

UDP6722直流电源

编程手册

REV 00
2023. 04

UNI-T[®]

保证和声明

版权

优利德中国科技有限公司

商标信息

UNI-T是优利德中国科技有限公司的注册商标。

文档编号

声明

- 公司产品受中国及其它国家和地区的专利（包括已取得的和正在申请的专利）保护。
- 本公司保留改变规格及价格的权利。
- 本手册提供的信息取代以往出版的所有资料。
- 本手册提供的信息如有变更，恕不另行通知。
- 对于本手册可能包含的错误，或因手册所提供的信息及演绎的功能以及因使用本手册而导致的任何偶然或继发的损失，UNI-T概不负责。
- 未经 UNI-T事先书面许可，不得影印、复制或改编本手册的任何部分。
-

产品认证

UNI-T认证本产品符合中国国家产品标准和行业产品标准及 ISO9001：2008 标准和 ISO14001：2004 标准，并进一步认证本产品符合其它国际标准组织成员的相关标准。

1 SCPI 指令简介

SCPI (Standard Commands for Programmable Instruments, 即可编程仪器标准命令集) 是一种建立在现有标准 IEEE 488.1 和 IEEE 488.2 基础上, 并遵循了 IEEE754 标准中浮点运算规则、ISO646 信息交换 7 位编码符号 (相当于 ASCII 编程) 等多种标准的标准化仪器编程语言。本节简介 SCPI 命令的格式、符号、参数和缩写规则。

1.1 命令串解析

主机可以发送一串命令给仪器, 仪器命令解析器在捕捉到结束符 (\n) 或输入缓冲区溢出后开始解析。

例如:

合法的命令串:

```
AAA:BBB CCC;DDD EEE;:FFF
```

仪器命令解析器负责所有命令解析和执行, 在编写程序前您必须首先对其解析规则有所了解。

1.2 命令解析规则

命令解析器只对 ASCII 码数据进行解析和响应。

在收到结束符后开始命令解析。本仪器接受以下内容作为结束符。

CR+LF

命令解析器在解析到错误后, 立即终止解析, 当前指令作废。

命令解析器对命令串的解析不区分大小写。

命令解析器支持命令缩写形式, 缩写规格参见之后章节。

RS485 模式 SCPI 协议前方加 ADDR□ 本机地址::□, 本机地址可设为 1-32。

方便多机通过 SCPI 协议进行通讯。

例如: ADDR□1::□IDN? □表示一个空格

仪器发送的数据 结束符默认为 0x0D 0x0A (CRLF)。

通过分号 ; 可以进行多指令发送。

1.3 符号约定和定义

本章使用了一些符号, 这些符号并不是命令树的一部分, 只是为了能更好的对命令串的理解。

<> 尖括号中的文字表示该命令的参数

[] 方括号中的文字表示可选命令

{ } 当大括号包含几个参数项目时, 表示只能从中选择一个项目。

() 参数的缩写形式放在小括号中。

大写字母 命令的缩写形式。

□ 空格字符, 表示一个空格, 仅用于阅读需要。

1.4 命令树结构

对 SCPI 命令采用树状结构的, 可向下三级 (注: 此仪器的命令解析器可向下解析任意层), 在这里最高级称为子系统命令。只有选择了子系统命令, 该其下级命令才有效, SCPI 使用冒号 (:) 来分隔高级命令和低级命令。

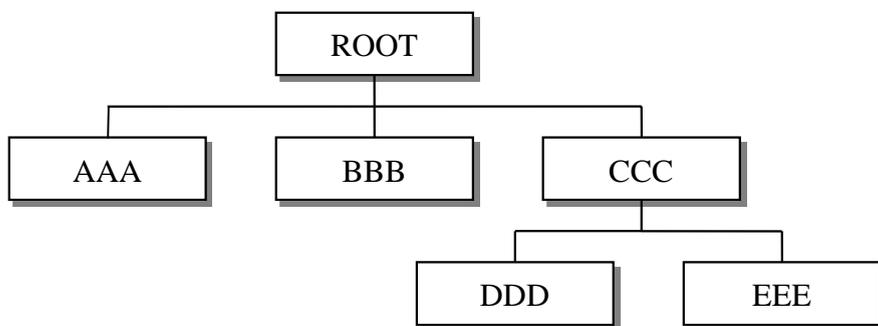


图 1-1 命令树结构

举例说明

ROOT:CCC:DDD ppp

ROOT	子系统命令
CCC	第二级
DDD	第三级
ppp	参数

1.5 命令和参数

一条命令树由命令和[参数]组成，中间用 1 个空格（ASCII：20H）分隔。

举例说明 AAA:BBB 1. 234

命令 [参数]

1.6 命令

命令字可以是长命令格式或缩写形式，使用长格式便于工程师更好理解命令串的含义；缩写形式适合书写。

1.7 参数

单命令字命令，无参数。

例如：AAA:BBB

参数可以是字符串形式，其缩写规则仍遵循上节的“命令缩写规则”。如：AAA:BBB 1. 23

参数可以是数值形式

<integer> 整数 123, +123, -123

<float> 浮点数

1. <fixfloat>: 定点浮点数: 1. 23, -1. 23

2. <Sciloat>: 科学计数法浮点数: 1. 23E+4, +1. 23e-4

3. <mpfloat>: 倍率表示的浮点数: 1. 23k, 1. 23M, 1. 23G, 1. 23u 表 0-1 倍率缩写

数值	倍率
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

提示：倍率不区分大小写，其写法与标准名称不同。

1.8 分隔符

仪器命令解析器只接收允许的分隔符，除此之外的分隔符命令解析器将产生“Invalid separator (非法分割符)”错误。这些分隔符包括：

； 分号，用于分隔两条命令。

例如：AAA:BBB 100.0 ； CCC:DDD

： 冒号，用于分隔命令树，或命令树重启动。

例如：AAA : BBB : CCC 123.4; : DDD : EEE 567.8

? 问号，用于查询。

例如：AAA ?

□ 空格，用于分隔参数。

例如：AAA:BBB□1.234

2 SCPI 命令参考

所有命令都是按子系统命令顺序进行解释，下面列出了所有子系统

DISPly	显示子系统
SYSTem	系统子系统
OUTPut	输出子系统
CURRent	电流设置子系统
VOLTage	电压设置子系统
APPLy	电压电流设置子系统
MEASure	获取结果子系统
FETCH?	获取结果子系统
LIST	列表设置子系统
DELAyer	延时器设置子系统
FILE	文件子系统

公共命令：

IDN?	仪器信息查询子系统
------	-----------

2.1 DISPlay 显示子系统

DISPlay 子系统可以用来切换不同的显示页面。

图 1-1 DISPlay 子系统树

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

DISPlay:PAGE

DISP:PAGE 用来切换到指定页面。

命令语法	DISPlay:PAGE <页面名称>	
参数	<页面名称> 包括：	
	MEAS	测量显示页
	MSET	测量设置页
	LIST	列表设置页
	LISTFile	列表文件页
	DELA	延时器设置页
	DELAFile	延时器文件页
	SYST	系统设置页
	FILE	文件管理页
例如	发送>disp:page mset //切换到设置页面	
查询语法	DISPlay:PAGE?	
查询响应	<页面名称>	
	MEAS	测量显示页
	MSET	测量设置页
	LIST	列表设置页
	LISTFILE	列表文件页
	DELA	延时器设置页
	DELAFILE	延时器文件页
	SYST	系统设置页
	FILE	文件管理页
例如	发送>DISP:PAGE? 返回> MEAS	

2.2 SYSTem 子系统

SYSTem 子系统命令主要用于系统参数设置。



注意：
SYSTem 子系统设置的参数不会自动存储到文件中，
设置好参数后，需要调用 FILE 子系统进行保存到机内文件中。

SYSTem 子系统树：

SYSTem	:LANGuage	{ENGLISH, CHINESE, EN, CN}	语言设置
	:TIME	<YEAR>, <MONTH>, <DAY>, <HOUR>, <MINUTE>, <SECOND>	日期时间设置
	:KEYSound	{OFF, ON, 0, 1}	按键声音设置

SYSTem:LANGuage 系统语言设置

命令语法	SYSTem:LANGuage {ENGLISH, CHINESE, EN, CN}
例如	发送>SYST:LANG EN //设置为英文显示
查询语法	SYST:LANG?
查询响应	{ENGLISH, CHINESE}

