
UT3500 系列电池内阻测试仪
SCPI&Modbus 协议
包含型号 UT3562 和 UT3563

目录

目录.....	2
1. SCPI 命令参考.....	6
1.1 握手协议.....	6
1.2 结束符.....	6
1.3 命令串解析.....	7
1.3.1 命令解析规则.....	7
1.3.2 符号约定和定义.....	7
1.3.3 命令树结构.....	7
1.4 命令和参数.....	8
1.4.1 命令.....	8
1.4.2 参数.....	8
1.4.3 分隔符.....	8
1.4.4 错误码.....	9
1.5 Display 显示页面子系统.....	9
1.5.1 DISPLAY:PAGE.....	9
1.5.2 DISPLAY:LINE.....	10
1.6 FUNcTion 测量功能子系统.....	10
1.6.1 FUNcTion 测量参数设置.....	10
1.6.2 FUNcTion:MONitor 监视参数设置.....	11
1.7 RESistance 电阻设置子系统.....	11
1.7.1 RESistance:RANGe 电阻量程设置.....	11
1.7.2 RESistance:RANGe:No 电阻量程号设置.....	11
1.7.3 RESistance:RANGe:MODE 设置电阻比较方式.....	12
1.7.4 RESistance:LiMiT 设置电阻极限.....	12
1.7.5 RESistance:LiMiT:STATe 电阻比较器状态设置.....	12
1.7.6 RESistance:LiMiT:MODE 电阻比较方式.....	13
1.7.7 RESistance:LiMiT:NOMinal 电阻标称值设置.....	13
1.7.8 RESistance:LiMiT:SEQ 电阻直读值极限.....	13
1.7.9 RESistance:LiMiT:ABS 电阻绝对值极限.....	14
1.7.10 RESistance:LiMiT:PER 电阻百分比上下限.....	14
1.8 VOLTage 电压设置子系统.....	15
1.8.1 VOLTage:RANGe 电压量程设置.....	15
1.8.2 VOLTage:RANGe:No 电阻量程号设置.....	15
1.8.3 VOLTage:RANGe:MODE.....	15
1.8.4 VOLTage:LiMiT 设置电阻极限.....	16
1.8.5 VOLTage:LiMiT:STATe 电压比较器状态设置.....	16
1.8.6 VOLTage:LiMiT:MODE 电阻比较方式.....	16
1.8.7 VOLTage:LiMiT:NOMinal 电阻标称值设置.....	17
1.8.8 VOLTage:LiMiT:SEQ 电压直读值极限.....	17
1.8.9 VOLTage:LiMiT:ABS 电压绝对值极限.....	17
1.8.10 VOLTage:LiMiT:PER 电压百分比上下限.....	18

1.9	AUTorange 自动量程子系统*	18
1.9.1	AUTorange 自动量程设置*	18
1.10	ADJust 清零子系统	18
1.10.1	ADJust:CLEAR 取消清零功能	18
1.10.2	ADJust 执行清零	18
1.11	SAMPlE 采样子系统	19
1.11.1	SAMPlE:RATE 测试速度设置	19
1.11.2	SAMPlE:AVERAge 平均次数设置	19
1.12	CALCulate 运算子系统	20
1.12.1	CALCulate:AVERAge:STATe 平均功能*	20
1.12.2	CALCulate:AVERAge 设置平均值*	20
1.12.3	CALCulate:LIMit:STATe 比较器总开关设置*	21
1.12.4	CALCulate:LIMit:BEEPer 讯响设置设置	21
1.12.5	CALCulate:LIMit:RESistance:MODE 电阻比较方式*	21
1.12.6	CALCulate:LIMit:RESistance:UPPer 电阻上限设置*	22
1.12.7	CALCulate:LIMit:RESistance:LOWer 电阻下限设置*	22
1.12.8	CALCulate:LIMit:RESistance:REFerence 电阻标称值设置*	23
1.12.9	CALCulate:LIMit:RESistance:PERCent 电阻百分比极限*	23
1.12.10	CALCulate:LIMit:VOLTage:MODE 电压比较方式*	24
1.12.11	:CALCulate:LIMit:VOLTage:UPPer 电压上限设置*	24
1.12.12	:CALCulate:LIMit:VOLTage:LOWer 电压下限设置*	24
1.12.13	:CALCulate:LIMit:VOLTage:REFerence 电压标称值设置*	25
1.12.14	:CALCulate:LIMit:VOLTage:PERCent 电压百分比极限*	25
1.12.15	:CALCulate:LIMit:ABS 电压绝对偏差比较方式*	26
1.12.16	:CALCulate:STATistics[:STATe] 统计功能	26
1.12.17	CALCulate:STATistics:RESistance:NUMBer? 电阻统计数量	26
1.12.18	CALCulate:STATistics:RESistance:MEAN? 平均值	26
1.12.19	CALCulate:STATistics:RESistance:MAXimum? 最大值	27
1.12.20	CALCulate:STATistics:RESistance:MINimum? 最小值	27
1.12.21	CALCulate:STATistics:RESistance:LIMit? 档计数	27
1.12.22	CALCulate:STATistics:RESistance:DEViation?	27
1.12.23	CALCulate:STATistics:RESistance:Cp?	27
1.12.24	CALCulate:STATistics:VOLTage:NUMBer? 电压统计数量	28
1.12.25	CALCulate:STATistics:VOLTage:MEAN? 平均值	28
1.12.26	CALCulate:STATistics:VOLTage:MAXimum? 最大值	28
1.12.27	CALCulate:STATistics:VOLTage:MINimum? 最小值	28
1.12.28	CALCulate:STATistics:VOLTage:LIMit? 档计数	28
1.12.29	CALCulate:STATistics:VOLTage:DEViation?	29
1.12.30	CALCulate:STATistics:VOLTage:Cp?	29
1.13	LOGger (MEMory) 子系统	29
1.13.1	LOGger[:STATe] 或 MEMory[:STATe] 数据记录和统计模式	29
1.13.2	LOGger:START 或 MEMory:START 数据记录启动	29
1.13.3	LOGger:SIZE 或 MEMory:SIZE 数据记录缓冲区大小设置	30
1.13.4	LOGger:COUNT? 或 MEMory:COUNT? 数据缓冲区已记录的总数	30
1.13.5	LOGger:DATA? 或 MEMory:DATA? 数据缓冲区数据	30

