UNI-T®

Instruments.uni-trend.com

Programming Manual
UDP6722 DC Power Supply

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

SCPI（Standard Commands for Programmable Instruments）is a standardized instrument programming language that builds on existing standards IEEE 488.1 and IEEE 488.2 and follows the floating point rules of IEEE 754 standard, ISO 646 message exchange 7-bit encoding notation (equivalent to ASCII programming) and many other standards.

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

1.1 Command String Parse

The host computer can send a string of commands to the instrument and the command parser of the instrument starts to parsing after catching the terminator (\n) or an input buffer overflow.

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.

1.2 Command Parse Rule

Command parser only parses and responds to ASCII data.
The command parser starts to executing when receive the end mark. The instrument receives the following contents as end mark.
CR+LF
The command parser will terminate the parsing immediately after parsing an error, and the current command will be invalidated.
The command parser is case-insensitive for parsing command strings.
The command parser supports abbreviated form of command and the detailed see the following section.
Add ADDR□Local address::□ in front of RS485 mode, SCPI protocol, the local address can set to 1-32.
It is convenient for communicating with multiple machines via SCPI protocol.
For Example: ADDR□1::□IDN? □ represents a space
The instrument sends
The end mark of data send by the instrument
The default data end mark sent by the instrument is 0x0D 0x0A (CRLF).
Use semicolon ";" can send multiple commands.

1.3 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.

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

[ ] 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.

( ) The abbreviated form of the parameter is enclosed in parentheses.

Capital letter Abbreviated form of the command.

☐ Space character It represents a space and only for reading.

1.4 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. SCPI uses a colon (:) to separate high level commands from low level commands.

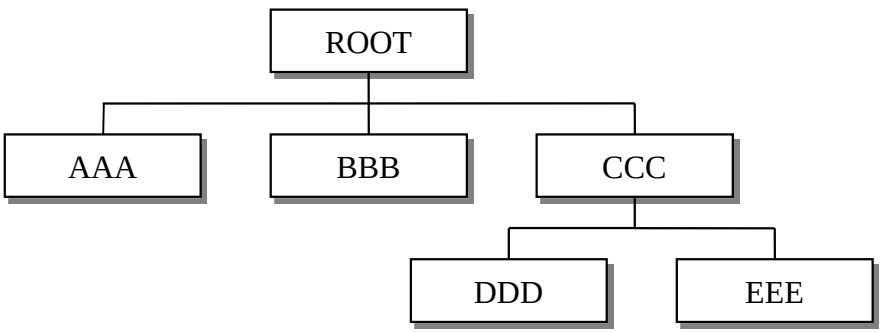


Figure 1-1 Command Tree Structure

For Example

ROOT:CCC:DDD ppp

ROOT Subsystem command

CCC Second level

DDD Third level

ppp Parameter

1.5 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]

1.6 Command

Command words 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.

1.7 Parameter

Single character command, no parameter
For Example AAA:BBB
Parameter can be string format and its abbreviated form is also follow the last section " command abbreviated rule" For example AAA:BBB 1.23
Parameter can be numerical value format
<integer> integer 123, +123, -123
<float> floating point number
1. <fixfloat>: fixed floating point number: 1.23, -1.23
2. <Sciloat>: floating point number represented by scientific notation: 1.23E+4, +1.23e-4
3. <mpfloat>: floating point number represented by multiplying power: 1.23k, 1.23M, 1.23G, 1.23u

Table 0-1 Abbreviation of Multiplying Power

Numerical Value	Multiplying Power
1E18(EXA)	EX
1E15(PETA)	PE
1E12(TERA)	T
1E9(GIGA)	G
1E6(MEGA)	MA
1E3(KILO)	K
1E-3(MILLI)	M
1E-6(MICRO)	U
1E-9(NANO)	N
1E-12(PICO)	P
1E-15(PEMTO)	F
1E-18(ATTO)	A

Note: The multiplying power is case-insensitive, and the writing is differently from the standard name.

1.8 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.

For Example AAA:BBB 100.0 ; CCC:DDD

: Colon is for separating command tree or restart the command tree.

For Example AAA : BBB : CCC 123.4; : DDD : EEE 567.8

? Question mark is for querying.

For Example AAA ?

☐ **Blank** is for separating the parameter.

For Example AAA:BBB☐1.234

2. SCPI Command Reference

All commands is explained by the subsystem command order. All subsystems are listed as follows.

DISPlay	Display subsystem
SYSTem	System subsystem
OUTPut	Output subsystem
CURRent	Current setup subsystem
VOLTage	Voltage setup subsystem
APPLy	Voltage and current setup subsystem
MEASure	Measurement subsystem
FETCh?	Fetch result subsystem
LIST	List setup subsystem
DELAyer	Delayer setup subsystem
FILE	File subsystem

Common Command

IDN?	Instrument query subsystem

2.1 DISPlay Subsystem

DISPlay subsystem is used to switch to different display pages.

Figure 1-1 DISPlay subsystem tree

DISPlay	:PAGE	{MEAS,MSET,LIST,LISTFILE, SYST2, SINF}

DISPlay:PAGE

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

Command Syntax	DISPlay:PAGE <page name>
Parameter	< page name > includes MEAS Measurement display page MSET Measurement setup page LIST List setup page LISTFile List file page DELA Delayer setup page DELAFile Delayer file page SYST System setup page FILE File management page
For Example	Send >disp:page mset //Switching to the measurement setup page.
Query Syntax	DISPlay:PAGE?
Query Respond	< page name > MEAS Measurement display page MSET Measurement setup page LIST List setup page LISTFile List file page DELA Delayer setup page DELAFile Delayer file page SYST System setup page FILE File management page

For Example	Send >DISP:PAGE? Return > MEAS

2.2 SYSTem Subsystem

SYSTem subsystem command is used to set system parameter.

Notes:
The parameter set by SYSTem subsystem command will not automatic save in the file.
After the parameter is set, it need use FILE subsystem command to save the setting to internal file.

SYSTem subsystem tree

SYSTem	:LANGuage	{ENGLISH,CHINESE,EN,CN}	Language setting
	:TIME	<YEAR>,<MONTH>,<DAY>,<HOUR>,<MINUTE>,<SECOND>	Time setting
	:KEYSound	{OFF,ON,0,1}	Key sound setting

SYSTem:LANGuage

Language Setting

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

SYSTem:TIME

Time Setting

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 Respond	<YEAR>-<MONTH>-<DAY> <HOUR>:<MINUTE>:<SECOND>
For Example	Send >SYST:TIME? Return >2023-04-01 09:30:49

SYSTem:KEYSound

Key Sound Setting

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

2.3 OUTPut Subsystem

OUTPut subsystem command is used to stop the output.

OUTPut subsystem tree

OUTPut	{OFF,ON,0,1}			Language setting
	:CVCC?			Query CVCC
	:TIMer	{OFF,ON,0,1}		Set the output status of the timer
	:TIMer	:DATA	<float>	Set the output numerical value of the timer
	:POUT	{OFF,ON,0,1}		Set the status of boot output

OUTPut

Turn on power output

Command Syntax	OUTPut {OFF,ON,0,1}
For Example	Send >OUTP ON //Turn on power output.
Query Syntax	OUTPut?
Query Respond	{OFF,ON}

OUTPut:CVCC

Query power CV CC status

Command Syntax	OUTPut:CVCC?
Query Respond	{CV,CC}

OUTPut:TIMer

Set the output status of the timer

This command is invalid if the instrument is operating the fixed output.

Command Syntax	OUTPut:TIMer {OFF,ON,0,1}
For Example	Send >OUTP:TIM ON //enable to output the timer.
Query Syntax	OUTPut:TIMer?
Query Respond	{OFF,ON}

OUTPut:TIMer:DATA

Set the output numerical value of the timer

This command is invalid if the instrument is operating the fixed output.

Command Syntax	OUTPut:TIMer:DATA <float>
For Example	Send >OUTP:TIM:DATA 10.1
Query Syntax	OUTPut:TIM:DATA?
Query Respond	<float>

OUTPut:TIMer:POUT

Set the status of boot output

Command Syntax	OUTPut:POUT {OFF,ON,0,1}
For Example	Send >OUTP:POUT OFF
Query Syntax	OUTPut:POUT?
Query Respond	{OFF,ON}

2.4 [SOURce:]CURRent Subsystem

CURRent subsystem command is used to set the parameter of output current. This subsystem command is invalid **if the instrument is operating list or delayer mode**.