SYSTem:TIME 设置日期时间

命令语法	SYSTem:TIME <YEAR>, <MONTH>, <DAY>, <HOUR>, <MINUTE>, <SECOND>
例如	发送>SYST:TIME 2022, 1, 17, 11, 15, 20 //2022-1-17 11:15:20
查询语法	SYSTem:TIME?
查询响应	<YEAR>-<MONTH>-<DAY> <HOUR>:<MINUTE>:<SECOND>
例如	发送>SYST:TIME? 返回>2023-04-01 09:30:49

SYSTem:KEYSound 设置按键音

命令语法	SYSTem:KEYSound {OFF, ON, 0, 1}
例如	发送>SYST:KEYS ON //按键音打开
查询语法	SYSTem:KEYSound?
查询响应	{OFF, ON}

2.3 OUTPut 子系统

OUTPut 子系统命令主要用于启动停止输出。

OUTPut 子系统树：

OUTPut	{OFF, ON, 0, 1}		语言设置	
	:CVCC?		查询 CVCC	
	:TImer	{OFF, ON, 0, 1}	设置输出定时器状态	
	:TImer	:DATA	<float>	设置输出定时器数值
	:POUT	{OFF, ON, 0, 1}	设置开机输出	

OUTPut 启动电源输出

命令语法	OUTPut {OFF, ON, 0, 1}
例如	发送>OUTP ON //启动电源输出
查询语法	OUTPut?
查询响应	{OFF, ON}

OUTPut:CVCC 查询电源 CV CC 状态

命令语法	OUTPut:CVCC?
查询响应	{CV, CC}

OUTPut:TIMer 设置输出定时器状态

当电源已经定数输出时，本指令无效。

命令语法	OUTPut:TIMer {OFF, ON, 0, 1}
例如	发送>OUTP:TIM ON //使能输出定时器
查询语法	OUTPut:TIMer?
查询响应	{OFF, ON}

OUTPut:TIMer:DATA 设置输出定时器数值

当电源已经定数输出时，本指令无效。

命令语法	OUTPut:TIMer:DATA <float>
例如	发送>OUTP:TIM:DATA 10.1
查询语法	OUTPut:TIM:DATA?
查询响应	<float>

OUTPut:TIMer:POUT 设置开机输出状态

命令语法	OUTPut:POUT {OFF, ON, 0, 1}
例如	发送>OUTP:POUT OFF
查询语法	OUTPut:POUT?
查询响应	{OFF, ON}

2.4 [SOURce:]CURRent 子系统

CURRent 子系统用来设置输出电流相关参数，当电源工作在列表或延时器模式时，本子系统内的指令无效。

CURRent 子系统树：

[SOURce:]CURRent	{MINimum, MAXimum, DEFault, <float>}		设置电流值
	:PROTection	<float>	设置过流保护值
	:PROTection	:STATe {OFF, ON, 0, 1}	设置过流保护状态
	:PROTection	TRIPed?	查询是否过流保护
	:PROTection	CLear	清除过流保护状态

[SOURCE:]CURRENT 设置输出电流值

命令语法	[SOURCE:]CURRENT {MINimum, MAXimum, DEFault, <float>}
例如	发送>CURR 5.1
查询语法 1	[SOURCE:]CURRENT?
查询语法 2	[SOURCE:]CURRENT? {MINimum, MAXimum, DEFault}
查询响应	<float>

[SOURCE:]CURRENT:PROTECTION 设置过流保护值

命令语法	[SOURCE:]CURRENT:PROTECTION {MINimum, MAXimum, <float>}
例如	发送>CURR:PROT 10.1
查询语法 1	[SOURCE:]CURRENT:PROTECTION?
查询语法 2	[SOURCE:]CURRENT:PROTECTION? {MINimum, MAXimum}
查询响应	<float>

[SOURCE:]CURRENT:PROTECTION:STATE 设置过流保护状态

命令语法	[SOURCE:]CURRENT:PROTECTION:STATE {OFF, ON, 0, 1}
例如	发送>CURR:PROT:STAT OFF
查询语法	[SOURCE:]CURRENT:PROTECTION:STATE?
查询响应	{OFF, ON}

[SOURCE:]CURRENT:PROTECTION:TRIPed? 设置过流保护状态

命令语法	[SOURCE:]CURRENT:PROTECTION:TRIPed?
例如	发送>CURR:PROT:TRIP?
查询响应	{0, 1} //0:未过流保护, 1:过流保护

[SOURCE:]CURRENT:PROTECTION:CLEAR 清除过流保护提示

命令语法	[SOURCE:]CURRENT:PROTECTION:CLEAR
例如	发送>CURR:PROT:CLEAR

2.5 [SOURCE:]VOLTage 子系统

VOLTage 子系统用来设置输出电压相关参数，当电源工作在列表或延时器模式时，本子系统内的指令无效。

VOLTage 子系统树：

[SOURCE:]VOLTage	{MINimum, MAXimum, DEFault, <float>}		设置电压值	
	:PROTECTION	<float>	设置过压保护值	
	:PROTECTION	:STATE	{OFF, ON, 0, 1}	设置过压保护状态
	:PROTECTION	:TRIPed?		查询是否过压保护
	:PROTECTION	:CLEAR		清除过压保护提示

[SOURce:]VOLTage 设置输出电压值

命令语法	[SOURce:]VOLTage {MINimum, MAXimum, DEFault, <float>}
例如	发送>VOLT 5.1
查询语法 1	[SOURce:]VOLTage?
查询语法 2	[SOURce:]VOLTage? {MINimum, MAXimum, DEFault}
查询响应	<float>

[SOURce:]VOLTage:PROTection 设置过压保护值

命令语法	[SOURce:]VOLTage:PROTection {MINimum, MAXimum, <float>}
例如	发送>VOLT:PROT 10.1
查询语法 1	[SOURce:]VOLTage:PROTection?
查询语法 2	[SOURce:]VOLTage:PROTection? {MINimum, MAXimum}
查询响应	<float>

[SOURce:]VOLTage:PROTection:STATe 设置过压保护状态

命令语法	[SOURce:]VOLTage:PROTection:STATe {OFF, ON, 0, 1}
例如	发送>VOLT:PROT:STAT OFF
查询语法	[SOURce:]VOLTage:PROTection:STATe?
查询响应	{OFF, ON}

[SOURce:]VOLTage:PROTection:TRIPed? 查询是否过压保护

命令语法	[SOURce:]VOLTage:PROTection:TRIPed?
例如	发送>VOLT:PROT:TRIP?
查询响应	{0, 1} //0:未过压保护, 1:过压保护

[SOURce:]VOLTage:PROTection:CLEAr 清除过压保护提示

命令语法	[SOURce:]VOLTage:PROTection:CLEAr
例如	发送>VOLT:PROT:CLE

2.6 [SOURce:]APPLy 子系统

APPLy 子系统用来同时设置电压电流相关参数，当电源工作在列表或延时器模式时，本子系统内的指令无效。

APPLy 子系统树：

APPLy	{MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, DEFault, <float>}	设置电压电流值
	:ALL {MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, <float>}, {MINimum, MAXimum, <float>}	设置电压电流过压过流值

[SOURce:]APPLy 设置电压电流值

命令语法	[SOURce:]APPLy {MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, DEFault, <float>} //电压, 电流
例如	发送>APPL 80, 5 //电压 80V, 电流 5A
查询语法 1	[SOURce:]APPLy?
查询语法 2	[SOURce:]APPLy? {MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, DEFault, <float>}
查询响应	<float>, <float>
例如 1	发送>APPL? 返回>80, 5 //电压 80V, 电流 5A
例如 2	发送>APPL? MAX, MAX 返回>85.00, 20.5 //最大电压 80V, 最大电流 20.5A

[SOURce:]APPLy:ALL 设置电压电流过压过流值

命令语法	[SOURce:]APPLy:ALL {MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, <float>}, {MINimum, MAXimum, <float>}
例如	发送>APPL:ALL 80, 5, 85, 20 //电压 80V, 电流 5A, 过压 85V, 过流 20A
查询语法 1	[SOURce:]APPLy:ALL?
查询语法 2	[SOURce:]APPLy? {MINimum, MAXimum, DEFault}, {MINimum, MAXimum, DEFault}, {MINimum, MAXimum}, {MINimum, MAXimum}
查询响应	<float>, <float>, <float>, <float> //电压, 电流, 过压, 过流