1.14	SYSTem 子系统.....	31
1.14.1	SYSTem:LANGUage 系统语言.....	31
1.14.2	SYSTem:TIME 系统时间设置.....	31
1.14.3	SYSTem:KEYLock 或 SYSTem:KLOCK 键盘锁设置.....	31
1.14.4	SYSTem:CODE 错误码返回.....	32
1.14.5	SYSTem:BEEPer 按键音.....	32
1.14.6	SYSTem:SHAKhand 或 SYSTem:HEADer 通讯握手指令 (数据头返回).....	32
1.14.7	SYSTem:CURRent 电流输出设置.....	32
1.14.8	SYSTem:CALibration 仪器自校准功能.....	33
1.14.9	SYSTem:CALibration:AUTO 仪器自校准开关.....	33
1.14.10	SYSTem:RESult 测试结果发送.....	33
1.14.11	SYSTem:DATAout 测试结果发送*.....	33
1.14.12	SYSTem:BACKup 保存测量参数到当前文件里*.....	34
1.14.13	SYSTem:RESet 出厂设置.....	34
1.15	TRIGger 子系统.....	34
1.15.1	TRIGger:[IMMediate].....	34
1.15.2	TRIGger:SOURce.....	34
1.15.3	TRIGger:DELAy.....	34
1.15.4	TRIGger:DELAy:STATe.....	34
1.15.5	TRG.....	35
1.16	FETCh 和 READ 子系统.....	35
1.16.1	FETCh? 或 READ?获取测量数据.....	35
1.16.2	FETCh:FULL? 或 READ:FULL? 获取完整测量数据.....	35
1.17	CORRection 子系统.....	36
1.17.1	CORRection:SHORt.....	36
1.18	FILE(MMEM) 子系统.....	36
1.18.1	FILE:SAVE 保存文件.....	36
1.18.2	FILE:LOAD 读取文件.....	36
1.18.3	FILE:DELeTe 删除指定文件.....	36
1.19	IDN? 子系统.....	37
1.20	ERRor 子系统.....	37
1.21	SAV 子系统.....	37
2.	Modbus (RTU) 通讯协议.....	38
2.1	数据格式.....	38
2.1.1	指令帧.....	38
2.1.2	CRC-16 计算方法.....	39
2.1.3	响应帧.....	39
2.1.4	无响应.....	40
2.1.5	错误码.....	40
2.2	功能码.....	41
2.3	寄存器.....	41
2.4	读出多个寄存器.....	41
2.5	写入多个寄存器.....	42
2.6	回波测试.....	43
3.	Modbus (RTU) 指令集.....	44

3.1	寄存器总览	44
3.2	获取测量数据	46
3.2.1	获取测量结果	46
3.2.2	获取比较器结果【2004】	47
3.3	参数设置	47
3.3.1	功能寄存器【3000】	47
3.3.2	电阻量程寄存器【3001】	48
3.3.3	电压量程寄存器【3002】	48
3.3.4	电阻量程方式寄存器【3003】	49
3.3.5	电压量程方式寄存器【3004】	49
3.3.6	测试速度寄存器【3005】	50
3.3.7	平均次数寄存器【3006】	50
3.3.8	触发方式寄存器【3007】	51
3.3.9	触发延时寄存器【3008】	51
3.3.10	触发沿寄存器【3009】	52
3.3.11	自校准开关寄存器【300A】	52
3.3.12	电流模式寄存器【300B】	53
3.3.13	文件开机调用寄存器【300C】	53
3.3.14	自动保存【300D】	54
3.4	比较器设置	55
3.4.1	电阻比较器状态寄存器【3100】	55
3.4.2	电压比较器状态寄存器【3101】	55
3.4.3	电阻比较器方式寄存器【3102】	56
3.4.4	电阻比较器方式寄存器【3103】	56
3.4.5	讯响寄存器【3104】	57
3.4.6	电阻标称值寄存器【3110】	57
3.4.7	电压标称值寄存器【3112】	58
3.4.8	电阻极限值寄存器【3114-3117】	58
3.4.9	电压极限值寄存器【3184-3187】	59
3.5	文件操作	59
3.5.1	保存到当前文件【4000】	59
3.5.2	保存到指定文件【4008】	60
3.5.3	重新载入当前文件【4010】	60
3.5.4	载入指定文件【4018】	60
3.6	清零【5000】	61

1. SCPI 命令参考

本章包括以下几方面的内容：

- 命令解析器——了解命令解析器的一些规则。
- 命令语法——命令行的书写规则
- 查询语法——查询命令的书写规则
- 查询响应——查询响应的格式
- 命令参考

本章节提供了仪器使用的所有的 SCPI 命令，通过这些 SCPI 命令，可以完全控制仪器所有功能。

1.1 握手协议

由于仪器使用了 RS-232 标准的最小子集，不使用硬件握手信号，因此为了减小通讯中可能的数据丢失或数据错误的现象，仪器可启用软件握手，高级语言软件工程师应严格按以下握手协议，进行计算机通讯软件的编制：

- 仪器命令解析器只接受 ASCII 格式，命令响应也返回 ASCII 码。
- **主机发送的命令串必须以结束符结尾，仪器命令解析器在收到结束符后才开始执行命令串。**
- 仪器可设置指令握手：仪器在每接受到一个字符后，立即将该字符回送给主机，主机只有接收到这个回送字符后才能继续发送下一个字符。

提示：如果主机无法接受到仪器返回的数据，您可以使用以下方法来试图解决：

1. 软件握手被关闭，请参考仪器<系统设置>页将其开启。
 2. 串行口连接故障，请查看电缆连接。
 3. 计算机端高级语言程序通信格式错误。请试着检查串行口端口号、通信格式是否正确以及波特率是否和仪器设置的相同。
 4. 如果仪器正在解析上次命令，主机也无法接受到仪器的响应，请稍候再试。
- <问题仍无法解决，请立即咨询优利德仪器技术工程师>



1.2 结束符

仪器支持 4 种结束符：

LF (十六进制：0x0A)

CR(十六进制：0x0D)

CR+LF (十六进制：0x0D 0x0A)

NUL (十六进制：0x00)

结束符可以在系统配置页中进行选择，仪器出厂默认为 LF。

注意：

仪器允许主机发送的指令不带结束符，但建议指令末尾加上结束符，否则会导致每次接收指令后增加超时等待（根据波特率不同，指令超时在 10ms~50ms）。



1.3 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。

1.3.1 命令解析规则

1. 命令解析器只对 ASCII 码数据进行解析和响应。
2. **SCPI 命令串必须以 NL(' \n' ASCII 0x0A)为结束符**，命令解析器在收到结束符后或缓冲区溢出才开始执行命令串。
3. 如果指令握手打开，命令解析器在每接受到一个字符后，立即将该字符回送给主机，主机只有接收到这个回送字符后才能继续发送下一个字符。
4. 命令解析器在解析到错误后，立即终止解析，当前指令作废。
5. 命令解析器在解析到查询命令后，终止本次命令串解析，其后字符串被忽略。
6. 命令解析器对命令串的解析不区分大小写。
7. 命令解析器支持命令缩写形式，缩写规格参见之后章节。