CURRent subsystem tree

[SOURce:]CURRent	{MINimum,MAXimum,DEFault,<float>}			Set the current value
	:PROTection	<float>		Set the OCP value
	:PROTection	:STATe	{OFF,ON,0,1}	Set the status of OCP
	:PROTection	TRIPed?		Query whether is OCP
	:PROTection	CLEar		Delete the alarm of OCP

[SOURce:]CURRent

Set the OCP value

Command Syntax	[SOURce]:CURRent {MINimum,MAXimum,DEFault,<float>}
For Example	Send >CURR 5.1
Query Syntax1	[SOURce:]CURRent?
Query Syntax2	[SOURce:]CURRent? {MINimum,MAXimum,DEFault}
Query Respond	<float>

[SOURce:]CURRent:PROTection

Set the output current value

Command Syntax	[SOURce:]CURRent:PROTection {MINimum,MAXimum,<float>}
For Example	Send >CURR:PROT 10.1
Query Syntax1	[SOURce:]CURRent:PROTection?
Query Syntax2	[SOURce:]CURRent:PROTection? {MINimum,MAXimum}
Query Respond	<float>

[SOURce:]CURRent:PROTection:STATe

Set the status of OCP

Command Syntax	[SOURce:]CURRent:PROTection:STATe {OFF,ON,0,1}
For Example	Send >CURR:PROT:STAT OFF

Query Syntax	[SOURce:]CURRent:PROTection:STATe?
Query Respond	{OFF,ON}

[SOURce:]CURRent:PROTection:TRIPed?

Query whether is OCP

Command Syntax	[SOURce:]CURReng:PROTection:TRIPed?
For Example	Send >CURR:PROT:TRIP?
Query Respond	{0,1} //0: not OCP, 1: OCP

[SOURce:]CURRent:PROTection:CLEar

Delete the alarm of OCP

Command Syntax	[SOURce:]CURRent:PROTection:CLEar
For Example	Send >CURR:PROT:CLE

2.5 [SOURce:]VOLTage Subsystem

VOLTage subsystem command is used to set the parameter of output voltage. This subsystem command is invalid **if the instrument is operating list or delayer mode.**

VOLTage subsystem tree

[SOURce:]VOLTage	{MINimum,MAXimum,DEFault,<float>}			Set the voltage value
	:PROTection	<float>		Set the OVP value
	:PROTection	:STATe	{OFF,ON,0,1}	Set the status of OVP
	:PROTection	:TRIPed?		Query whether is OVP
	:PROTection	:CLEar		Delete the alarm of OVP

[SOURce:]VOLTage

Set the voltage value

Command Syntax	[SOURce]:VOLTage {MINimum,MAXimum,DEFault,<float>}
For Example	Send >VOLT 5.1
Query Syntax1	[SOURce:]VOLTage?
Query Syntax2	[SOURce:]VOLTage? {MINimum,MAXimum,DEFault}
Query Respond	<float>

[SOURce:]VOLTage:PROTection

Set the OVP value

Command Syntax	[SOURce:]VOLTage:PROTection {MINimum,MAXimum,<float>}
For Example	Send >VOLT:PROT 10.1

Query Syntax1	[SOURce:]VOLTage:PROTection?
Query Syntax2	[SOURce:]VOLTage:PROTection? {MINimum,MAXimum}
Query Respond	<float>

[SOURce:]VOLTage:PROTection:STATe

Set the status of OVP

Command Syntax	[SOURce:]VOLTage:PROTection:STATe {OFF,ON,0,1}
For Example	Send >VOLT:PROT:STAT OFF
Query Syntax	[SOURce:]VOLTage:PROTection:STATe?
Query Respond	{OFF,ON}

[SOURce:]VOLTage:PROTection:TRIPed?

Query whether is OVP

Command Syntax	[SOURce:]VOLTage:PROTection:TRIPed?
For Example	Send >VOLT:PROT:TRIP?
Query Respond	{0,1} //0: not OVP,1:OVP

[SOURce:]VOLTage:PROTection:CLEar

Delete the alarm of OVP

Command Syntax	[SOURce:]VOLTage:PROTection:CLEar
For Example	Send >VOLT:PROT:CLE

2.6 [SOURce:] APPLy Subsystem

APPLy subsystem is used to set the parameter of voltage and current. This subsystem command is invalid **if the instrument is operating list or delayer mode.**

APPLy subsystem tree

APPLy	{MINimum,MAXimum,DEFault,<float>},{MINimum,MAXimum,DEFault,<float>}		Set the voltage and the current value
	:ALL	{MINimum,MAXimum,DEFault,<float>},{MINimum,MAXimum,DEFault,<float>},{MINimum,MAXimum,<float>},{MINimum,MAXimum,<float>}	Set the OVP and the OCP value

[SOURce:]APPLy

Set the voltage and the current value

Command Syntax	[SOURce:]APPLy {MINimum,MAXimum,DEFault,<float>},{MINimum,MAXimum,DEFault,<flo> // Voltage, current
For Example	Send >APPL 80,5 //Set the voltage to 80V, the current to 5A.
Query Syntax1	[SOURce:]APPLy?

Query Syntax2	[SOURce:]APPLy? {MINimum,MAXimum,DEFault,<float>},MINimum,MAXimum,DEFault,<float>}
Query Respond	<float>,<float>
For Example1	Send >APPL? Return >80,5 //Set the voltage to 80V, the current to 5A.
For Example2	Send >APPL? MAX,MAX Return >85.00,20.5 //Set the maximum voltage to 80V, the maximum current 20.5A

[SOURce:]APPLy:ALL

Set the OVP and the OCP value

Command Syntax	[SOURce:]APPLy:ALL {MINimum,MAXimum,DEFault,<float>},{MINimum,MAXimum,DEFault,<float>},{MINimum,MAXimum,<float>},{MINimum,MAXimum,<float>}
For Example	Send >APPL:ALL 80,5,85,20 //Set the voltage to 80V, the current to 5A, the OVP to 85V, the OCP to 20A.
Query Syntax1	[SOURce:]APPLy:ALL?
Query Syntax2	[SOURce:]APPLy? {MINimum,MAXimum,DEFault},{MINimum,MAXimum,DEFault},{MINimum,MAXimum},{MINimum,MAXimum}
Query Respond	<float>,<float>,<float>,<float> //Voltage, current, OVP, OCP

2.7 [SOURce:] APPLy Subsystem

APPLy subsystem command is used to set the voltage and the current value.

Command Syntax	[SOURce:]APPLy {MINimum,MAXimum,DEFault,<float>},{MINimum,MAXimum,DEFault,<float>}
For Example	Send > APPL 80,5 //Set the voltage to 80V, the current to 5A.
Query Syntax1	[SOURce:]APPLy?
Query Syntax2	[SOURce:]APPLy? {MINimum,MAXimum,DEFault},{MINimum,MAXimum,DEFault}
Query Respond	<float>,<float> //Voltage, current

2.8 MEASure Subsystem

MEASure subsystem command is used to query the readback data.

MEASure subsystem tree

MEASure	[:VOLTage]?	Query the readback voltage value
	:CURRent?	Query the current voltage value
	:POWer?	Query the readback power value
	:ALL?	Query all the readback values

MEASure[:VOLTage]?

Query the readback voltage value

Query Syntax	MEASure[:VOLTage]?
Query Respond	<float>

MEASure:CURRent?

Query the current voltage value

Query Syntax	MEASure:CURRent?
Query Respond	<float>

MEASure:POWer?

Query the readback power value

Query Syntax	MEASure:POWer?
Query Respond	<float>

MEASure:ALL?

Query all the readback values (voltage, current and power)

Query Syntax	MEASure:ALL?
Query Respond	<float>,<float>,<float> // Voltage, current, power

2.9 FETCh Subsystem

FETCh subsystem command is used to query the readback data. It is the same as MEASure subsystem command.

FETCh subsystem tree

FETCh	[:VOLTage]?	Query the readback voltage value
	:CURRent?	Query the readback current value
	:POWer?	Query the readback power value
	:ALL?	Query all the readback values

FETCh[:VOLTage]?

Query the readback voltage value

Query Syntax	FETCh[:VOLTage]?
Query Respond	<float>

FETCh:CURRent?

Query the readback current value

Query Syntax	FETCh:CURRent?
Query Respond	<float>

FETCh:POWer?

Query the readback power value

Query Syntax	FETCh:POWer?
Query Respond	<float>

FETCh:ALL?