2.7 [SOURce:]APPLy 子系统

APPLy 子系统用来同时设置电压和电流值

命令语法	[SOURce:]APPLy {MINimum, MAXimum, DEFault, <float>}, {MINimum, MAXimum, DEFault, <float>}
例如	发送> APPL 80, 5 //电压 80V, 电流 5A
查询语法 1	[SOURce:]APPLy?
查询语法 2	[SOURce:]APPLy? {MINimum, MAXimum, DEFault}, {MINimum, MAXimum, DEFault}
查询响应	<float>, <float> //电压, 电流

2.8 MEASure 子系统

MEASure 子系统用来查询回读数据

MEASure 子系统树:

MEASure	[:VOLTage]?	查询回读电压值
	:CURRent?	查询回读电流值
	:POWer?	查询回读功率值
	:ALL?	查询所有回读值

MEASure[:VOLTage]? 查询回读电压值

查询语法	MEASure[:VOLTage]?
查询响应	<float>

MEASure:CURRent? 查询回读电流值

查询语法	MEASure:CURRent?
查询响应	<float>

MEASure:POWer? 查询回读功率值

查询语法	MEASure:POWer?
查询响应	<float>

MEASure:ALL? 查询回读电压电流功率值

查询语法	MEASure:ALL?
查询响应	<float>, <float>, <float> //电压、电流、功率

2.9 FETCh 子系统

FETCh 子系统用来查询回读数据等同于 MEASure 子系统

FETCh 子系统树:

FETCh	[:VOLTage]?	查询回读电压值
	:CURRent?	查询回读电流值
	:POWer?	查询回读功率值
	:ALL?	查询所有回读值

FETCh[:VOLTage]? 查询回读电压值

查询语法	FETCh[:VOLTage]?
查询响应	<float>

FETCh:CURRent? 查询回读电流值

查询语法	FETCh:CURRent?
查询响应	<float>

FETCh:POWer? 查询回读功率值

查询语法	FETCh:POWer?
查询响应	<float>

FETCh:ALL? 查询回读电压电流功率值

查询语法	FETCh:ALL?
查询响应	<float>, <float>, <float> //电压、电流、功率

2.10 LIST 列表子系统

LIST 子系统用于列表输出的相关设置，当电源在列表输出时，本子系统的指令无效。当电源在延时器输出时，FUNCTION 和 LOAD 指令无效。

LIST 子系统树：

LIST	:STARtno	<integer>	设置起始组号
	:GROUps	<integer>	设置输出组数
	:REPEat	<integer>	设置循环次数
	:FINIsh	{STOP, HOLD}	设置停止状态
	:FUNctioN	{OFF, ON, 0, 1}	设置列表使能
	:STEP	<integer>, <float>, <float>, <float> //组号, 电压, 电流, 时间	单步参数设置
	:VOLTage	<integer>, <float> //组号, 电压	单步电压设置
	:CURRent	<integer>, <float> //组号, 电流	单步电流设置
	:TIMer	<integer>, <float> //组号, 时间	单步时间设置
	:LOAD	<integer>	加载列表文件
	:SAVE	<integer>	保存列表文件
	:DELete	<integer>	删除列表文件
	:REName	<integer>, <string>	重命名列表文件
	:PLOad	<integer>, {OFF, ON, 0, 1}	列表文件开机调用
	:AUTOSave	<integer>, {OFF, ON, 0, 1}	列表文件自动保存

LIST:STARtno 用来设置列表起始步骤。

命令语法	LIST:STARtno <integer>
例如	发送>LIST:STAR 1
查询语法	LIST:STARtno?
查询响应	<integer>

LIST:GROUps 用来设置列表输出组数。

命令语法	LIST:GROUps <integer>
例如	发送>LIST:GROUps 1
查询语法	LIST:GROUps?
查询响应	<integer>

LIST:REPEat 用来设置列表重复次数。

命令语法	LIST:REPEat <integer>
例如	发送>LIST:REPEat 1
查询语法	LIST:REPEat?
查询响应	<integer>

LIST:FINIsh 用来设置列表结束状态。

命令语法	LIST:FINIsh {STOP, HOLD}
例如	发送>LIST:FINIsh STOP
查询语法	LIST:FINIsh?
查询响应	{STOP, HOLD}

LIST:FUNcTion 用来使能列表。

命令语法	LIST:FUNcTion {OFF, ON, 0, 1}
例如	发送>LIST:FUNc ON
查询语法	LIST:FUNcTion?
查询响应	{OFF, ON}

LIST:STEP 用来设置列表数据。

命令语法	LIST:STEP <integer>, <float>, <float>, <float>
例如	发送>LIST:STEP 1, 80, 5, 10 //列表第 1 步参数设置为 80V, 5A, 10 秒
查询语法	LIST:STEP? <integer>
查询响应	<integer>, <float>, <float>, <float> //组号, 电压, 电流, 时间
例如	发送>LIST:STEP? 1 返回>1, 80.00, 5.00, 10.0 //组号, 电压, 电流, 时间

LIST:VOLTage 用来设置列表电压数据。

命令语法	LIST:VOLTage <integer>, <float> //组号, 电压
例如	发送>LIST:VOLT 1, 80 //列表第 1 组电压设置为 80V
查询语法	LIST:VOLTage? <integer>
查询响应	<float>
例如	发送>LIST:VOLT? 1 返回>80

LIST:CURRent 用来设置列表电流数据。

命令语法	LIST:CURRent <integer>, <float> //组号, 电流
例如	发送>LIST:CURR 1, 5 //列表第 1 组电流设置为 5A
查询语法	LIST:CURR? <integer>
查询响应	<float>

例如	发送>LIST:CURR? 1 返回>5
----	-------------------------

LIST:TIMer 用来设置列表时间数据。

命令语法	LIST:TIMer <integer>,<float> //组号, 时间
例如	发送>LIST:TIM 1,10 //列表第 1 组时间设置为 10 秒
查询语法	LIST:TIM? <integer>
查询响应	<float>
例如	发送>LIST:TIM? 1 返回> 10

LIST:LOAD 用来加载列表文件。

当电源已经定数输出时，本指令无效。

命令语法	LIST:LOAD <integer>
例如	发送>LIST:LOAD 1 //加载列表文件 1

LIST:SAVE 用来保存列表文件。

命令语法	LIST:SAVE <integer>
例如	发送>LIST:SAVE 1 //将当前设置保存到列表文件 1 中

LIST:DELeTe 用来删除列表文件。

命令语法	LIST:DELeTe <integer>
例如	发送>LIST:DEL 1 //删除列表文件 1

LIST:REName 用来删除列表文件。

命令语法	LIST:REName <integer>,<string>
例如	发送>LIST:REN 1," ABC" //将列表文件 1 命名为 ABC

LIST:PL0ad 用来设置开机调用。

将对应的文件设置为开机加载。同时只能有一个文件被设置成开机加载，例如文件1 是开机加载，若设置文件2 是开机加载则，文件1 的开机加载会被取消，若删除了开机加载的文件，则文件0 会被设置成开机加载。

命令语法	LIST:PL0ad <integer>
例如	发送>LIST:PL0ad 1 //列表文件 1 设置为开机调用
查询语法 1	LIST:PL0ad?
查询响应 1	<integer>
查询语法 2	LIST:PL0ad? <integer> //查询指定文件是否是开机调用
查询响应 2	{OFF, ON}

例如 1	发送>LIST:PLoad? 响应>3	//查询开机调用文件号 //文件 3 是开机调用
例如 2	发送 2>LIST:PLoad? 1 响应 2>OFF	//查询文件 1 是否是开机调用 //文件 1 没有开机调用

LIST:AUTOsave 用来设置自动保存。

设置开机调用文件的自动保功能。若使能了自动保存，手动设置的参数会立刻保存到对应的文件中。

命令语法	LIST:AUTOsave {OFF, ON, 0, 1}
例如	发送>LIST:AUTOsave ON //使能自动保存
查询语法	LIST:AUTOsave?
查询响应	{OFF, ON}