1.3.2 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。

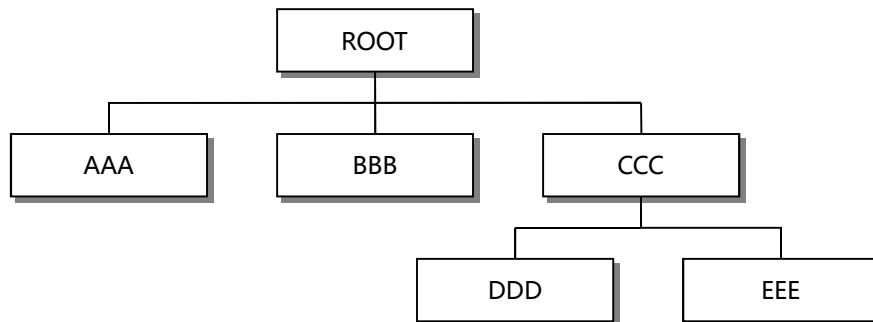
<>	尖括号中的文字表示该命令的参数
[]	方括号中的文字表示可选命令
{}	当大括号包含几个参数项目时，表示只能从中选择一个项目。
()	参数的缩写形式放在小括号中。
大写字母	命令的缩写形式。

1.3.3 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 1-3-3

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

1.4 命令和参数

一条命令由 **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]

1.4.1 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。

1.4.2 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”。
例如：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. *<Scioat>*: 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 1-4-2

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。

1.4.3 分隔符

仪器命令解析器只接受允许的分隔符，除此之外的分隔符命令解析器将产生 “Invalid separator(非法

BinSETup (BSET)	比较器设置页
CORRection (CSET)	短路清零页
CATALog (FILE)	文件页
SYSTem	系统配置页
SYSTEMINFO (SINF)	系统信息页

例如: 发送> disp:page setup<NL> //切换到设置页面

查询语法: DISP:PAGE?

查询响应: <页面名称>缩写
 Test
 enla
 mset
 bset
 cset
 cata
 syst
 sinf

例如: 发送> disp:page?
 返回> Test
 发送> disp:page Test;page?
 返回> Test

1.5.2 DISPLAY:LINE

DISP:LINE 用来在页面底部的提示栏显示一串文本。文本最多可以显示 30 个字符，文本将停留 10s。

命令语法: DISPLAY:LINE <string>

参数: <string> 最多 30 个字符

例如: 发送> DISP:LINE "This is a Comment."

查询语法: DISPLAY:LINE?

查询响应: 屏幕上提示栏文本，如果提示栏是空的，将返回 NULL。

1.6 FUNCTION 测量功能子系统

图 1-6 FUNCTION 子系统树

FUNCTION	{RV, RESistance (R), VOLTage (V)}	
	:MON	{RABS, RPER, VABS, VPER, OFF}

1.6.1 FUNCTION 测量参数设置

命令语法: FUNCTION {RV, RESistance (R), VOLTage (V)}

参数: RV 电阻和电压测量功能
 RESistance 或 R 仅电阻测量功能
 VOLTage 或 V 仅电压测量功能

例如: 发送> FUNC RES //选择电阻测量功能
 发送> FUNC R //选择电阻测量功能
 发送> FUNC V //选择电压测量功能
 发送> FUNC RV //选择电阻+电压测量功能

查询语法: FUNCTION?

查询响应: RV
 RESistance
 VOLTage

例如: 发送> FUNC?
 接收> RESISTANCE

1.6.2 FUNction:MONitor 监视参数设置

命令语法: **FUNction:MONitor {OFF,RABS,RPER,VABS,VPER}**

参数: **OFF** 监视功能关闭
RABS 电阻绝对偏差 (R Δ) 值
RPER 电阻相对偏差 (R%) 值
vABS 电压绝对偏差 (V Δ) 值
vPER 电压相对偏差 (V%) 值

例如: 发送> **FUNC:MON RPER** //监视参数设置为电阻相对偏差值(R%)

查询语法: **FUNction:MONitor?**

查询响应: **{OFF,RABS,RPER,VABS,VPER}**

例如: 发送> **FUNC:MON?**
 接收> **OFF**

1.7 RESistance 电阻设置子系统

图 1-7 RESistance 子系统树

RESistance	:RANGe	<0~3.1k>	
		:NO	{0,1,2,3,4,5,6}
		:MODE	{AUTO,NOM,HOLD}
	:LiMiT	:STATE	{ON,OFF}
		:MODE	{SEQ,PER,ABS}
		:NOminal	<float>
		:SEQ	<lower>,<upper>
		:PER	<lower>,<upper>
		:ABS	<lower>,<upper>
			<lower>,<upper>

1.7.1 RESistance:RANGe 电阻量程设置

根据电阻值设置量程。

命令语法: **RESistance:RANGe <0 to 3100>**

参数: <0-3100> 浮点数表示的电阻值

例如: 发送> **:RES:RANG 100E-3** //选择 100m Ω 所在量程
 发送> **:RES:RANG 10m** //选择 10m Ω 所在量程

查询语法: **:RESistance:RANGe?**

查询响应: 返回测试量程:

3.0000E-3, 30.000E-3, 300.00E-3, 3.0000E+0, 30.000E+0, 300.00E+0, 3.0000E+3

例如: 发送> **RES:RANG?**
 接收> **300.00E-3**
 发送> **RES:100m;RANG?** //设置量程并查询结果
 接收> **300.00E-3**

1.7.2 RESistance:RANGe:No 电阻量程号设置

RES:RANG:NO 用来设置量程号

命令语法: **RESistance:RANGe:No {<量程号>,<min>,<max>}**

参数: 其中, <量程号> 0~6

min 最小量程=0

max 最大量程=6

例如: 发送> **RES:RANG:NO 3<NL>** //切换到 3 量程 (3 Ω)

查询语法: RES:RANG:NO?

查询响应: 量程号 0~6

例如: 发送> RES:RANGE:NO?
返回> 5
发送> RES:RANGE:NO 2;NO? //发送指令并查询结果
返回> 2

1.7.3 RESistance:RANGe:MODE 设置电阻比较方式

RES:RANG:MODE 用来设置量程方式

命令语法: RESistance:RANGe:MODE {AUTO,HOLD,NOMinal}

例如: 发送> RES:RANG:MODE AUTO //切换到自动量程方式

查询语法: RES:RANG:MODE?