Query all the readback values (voltage, current and power)

Query Syntax	FETCh:ALL?
Query Respond	<float>,<float>,<float> // Voltage, current, power

2.10 LIST Subsystem

LIST subsystem command is used to set the list output. **This subsystem command is invalid if the instrument is operating the list output. FUNCtionA and LOAD command is invalid if the instrument is in the delayer output mode.**

LIST subsystem tree

LIST	:STARtno	<integer>	Set the initial group number
	:GROUps	<integer>	Set the output group number
	:REPEat	<integer>	Set the repeat times
	:FINIsh	{STOP,HOLD}	Set the stop status
	:FUNCtion	{OFF,ON,0,1}	Set to enable the list
	:STEP	<integer>,<float>,<float>,<float> //Group number, voltage, current, time	Parameter setup of the single step
	:VOLTage	<integer>,<float> //Group number, voltage	Voltage setup of the single step
	:CURRent	<integer>,<float> //Group number, current	Current setup of the single step
	:TIMer	<integer>,<float> //Group number, time	Time setup of the single step
	:LOAD	<integer>	Loading the list file
	:SAVE	<integer>	Save the list file
	:DELete	<integer>	Delete the list file
	:REName	<integer>,<string>	Rename the list file
	:PLOad	<integer>,{OFF,ON,0,1}	Boot recalling the list file
	:AUTOSave	<integer>,{OFF,ON,0,1}	Auto save the list file

LIST:STARtno

Set the initial step of the list

Command Syntax	LIST:STARtno <integer>
For Example	Send >LIST:STAR 1
Query Syntax	LIST:STARtno?
Query Respond	<integer>

LIST:GROUps

Set the output group number of the list

Command Syntax	LIST:GROUps <integer>
For Example	Send >LIST:GROUps 1
Query Syntax	LIST:GROUps?
Query Respond	<integer>

LIST:REPEat

Set the repeat times of the list

Command Syntax	LIST:REPEat <integer>
For Example	Send >LIST:REPEat 1
Query Syntax	LIST:REPEat?
Query Respond	<integer>

LIST:FINIsh

Set the stop status of the list

Command Syntax	LIST:FINIsh {STOP,HOLD}
For Example	Send >LIST:FINIsh STOP
Query Syntax	LIST:FINIsh?
Query Respond	{STOP,HOLD}

LIST:FUNCtion

Set to enable the list

Command Syntax	LIST:FUNCtion {OFF,ON,0,1}
For Example	Send >LIST:FUNC ON
Query Syntax	LIST:FUNCtion?
Query Respond	{OFF,ON}

LIST:STEP

Set the list data

Command Syntax	LIST:STEP <integer>,<float>,<float>,<float>
For Example	Send >LIST:STEP 1,80,5,10 //Set the parameter of the first step to 80V,5A,10s
Query Syntax	LIST:STEP? <integer>
Query Respond	<integer>,<float>,<float>,<float> //Group number, voltage, current, time
For Example	Send >LIST:STEP? 1 Return >1,80.00,5.00,10.0 //Group number, voltage, current, time

LIST:VOLTage

Set the voltage data of the list

Command Syntax	LIST:VOLTage <integer>,<float> //Group number, voltage
For Example	Send >LIST:VOLT 1,80 //Set the voltage of the group 1 in the list to 80V.
Query Syntax	LIST:VOLTage? <integer>
Query Respond	<float>
For Example	Send >LIST:VOLT? 1 Return >80

LIST:CURRent

Set the current data of the list

Command Syntax	LIST:CURRent <integer>,<float> //Group number, current
For Example	Send >LIST:CURR 1,5 //Set the current of the group 1 in the list to 5A.
Query Syntax	LIST:CURR? <integer>
Query Respond	<float>
For Example	Send >LIST:CURR? 1 Return >5

LIST:TIMer

Set the time data of the list

Command Syntax	LIST:TIMer <integer>,<float> //Group number, time
For Example	Send >LIST:TIM 1,10 //Set the time of the first group in the list to10s.
Query Syntax	LIST:TIM? <integer>
Query Respond	<float>
For Example	Send >LIST:TIM? 1 Return > 10

LIST:LOAD

Loading the list file

This command is invalid if the instrument is operating the fixed output.

Command Syntax	LIST:LOAD <integer>
For Example	Send >LIST:LOAD 1 //Loading the list file 1.

LIST:SAVE

Save the list file

Command Syntax	LIST:SAVE <integer>
For Example	Send >LIST:SAVE 1 //Save the currently setting into list file 1.

LIST:DELete

Delete the list file

Command Syntax	LIST:DELete <integer>
For Example	Send >LIST:DEL 1 //Delete the list file 1.

LIST:REName

Rename the list file

Command Syntax	LIST:REName <integer>,<string>
For Example	Send >LIST:REN 1,"ABC" //Rename the list file 1 to ABC.

LIST:PLOad

Set the boot loading

Set the corresponding file to boot loading. Only one file can be set to boot loading at the same time. For Example, file 1 is boot loading, if file 2 sets to boot loading, the boot loading of file 1 will be cancelled, if the boot load file is deleted, then file 0 will be set to boot loading.

Command Syntax	LIST:PLOad <integer>
For Example	Send >LIST:PLOad 1 //Set the list file 1 to boot loading.
Query Syntax1	LIST:PLOad? //Query the file number of the boot loading.
Query Respond1	<integer>
Query Syntax2	LIST:PLOad? <integer>//Query the specified file whether is boot loading.
Query Respond2	{OFF,ON}
For Example1	Send >LIST:PLOad? // Query the file number of the boot loading. Respond >3 //File 3 is to boot loading.
For Example2	Send 2>LIST:PLOad? 1 // Query the file 1 whether is boot loading. Respond 2 >OFF //File 1 is not boot loading.

LIST:AUTOSave

Set the automatic storage function

Set the automatic storage function for boot loading of the file. If the automatic storage function is enabled, the parameter set by manual will immediately save in the corresponding file.

Command Syntax	LIST:AUTOSave {OFF,ON,0,1}
For Example	Send >LIST:AUTOS ON //Enable the automatic storage function.
Query Syntax	LIST:AUTOSave?
Query Respond	{OFF,ON}

2.11 DELAyer Subsystem

DELAyer subsystem command is used to set the delayer output. This subsystem command is invalid **if the instrument is operating the delayer output. And when** the instrument is operating the list output, **FUNCtion and LOAD command is invalid.**

DELAyer subsystem tree

DELAyer	:STARtno	<integer>	Set the initial group number
	:GROUps	<integer>	Set the output group number
	:REPEat	<integer>	Set the repeat times
	:FINIsh	{STOP,HOLD}	Set the stop status
	:FUNCtion	{OFF,ON,0,1}	Enable the delayer
	:STEP	<integer>,{OFF,ON,0.1},<float> //group number, status, time	Parameter setup of the single step
	:STATe	<integer>,{OFF,ON,0.1} //group number, status	Status setuo of the single step
	:TIMer	<integer>,<float> // group number, time	Time setup of the single step
	:LOAD	<integer>	Loading the delayer file
	:SAVE	<integer>	Save the delayer file

	:DELete	<integer>	Delete the delayer file
	:REName	<integer>,<string>	Rename the delayer file
	:PLOad	<integer>,{OFF,ON,0,1}	Boot loading the delayer file
	:AUTOSave	<integer>,{OFF,ON,0,1}	Auto save the delayer file

DELAyer:STARtno

Set the initial group number of the delayer

Command Syntax	DELAyer:STARtno <integer>
For Example	Send >DELA:STAR 1
Query Syntax	DELAyer:STARtno?
Query Respond	<integer>

DELAyer:GROUps

Set the output group number of the delayer

Command Syntax	DELAyer:GROUps <integer>
For Example	Send >DELA:GROU 1
Query Syntax	DELAyer:GROUps?
Query Respond	<integer>

DELAyer:REPEat

Set the repeat times of the delayer

Command Syntax	DELAyer:REPEat <integer>
For Example	Send >DELA:REPE 1
Query Syntax	DELAyer:REPEat?
Query Respond	<integer>

DELAyer:FINIsh

Set the stop status of the delayer

Command Syntax	DELAyer:FINIsh {STOP,HOLD}
For Example	Send >DELA:FINIsh STOP
Query Syntax	DELAyer:FINIsh?
Query Respond	{STOP,HOLD}

DELAyer:FUNCtion

Enable the delayer

Command Syntax	DELAyer:FUNCtion {OFF,ON,0,1}
For Example	Send >DELA:FUNC ON
Query Syntax	DELAyer:FUNCtion?
Query Respond	{OFF,ON}