2.11 DELayer 延时器子系统

DELAyer 子系统用于延时器输出相关的设置，当电源在延时器输出时，本子系统的指令无效。当电源在列表输出时，FUNCTION 和 LOAD 指令无效。

DELAyer 子系统树：

DELAyer	:STARtno	<integer>	设置起始组号	
	:GROUps	<integer>	设置输出组数	
	:REPEat	<integer>	设置循环次数	
	:FINIsh	{STOP, HOLD}	设置停止状态	
	:FUNctIon	{OFF, ON, 0, 1}	延时器使能	
	:STEP	<integer>, {OFF, ON, 0, 1}, <float>	//组号, 状态, 时间	单步参数设置
	:STATe	<integer>, {OFF, ON, 0, 1}	//组号, 状态	单步状态设置
	:TIMer	<integer>, <float>	//组号, 时间	单步时间设置
	:LOAD	<integer>		加载延时器文件
	:SAVE	<integer>		保存延时器文件
	:DELete	<integer>		删除延时器文件
	:RENAmE	<integer>, <string>		重命名延时器文件
	:PLOad	<integer>, {OFF, ON, 0, 1}		延时器文件开机调用
	:AUTOSave	<integer>, {OFF, ON, 0, 1}		延时器文件自动保存

DELAyer:STARtno 用来设置延时器起始组号。

命令语法	DELAyer:STARtno <integer>
例如	发送>DELA:STAR 1
查询语法	DELAyer:STARtno?
查询响应	<integer>

DELAyer:GROUps 用来设置延时器输出组数。

命令语法	DELAyer:GROUps <integer>
例如	发送>DELA:GROU 1
查询语法	DELAyer:GROUps?
查询响应	<integer>

DELAyer:REPEat 用来设置延时器重复次数。

命令语法	DELAyer:REPEat <integer>
例如	发送>DELA:REPE 1
查询语法	DELAyer:REPEat?
查询响应	<integer>

DELAyer:FINIsh 用来设置延时器结束状态。

命令语法	DELAyer:FINIsh {STOP,HOLD}
例如	发送>DELA:FINIsh STOP
查询语法	DELAyer:FINIsh?
查询响应	{STOP,HOLD}

DELAyer:FUNcTion 用来使能延时器。

命令语法	DELAyer:FUNcTion {OFF,ON,0,1}
例如	发送>DELA:FUNc ON
查询语法	DELAyer:FUNcTion?
查询响应	{OFF,ON}

DELAyer:STEP 用来设置延时器数据。

命令语法	DELAyer:STEP <integer>, {OFF,ON,0,1}, <float>
例如	发送>DELA:STEP 1, ON, 10.1 //延时器第 1 组使能输出 10.1 秒
查询语法	DELAyer:STEP? <integer>
查询响应	<integer>, {OFF,ON}, <float>
例如	发送>DELA:STEP? 1 返回>1, ON, 10.1 //组号, 状态, 时间

DELAyer:STATe 用来设置延时器每步的输出状态。

命令语法	DELAyer:STATe <integer>, {OFF,ON,0,1}
例如	发送>DELA:STAT 1, ON //延时器第 1 组使能输出
查询语法	DELAyer:STATe? <integer>
查询响应	{OFF,ON}

DELAyer:TIMer 用来设置延时器每步的输出时间。

命令语法	DELAyer:TIMer <integer>, [<float>, MINimum, MAXimum]
例如	发送>DELA:TIM 1, 10.1 //延时器第 1 组输出 10.1 秒
查询语法	DELAyer:TIMer? <integer>
查询响应	<float>

DELAyer:LOAD 用来加载列表文件。

命令语法	DELAyer:LOAD <integer>
例如	发送>DELAyer:LOAD 1 //加载延时器文件 1

DELAyer:SAVE 用来保存延时器文件。

命令语法	DELAyer:SAVE <integer>
例如	发送>DELAyer:SAVE 1 //将当前设置保存到延时器文件 1 中

DELAyer:DELeTe 用来删除延时器文件。

命令语法	DELAyer:DELeTe <integer>
例如	发送>DELAyer:DEL 1 //删除延时器文件 1

DELAyer:REName 用来删除延时器文件。

命令语法	DELAyer:REName <integer>, <string>
例如	发送>DELAyer:REN 1, "ABC" //将延时器文件 1 命名为 ABC

DELAyer:PL0ad 用来设置开机调用。

将对应的文件设置为开机加载。同时只能有一个文件被设置成开机加载，例如文件1 是开机加载，若设置文件2 是开机加载则，文件1 的开机加载会被取消，若删除了开机加载的文件，则文件0 会被设置成开机加载。

命令语法	DELAyer:PL0ad <integer>
例如	发送>DELAyer:PL0 1 //延时器文件 1 设置为开机调用
查询语法 1	DELAyer:PL0ad?
查询响应 1	<integer> //查询开机调用的文件号
查询语法 2	DELAyer:PL0ad? <integer>
查询响应 2	{OFF, ON} //查询指定文件是否是开机调用
例如 1	发送>DELA:PL0ad? //查询开机调用文件号
	响应>3 //文件 3 是开机调用
例如 2	发送 2>DELA:PL0ad? 1 //查询文件 1 是否是开机调用
	响应 2>OFF //文件 1 没有开机调用

DELAyer:AUTOsave 用来设置开机调用。

设置开机调用文件的自动保功能。若使能了自动保存，手动设置的参数会立刻保存到对应的文件中。

命令语法	DELAyer:AUTOsave {OFF, ON, 0, 1}
例如	发送>DELAyer:AUTOS ON //使能自动保存
查询语法	DELAyer:AUTOsave?
查询响应	{OFF, ON}

2.12 FILE 文件子系统

FILE 子系统树:

FILE	:LOAD	<integer>	加载系统文件
	:SAVE	<integer>	保存系统文件
	:DELeTe	<integer>	删除系统文件
	:REName	<integer>, <string>	重命名系统文件
	:PL0ad	<integer>, {OFF, ON, 0, 1}	系统文件开机调用
	:AUTOSave	<integer>, {OFF, ON, 0, 1}	系统文件自动保存

FILE:LOAD 加载系统文件

命令语法	FILE:LOAD <integer>
例如	发送>FILE:LOAD 1 //加载系统文件 1

FILE:SAVE 保存系统文件

命令语法	FILE:SAVE <integer>
例如	发送>FILE:SAVE 1 //将当前设置保存到系统文件 1 中

FILE:DELeTe 删除系统文件

命令语法	FILE:DELeTe <integer>
例如	发送>FILE:DEL 1 //删除系统文件 1

FILE:REName 重命名系统文件

命令语法	FILE:REName <integer>, <string>
例如	发送>FILE:REN 1, "ABC" //将系统文件 1 命名为 ABC

FILE:PL0ad 设置开机调用文件号

将对应的文件设置为开机加载。同时只能有一个文件被设置成开机加载，例如文件1 是开机加载，若设置文件2 是开机加载则，文件1 的开机加载会被取消，若删除了开机加载的文件，则文件0 会被设置成开机加载。

命令语法	FILE:PL0ad <integer>
例如	发送>FILE:PL0 1 //系统文件 1 设置为开机调用
查询语法 1	FILE:PL0ad? //查询开机调用的文件号
查询响应 1	<integer>
查询语法 2	FILE:PL0ad? <integer> //查询指定文件是否是开机调用
查询响应 2	{OFF, ON}
例如 1	发送>FILE:PL0ad? 响应>3 //查询开机调用文件号 //文件 3 是开机调用
例如 2	发送 2>FILE:PL0ad? 1 //查询文件 1 是否是开机调用

响应 2>OFF	//文件 1 没有开机调用
----------	---------------

FILE:AUTOsave 设置自动保存文件号

设置开机调用文件的自动保功能。若使能了自动保存，手动设置的参数会立刻保存到对应的文件中。

命令语法	FILE:AUTOsave {OFF, ON, 0, 1}
例如	发送>FILE:AUTOS 1, ON //系使能自动保存
查询语法	FILE:AUTOsave?
查询响应	{OFF, ON}

2.13 *IDN? 子系统

*IDN? 子系统用来返回仪器的版本号。

查询语法	*IDN?
查询响应	<Manufacturer>, <MODEL>, <Sn>, <Revision>
例如	发送> *IDN? 返回> UNIT, UDP6722, UNLICENSED, REV1. 21