查询响应: {AUTO,HOLD,NOM}

例如: 发送> RES:RANGE:MODE?
返回> AUTO
发送> RES:RANGE:MODE AUTO;MODE? //发送指令并查询结果
返回> AUTO

注意: 如果设置为 NOM 方式, 仪器会根据比较器方式设置量程:
当比较器方式为 SEQ 时, 会根据比较器上限设置量程;
当比较器方式为 ABS 和 PER 时, 将根据标称值设置量程。

1.7.4 RESistance:LiMiT 设置电阻极限

设置当前比较方式下的电阻上下限。

命令语法: RESistance:LiMiT <lower,upper>
RESistance:LMT <lower,upper>

参数: lower: 上限浮点数
upper: 下限浮点数

数据值对应当前使用的比较方式, 直读比较 (SEQ) 和绝对值 (ABS) 时代表电阻值 (Ω), 百分比 (PER) 比较时代表百分比值 (%)。

例如: 发送> RES:LMT 1,10
发送> RES:LMT 10m,100m
发送> RES:LMT 1e-3,1e-2

查询语法: RESistance:LiMiT?

查询响应: <lower>,<upper> 当前数据值对应当前比较方式下的上下限值。

数据格式: $\pm\#.####E\pm\#$

每个数据的总位数 10 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E-3(m), E+0, E+3(k) 等格式以对应倍率单位值。

例如: 发送> RES:LMT?
接收> +1.0000E-3,+10.000E-3
发送> RES:LMT 10m,12m;LMT?
接收> +10.000E-3,+12.000E-3

1.7.5 RESistance:LiMiT:STATe 电阻比较器状态设置

电阻比较器打开或关闭设置。

命令语法: RESistance:LiMiT:STATe {ON/1,OFF/0}

例如: 发送> RES:LMT:STAT OFF

查询语法: RESistance:LiMiT:STATe?

查询响应: {on,off}

例如: 发送> RES:LMT:STAT?
接收> on

1.7.6 RESistance:LiMiT:MODE 电阻比较方式

电压比较方式设置。

命令语法: **RESistance:LiMiT:MODE {SEQ,PER,ABS}**

参数: SEQ: 上下限直读比较方式
PER: 百分比比较方式 (相对偏差比较)
ABS: 绝对偏差比较 (Δ)

例如: 发送> RES:LMT:MODE PER //百分比比较

查询语法: **RESistance:LiMiT:MODE?**

查询响应: {SEQ,PER,ABS}

例如: 发送> RES:LMT:MODE?
接收> SEQ

1.7.7 RESistance:LiMiT:NOMinal 电阻标称值设置

设置电阻标称值, 标称值只有在 ABS 和 PER 模式下才参与运算。

命令语法: **RESistance:LiMiT:NOMinal <float>**

参数: float: 标称值浮点数, 单位 (Ω)

例如: 发送> RES:LMT:NOM 12.345m
发送> RES:LMT:NOM 12.345e-3

查询语法: **RESistance:LiMiT:NOMinal?**

查询响应: <float>

数据格式: $\pm\#.####e\pm\#$

每个数据的总位数 10 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E-3, E+0, E+3 等格式以对应单位值。

例如: 发送> RES:LIM:NOM?
接收> +100.00e-3 //100.00m Ω
发送> RES:LIM:NOM 100.00m;NOM? //设置标称值并且查询
接收> +100.00e-3 //100.00m Ω



注意:

如果当前电阻量程设置为标称量程 (NOM RANGE), 并且电阻比较方式为 PER 或 ABS, 电阻量程会根据标称值切换到最佳量程。

但是电阻比较方式为 SEQ 时, 不会根据标称值切换量程。(SEQ 时将根据上限选择量程。)

1.7.8 RESistance:LiMiT:SEQ 电阻直读值极限

设置当前直读比较方式下的电阻上下限。

命令语法: **RESistance:LiMiT:SEQ <lower,upper>**

参数: lower: 上限浮点数
upper: 下限浮点数

设置直读比较 (SEQ) 的上下限值。

例如: 发送> RES:LMT:SEQ 1m,10m
发送> RES:LMT:SEQ 1e-3,10e-3

查询语法: **RESistance:LiMiT:SEQ?**

查询响应: <lower>,<upper>

数据格式: $\pm\#.####E\pm\#$

每个数据的总位数 11 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E-03, E+00, E+03 等格式以对应单位值。

例如: 发送> RES:LMT:SEQ?

```
接收> +1.0000e-03,+10.000e-03
发送> RES:LMT:SEQ 10m,100m;SEQ? //设置标称值并且查询
接收> +100.00e-03 //100.00mΩ
```



注意:

CALCulate:LIMit:RESistance:SEQ 会将电阻比较方式强制切换到 SEQ 方式。
但是, CALCulate:LIMit:RESistance:SEQ? 指令不会切换比较方式。

1.7.9 RESistance:LiMiT:ABS 电阻绝对值极限

设置当前绝对值比较方式下的电阻上下限。

命令语法: **RESistance:LiMiT:ABS <lower,upper>**

参数: lower: 上限浮点数
upper: 下限浮点数

设置绝对值比较 (ABS) 的上下限值。

例如: 发送> RES:LMT:ABS -1.23m,1.23m
发送> RES:LMT:ABS -1.23e-3,12.3e-3

查询语法: **RESistance:LiMiT:ABS?**

查询响应: <lower>,<upper>

数据格式: ±#.####E±#

每个数据的总位数 10 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E-03, E+00, E+03 等格式以对应单位值。

例如: 发送> RES:LMT:ABS?
接收> -1.2300e-3,+12.300e-3
发送> RES:LMT:ABS -1.23m,12.3m;ABS?
接收> -1.2300e-3,+12.300e-3



注意:

CALCulate:LIMit:RESistance:ABS 会将电阻比较方式强制切换到 ABS 方式。
但是, :CALCulate:LIMit:RESistance:ABS? 指令不会切换比较方式。

1.7.10 RESistance:LiMiT:PER 电阻百分比上下限

设置当前百分比比较方式下的电阻上下限百分比值。

命令语法: **RESistance:LiMiT:PER <lower,upper>**

参数: lower: 上限浮点数
upper: 下限浮点数

设置百分比比较 (PER) 的上下限百分比值。

例如: 发送> RES:LMT:PER -10.1,10.1

查询语法: **RESistance:LiMiT:PER?**

查询响应: <lower>,<upper>

数据格式: ±#.####E+0

每个数据的总位数 10 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E+0 格式。

例如: 发送> RES:LMT:PER?
接收> -10.000E+0,+10.000E+0
发送> RES:LMT:PER -10,10;PER? //设置并查询返回
接收> -10.000E+0,+10.000E+0