DELAyer:STEP

Set the delayer's data

Command Syntax	DELAyer:STEP <integer>,{OFF,ON,0.1},<float>
For Example	Send >DELA:STEP 1,ON,10.1 //Enable the first group of the delayer to 10.1s
Query Syntax	DELAyer:STEP? <integer>
Query Respond	<integer>,{OFF,ON},<float>
For Example	Send >DELA:STEP? 1 Return >1,ON,10.1 //Group number, status, time

DELAyer:STATe

Set the output status for each step of the delayer

Command Syntax	DELAyer:STATe <integer>,{OFF,ON,0,1}
For Example	Send >DELA:STAT 1,ON //Enable the first group of the delayerto output
Query Syntax	DELAyer:STATe? <integer>
Query Respond	{OFF,ON}

DELAyer:TIMer

Set the output time for each step of the delayer

Command Syntax	DELAyer:TIMer <integer>,{<float>,MINimum,MAXimum}
For Example	Send >DELA:TIM 1,10.1 //Enable the first group of the delayer to 10.1s
Query Syntax	DELAyer:TIMer? <integer>
Query Respond	<float>

DELAyer:LOAD

Loading the delayer file

Command Syntax	DELAyer:LOAD <integer>
For Example	Send >DELAyer:LOAD 1 //Loading the delayer file 1.

DELAyer:SAVE

Save the delayer file

Command Syntax	DELAyer:SAVE <integer>
For Example	Send >DELAyer:SAVE 1 //Save the currently setting into the delayer file 1

DELAyer:DELete

Delete the delayer file

Command Syntax	DELAyer:DELete <integer>
For Example	Send >DELAyer:DEL 1 //Delete the delayer file 1.

DELAyer:REName

Rename the delayer file

Command Syntax	DELAyer:REName <integer>,<string>
For Example	Send >DELAyer:REN 1,"ABC" //Rename the delayer file 1 to ABC.

DELAyer:PLOad

Set the boot loading

Set the corresponding file to boot loading. Only one file can be set to boot loading at the same time. For Example, file 1 is boot loading, if file 2 sets to boot loading, the boot loading of file 1 will be cancelled, if the boot load file is deleted, then file 0 will be set to boot loading.

Command Syntax	DELAyer:PLOad <integer>
For Example	Send >DELAyer:PLO 1 //Set the delayer file 1 to boot loading.
Query Syntax1	DELAyer:PLOad? //Query the file number of the boot loading.
Query Respond1	<integer>
Query Syntax2	DELAyer:PLOad? <integer> //Query the specified file whether is boot loading.
Query Respond2	{OFF,ON} //Query the delayer file 1 whether is boot loading.
For Example1	Send >DELA:PLOad? //Query the file number of the boot loading. Respond >3 //File 3 is to boot loading.
For Example2	Send 2>DELA:PLOad? 1 //Query the file 1 whether is boot loading. Respond 2>OFF //File 1 is not boot loading.

DELAyer:AUTOSave

Set the automatic storage function

Set the automatic storage function for boot loading of the file. If the automatic storage function is enabled, the parameter set by manual will immediately save in the corresponding file.

Command Syntax	DELAyer:AUTOSave {OFF,ON,0,1}
For Example	Send >DELAyer:AUTOS ON //Enable the automatic storage function.
Query Syntax	DELAyer:AUTOSave?
Query Respond	{OFF,ON}

2.12 FILE Subsystem

FILE subsystem tree

FILE	:LOAD	<integer>	Loading the system file
	:SAVE	<integer>	Save the system file
	:DELete	<integer>	Delete the system file
	:REName	<integer>,<string>	Rename the system file
	:PLOad	<integer>,{OFF,ON,0,1}	Boot loading the system file
	:AUTOSave	<integer>,{OFF,ON,0,1}	Auto save the system file

FILE:LOAD

Loading the system file

Command Syntax	FILE:LOAD <integer>
For Example	Send >FILE:LOAD 1 // Loading the system file 1.

FILE:SAVE

Save the system file

Command Syntax	FILE:SAVE <integer>
For Example	Send >FILE:SAVE 1 //Save the currently setting to the system file 1.

FILE:DELete

Delete the system file

Command Syntax	FILE:DELete <integer>
For Example	Send >FILE:DEL 1 //Delete the system file 1.

FILE:RENname

Rename the system file

Command Syntax	FILE:RENname <integer>,<string>
For Example	Send >FILE:REN 1,"ABC" //Rename the system file 1 to ABC.

FILE:PLOad

Set the file number of the boot loading

Set the corresponding file to boot loading. Only one file can be set to boot loading at the same time. For Example, file 1 is boot loading, if file 2 sets to boot loading, the boot loading of file 1 will be cancelled, if the boot load file is deleted, then file 0 will be set to boot loading.

Command Syntax	FILE:PLOad <integer>
For Example	Send >FILE:PLO 1 //Set the system file 1 to boot loading.
Query Syntax1	FILE:PLOad? //Query the file number of the boot loading.
Query Respond1	<integer>
Query Syntax2	FILE:PLOad? <integer> //Query the specified file whether is boot loading.
Query Respond2	{OFF,ON}
For Example1	Send >FILE:PLOad? //Query the file number of the boot loading. Respond >3 //File 3 is to boot loading.
For Example2	Send 2>FILE:PLOad? 1 //Query the file 1 whether is boot loading. Respond 2>OFF //File 1 is not boot loading.

FILE:AUTOSave

Set the automatic storage function

Set the automatic function for boot loading file. If the automatic function is enabled, the parameter set by manual will immediately save in the corresponding file.

Command Syntax	FILE:AUTOSave {OFF,ON,0,1}
For Example	Send >FILE:AUTOS 1,ON //Enable the system to automatic storage function.
Query Syntax	FILE:AUTOSave?
Query Respond	{OFF,ON}

2.13 *IDN？ Subsystem

*IDN？ subsystem command is used to return the version number of the instrument.

Query Syntax	*IDN?
Query Respond	<Manufacturer>,<MODEL>,<Sn>,<Revision>
For Example	Send > *IDN? Return > UNIT,UDP6722,UNLICENSED,REV1.21

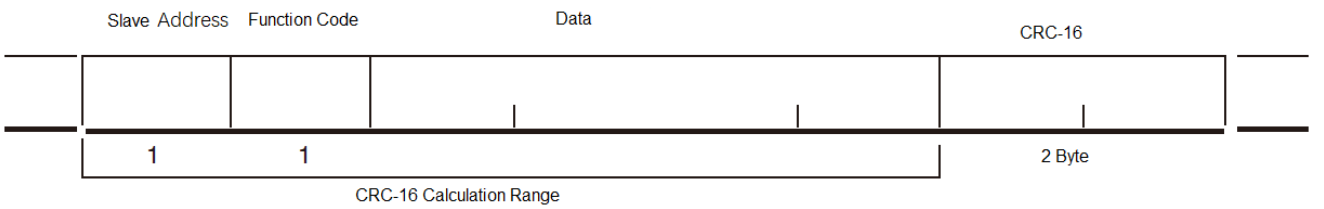
3. Modbus (RTU) Communication Protocol

3.1 Data Format

The instrument follows Modbus (RTU) communication protocol, so the instrument will respond the instruction of the upper computer and return the respond frame.

Instruction Frame

Figure 0-1 Modbus Instruction Frame



Description of Instruction Frame

	It need the squelch interval of 3.5 characters at least.
Slave Station	1 byte Modbus supports the slave station of 00~0x63 Assign to 00 in unified broadcasting
Function Code	1 Byte 0x03: Read multiple registers 0x04: =03H, not use 0x06: Write single register, it can be replace by 10H 0x08: Echo test (only for debugging) 0x10: Write multiple registers
Data	To assign the address, quantity and contents of the registers
CRC-16	2 Bytes, low bit in the front CyclicRedundancy Check Caculating all the data from slave address to the end of data, get CRC16 check code
	It need the squelch interval of 3.5 characters at least.

Respond Frame

Except the command is broadcast from 00H slave address, the instrument will return a response frame for any other slave address.