3 Modbus (RTU) 通讯协议

3.1 数据格式

我们遵循 Modbus (RTU) 通讯协议，仪器将响应上位机的指令，并返回标准响应帧。

指令帧

图 0-1 Modbus 指令帧



指令帧说明

	至少需要 3.5 字符时间的静噪间隔
从站地址	1 字节 Modbus 可以支持 00~0x63 个从站 统一广播时指定为 00
功能码	1 字节 0x03: 读出多个寄存器 0x04: =03H, 不使用 0x06: 写入单个寄存器, 可以用 10H 替代 0x08: 回波测试 (仅用于调试时使用) 0x10: 写入多个寄存器

数据	指定寄存器地址、数量和内容
CRC-16	2 字节，低位在前 CyclicRedundancy Check 将从站地址到数据末尾的所有数据进行计算，得到 CRC16 校验码 至少需要 3.5 字符时间的静噪间隔

响应帧

除非是 00H 从站地址广播的指令，其它从站地址仪器都会返回响应帧。

图 0-2 正常响应帧

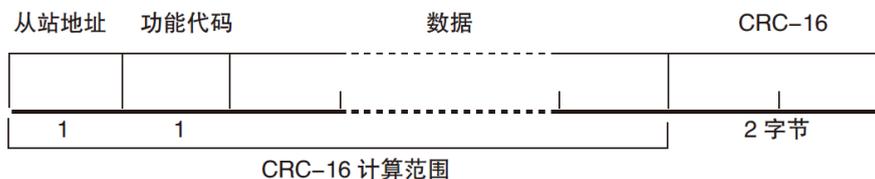


图 0-3 异常响应帧



异常响应帧说明

从站地址	1 字节 从站地址原样返回
功能码	1 字节 指令帧的功能码逻辑或 (OR) 上 BIT7 (0x80)，例如：0x03 OR 0x80 = 0x83
错误码	异常代码： 0x01 功能码错误（功能码不支持） 0x02 寄存器错误（寄存器不存在） 0x03 数据错误 0x04 执行错误
CRC-16	2 字节，低位在前 CyclicRedundancy Check 将从站地址到数据末尾的所有数据进行计算，得到 CRC16 校验码

无响应

以下情况，仪器将不进行任何处理，也不响应，导致通讯超时。

1. 从站地址错误
2. 传输错误
3. CRC-16 错误
4. 位数错误，例如：功能码 0x03 总位数必须为 8，而接受到的位数小于 8 或大于 8 个字节。
5. 从站地址为 0x00 时，代表广播地址，仪器不响应。

错误码错误码说明

错误码	名称	说明	优先级
0x01	功能码错误	功能码不存在	1

0x02	寄存器错误	寄存器不存在	2
0x03	数据错误	寄存器数量或字节数量错误	3
0x04	执行错误	数据非法, 写入的数据不在允许范围内	4

3.2 功能码

仪器仅支持以下几个功能码, 其它功能码未做支持。

功能码	名称	说明
0x03	读出多个寄存器	读出多个连续寄存器数据
0x10	写入多个寄存器	写入多个连续寄存器

3.3 寄存器

仪器的寄存器数量为 2 字节模式, 即每次必须写入 2 个字节, 例如: 速度的寄存器为 0x3002, 数据为 2 字节, 数值必须写入 0x0001

数据:

仪器支持以下几种数值:

- 1 个寄存器, 双字节 (16 位) 整数, 例如: 0x64 → 00 64
- 2 个寄存器, 四字节 (32 位) 整数, 例如: 0x12345678 → 12 34 56 78
- 2 个寄存器, 四字节 (32 位) 单精度浮点数, 3.14 → 40 48 F5 C3

3.4 读出多个寄存器

图 0-4 读出多个寄存器 (0x03)

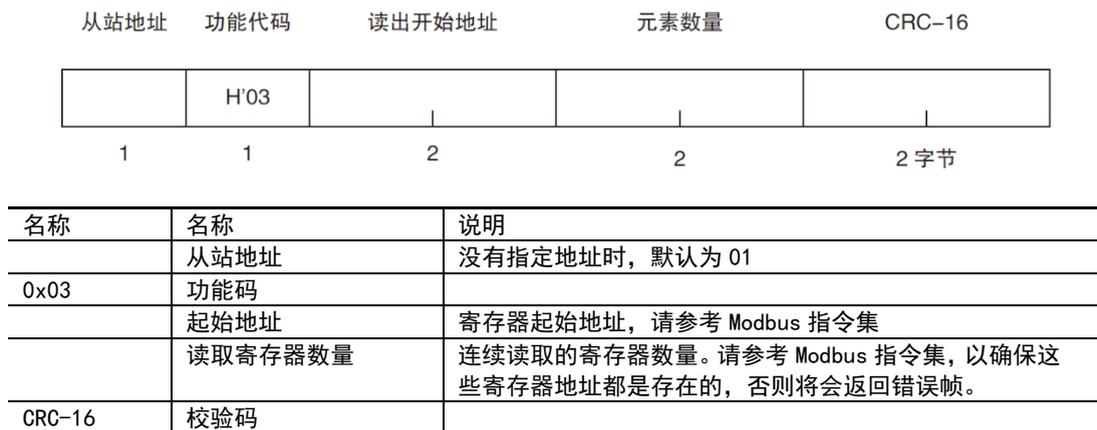
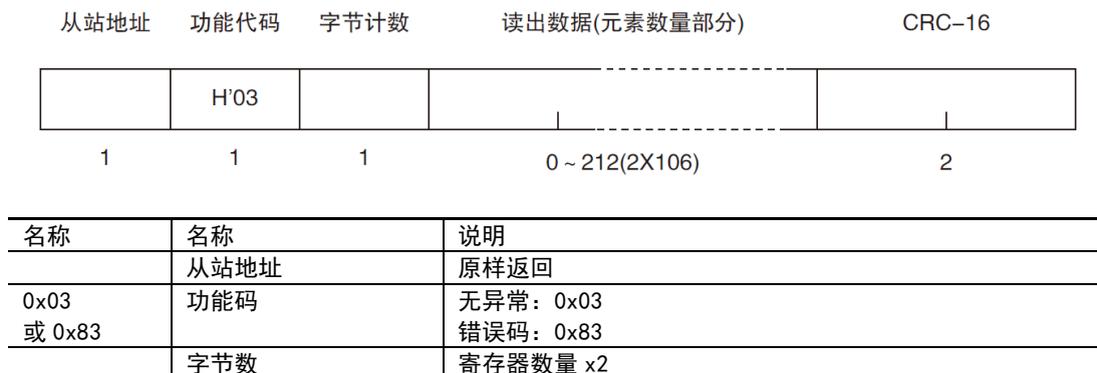


图 0-5 读出多个寄存器 (0x03) 响应帧

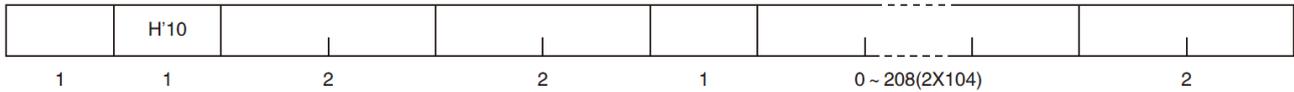


		例如：1 个寄存器返回 02
	数据	读取的数据
CRC-16	校验码	

3.5 写入多个寄存器

图 0-6 写入多个寄存器 (0x10)

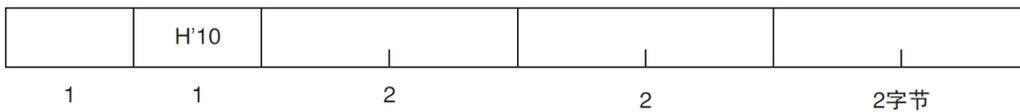
从站地址 功能代码 读出开始地址 元素数量 字节计数 写入数据(元素数量部分) CRC-16



名称	名称	说明
	从站地址	默认为 01
0x10	功能码	
	起始地址	寄存器起始地址，请参考 Modbus 指令集
	写入寄存器数量 0001~0068 (104)	连续读取的寄存器数量。请参考 Modbus 指令集，以确保这些寄存器地址都是存在的，否则将会返回错误帧。
	字节数	=寄存器数量 x2
CRC-16	校验码	