注意:

CALCulate:LIMit:RESistance:PER 会将电阻比较方式强制切换到 PER 方式。
CALCulate:LIMit:RESistance:PER? 指令不会切换比较方式。

1.8 VOLTage 电压设置子系统

图 1-8 VOLTage 子系统树

VOLTage	:RANGE	<-300~300>	
		:NO	{0,1,2}
		:MODE	{AUTO,NOM,HOLD}
	:LIMiT	:STATE	{ON,OFF}
		:MODE	{SEQ,PER,ABS}
		:NOMinal	<float>
		:SEQ	<lower>,<upper>
		:PER	<lower>,<upper>
		:ABS	<lower>,<upper>
		<lower>,<upper>	

1.8.1 VOLTage:RANGe 电压量程设置

电压量程设置。

命令语法: `VOLTage:RANGe <-300 to 300>`

参数: <0~3100> 浮点数表示的电压值

例如: 发送> `:VOLT:RANG 10` //选择 10V 所在量程

查询语法: `:VOLTage:RANGe?`

查询响应: 返回测试量程:
6.00000E+0, 60.0000E+0, 300.000E+0

例如: 发送> `VOLT:RANG?`
接收> `60.0000E+0`
发送> `VOLT:RANG 3?`

1.8.2 VOLTage:RANGe:No 电阻量程号设置

VOLT:RANG:NO 用来设置量量程号

命令语法: `VOLTage:RANGe:No {<0~2>,min,max}`

参数: 其中, <量程号> 0~2

`min` 最小量程 = 0

`max` 最大量程 = 2

例如: 发送> `VOLT:RANG:NO 1` //切换到 1 量程

查询语法: `VOLTage:RANGe:NO?`

查询响应: 量程号 0~2

例如: 发送> `VOLT:RANG:NO?`
返回> `1`
发送> `VOLT:RANG:NO 1:NO?` //发送指令并查询结果
返回> `1`

1.8.3 VOLTage:RANGe:MODE

VOLT:RANG:MODE 用来设置量程方式

命令语法: `VOLTage:RANGe:MODE {AUTO,HOLD,NOMinal}`

例如: 发送> `VOLT:RANG:MODE AUTO` //切换到自动量程方式

查询语法: `VOLTage:RANGe:MODE?`

查询响应: {AUTO,HOLD,NOM}

例如: 发送> `VOLT:RANG:MODE?`
返回> `AUTO`

发送> VOLT:RANGE:MODE AUTO;MODE? //发送指令并查询结果

返回> AUTO

注意: 如果设置为 NOM 方式, 仪器会根据比较器方式设置量程:
当比较器方式为 SEQ 时, 会根据比较器上限设置量程;
当比较器方式为 ABS 和 PER 时, 将根据标称值设置量程。

1.8.4 VOLTage:LiMiT 设置电阻极限

设置当前比较方式下的电压上下限。

命令语法: **VOLTage:LiMiT <lower, upper>**

参数: lower: 上限浮点数
upper: 下限浮点数

数据值对应当前使用的比较方式, 直读比较 (SEQ) 和绝对值 (ABS) 时代表电压值 (V), 百分比 (PER) 比较时代表百分比值 (%)。

例如: 发送> VOLTage:LMT 10,20

查询语法: **VOLTage:LiMiT?**

查询响应: <lower>, <upper> 当前数据值对应当前比较方式下的上下限值。

数据格式: ±#.#####E+0

每个数据的总位数 11 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E+0 格式以对应倍率单位值。

例如: 发送> VOLT:LMT?

接收> +10.0000E+0,+20.0000E+0

发送> VOLT:LMT 10,20;LMT?

接收> +10.0000E+0,+20.0000E+0

1.8.5 VOLTage:LiMiT:STATe 电压比较器状态设置

电压比较器打开或关闭设置。

命令语法: **VOLTage:LiMiT:STATe {ON/1,OFF/0}**

例如: 发送> VOLT:LMT:STAT OFF

查询语法: **VOLTage:LiMiT:STATe?**

查询响应: {on, off}

例如: 发送> VOLT:LMT:STAT?

接收> on

1.8.6 VOLTage:LiMiT:MODE 电阻比较方式

电压比较方式设置。

命令语法: **VOLTage:LiMiT:MODE {SEQ, PER, ABS}**

参数: SEQ: 上下限直读比较方式
PER: 百分比比较方式 (相对偏差比较)
ABS: 绝对偏差比较 (Δ)

例如: 发送> VOLT:LMT:MODE PER //百分比比较

查询语法: **VOLTage:LiMiT:MODE?**

查询响应: {SEQ, PER, ABS}

例如: 发送> VOLT:LMT:MODE?

接收> SEQ

1.8.7 VOLTage:LiMiT:NOMinal 电阻标称值设置

设置电压标称值，标称值只有在 ABS 和 PER 模式下才参与运算。

命令语法: **VOLTage:LiMiT:NOMinal <float>**

参数: float: 标称值浮点数, 单位 (V)

例如: 发送> VOLT:LMT:NOM 10.1234

查询语法: **VOLTage:LiMiT:NOMinal?**

查询响应: <float>

数据格式: ±#.#####E+0

每个数据的总位数 11 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E+0 格式以对应单位值。

例如: 发送> VOLT:LIM:NOM?

接收> +10.0000E+0 //10.0000

发送> VOLT:LIM:NOM 3.6;NOM? //设置标称值并且查询

接收> +3.6000E+0



注意:

如果当前电压量程设置为标称量程 (NOM RANGE), 并且电压比较方式为 PER 或 ABS, 电压量程会根据标称值切换到最佳量程。

但是电压比较方式为 SEQ 时, 不会根据标称值切换量程。(SEQ 时将根据上限选择量程。)

1.8.8 VOLTage:LiMiT:SEQ 电压直读值极限

命令语法: **VOLTage:LiMiT:SEQ <lower, upper>**

参数: lower: 上限浮点数

upper: 下限浮点数

设置直读比较 (SEQ) 的上下限值。

例如: 发送> VOLT:LMT:SEQ 1.23456, 3.45678

查询语法: **VOLTage:LiMiT:SEQ?**

查询响应: <lower>, <upper>

数据格式: ±#.#####E+0

每个数据的总位数 11 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E+0 格式。

例如: 发送> VOLT:LMT:SEQ?

