH20	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0334	Multi-channel switch CH21	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0335	Multi-channel switch CH22	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0336	Multi-channel switch CH23	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.

0337	Multi-channel switch CH24	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0338	Multi-channel switch CH25	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
0339	Multi-channel switch CH26	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
033A	Multi-channel switch CH27	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
033B	Multi-channel switch CH28	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
033C	Multi-channel switch CH29	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.
033D	Multi-channel switch CH30	00 00: Close channel 00 01: Open channel	Read and write register, 2-byte integer.

4.2 Fetch Measured Data

Fetch Measured Data (AABB CCDD)[0200]

The register 0200-0201 is used to fetch the measured data from the instrument.

For example, fetch the measured data

Command

1	2	3	4	5	6	7	8
01	03	0200		0002		CRC-16	
Slave station address	Read	Register address		Quantity of register		Check code	

Response

1	2	3	4	5	6	7	8	9

01	03	Byte	Single-precision floating-point number	CRC-16

Send

1	2	3	4	5	6	7	8
01	03	02	00	00	02	C5	B3

Response

1	2	3	4	5	6	7	8	9
01	03	04	42	C7	F9	9E	9C	4E

Where B4-B7 is the measurement data: 42C7F99E represents a single-precision floating-point number, LSB.

The byte order AA BB CC DD converted to a decimal number is 99.987564.

Fetch Comparator Results [0202]

The register 0202-0203 is used to fetch the measured data from the instrument.

The returned 4-byte integer data represents the comparator results.

00: Fail

01: BIN 1 (Pass)

02: BIN 2 (Pass)

03: BIN 3 (Pass)

04: BIN 4 (Pass)

05: BIN 5 (Pass)

06: BIN 6 (Pass)

Send

1	2	3	4	5	6	7	8
01	03	02	02	00	02	64	73

Response

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	00	FA	33

Fetch Measured Data (CCDD AABB) [0204]

The register 0204-0205 is used to fetch the measured data from the instrument.

For example, fetch the measured data

Command

1	2	3	4	5	6	7	8
01	03	0204		0002		CRC-16	
Slave station address	Read	Register address		Quantity of register		Check code	

Response

1	2	3	4	5	6	7	8	9
01	03	Byte	Single-precision floating-point number				CRC-16	

Send

1	2	3	4	5	6	7	8
01	03	02	04	00	02	84	72

Response

1	2	3	4	5	6	7	8	9
01	03	04	F9	A2	42	C7	1A	7F

Where B4-B7 is the measurement data: F9 A2 42 C7 represents a single-precision floating-point number, MSB.

The byte order is CC DD AA BB. When exchanging word order to AABBCCDD (LSB), 42 C7 F9 A2 converts to the decimal number 99.987564.

Trigger Once and Return Measured Results (AABB CCDD) [0206]

The register 0206-0207 is used to fetch the measured data from the instrument.

For example, fetch the measured data

Command

1	2	3	4	5	6	7	8
01	03	0206		0002		CRC-16	
Slave station address	Read	Register address		Quantity of register		Check code	

Response

1	2	3	4	5	6	7	8	9
01	03	Byte	Single-precision floating-point number				CRC-16	

Send

1	2	3	4	5	6	7	8
01	03	02	06	00	02	25	B2

Response

1	2	3	4	5	6	7	8	9
01	03	04	42	C7	F9	A2	9C	5F

Where B4-B7 is the measurement data: 42 C7 F9 A2 represents a single-precision floating-point number, LSB.

The byte order AA BB CC DD converted to a decimal number is 99.987572.

Note: This command is only available in external trigger mode. The instrument will trigger once and return the measured results.

Trigger Once and Return Measured Results (CCDD AABB) [0208]

The register 0208-0209 is used to fetch the measured data from the instrument.

For example, fetch the measured data

Command

1	2	3	4	5	6	7	8
01	03	0208		0002		CRC-16	
Slave station address	Read	Register address		Quantity of register		Check code	

Response

1	2	3	4	5	6	7	8	9
01	03	Byte	Single-precision floating-point number				CRC-16	

Send

1	2	3	4	5	6	7	8
01	03	02	08	00	02	44	71

Response

1	2	3	4	5	6	7	8	9
01	03	04	F9	A2	42	C7	EB	07

Where B4-B7 is the measurement data: F9 A2 42 C7 represents a single-precision floating-point number, MSB.