Figure 0-2 Normal Respond Frame

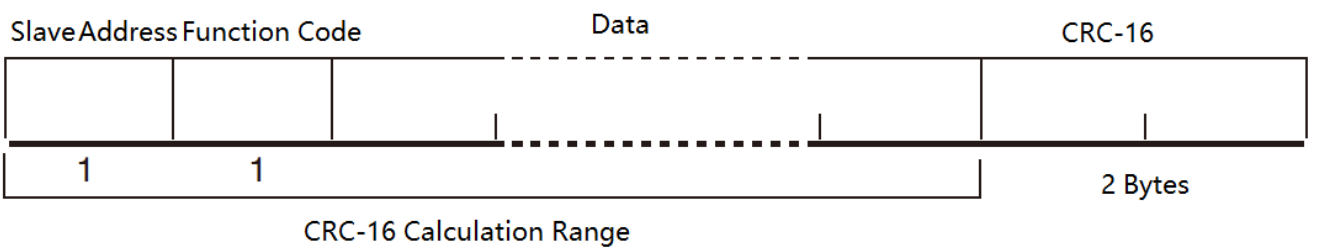
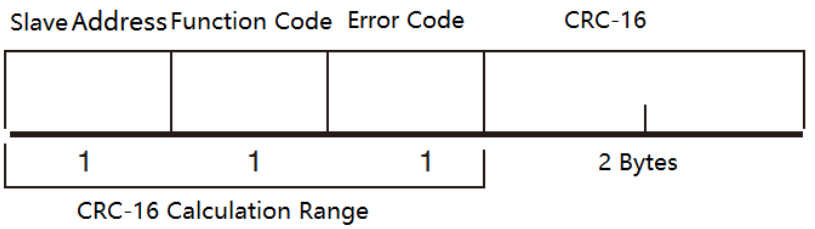


Figure 0-3 Abnormal Respond Frame



Description of Abnormal Respond Frame

Slave Station	1 byte Original return from slave address
Function Code	1 byte Function code logical (OR) BIT7 of command frame (0x80) For Example: 0x03 OR 0x80 = 0x83
Error Code	Exception code 0x01 Function code error (function code is not support) 0x02 Register error (Register is not exist) 0x03 Data error 0x04 Execution error
CRC-16	2 bytes, low bit in the front CyclicRedundancy Check Caculating all the data from slave address to the end of data, get CRC16 check code

No Response

In the following cases,the instrument will not perform any processing and will not respond, resulting in a communication timeout.

1. Slave address error
2. Transmission error
3. CRC-16 error
4. Bit error, For Example: the total bit of function code 0x03 must be 8, and received bit is less than 8 or great than 8 bytes.
5. Slave address is 0x00, it represents the broadcast address, the instrument is no respond.

Error Code

Error Code	Name	Description	Priority
0x01	Function code error	Function code is not exist	1
0x02	Register error	Register is not exist	2
0x03	Data error	Quantity of registers or quantity of bytes error	3
0x04	Execution error	Data is illegal, written data is not in the allowable range	4

3.2 Function Code

The instrument only supports the serval function code in the following table.

Function Code	Name	Description
0x03	Read multiple registers	Read multiple consecutive register data
0x10	Write multiple registers	Write multiple consecutive register data

3.3 Register

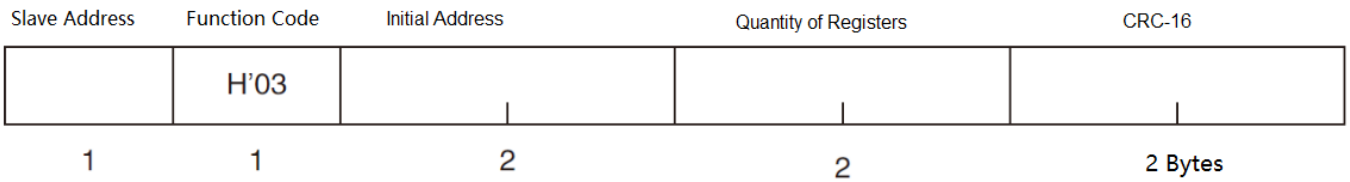
The quantity of registers in the instrument is 2 bytes mode, that is 2 bytes must be written each time. For Example: speed register is 0x3002, data shoud be 2 bytes and the numerical value must be written 0x0001.

Data
The instrument supports the following numerical value.

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

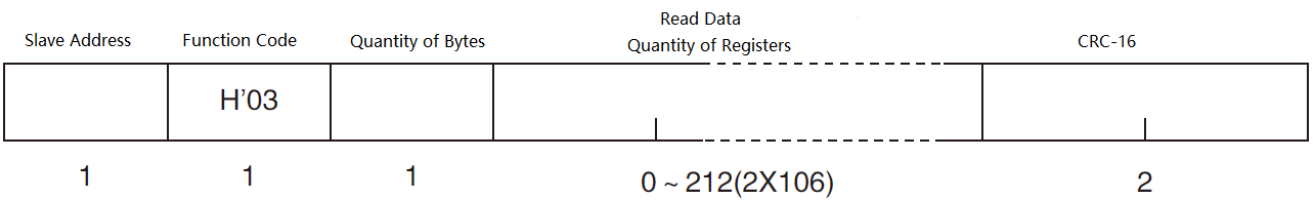
3.4 Read Multiple Registers

Figure 0-4 Read Multiple Registers (0x03)



Name	Field Name	Description
	Slave address	If there no specified address, it defaults to 01
0x03	Function Code	
	Initial Address	Initial address of the register refers to Modbus Command Set
	Read the quantity of registers	The quantity of registers that aree read consecutively, please refers to Modbus Command Set. To make sure these register address are exist, otherwise it will return error frame.
CRC-16	Check Code	

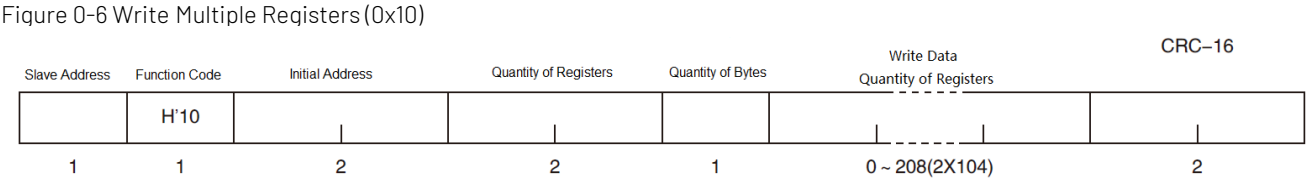
Figure 0-5 Read the respond frame of multiple registers (0x03)



Name	Field Name	Description
	Slave Station	Original return
0x03 or 0x83	Function Code	No abnormal: 0x03 Error code: 0x83
	Quantity of Bytes	Quantity of registers x2 For Example: 1 register returns 02
	Data	Read data
CRC-16	Check Code	

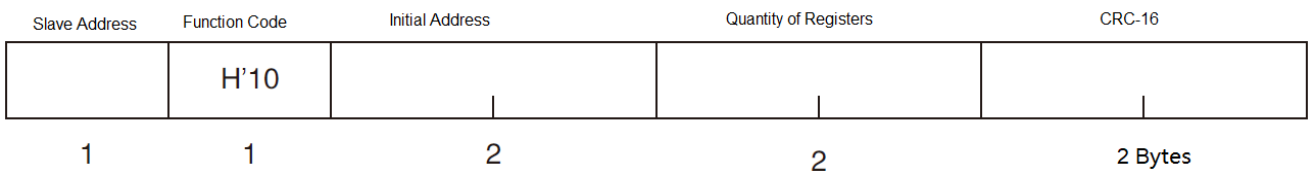
3.5 Write Multiple Registers

Figure 0-6 Write Multiple Registers (0x10)



Name	Field Name	Description
	Slave Station	The default is 01
0x10	Function Code	
	Initial Address	Initial address of the register refers to Modbus Command Set
	Write the quantity of registers 0001~0068（104）	The quantity of registers that are write consecutively, please refers to Modbus Command Set. To make sure these register address are exist, otherwise it will return error frame.
	Quantity of Bytes	= Quantity of registers x2
CRC-16	Check Code	

Figure 0-7 Write the respond frame of multiple registers (0x10)



Name	Field Name	Description
	Slave Station	Original return
0x10 or 0x90	Function Code	No abnormal: 0x10 Error code：0x90
	Initial Address	
	Quantity of Registers	
	CRC-16 Check Code	

4. Modbus(RTU) Command Set

4.1 Register Overview

All registers address used by the instrument are listed in the following table.