图 0-7 写入多个寄存器 (0x10) 响应帧

从站地址 功能代码 写入开始地址 元素数量 CRC-16



名称	名称	说明
	从站地址	原样返回
0x10 或 0x90	功能码	无异常：0x10 错误码：0x90
	起始地址	
	寄存器数量	
	CRC-16 校验码	

4 Modbus (RTU) 指令集

4.1 寄存器总览

以下列出了仪器使用的所有寄存器地址。



注意：1、除非特别说明，以下说明中指令和响应帧的数值都是 16 进制数据。

2、寄存器只包含获取测量结果和启动/停止测试，如需定制其他指令，请联系浩仪销售部。

3、浮点数在线转换，请参考网站 http://www.binaryconvert.com/convert_float.html

寄存器地址	名称	数值	占用寄存器	权限
0x0200	启动或停止电源输出	2 字节整数 0x0000: 停止 0x0001: 启动	1	读写
0x0201	读取 CV/CC 状态	2 字节整数 0x0000: CV 0x0001: CC	1	只读
0x0202	读取回读电压值	4 字节浮点数	2	只读
0x0204	读取回读电流值	4 字节浮点数	2	只读
0x0206	读取回读功率值	4 字节浮点数	2	只读
0x0208	设置电压值	4 字节浮点数	2	
0x020A	设置电流值	4 字节浮点数	2	读写
0x020C	设置过压保护值	4 字节浮点数	2	读写
0x020E	设置过流保护值	4 字节浮点数	2	读写
0x0210	设置输出时间值	4 字节浮点数	2	读写
0x0212	设置过压保护状态	2 字节整数 0x0000: OFF 0x0001: ON	1	读写
0x0213	设置过流保护状态	2 字节整数 0x0000: OFF 0x0001: ON	1	读写
0x0214	设置输出定时器状态	2 字节整数 0x0000: OFF 0x0001: ON	1	读写
0x0215	设置开机输出	2 字节整数 0x0000: OFF 0x0001: ON	1	读写
0x0216	设置列表起始组号	2 字节整数	1	读写
0x0217	设置列表输出组数	2 字节整数	1	读写
0x0218	设置列表重复次数	2 字节整数	1	读写
0x0219	设置列表结束状态	2 字节整数 0x0000: STOP 0x0001: HOLD	1	读写
0x021A	设置列表使能状态	2 字节整数 0x0000: OFF 0x0001: ON	1	读写
0x021B	选择列表当前步骤	2 字节整数	1	读写
0x021C	设置列表当前步骤下的电压值	4 字节浮点数	2	读写
0x021E	设置列表当前步骤下的电流值	4 字节浮点数	2	读写
0x0220	设置列表当前步骤下的时间值	4 字节浮点数	2	读写
0x0221	加载列表文件	2 字节整数	1	只写
0x0222	保存列表文件	2 字节整数	1	只写
0x0223	删除列表文件	2 字节整数	1	只写
0x0224	列表文件开机加载	2 字节整数	1	读写
0x0225	列表文件实时保存	2 字节整数	1	读写
0x0226	设置延时器起始组号	2 字节整数	1	读写
0x0227	设置延时器输出组数	2 字节整数	1	读写
0x0228	设置延时器重复次数	2 字节整数	1	读写
0x0229	设置延时器停止状态	2 字节整数	1	读写
0x022A	设置延时器使能状态	2 字节整数 0x0000: STOP 0x0001: HOLD	1	读写
0x022B	选择延时器当前步骤	2 字节整数	1	读写
0x022C	设置延时器当前步骤下的状态	2 字节整数	1	读写
0x022D	设置延时器当前步骤下的时间值	4 字节浮点数	2	读写
0x022F	加载延时器文件	2 字节整数	1	只写

0x0230	保存延时器文件	2 字节整数	1	只写
0x0231	删除延时器文件	2 字节整数	1	只写
0x0232	设置延时器文件开机加载	2 字节整数	1	读写
0x0233	设置延时器文件实时保存	2 字节整数	1	读写
0x0234	系统文件开机调用	2 字节整数	1	读写
0x0235	保存系统文件	2 字节整数	1	只写
0x0236	删除系统文件	2 字节整数	1	只写
0x0237	设置系统文件开机加载	2 字节整数	1	读写
0x0238	设置系统文件实时保存	2 字节整数	1	读写
0x0239	仪器页面设置	2 字节整数 0x0000: 测量显示 0x0001: 测量设置 0x0002: 列表设置 0x0003: 列表保存 0x0004: 延时器设置 0x0005: 延时器保存 0x0006: 系统设置 0x0007: 文件管理	1	读写
0x023A	设置当前语言	2 字节整数 0x0000: English 0x0001: Chinese	1	读写
0x023B	设置年	2 字节整数	1	读写
0x023C	设置月	2 字节整数	1	读写
0x023D	设置日	2 字节整数	1	读写
0x023E	设置时	2 字节整数	1	读写
0x023F	设置分	2 字节整数	1	读写
0x0240	设置秒	2 字节整数	1	读写
0x0241	设置按键音	2 字节整数	1	读写
0x0242	查询与清除过压保护提示	2 字节整数	1	读写
0x0243	查询与清除过压保护提示	2 字节整数	1	读写

4.2 电源状态查询

启动/停止电源输出

写入

1	2	3	4	5	6	7	8	9	10	11
01	10	02	00	00	01	02	00	01	44	50
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8~B9:

0x0000: 停止电源输出

0x0001: 启动电源输出

响应

1	2	3	4	5	6	7	8
01	10	02	00	00	01	00	71
从站	写	寄存器		寄存器数量		CRC16	

查询输出状态

发送:

1	2	3	4	5	6	7	8
01	03	02	00	00	01	85	B2
从站	读	寄存器		寄存器数量		校验码	

响应

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

B4~B5:

0X0000 电源已停止

0X0001 电源在输出

查询 CV/CC 状态

发送

1	2	3	4	5	6	7	8
01	03	02	01	00	01	79	84
从站	读	寄存器		寄存器数量		校验码	

响应

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

B4~B5

0X0000 CV

0X0001 CC

查询回读电压值

发送

1	2	3	4	5	6	7	8
01	03	02	02	00	02	64	73
从站	读	寄存器		寄存器数量		校验码	

响应:

1	2	3	4	5	6	7	8	9	
01	03	04	41	9F	F3	63	DA	F8	
01	03	字节	单精度浮点数				CRC-16		

B4~B7: float 类型的数据, 0x419FF363=19.993841, 当前仪器回读电压 19.99V

查询回读电流值

发送

1	2	3	4	5	6	7	8
01	03	02	04	00	02	84	72
从站	读	寄存器		寄存器数量		校验码	

响应:

1	2	3	4	5	6	7	8	9	
01	03	04	40	9F	E8	64	90	36	
01	03	字节	单精度浮点数				CRC-16		

B4~B7: float 类型的数据, 0x409FE864=4.997118, 当前仪器回读电流 4.99A

查询回读功率值

发送

1	2	3	4	5	6	7	8
01	03	02	06	00	02	25	B2
从站	读	寄存器		寄存器数量		校验码	

响应:

1	2	3	4	5	6	7	8	9
01	03	04	00	00	00	00	FA	33
01	03	字节	单精度浮点数				CRC-16	

B4~B7: float 类型的数据, 0x00000000=0, 当前仪器回读功率 0W

4.3 电源输出参数设置

设置电压值

当电源工作在列表或延时器模式时, 本指令无效。

写入

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数				CRC16	

B8~B11 为单精度浮点数, 0x41200000 = 10, 即电压设置为 10.00V

响应

1	2	3	4	5	6	7	8
01	10	02	08	00	02	00	71
从站	写	寄存器		寄存器数量		CRC16	

设置电流值

当电源工作在列表或延时器模式时, 本指令无效。

写入

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数				CRC16	

B8~B11 为单精度浮点数, 0x40A00000 = 5, 即电压设置为 5.00A

响应

1	2	3	4	5	6	7	8
01	10	02	0A	00	02	00	71
从站	写	寄存器		寄存器数量		CRC16	

设置过压保护值

当电源工作在列表或延时器模式时, 本指令无效。

写入

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数				CRC16	

B8~B11 为单精度浮点数, 0x41A00000 = 20, 即过压保护值设置为 20.00V

响应

1	2	3	4	5	6	7	8
01	10	02	0A	00	02	00	71
从站	写	寄存器		寄存器数量		CRC16	

设置过流保护值

当电源工作在列表或延时器模式时, 本指令无效。

写入

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数			CRC16		

B8~B11 为单精度浮点数, 0x41A00000 = 20, 即过流保护值设置为 20.00A

响应

1	2	3	4	5	6	7	8
01	10	02	0E	00	02	21	B3
从站	写	寄存器		寄存器数量		CRC16	

设置定时输出值

当电源在定时输出时, 本指令无效。

写入

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数			CRC16		

B8~B11 为单精度浮点数, 0x41A00000 = 20, 即定时输出值设置为 20.00s

响应

1	2	3	4	5	6	7	8
01	10	02	10	00	02	41	B5
从站	写	寄存器		寄存器数量		CRC16	

设置过流保护状态

当电源工作在列表或延时器模式时, 本指令无效。

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	13	00	01	02	00	01	46	F3
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8~B9:

0x0000: 停止过流保护

0x0001: 启用过流保护

响应

1	2	3	4	5	6	7	8
01	10	02	13	00	01	F1	B4
从站	写	寄存器		寄存器数量		CRC16	

设置过压保护状态

当电源工作在列表或延时器模式时, 本指令无效。

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	12	00	01	02	00	01	47	22
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8~B9:

0x0000: 停止过压保护

0x0001: 启用过压保护

响应

1	2	3	4	5	6	7	8
01	10	02	12	00	01	A0	74
从站	写	寄存器		寄存器数量		CRC16	

设置输出定时器状态

当电源在定时输出时，本指令无效。

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	14	00	01	02	00	01	47	44
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8~B9:

0x0000: 停止定时输出

0x0001: 启用定时输出

响应

1	2	3	4	5	6	7	8
01	10	02	14	00	01	40	74
从站	写	寄存器	寄存器数量	CRC16			

设置开机输出状态

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	15	00	01	02	00	01	46	95
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8~B9:

0x0000: 停止开机输出

0x0001: 启用开机输出

响应

1	2	3	4	5	6	7	8
01	10	02	15	00	01	11	B5
从站	写	寄存器	寄存器数量	CRC16			

查询是否过压保护

发送

1	2	3	4	5	6	7	8
01	03	02	42	00	01	25	A6
从站	读	寄存器	寄存器数量	校验码			

响应

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

B4~B5

0X0000 未过压保护

0X0001 已过压保护

清除过压保护提示

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	42	00	01	02	00	01	4B	72
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

响应

1	2	3	4	5	6	7	8
01	10	02	42	00	01	A0	65
从站	写	寄存器	寄存器数量	CRC16			

设查询是否过流保护

发送

1	2	3	4	5	6	7	8
01	03	02	43	00	01	74	66
从站	读	寄存器	寄存器数量	校验码			

响应

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

B4~B5

0X0000 未过流保护

0X0001 已过流保护

清除过流保护提示

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	43	00	01	02	00	01	4A	A3
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

响应

1	2	3	4	5	6	7	8
01	10	02	43	00	01	F1	A5
从站	写	寄存器	寄存器数量	CRC16			

4.4 列表参数设置

当电源在列表输出时，本子系统的指令无效。当电源在延时器输出时，使能列表和加载文件指令无效。

设置起始组号

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	16	00	01	02	00	02	06	A7
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9=0X0002，起始组号设置为 2

响应

1	2	3	4	5	6	7	8
01	10	02	16	00	01	E1	B5
从站	写	寄存器	寄存器数量	CRC16			

设置输出组数

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	17	00	01	02	00	02	07	76
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9=0X0002，输出组数设置为 2

响应

1	2	3	4	5	6	7	8
01	10	02	17	00	01	B0	75
从站	写	寄存器	寄存器数量	CRC16			

设置重复次数

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	18	00	01	02	00	01	47	88
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9=0X0001, 重复次数设置为 1

响应

1	2	3	4	5	6	7	8
01	10	02	18	00	01	80	76
从站	写	寄存器	寄存器数量	CRC16			

设置结束状态

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	19	00	01	02	00	00	87	99
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9

0x0000: STOP

0X0001: HOLD

响应

1	2	3	4	5	6	7	8
01	10	02	19	00	01	D1	B6
从站	写	寄存器	寄存器数量	CRC16			

使能列表功能

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	1A	00	01	02	00	01	87	99
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9

0x0000: 停止列表功能

0X0001: 使能列表功能

响应

1	2	3	4	5	6	7	8
01	10	02	1A	00	01	21	B6
从站	写	寄存器	寄存器数量	CRC16			

选择列表当前步骤

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	1B	00	01	02	00	01	47	BB
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9=0X0001, 选择表格中的第 1 步

响应

1	2	3	4	5	6	7	8
01	10	02	1B	00	01	70	76
从站	写	寄存器	寄存器数量	CRC16			

设置当前步骤下的电压值

发送

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数			CRC16		

B8~B11 是单精度浮点数, =0X41A00000=20, 即列表第 1 步的电压值设置为 20.00V

当前步骤是由 0x021B 的寄存器值决定的

响应

1	2	3	4	5	6	7	8
01	10	02	1C	00	02	81	B6
从站	写	寄存器		寄存器数量		CRC16	

设置当前步骤下的电流值

发送

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数			CRC16		

B8~B11 是单精度浮点数, =0X41A00000=20, 即列表第 1 步的电压值设置为 20.00A

当前步骤是由 0x021B 的寄存器值决定的

响应

1	2	3	4	5	6	7	8
01	10	02	1E	00	02	20	76
从站	写	寄存器		寄存器数量		CRC16	

设置当前步骤下的时间值

发送

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数			CRC16		

B8~B11 是单精度浮点数, =0X41A00000=20, 即列表第 1 步的电压值设置为 20.00s

当前步骤是由 0x021B 的寄存器值决定的

响应

1	2	3	4	5	6	7	8
01	10	02	20	00	02	41	BA
从站	写	寄存器		寄存器数量		CRC16	

同时设置指定步骤下的电压、电流、时间值

发送

01	10	02	1B	00	07	0E	00	01	41	A0	00	00	41	A0	00	00	41	A0	00	00	87	A3
		寄存器		寄存器数量		字节	整数	单精度浮点数			单精度浮点数			单精度浮点数			CRC-16					
							步骤	电压值			电流值			时间值								

将第 1 步的电压设置成 20V, 电流 20A, 时间 20s

响应

1	2	3	4	5	6	7	8
01	10	02	1B	00	07	F0	74
从站	写	寄存器		寄存器数量		CRC16	

加载列表文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	21	00	01	02	00	01	42	E1
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	21	00	01	50	7B
从站	写	寄存器	寄存器数量	CRC16			

保存列表文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	22	00	01	02	00	01	42	D2
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	22	00	01	A0	7B
从站	写	寄存器	寄存器数量	CRC16			

删除列表文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	23	00	01	02	00	01	43	03
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	23	00	01	F1	BB
从站	写	寄存器	寄存器数量	CRC16			

开机加载

将对应的文件设置为开机加载。同时只能有一个文件被设置成开机加载，例如文件1 是开机加载，若设置文件2 是开机加载则，文件1 的开机加载会被取消，若删除了开机加载的文件，则文件0 会被设置成开机加载。

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	24	00	01	02	00	01	42	B4
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	24	00	01	40	7A
从站	写	寄存器	寄存器数量	CRC16			

自动保存

设置开机调用文件的自动保功能。若使能了自动保存，手动设置的参数会立刻保存到对应的文件中。

1	2	3	4	5	6	7	8	9	10	11
01	10	02	25	00	01	02	00	01	43	65
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9:

0000 禁用自动保存

0001 禁用自动保存

响应

1	2	3	4	5	6	7	8
01	10	02	25	00	01	11	BA
从站	写	寄存器	寄存器数量	CRC16			

4.5 延时器参数设置

当电源在延时器输出时，本子系统的指令无效。当电源在列表输出时，使能延时器和加载文件指令无效

设置起始组号

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	26	00	01	02	00	02	82	96
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9=0X0002, 起始组号设置为 2

响应

1	2	3	4	5	6	7	8
01	10	02	26	00	01	E1	BA
从站	写	寄存器	寄存器数量	CRC16			

设置输出组数

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	27	00	01	02	00	02	02	86
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9=0X0002, 输出组数设置为 2