The byte order is CC DD AA BB. When exchanging word order to AABBCCDD (LSB), 42 C7 F9 A2 converts to the decimal number 99.987564.

Note: This command is only available in external trigger mode. The instrument will trigger once and return the measured results.

Read Multi-channel Measured Results (AABB CCDD) [0250]

The register 0250-028B is used to fetch the measured data from the multi-channel.

For example, fetch the measured data of CH1.

Command

1	2	3	4	5	6	7	8
01	03	0250		0002		CRC-16	
Slave station address	Read	Register address		Quantity of register		Check code	

Respond

1	2	3	4	5	6	7	8	9
01	03	Byte	Single-precision floating-point number				CRC-16	

Fetch the measured data

Send

1	2	3	4	5	6	7	8
01	03	02	50	00	02	C5	A2

Respond

1	2	3	4	5	6	7	8	9
01	03	04	42	C7	F9	9E	9C	4E

Where B4-B7 is the measurement data: 42C7F99E represents a single-precision floating-point number, LSB.

The byte order AA BB CC DD converted to a decimal number is 99.987564.

Fetch Multi-channel Comparator Results [0290]

The register 0290-0293 is used to fetch the measured data from the multi-channel comparator. The query returned 8-byte data from LSB to MSB, where each 2-bit represents a comparator result for one channel. The lowest bit corresponds to the higher-numbered channel. For example, in the S30 model, the lowest bit represents CH30. Extra bits in the higher positions are replaced with 0.

For example, fetch the measured data from the multi-channel comparator:

1. Open the comparator.
2. A query for CH1 returns the result PASS.
3. Queries for other channels return the result HIGH

Send

1	2	3	4	5	6	7	8
01	03	02	90	00	04	45	9C

Respond

1	2	3	4	5	6	7	8	9	10	11	12	13
01	03	08	07	FF	FF	FF	FF	FF	FF	FF	DA	B1

Trigger Multi-channel Scan Measurement Once (AABB CCDD) [028C]

The register 028C-028D is used to fetch the measured data from the multi-channel.

For example, fetch the measured data from the multi-channel

Command

1	2	3	4	5	6	7	8
01	03	028C		0002		CRC-16	
Slave station address	Read	Register address		Quantity of register		Check code	

Respond

1	2	3	4	5	6	7	8	9
01	03	02	00	00	00	01	B3	F3

The query returns 00 00 00 01, indicating that a trigger has been completed and that the measured data can be fetched from the result register of the multi-channel.

4.3 Parameter Setting

Range Scale [020A]

Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	0A	00	02	04	00	00	00	02	EB	71
	Write	Register address		Quantity of register		Byte	Data				CRC	

Response

1	2	3	4	5	6	7	8
01	10	02	0A	00	02	60	72
		Register address		Quantity of register		CRC	

Read

1	2	3	4	5	6	7	8
01	03	02	0A	00	02	E5	B1
	Read	Register address		Quantity of register		CRC	

Response

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	02	7B	F2
		Byte	Data				CRC	

The input range is from 0000 to 0008.

Range Type [020C]

Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	0C	00	02	04	00	00	00	00	EA	9A
	Write	Register address		Quantity of register		Byte	Data				CRC	

Response

1	2	3	4	5	6	7	8
01	10	02	0C	00	02	80	73
		Register address		Quantity of register		CRC	

Read

1	2	3	4	5	6	7	8
01	03	02	0C	00	02	05	B0
	Read	Register address		Quantity of register		CRC	

Response

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	00	FA	33
		Byte	Data				CRC	

The input range is from 0000 to 0002.

Multi-channel Switch [0320-033E]

The multi-channel switch values start from 02A0 and end at 0317. Each channel switch can be controlled separately.