Notes
1. Unless other stated, the numerical value of the command and the respond frame are in hexadecimal.
2. Register only contains the command of fetch the test result and start/stop the test. If user want to have other commands, please contact UNI-T sales department.
3. Floating-point on-line conversion can refer to website http://www.binaryconvert.com/convert_float.html

Register Address	Name	Numerical Value	Quantity of Register	Authority
0x0200	Start or stop the power output	2 bytes integer 0x0000：OFF 0x0001：ON	1	Read and Write
0x0201	Read the statud of CV/CC	2 bytes integer 0x0000：CV 0x0001：CC	1	Read Only
0x0202	Read the readback voltage value	4 bytes floating-point number	2	Read Only
0x0204	Read the readback current value	4 bytes floating-point number	2	Read Only
0x0206	Read the readback power value	4 bytes floating-point number	2	Read Only
0x0208	Set the voltage value	4 bytes floating-point number	2	
0x020A	Set the current value	4 bytes floating-point number	2	Read and Write
0x020C	Set the OVP value	4 bytes floating-point number	2	Read and Write
0x020E	Set the OCP value	4 bytes floating-point number	2	Read and Write
0x0210	Set the output time value	4 bytes floating-point number	2	Read and Write
0x0212	Set the statud of OVP	2 bytes integer 0x0000：OFF 0x0001：ON	1	Read and Write
0x0213	Set the statud of OCP	2 bytes integer 0x0000：OFF 0x0001：ON	1	Read and Write
0x0214	Set the statud of the output timer	2 bytes integer 0x0000：OFF 0x0001：ON	1	Read and Write
0x0215	Set the boot output	2 bytes integer 0x0000：OFF 0x0001：ON	1	Read and Write
0x0216	Set the initial group number of the list	2 bytes integer	1	Read and Write
0x0217	Set the output group number of the list	2 bytes integer	1	Read and Write
0x0218	Set the repeat times of the list	2 bytes integer	1	Read and Write
0x0219	Set the stop status of the list	2 bytes integer 0x0000：STOP 0x0001：HOLD	1	Read and Write
0x021A	Set the enable status of the list	2 bytes integer 0x0000：OFF	1	Read and Write

		0x0001：ON		
0x021B	Select the current step of the list	2 bytes integer	1	Read and Write
0x021C	Set the voltage value of the current step in the list	4 bytes floating-point number	2	Read and Write
0x021E	Set the current value of the current step in the list	4 bytes floating-point number	2	Read and Write
0x0220	Set the time value of the current step in the list	4 bytes floating-point number	2	Read and Write
0x0221	Loading the list file	2 bytes integer	1	Write Only
0x0222	Save the list file	2 bytes integer	1	Write Only
0x0223	Delete the list file	2 bytes integer	1	Write Only
0x0224	Boot loading the list file	2 bytes integer	1	Read and Write
0x0225	Save the list file in real time	2 bytes integer	1	Read and Write
0x0226	Set the initial group number of the delayer	2 bytes integer	1	Read and Write
0x0227	Set the output group number of the delayer	2 bytes integer	1	Read and Write
0x0228	Set the repeat times of the delayer	2 bytes integer	1	Read and Write
0x0229	Set the stop status of the delayer	2 bytes integer	1	Read and Write
0x022A	Set the enable status of the delayer	2 bytes integer 0x0000：STOP 0x0001：HOLD	1	Read and Write
0x022B	Set the current step of the delayer	2 bytes integer	1	Read and Write
0x022C	Set the status of the current step in delayer mode	2 bytes integer	1	Read and Write
0x022D	Set the time value of the current step in delayer mode	4 bytes floating-point number	2	Read and Write
0x022F	Load the delayer file	2 bytes integer	1	Write Only
0x0230	Save the delayer file	2 bytes integer	1	Write Only
0x0231	Delete the delayer file	2 bytes integer	1	Write Only
0x0232	Boot loading the delayer file	2 bytes integer	1	Read and Write
0x0233	Save the delayer file in real time	2 bytes integer	1	Read and Write
0x0234	Boot loading the system file	2 bytes integer	1	Read and Write
0x0235	Save the system file	2 bytes integer	1	Write Only
0x0236	Delete the system file	2 bytes integer	1	Write Only
0x0237	Set the boot loading of the system file	2 bytes integer	1	Read and Write
0x0238	Save the system file in real time	2 bytes integer	1	Read and Write
0x0239	Instrument's interface setup	2 bytes integer 0x0000：Measurement display 0x0001：Measurement setup 0x0002：List setup 0x0003：Save List	1	Read and Write

		0x0004：Delayer setup 0x0005：Save Delayer 0x0006：System setup 0x0007：File management		
0x023A	Set the language	2 bytes integer 0x0000：English 0x0001：Chinese	1	Read and Write
0x023B	Year	2 bytes integer	1	Read and Write
0x023C	Month	2 bytes integer	1	Read and Write
0x023D	Date	2 bytes integer	1	Read and Write
0x023E	Time	2 bytes integer	1	Read and Write
0x023F	Minute	2 bytes integer	1	Read and Write
0x0240	Second	2 bytes integer	1	Read and Write
0x0241	Key sound	2 bytes integer	1	Read and Write
0x0242	Query and delete the alarm of OVP	2 bytes integer	1	Read and Write
0x0243	Query and delete the alarm of OCP	2 bytes integer	1	Read and Write

4.2 Query Power Status

Start/Stop Power Output

Write

1	2	3	4	5	6	7	8	9	10	11
01	10	02	00	00	01	02	00	01	44	50
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8~B9:
0x0000: Stop the power output
0x0001: Start the power output

Respond

1	2	3	4	5	6	7	8
01	10	02	00	00	01	00	71
Slave station	Write	Register		Quantity of registers		CRC16	

Query Output Status

Send :

1	2	3	4	5	6	7	8
01	03	02	00	00	01	85	B2
Slave station	Read	Register		Quantity of registers		Check Code	

Respond

1	2	3	4	5	8	9
01	03	02	00	00	B8	44
01	03	Byte	Integer		CRC-16	

B4~B5:
0X0000: the power is stop
0X000: the power is outputting

Query CV/CC Status

Send

1	2	3	4	5	6	7	8
01	03	02	01	00	01	79	84
Slave station	Read	Register		Quantity of registers		Check Code	

Respond

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

B4~B5
0X0000 CV
0X0001 CC

Query Readback Voltage Value

Send

1	2	3	4	5	6	7	8
01	03	02	02	00	02	64	73
Slave station	Read	Register		Quantity of registers		Check Code	

Respond:

1	2	3	4	5	6	7	8	9
01	03	04	41	9F	F3	63	DA	F8
01	03	Byte	Float-point number with single precision				CRC-16	

B4~B7: data of float type, 0x419FF363=19.993841, the current readback voltage value of the instrument is 19.99V

Query Readback Current Value

Send

1	2	3	4	5	6	7	8
01	03	02	04	00	02	84	72
Slave station	Read	Register		Quantity of registers		Check Code	

Respond:

1	2	3	4	5	6	7	8	9
01	03	04	40	9F	E8	64	90	36
01	03	Byte	Float-point number with single precision				CRC-16	

B4~B7: data of float type, 0x409FE864=4.997118, the current readback current value of the instrument is 4.99A

Query Readback Power Value

Send

1	2	3	4	5	6	7	8
01	03	02	06	00	02	25	B2
Slave station	Read	Register		Quantity of registers		Check Code	

Respond:

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	00	FA	33
01	03	Byte	Float-point number with single precision				CRC-16	

B4~B7: data of float type, 0x00000000=0, the current readback power value of the instrument is 0W

4.3 Parameter Setup of Power Output

Voltage Value

This command is invalid if the instrument is operating list or delayer mode.

Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	08	00	02	04	41	20	00	00	FE	9F
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, 0x41200000 = 10, i.e. set the voltage to 10.00V

Respond

1	2	3	4	5	6	7	8
01	10	02	08	00	02	00	71
Slave station	Write	Register		Quantity of registers		CRC16	

Current Value

This command is invalid if the instrument is operating list or delayer mode.

Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	0A	00	02	04	40	A0	00	00	7F	52
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, 0x40A00000 = 5, i.e. set the current to 5.00A

Respond

1	2	3	4	5	6	7	8
01	10	02	0A	00	02	00	71
Slave station	Write	Register		Quantity of registers		CRC16	

OVP Value

This command is invalid if the instrument is operating list or delayer mode.

Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	0C	00	02	04	41	A0	00	00	FE	84
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, 0x41A00000 = 20, i.e. set the OVP to 20.00V

Respond

1	2	3	4	5	6	7	8
01	10	02	0A	00	02	00	71
Slave station	Write	Register		Quantity of registers		CRC16	

OCP Value

This command is invalid if the instrument is operating list or delayer mode.

Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	0E	00	02	04	41	A0	00	00	7F	5D
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, 0x41A00000 = 20, i.e. set the OCP to 20.00A

Respond

1	2	3	4	5	6	7	8
01	10	02	0E	00	02	21	B3
Slave station	Write	Register		Quantity of registers		CRC16	

Output value of Timing

This command is invalid if the instrument is operating the fixed output.

Write

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	10	00	02	04	41	A0	00	00	FF	DD
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, 0x41A00000 = 20, output value of time setting is 20s

Respond

1	2	3	4	5	6	7	8
01	10	02	10	00	02	41	B5
Slave station	Write	Register		Quantity of registers		CRC16	

Set the status of OCP

This command is invalid if the instrument is operating list or delayer mode.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	13	00	01	02	00	01	46	F3
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8~B9:
0x0000: Stop OCP
0x0001: Start OCP

Respond

1	2	3	4	5	6	7	8
01	10	02	13	00	01	F1	B4
Slave station	Write	Register		Quantity of registers		CRC16	

Set the status of OVP

This command is invalid if the instrument is operating list or delayer mode.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	12	00	01	02	00	01	47	22
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8~B9:
0x0000: Stop OVP
0x0001: Start OVP

Respond

1	2	3	4	5	6	7	8
01	10	02	12	00	01	A0	74
Slave station	Write	Register		Quantity of registers		CRC16	

Set the Status of Output Timer

This command is invalid if the instrument is operating the fixed output.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	14	00	01	02	00	01	47	44
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8~B9:
0x0000: Stop the timing output
0x0001: Start the timing output

Respond

1	2	3	4	5	6	7	8
01	10	02	14	00	01	40	74
Slave station	Write	Register		Quantity of registers		CRC16	

Set the Status of Boot Output

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	15	00	01	02	00	01	46	95
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8~B9:
0x0000: Stop the boot output
0x0001: Start the boot output

Respond

1	2	3	4	5	6	7	8
01	10	02	15	00	01	11	B5
Slave station	Write	Register		Quantity of registers		CRC16	

Query Whether is OVP

Send

1	2	3	4	5	6	7	8
01	03	02	42	00	01	25	A6
Slave station	Read	Register		Quantity of registers		Check Code	

Respond

1	2	3	4	5	8	9
01	03	02	00	00	B8	44
01	03	Byte	Integer		CRC-16	

B4~B5
0X0000: not has OVP
0X0001: has OVP

Clear the alarm of OVP

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	42	00	01	02	00	01	4B	72
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

Respond

1	2	3	4	5	6	7	8
01	10	02	42	00	01	A0	65
Slave station	Write	Register		Quantity of registers		CRC16	

Query Whether is OCP

Send

1	2	3	4	5	6	7	8
01	03	02	43	00	01	74	66
Slave station	Read	Register		Quantity of registers		Check Code	

Respond

1	2	3	4	5	8	9
01	03	02	00	00	B8	44
01	03	Byte	Integer		CRC-16	

B4~B5
0X0000: not has OCP
0X0001: has OCP

Clear the alarm of OCP

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	43	00	01	02	00	01	4A	A3
Station number	Writ e	Register		Quantity of registers		Byte	Integer		CRC16	

Respond

1	2	3	4	5	6	7	8
01	10	02	43	00	01	F1	A5
Slave station	Write	Register		Quantity of registers		CRC16	

4.4 List Parameter Setup

This subsystem command is invalid if the instrument is operating the list output. And when the instrument is operating the delayer output, enable list and LOAD File command is invalid.

Initial Group Number

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	16	00	01	02	00	02	06	A7
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0X0002, set the initial group number to 2

Respond

1	2	3	4	5	6	7	8
01	10	02	16	00	01	E1	B5
Slave station	Write	Register		Quantity of registers		CRC16	

Output Group Number

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	17	00	01	02	00	02	07	76

Station number	Write	Register	Quantity of registers	Byte	Integer	CRC16

B8B9=0X0002, set the output group number to 2

Respond

1	2	3	4	5	6	7	8
01	10	02	17	00	01	B0	75
Slave station	Write	Register		Quantity of registers		CRC16	

Repeat Times

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	18	00	01	02	00	01	47	88
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0X0001, set the repeat times to 1

Respond

1	2	3	4	5	6	7	8
01	10	02	18	00	01	80	76
Slave station	Write	Register		Quantity of registers		CRC16	

Stop Status

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	19	00	01	02	00	00	87	99
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0x0000: STOP
0X0001: HOLD

Respond

1	2	3	4	5	6	7	8
01	10	02	19	00	01	D1	B6
Slave station	Write	Register		Quantity of registers		CRC16	

Enable List Function

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	1A	00	01	02	00	01	87	99
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0x0000: Stop the list function
0X0001: Enable the list function

Respond

1	2	3	4	5	6	7	8
01	10	02	1A	00	01	21	B6
Slave station	Write	Register		Quantity of registers		CRC16	

Select the step of List

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	1B	00	01	02	00	01	47	BB
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0X0001, select the first step in the list

Respond

1	2	3	4	5	6	7	8
01	10	02	1B	00	01	70	76
Slave station	Write	Register		Quantity of registers		CRC16	

Set Voltage Value of the Step

Send

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	1C	00	02	04	41	A0	00	00	FF	88
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, =0X41A00000=20, i.e. set the voltage value of the first step in the list to 20.00V
The current step is decided by the register of 0x021B.

Respond

1	2	3	4	5	6	7	8
01	10	02	1C	00	02	81	B6
Slave station	Write	Register		Quantity of registers		CRC16	

Set Current Value of the Step

Send

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	1E	00	02	04	41	A0	00	00	7E	51
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, =0X41A00000=20, i.e. set the current value of the first step in the list to 20.00A
The current step is decided by the register of 0x021B.

Respond

1	2	3	4	5	6	7	8
01	10	02	1E	00	02	20	76
Slave station	Write	Register		Quantity of registers		CRC16	

Set Time Value of the Step

Send

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	20	00	02	04	41	A0	00	00	FC	C9
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, =0X41A00000=20, i.e. set the time value of the first step in the list to 20.00s

The current step is decided by the register of 0x021B.

Respond

1	2	3	4	5	6	7	8
01	10	02	20	00	02	41	BA
Slave station	Write	Register		Quantity of registers		CRC16	

Set Voltage, Current, Time Value for the specified Step

Send

01	10	02	1B	00	07	0E	00	01	41	A0	00	00	41	A0	00	00	41	A0	00	00	87	A3
		Register		Quantity of registers		Byte	Integer		Float-point number with single precision				Float-point number with single precision				Float-point number with single precision				CRC-16	
							Step		Voltage value				Current value				Time value					

Set the parameter of the first step: the voltage value to 20V, the current value to 20A, the time value to 20s

Respond

1	2	3	4	5	6	7	8
01	10	02	1B	00	07	F0	74
Slave station	Write	Register		Quantity of registers		CRC16	

Loading List File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	21	00	01	02	00	01	42	E1
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	21	00	01	50	7B
Slave station	Write	Register		Quantity of registers		CRC16	

Save List File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	22	00	01	02	00	01	42	D2
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	22	00	01	A0	7B
Slave station	Write	Register		Quantity of registers		CRC16	

Delete List File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	23	00	01	02	00	01	43	03
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	23	00	01	F1	BB
Slave station	Write	Register		Quantity of registers		CRC16	

Boot Loading

Set the corresponding file to boot loading. Only one file can be set to boot loading at the same time. For Example, file 1 is boot loading, if file 2 sets to boot loading, the boot loading of file 1 will be cancelled, if the boot load file is deleted, then file 0 will be set to boot loading.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	24	00	01	02	00	01	42	B4
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	24	00	01	40	7A
Slave station	Write	Register		Quantity of registers		CRC16	

Auto Save

Set the automatic function for boot loading file. If the automatic function is enabled, the parameter set by manual will immediately save in the corresponding file.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	25	00	01	02	00	01	43	65
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0000: disable the automatic function
0001: enable the automatic function

Respond

1	2	3	4	5	6	7	8
01	10	02	25	00	01	11	BA
Slave station	Write	Register		Quantity of registers		CRC16	

4.5 Delayer Parameter Setup

This subsystem command is invalid if the instrument is operating the delayer output. And when the instrument is operating the list output, enable delayer and LOAD File command is invalid.