接收> +1.23456E+0, +3.45678E+0

发送> VOLT:LMT:SEQ 3.5, 4.2;SEQ? //设置并查询设置值

接收> +3.5000E+0, +4.2000E+0



注意

CALCulate:LIMit:VOLTage:SEQ 会将电压比较方式强制切换到 SEQ 方式。

CALCulate:LIMit: VOLTage:SEQ? 指令不会切换比较方式。

1.8.9 VOLTage:LiMiT:ABS 电压绝对值极限

命令语法: **VOLTage:LiMiT:ABS <lower, upper>**

参数: lower: 上限浮点数

upper: 下限浮点数

设置绝对值比较 (ABS) 的上下限值。

例如: 发送> VOLTage:LiMiT:ABS -1.2, 1.2

查询语法: **VOLTage:LiMiT:ABS?**

查询响应: <float>, <float>

数据格式: ±#.#####E+0

每个数据的总位数 11 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E+00 格式。

例如: 发送> VOLT:LMT:ABS?
接收> -1.23456E+0,+1.23456E+0
发送> VOLT:LMT:ABS -12,12;ABS? //设置并查询设置值
接收> -12.0000E+0,+12.0000E+0



注意

VOLTage:LiMiT:ABS 会将电压比较方式强制切换到 ABS 方式。
VOLTage:LiMiT:ABS? 指令不会切换比较方式。

1.8.10 VOLTage:LiMiT:PER 电压百分比上下限

命令语法: **VOLTage:LiMiT:PER <lower,upper>**

参数: lower: 上限浮点数
upper: 下限浮点数

设置百分比比较 (PER) 的上下限百分比值。

例如: 发送> :CALC:LIM:VOLT:PER -1,1

查询语法: **VOLTage:LiMiT:PER?**

查询响应: <float>,<float>

数据格式: ±#.#####E+0

每个数据的总位数 11 位, 第一位为符号位, 小数位置不固定, 为了兼容 PLC 等设备, 指数部分只使用 E+00 格式以对应单位值。

例如: 发送> VOLT:LMT:PER?
接收> -10.0000E+00,+10.0000E+00
发送> VOLT:LMT:PER -10,10;PER? //设置并查询设置值
接收> -10.0000E+00,+10.0000E+00

1.9 AUTorange 自动量程子系统*

(*新设计不建议使用)

1.9.1 AUTorange 自动量程设置*

自动量程设置, 此指令同时设置电压和电阻量程方式。单独设置量程方式, 请参考 RES:RANG:MODE 和 VOLT:RANG:MODE 指令。

命令语法: **AUTorange {1,0,ON,OFF}**

例如: 发送> AUT ON //切换到自动量程方式

查询语法: **AUTorange?**

查询响应: {ON,OFF}

1.10 ADJust 清零子系统

1.10.1 ADJust:CLEAR 取消清零功能

关闭清零功能。

命令语法: **:ADJust:CLEAR**

例如: 发送> :ADJ:CLEA

1.10.2 ADJust 执行清零

在发送此指令前, 必须将测试夹短路。

命令语法: **ADJust**

响应:	{0,1}
	0: 清零成功
	1: 清零失败
例如:	发送> :ADJ //开始执行清零 接收> 0 //清零成功
查询语法:	:ADJust?
查询响应:	{0,1}
	0: 最近一次清零成功
	1: 最近一次清零失败



注意

清零需要一定时间，所有返回值需要稍等一段时间才能返回。在自动量程下清零大概需要6s。

1.11 SAMPlE 采样子系统

SAMPLE	:RATE	{SLOW, MEDium, FAST, EXFAST}
	:AVERage	<integer 0~256>

1.11.1 SAMPlE:RATE 测试速度设置

命令语法:	SAMPlE:RATE {SLOW, MEDium, FAST, EXFast}
参数:	SLOW: 慢速 MEDium: 中速 FAST: 快速 EXFAST: 高速
例如:	发送> SAMP:RATE MED
查询语法:	SAMPlE:RATE?
查询响应:	{SLOW, MEDIUM, FAST, EXFAST}
例如:	发送> SAMP:RATE? 接收> FAST 发送> SAMP:RATE MED;RATE? 接收> MED

1.11.2 SAMPlE:AVERage 平均次数设置

平均次数设置。

命令语法:	SAMPlE:AVERage <integer 0~256> SAMPlE:AVG <integer 0~256>
参数:	0 或 1: 平均功能将关闭
例如:	发送> SAMP:AVER 10 发送> SAMP:AVG 5
查询语法:	SAMPlE:AVERage? SAMPlE:AVG?
查询响应:	{0~256}
例如:	发送> SAMP:AVER? 接收> 0 //=-OFF 发送> SAMP:AVER 2;AVER? 发送> 2

1.12 CALCulate 运算子系统

图 1-12 CALCulate 子系统树

CALCulate	:AVERage*	<integer 0~256>		
	:STATe	{OFF}		
	:LIMit	:STATe	{ON,OFF}	
		:BEEPer	{OFF,HL,IN}	
		:RESistance	:MODE	{HL,REF,ABS}
			:UPPer	float <integer>
			:LOWer	<integer>
			:REFerence	<integer>
			:PERCent	<lower,upper>
		:VOLTage	:MODE	{HL,REF,ABS}
			:UPPer	float <integer>
			:LOWer	<integer>
	:PERCent		<lower,upper>	
	:ABS	{ON(0),OFF(1)}		
	:STATistics	: [STATe]	{ON(1),OFF(0)}	
		:RESistance	:NUMBer? (:NO?)	
			:MEAN?	
			:MAXimum?	
			:MINimum?	
			:LIMit? (LMT?)	
			:DEViation?	
			:CP?	
		:VOLTage	:NUMBer? (:NO?)	
:MEAN?				
:MAXimum?				
:MINimum?				
:LIMit? (LMT?)				
:DEViation?				
:CP?				

(*新设计不建议使用)

1.12.1 CALCulate:AVERage:STATe 平均功能*

(*新设计不建议使用)

关闭平均功能，仪器平均次数将设置为 1，此指令建议使用 SAMPLE:AVERage 0 代替。

命令语法: **CALCulate:AVERage:STATe {OFF}**

参数: {OFF}

当设置为 OFF 时，平均次数为 1

例如: 发送 > :CALC:AVER:STAT OFF

查询语法: **CALCulate:AVERage:STATe?**

查询响应: {ON,OFF}

ON: 平均次数 > 1

OFF: 平均次数 = 1



注意

此指令没有打开功能，即 CALC:AVER:STAT ON 无效。开启平均功能请使用 SAMP:AVER <2~256> 代替。

1.12.2 CALCulate:AVERage 设置平均值*

(*新设计不建议使用)

此指令建议使用 SAMPLE:AVERage 代替。

命令语法: **CALCulate:AVERage <0 to 256>**

参数: 整数，输入从 1 到 256 的平均次数

当设置为 1 时, 平均功能将关闭

例如: 发送> CALC:AVER 10

查询语法: CALCulate:AVERage?