Write: Close CH1

1	2	3	4	5	6	7	8	9	12	13
01	10	02	A0	00	01	02	00	00	9D	30
	Write	Register address		Quantity of register		Byte	Data		CRC	

Respond

1	2	3	4	5	6	7	8
01	10	02	0A	00	02	60	72
		Register address		Quantity of register		CRC	

Read

1	2	3	4	5	6	7	8
01	03	02	A0	00	02	C5	91
	Read	Register address		Quantity of register		CRC	

Respond

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	00	7B	F2
		Byte	Data				CRC	

4.4 Comparator Setting

Nominal Value Setting [0222-0223]

Write

100 (Single-precision floating-point number: 0x42C80000)

Response

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	22	00	02	04	42	C8	00	00	FC	88
	Write	Register address		Quantity of register		Byte	Data				CRC	

1	2	3	4	5	6	7	8
01	10	02	22	00	02	E0	7A
		Register address		Quantity of register		CRC	

Read

1	2	3	4	5	6	7	8
01	03	02	22	00	02	65	B9

	Read	Register address	Quantity of register	CRC

Response

1	2	3	4	5	6	7	8	9
01	03	04	42	C8	00	00	FA	33
		Byte	Data 100				CRC	

Limit Value [0224-023B]

The limit values of the 6 BINs for the comparator range from 0224 to 023B. Each comparator uses 2 registers for the lower limit and 2 registers for the upper limit, totaling 4 registers. The lower limit and the upper limit can be set separately or at the same time.

Write

Lower limit: 1E-5

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	24	00	02	04	37	27	C5	AC	04	76
	Write	Register address		Quantity of register		Byte	Data				CRC	

Response

1	2	3	4	5	6	7	8
01	10	02	24	00	02	00	7B

Upper limit: 1.2E5

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	26	00	02	04	47	EA	60	00	75	BD
	Write	Register address		Quantity of register		Byte	Data				CRC	

Response

1	2	3	4	5	6	7	8
01	10	02	26	00	02	A1	BB

Read

1	2	3	4	5	6	7	8
01	03	02	24	00	02	85	B8

Response

1	2	3	4	5	6	7	8	9
01	03	04	37	27	C5	AC	17	61
		Byte	Data 1.2E5				CRC	

Read

1	2	3	4	5	6	7	8
01	03	02	26	00	02	24	78

Response

1	2	3	4	5	6	7	8	9
01	03	04	47	EA	60	00	E7	73
		Byte	Data 1E-5				CRC	

Multi-channel Limit Value [02A0-0317]

The limit values of the multi-channel comparator start at 02A0 and end at 0317. The lower limit of the comparator uses 2 registers, and the upper limit uses 2 registers per channel, totaling 4 registers. The upper and lower limits can be set separately or simultaneously.

Write

CH1 lower limit: 1E-5

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	A0	00	02	04	37	27	C5	AC	0D	E5
	Write	Register address		Quantity of register		Byte	Data				CRC	

Respond

1	2	3	4	5	6	7	8
01	10	02	24	00	02	00	7B

CH1 upper limit: 1.2E5

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	A2	00	02	04	47	EA	60	00	7C	2E
	Wri	Register		Quantity		Byt	Data				CRC	

	te	address	of register	e		

Respond

1	2	3	4	5	6	7	8
01	10	02	26	00	02	A1	BB

Read CH1 lower limit

1	2	3	4	5	6	7	8
01	03	02	A0	00	02	C5	91

Respond

1	2	3	4	5	6	7	8	9
01	03	04	37	27	C5	AC	17	61
		Byte	Data 1.2E5				CRC	

Read CH1 upper limit

1	2	3	4	5	6	7	8
01	03	02	A2	00	02	64	51

Respond

1	2	3	4	5	6	7	8	9
01	03	04	47	EA	60	00	E7	73
		Byte	Data 1E-5				CRC	

4.5 System Function

Zero Clearing [023C]

The instrument will start to execute short-circuit zero clearing when reading the register 023C.

Before zero clearing, the test wire should be short-circuited and the 0ADJ (zero clearing) function should be enabled; otherwise, zero clearing will fail.

The process of zero clearing takes few seconds.

During the execution of zero clearing or after zero clearing is completed, the state of zero clearing will be returned.

0000: Zero clearing is successful.

FFFF: Zero clearing is failed

0002: 0 ADJ (zero clearing) function is not enabled.

Read

1	2	3	4	5	6	7	8
01	03	02	3C	00	02	05	BF

Response

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	00	FA	33
		Byte	Zero clearing is successful				CRC	