Initial Group Number

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	26	00	01	02	00	02	82	96
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0X0002, set the initial group number to 2

Respond

1	2	3	4	5	6	7	8
01	10	02	26	00	01	E1	BA
Slave station	Write	Register		Quantity of registers		CRC16	

Output Group Number

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	27	00	01	02	00	02	02	86
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0X0002, set the output group number to 2

Respond

1	2	3	4	5	6	7	8
01	10	02	27	00	01	B0	7A
Slave station	Write	Register		Quantity of registers		CRC16	

Repeat Times

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	28	00	01	02	00	01	42	78
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0X0001, set the repeat times to 1

Respond

1	2	3	4	5	6	7	8
01	10	02	28	00	01	80	79
Slave station	Write	Register		Quantity of registers		CRC16	

Stop Status

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	29	00	01	02	00	00	82	69
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0x0000: STOP
0X0001: HOLD

Respond

1	2	3	4	5	6	7	8
01	10	02	29	00	01	D1	B9
Slave station	Write	Register		Quantity of registers		CRC16	

Enable Delayer Function

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2A	00	01	02	00	01	43	9A
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0x0000: Stop the delayer function
0X0001: Start the delayer function

Respond

1	2	3	4	5	6	7	8
01	10	02	2A	00	01	21	B9
Slave station	Write	Register		Quantity of registers		CRC16	

Select the step of Delayer

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2B	00	01	02	00	01	42	4B
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0X0001, select the first step in the list

Respond

1	2	3	4	5	6	7	8
01	10	02	2B	00	01	70	79
Slave station	Write	Register		Quantity of registers		CRC16	

Set the status of Step

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2C	00	01	02	00	01	42	4B
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9=0x0001, enable the output of the step
0x0000: OFF
0X0001: ON
The current step is decided by the register of 0x022B.

Respond

1	2	3	4	5	6	7	8
01	10	02	2C	00	01	C1	B8
Slave station	Write	Register		Quantity of registers		CRC16	

Set Time Value of the Step

Send

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	02	2D	00	02	04	41	A0	00	00	FC	C9
Station number	Write	Register		Quantity of registers		Byte	Float-point number with single precision				CRC16	

B8~B11 is float-point number with single precision, =0X41A00000=20, i.e. set the time value of the first step in the list to 20.00s
The current step is decided by the register of 0x022B.

Respond

1	2	3	4	5	6	7	8
01	10	02	20	00	02	41	BA
Slave station	Write	Register		Quantity of registers		CRC16	

Set the Status and Time Value of the Specified Step

Send

01	10	02	2B	00	04	08	00	01	00	01	41	A0	00	00	87	A3
		Register		Quantity of registers		Byte	Integer		Integer		Float-point number with single precision				CRC-16	
							Step		Status		Time					

Enable the setting in step 1 to output, time value is 20s

Respond

1	2	3	4	5	6	7	8
01	10	02	2B	00	04	B0	7A
Slave station	Write	Register		Quantity of registers		CRC16	

Loading Delayer File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2F	00	01	02	00	01	43	CF
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	2F	00	01	31	B8
Slave station	Write	Register		Quantity of registers		CRC16	

Save Delayer File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	30	00	01	02	00	01	41	A0
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	30	00	01	00	7E
Slave station	Write	Register		Quantity of registers		CRC16	

Delete Delayer File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	31	00	01	02	00	01	40	71
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	31	00	01	51	BE
Slave station	Write	Register		Quantity of registers		CRC16	

Boot Loading

Set the corresponding file to boot loading. Only one file can be set to boot loading at the same time. For Example, file 1 is boot loading, if file 2 sets to boot loading, the boot loading of file 1 will be cancelled, if the boot load file is deleted, then file 0 will be set to boot loading.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	32	00	01	02	00	01	40	42
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	32	00	01	A1	BE
Slave station	Write	Register		Quantity of registers		CRC16	

Auto Save

Set the automatic function for boot loading file. If the automatic function is enabled, the parameter set by manual will immediately save in the corresponding file.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	33	00	01	02	00	01	41	93
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0000: disable the automatic save
0001: enable the automatic save

Respond

1	2	3	4	5	6	7	8
01	10	02	33	00	01	F0	7E
Slave station	Write	Register		Quantity of registers		CRC16	

4.6 File Setup

Save the parameter setup except the parameter of the list and the delayer.

Loading File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	34	00	01	02	00	01	40	24
Station number	Writ e	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	34	00	01	41	BF
Slave station	Write	Register		Quantity of registers		CRC16	

Save File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	35	00	01	02	00	01	41	F5
Station number	Writ e	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	35	00	01	10	7F
Slave station	Write	Register		Quantity of registers		CRC16	

Delete File

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	36	00	01	02	00	01	41	C6
Station number	Writ e	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	36	00	01	E0	7F
Slave station	Write	Register		Quantity of registers		CRC16	

Boot Loading

Set the corresponding file to boot loading. Only one file can be set to boot loading at the same time. For Example, file 1 is boot loading, if file 2 sets to boot loading, the boot loading of file 1 will be cancelled, if the boot load file is deleted, then file 0 will be set to boot loading.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	37	00	01	02	00	01	40	17
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: internal file number, it not support USB file

Respond

1	2	3	4	5	6	7	8
01	10	02	37	00	01	B1	BF
Slave station	Write	Register		Quantity of registers		CRC16	

Auto Save

Set the automatic function for boot loading file. If the automatic function is enabled, the parameter set by manual will immediately save in the corresponding file.

Send

1	2	3	4	5	6	7	8	9	10	11
01	10	02	38	00	01	02	00	01	40	E8
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0000: disable the automatic save
0001: enabe the automatic save

Respond

1	2	3	4	5	6	7	8
01	10	02	38	00	01	81	BC
Slave station	Write	Register		Quantity of registers		CRC16	

4.7 System Setup

Page Switching

1	2	3	4	5	6	7	8	9	10	11
01	10	02	39	00	01	02	00	01	80	F9
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9：
0X0000: Measurement Display
0X0001: Measurement Setup
0X0002: List Setup
0X0003: List Save
0X0004: Delayer Setup
0X0005: List Save
0X0006: System Setup
0X0007: File Management

Respond

1	2	3	4	5	6	7	8
01	10	02	39	00	01	D0	7C
Slave station	Write	Register		Quantity of registers		CRC16	

Language Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3A	00	01	02	00	00	80	F9
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0X0000: ENGLISH
0X0001: CHINESE(Simplified)

Respond

1	2	3	4	5	6	7	8
01	10	02	3A	00	01	20	7C
Slave station	Write	Register		Quantity of registers		CRC16	

Year Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3B	00	01	02	00	17	C1	15
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: 0x0017 = 23, set year to 2023

Respond

1	2	3	4	5	6	7	8
01	10	02	3B	00	01	71	BC
Slave station	Write	Register		Quantity of registers		CRC16	

Month Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3C	00	01	02	00	08	81	7D
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: 0X0008 = 08, set month to August

Respond

1	2	3	4	5	6	7	8
01	10	02	3C	00	01	C0	7D
Slave station	Write	Register		Quantity of registers		CRC16	

Day Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3D	00	01	02	00	08	80	BB
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: 0X0008=8, set day to 8

Respond

1	2	3	4	5	6	7	8
01	10	02	3D	00	01	91	BD
Slave station	Write	Register		Quantity of registers		CRC16	

Time Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3E	00	01	02	00	08	80	88
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: 0X0008=8, set time to 8

Respond

1	2	3	4	5	6	7	8
01	10	02	3E	00	01	20	7C
Slave station	Write	Register		Quantity of registers		CRC16	

Minute Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3F	00	01	02	00	1E	80	F9
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: 0X001E = 30, set minute to 30 minutes

Respond

1	2	3	4	5	6	7	8
01	10	02	3F	00	01	20	7C
Slave station	Write	Register		Quantity of registers		CRC16	

Second Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	40	00	01	02	00	00	8B	50
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9: 0X0000=0, set second to 0

Respond

1	2	3	4	5	6	7	8
01	10	02	40	00	01	01	A5
Slave station	Write	Register		Quantity of registers		CRC16	

Key Sound Setting

1	2	3	4	5	6	7	8	9	10	11
01	10	02	41	00	01	02	00	00	8A	81
Station number	Write	Register		Quantity of registers		Byte	Integer		CRC16	

B8B9
0X0000: OFF

0X0001: ON

Respond

1	2	3	4	5	6	7	8
01	10	02	41	00	01	50	65
Slave station	Write	Register		Quantity of registers		CRC16	