查询响应: {0~256}

0 代表平均功能关闭

1.12.3 CALCulate:LIMit:STATe 比较器总开关设置*

(*新设计不建议使用)

比较器开关设置, 此指令会同时设置电压和电阻比较器开关。

命令语法: CALCulate:LIMit:STATe {0,1,ON,OFF}

参数: ON(1): 电阻比较器和电压比较器都打开, 并设置为 SEQ 模式。

OFF(0): 电阻比较器和电压比较器都关闭。

例如: 发送> CALC:LIM:STAT OFF

查询语法: CALCulate:LIMit:STATe?

查询响应: {ON,OFF}

当电阻比较器设置为 OFF, 同时电压比较器设置为 OFF 时, 返回 OFF

例如: 发送> CALC:LIM:STAT?

接收> ON



注意

CALCulate:LIMit:STATe 会将电压和电阻比较器同时开启或关闭。

如果需要单独关闭电阻或电压比较器, 使用 RES:LMT:STATe 和 VOLT:LMT:STATe 指令。

1.12.4 CALCulate:LIMit:BEEPer 讯响设置设置

比较器讯响设置。

命令语法: CALCulate:LIMit:BEEPer {0/OFF,HL/NG/FAIL,IN/OK/PASS}

参数: 0/OFF: 讯响关闭

HL/NG/FAIL: 不合格讯响, 蜂鸣器在不合格时鸣叫

IN/OK/PASS: 合格讯响, 蜂鸣器在合格时鸣叫

例如: 发送> CALC:LIM:BEEP HL //不合格讯响

发送> CALC:LIM:BEEP OK //合格讯响

查询语法: CALCulate:LIMit:BEEPer?

查询响应: {OFF,HL,IN}

例如: 发送> CALC:LIM:BEEP?

接收> IN

1.12.5 CALCulate:LIMit:RESistance:MODE 电阻比较方式*

(*新设计不建议使用)

电阻比较方式设置。此指令建议使用 RES:LMT:MODE 代替。

命令语法: CALCulate:LIMit:RESistance:MODE {HL,REF,ABS}

参数: HL/SEQ: 上下限直读比较方式

REF/PER: 百分比比较方式 (相对偏差%比较)

ABS: 绝对偏差比较 (Δ)

例如: 发送> CALC:LIM:RES:MODE HL //上下限直读值比较

发送> CALC:LIM:RES:MODE REF //百分比比较

查询语法: CALCulate:LIMit:RESistance:MODE?

查询响应: {HL, REF, ABS}

例如: 发送> CALC:LIM:RES:MODE?
接收> HL

1.12.6 CALCulate:LIMit:RESistance:UPPer 电阻上限设置*

(*新设计不建议使用)

建议使用支持浮点数输入的指令 RES:LMT 代替。

命令语法: **CALCulate:LIMit:RESistance:UPPer <0-99999>**

参数: <0-99999>: 正整数, 超过 99999 被强制为 99999, 符号被忽略。

小数和单位与当前量程有关:

量程	小数位数	单位	说明
0 (3mΩ)	4	mΩ	12345 = 1.2345mΩ
1 (30mΩ)	3	mΩ	12345 = 12.345mΩ
2 (300mΩ)	2	mΩ	12345 = 123.45mΩ
3 (3Ω)	4	Ω	12345 = 1.2345 Ω
4 (30Ω)	3	Ω	12345 = 12.345Ω
5 (300Ω)	2	Ω	12345 = 123.45Ω
6 (3kΩ)	4	Ω	12345 = 1.2345Ω

例如: 发送> CALC:LIM:RES:UPPer 12345 //根据上表得到对应值, 量程 1 时代表 1.2345mΩ

查询语法: **CALCulate:LIMit:RESistance:UPPer?**

查询响应: <正整数> #####

位数最大 5 位, 不带符号和小数点

例如: 发送> CALC:LIM:RES:UPP?
接收> 1234 //量程 0 时, 代表 0.1234mΩ
发送> CALC:LIM:RES:UPP 12345;UPP?
接收> 12345 //量程 1 时, 代表 12.345mΩ



注意

此指令发送数值与量程有关系, RES:LMT <lower>, <upper> 可以直接输入实际浮点数数值。
CALC:LIM:RES:UPP 会将电阻比较方式设置为 SEQ 方式。

1.12.7 CALCulate:LIMit:RESistance:LOWer 电阻下限设置*

(*新设计不建议使用)

建议使用支持浮点数输入的指令 RES:LMT 代替。

命令语法: **CALCulate:LIMit:RESistance:LOWer <0-99999>**

参数: <0-99999>: 正整数, 超过 99999 被强制为 99999

小数和单位与当前量程有关:

量程	小数位数	单位	说明
1 (3mΩ)	4	mΩ	12345 = 1.2345mΩ
2 (30mΩ)	3	mΩ	12345 = 12.345mΩ
3 (300mΩ)	2	mΩ	12345 = 123.45mΩ
4 (3Ω)	4	Ω	12345 = 1.2345 Ω
5 (30Ω)	3	Ω	12345 = 12.345Ω
6 (300Ω)	2	Ω	12345 = 123.45Ω
7 (3kΩ)	4	Ω	12345 = 1.2345Ω

例如: 发送> CALC:LIM:RES:LOWer 1000 //根据上表得到对应值, 量程 4 时代表 0.1000Ω

查询语法: **CALCulate:LIMit:RESistance:LOWer?**

查询响应: <正整数> #####

总位数 5 位, 不带符号和小数点

例如: 发送> :CALC:LIM:RES:LOW?

接收> 1000 //返回值根据量程而不同



注意

此指令发送数值与量程有关系, RES:LMT <lower>, <upper> 可以直接输入实际浮点数数值。

CALC:LIM:RES:UPP 会将电阻比较方式设置为 SEQ 方式。

1.12.8 CALCulate:LIMit:RESistance:REFerence 电阻标称值设置*

(*新设计不建议使用)

建议使用支持浮点数输入的指令 RES:LMT:NOM 代替。

命令语法: **CALCulate:LIMit:REFerence <0-99999>**

参数: <0-99999>: 正整数值, 超过 99999 被强制为 99999

小数和单位与当前量程有关:

量程	小数位数	单位	说明
0 (3mΩ)	4	mΩ	12345 = 1.2345mΩ
1 (30mΩ)	3	mΩ	12345 = 12.345mΩ
2 (300mΩ)	2	mΩ	12345 = 123.45mΩ
3 (3Ω)	4	Ω	12345 = 1.2345 Ω
4 (30Ω)	3	Ω	12345 = 12.345Ω
5 (300Ω)	2	Ω	12345 = 123.45Ω
6 (3kΩ)	4	Ω	12345 = 1.2345Ω

例如: 发送> :CALC:LIM:REF 10000 //根据上表得到对应值, 量程 4 时代表 1.0000Ω

查询语法: **:CALCulate:LIMit:RESistance:REFerence?**

查询响应: <正整数> #####

总位数 5 位, 不带符号和小数点

例如: 发送> :CALC:LIM:RES:REF?