响应

1	2	3	4	5	6	7	8
01	10	02	27	00	01	B0	7A
从站	写	寄存器	寄存器数量	CRC16			

设置重复次数

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	28	00	01	02	00	01	42	78
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9=0X0001, 重复次数设置为 1

响应

1	2	3	4	5	6	7	8
01	10	02	28	00	01	80	79
从站	写	寄存器	寄存器数量	CRC16			

设置结束状态

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	29	00	01	02	00	00	82	69
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9

0x0000: STOP

0X0001: HOLD

响应

1	2	3	4	5	6	7	8
01	10	02	29	00	01	D1	B9
从站	写	寄存器		寄存器数量		CRC16	

使能延时器功能

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2A	00	01	02	00	01	43	9A
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8B9

0x0000: 打开延时器功能

0X0001: 关闭延时器功能

响应

1	2	3	4	5	6	7	8
01	10	02	2A	00	01	21	B9
从站	写	寄存器		寄存器数量		CRC16	

选择延时器当前步骤

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2B	00	01	02	00	01	42	4B
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8B9=0X0001, 选择表格中的第 1 步

响应

1	2	3	4	5	6	7	8
01	10	02	2B	00	01	70	79
从站	写	寄存器		寄存器数量		CRC16	

设置当前步骤的状态

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2C	00	01	02	00	01	42	4B
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8B9=0x0001, 使能本步的输出

0x0000: OFF

0X0001: ON

当前步骤是由 0x022B 的寄存器值设定的。

响应

1	2	3	4	5	6	7	8
01	10	02	2C	00	01	C1	B8
从站	写	寄存器		寄存器数量		CRC16	

设置当前步骤下的时间值

发送

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
站号	写	寄存器		寄存器数量		字节	单精度浮点数				CRC16	

B8~B11 是单精度浮点数, =0X41A0000=20, 即延时器第 1 步的电压值设置为 20.00s

当前步骤是由 0x022B 的寄存器值决定的

响应

1	2	3	4	5	6	7	8
01	10	02	20	00	02	41	BA
从站	写	寄存器	寄存器数量	CRC16			

同时设置指定步骤下的状态、时间值

发送

01	10	02	2B	00	04	08	00	01	00	01	41	A0	00	00	87	A3
		寄存器	寄存器数量	字节	整数	整数	单精度浮点数				CRC-16					
					步骤	状态	时间									

将第 1 步的设置使能输出，时间 20s

响应

1	2	3	4	5	6	7	8
01	10	02	2B	00	04	B0	7A
从站	写	寄存器	寄存器数量	CRC16			

加载延时器文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	2F	00	01	02	00	01	43	CF
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9：内部文件编号，不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	2F	00	01	31	B8
从站	写	寄存器	寄存器数量	CRC16			

保存延时器文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	30	00	01	02	00	01	41	A0
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9：内部文件编号，不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	30	00	01	00	7E
从站	写	寄存器	寄存器数量	CRC16			

删除延时器文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	31	00	01	02	00	01	40	71
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9：内部文件编号，不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	31	00	01	51	BE
从站	写	寄存器	寄存器数量	CRC16			

开机加载

将对应的文件设置为开机加载。同时只能有一个文件被设置成开机加载，例如文件1 是开机加载，若设置文件2 是开机加载则，文件1 的开机加载会被取消，若删除了开机加载的文件，则文件0 会被设置成开机加载。

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	32	00	01	02	00	01	40	42
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	32	00	01	A1	BE
从站	写	寄存器	寄存器数量	CRC16			

自动保存

设置开机调用文件的自动保功能。若使能了自动保存，手动设置的参数会立刻保存到对应的文件中。

1	2	3	4	5	6	7	8	9	10	11
01	10	02	33	00	01	02	00	01	41	93
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9:

0000: 禁用自动保存

0001: 使能自动保存

响应

1	2	3	4	5	6	7	8
01	10	02	33	00	01	F0	7E
从站	写	寄存器	寄存器数量	CRC16			

4.6 文件设置

保存了除列表和延时器以外的参数设置。

加载文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	34	00	01	02	00	01	40	24
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	34	00	01	41	BF
从站	写	寄存器	寄存器数量	CRC16			

保存文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	35	00	01	02	00	01	41	F5
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	35	00	01	10	7F
从站	写	寄存器	寄存器数量	CRC16			

删除文件

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	36	00	01	02	00	01	41	C6
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	36	00	01	E0	7F
从站	写	寄存器	寄存器数量	CRC16			

开机加载

将对应的文件设置为开机加载。同时只能有一个文件被设置成开机加载，例如文件1 是开机加载，若设置文件2 是开机加载则，文件1 的开机加载会被取消，若删除了开机加载的文件，则文件0 会被设置成开机加载。

发送

1	2	3	4	5	6	7	8	9	10	11
01	10	02	37	00	01	02	00	01	40	17
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 内部文件编号, 不支持 U 盘文件

响应

1	2	3	4	5	6	7	8
01	10	02	37	00	01	B1	BF
从站	写	寄存器	寄存器数量	CRC16			

自动保存

设置开机调用文件的自动保功能。若使能了自动保存，手动设置的参数会立刻保存到对应的文件中。

1	2	3	4	5	6	7	8	9	10	11
01	10	02	38	00	01	02	00	01	40	E8
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9:

0000: 禁用自动保存

0001: 使能自动保存

响应

1	2	3	4	5	6	7	8
01	10	02	38	00	01	81	BC
从站	写	寄存器	寄存器数量	CRC16			

4.7 系统设置

页面切换

1	2	3	4	5	6	7	8	9	10	11
01	10	02	39	00	01	02	00	01	80	F9
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9:

0X0000: 测量显示

0X0001: 测量设置
 0X0002: 列表设置
 0X0003: 列表保存
 0X0004: 延时器设置
 0X0005: 延时器保存
 0X0006: 系统设置
 0X0007: 文件管理

响应

1	2	3	4	5	6	7	8
01	10	02	39	00	01	D0	7C
从站	写	寄存器		寄存器数量		CRC16	

语言设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3A	00	01	02	00	00	80	F9
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8B9:

0X0000: ENGLISH

0X0001: CHINESE(Simplified)

响应

1	2	3	4	5	6	7	8
01	10	02	3A	00	01	20	7C
从站	写	寄存器		寄存器数量		CRC16	

日期年设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3B	00	01	02	00	17	C1	15
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8B9: 0x0017 = 23, 年份设置成 23 年

响应

1	2	3	4	5	6	7	8
01	10	02	3B	00	01	71	BC
从站	写	寄存器		寄存器数量		CRC16	

日期月设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3C	00	01	02	00	08	81	7D
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8B9: 0X0008 = 08, 月份设置成 8 月

响应

1	2	3	4	5	6	7	8
01	10	02	3C	00	01	C0	7D
从站	写	寄存器		寄存器数量		CRC16	

日期日设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3D	00	01	02	00	08	80	BB
站号	写	寄存器		寄存器数量		字节	整数		CRC16	

B8B9: 0X0008=8, 日设置为 8

响应

1	2	3	4	5	6	7	8
01	10	02	3D	00	01	91	BD

从站	写	寄存器	寄存器数量	CRC16
----	---	-----	-------	-------

时间时设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3E	00	01	02	00	08	80	88
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 0X0008=8, 时设置为 8

响应

1	2	3	4	5	6	7	8
01	10	02	3E	00	01	20	7C
从站	写	寄存器	寄存器数量	CRC16			

时间分设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	3F	00	01	02	00	1E	80	F9
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 0X001E = 30, 分设为 30 分

响应

1	2	3	4	5	6	7	8
01	10	02	3F	00	01	20	7C
从站	写	寄存器	寄存器数量	CRC16			

时间秒设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	40	00	01	02	00	00	8B	50
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9: 0X0000=0, 秒设置为 0

响应

1	2	3	4	5	6	7	8
01	10	02	40	00	01	01	A5
从站	写	寄存器	寄存器数量	CRC16			

按键音设置

1	2	3	4	5	6	7	8	9	10	11
01	10	02	41	00	01	02	00	00	8A	81
站号	写	寄存器	寄存器数量	字节	整数	CRC16				

B8B9:

0X0000: OFF

0X0001: ON

响应

1	2	3	4	5	6	7	8
01	10	02	41	00	01	50	65
从站	写	寄存器	寄存器数量	CRC16			