接收> 10000 //返回值根据量程而不同, 量程 1 时代表

1.12.9 CALCulate:LIMit:RESistance:PERCent 电阻百分比极限*

(*新设计不建议使用)

建议使用支持浮点数输入的指令 RES:LMT:PER 代替。

命令语法: **CALCulate:LIMit:RESistance:PERCent <%>**

参数: %: 百分比值

设置百分比比较 (PER) 的百分比值。

例如: 发送> :CALC:LIM:RES:PERC 1.1 //设置百分比极限为 -1.1%, 1.1%

查询语法: **CALCulate:LIMit:RESistance:PERCent?**

查询响应: <float> #.###

总位数为 4 位浮点数, 不带符号。

例如: 发送> :CALC:LIM:RES:PERC?

接收> 1.100

发送> :CALC:LIM:RES:PERC?

接收> 1.100



CALCulate:LIMit:RESistance:PERC 会将电压比较方式强制切换到 PER 方式。

CALCulate:LIMit:RESistance:PERC? 指令不会切换比较方式。

1.12.10

CALCulate:LIMit:VOLTage:MODE 电压比较方式*

(*新设计不建议使用)

电压比较方式设置。此指令建议使用 VOLT:LMT:MODE 代替。

命令语法: **CALCulate:LIMit:VOLTage:MODE {OFF,HL/SEQ,REF/PER,ABS}**

参数: OFF: 电压比较器关闭
HL: 上下限直读比较方式
REF: 百分比比较方式 (相对偏差比较)
ABS: 绝对偏差比较 (Δ)

例如: 发送> :CALC:LIM:VOLT:MODE HL //上下限直读值比较
发送> :CALC:LIM:VOLT:MODE OFF //电阻比较器关闭
发送> :CALC:LIM:VOLT:MODE REF //百分比比较
发送> :CALC:LIM:VOLT:MODE PER //百分比比较

查询语法: **CALCulate:LIMit:VOLTage:MODE?**

查询响应: {OFF,HL,REF,ABS}

例如: 发送> :CALC:LIM:VOLT:MODE?
接收> HL



注意: :CALCulate:LIMit:VOLTage:PER 会将电压比较方式强制切换到 PER 方式。
:CALCulate:LIMit: VOLTage:PER? 指令不会切换比较方式。

1.12.11

:CALCulate:LIMit:VOLTage:UPPer 电压上限设置*

(*新设计不建议使用)

此指令建议使用 VOLT:LMT <lower>,<upper>代替。

命令语法: **:CALCulate:LIMit:VOLTage:UPPer <0-999999>**

参数: <0-999999>: 正整数数值, 超过 999999 被强制为 999999, 符号被忽略。
小数和单位与当前量程有关:

量程	小数位数	说明
0 (8V)	5	123456 = 1.23456V
1 (80V)	4	123456 = 12.3456V
2 (400V)	3	123456 = 123.456V

例如: 发送> :CALC:LIM:VOLT:UPPer 123456 //根据上表得到对应值, 量程 1 时代表 1.23456V

查询语法: **:CALCulate:LIMit:VOLTage:UPPer?**

查询响应: <正整数> #####

位数最大 6 位, 不带符号和小数点

例如: 发送> :CALC:LIM:VOLT:UPP?
接收> 123456 //量程 2 时, 代表 123.456V
发送> :CALC:LIM:VOLT:UPP 12345;UPP?
接收> 12345 //量程 2 时, 代表 12.345V



注意

此指令发送数值与量程有关系, VOLT:LMT <lower>,<upper> 可以直接输入实际浮点数数值。
CALC:LIM:VOLT:UPP 会将电压比较方式设置为 SEQ 方式。

1.12.12

:CALCulate:LIMit:VOLTage:LOWer 电压下限设置*

(*新设计不建议使用)

此指令建议使用 VOLT:LMT <lower>,<upper>代替。

命令语法: `:CALCulate:LIMit:VOLTage:LOWer <0-999999>`

参数: `<0-999999>`: 正整数数值, 超过 999999 被强制为 999999, 符号被忽略。

小数和单位与当前量程有关:

量程	小数位数	说明
0	5	123456 = 1.23456V
1	4	123456 = 12.3456V
2	3	123456 = 123.456V

例如: 发送 > `:CALC:LIM:VOLT:LOW 100000` //根据上表得到对应值, 量程 1 时代表 1.00000V

查询语法: `:CALCulate:LIMit:VOLTage:LOWer?`

查询响应: `<正整数> #####`

位数最大 6 位, 不带符号和小数点

例如: 发送 > `:CALC:LIM:VOLT:LOW?`

接收 > 100000 //量程 1 时, 代表 1.00000V



注意

此指令发送数值与量程有关系, `VOLT:LMT <lower>, <upper>` 可以直接输入实际浮点数数值。
`CALC:LIM:VOLT:LOW` 会将电压比较方式设置为 SEQ 方式。

1.12.13 :CALCulate:LIMit:VOLTage:REference 电压标称值设置*

(*新设计不建议使用)

此指令建议使用 `VOLT:LMT:NOM` 代替

命令语法: `:CALCulate:LIMit:VOLTage:REference <0-999999>`

参数: `<0-999999>`: 正整数数值, 超过 999999 被强制为 999999, 符号被忽略。

小数和单位与当前量程有关:

量程	小数位数	说明
0	5	123456 = 1.23456V
1	4	123456 = 12.3456V
2	3	123456 = 123.456V

例如: 发送 > `:CALC:LIM:VOLT:REF 100000` //根据上表得到对应值, 量程 2 时代表 10.0000V

查询语法: `:CALCulate:LIMit:VOLTage:REference?`

查询响应: `<正整数> #####`

位数最大 6 位, 不带符号和小数点

例如: 发送 > `:CALC:LIM:VOLT:REF?`

接收 > 100000 //量程 2 时, 代表 10.0000V

1.12.14 :CALCulate:LIMit:VOLTage:PERCent 电压百分比极限*

(*新设计不建议使用)

此指令建议使用 `VOLT:LMT:PER <lower>, <upper>` 代替。

命令语法: `:CALCulate:LIMit:VOLTage:PERCent <float>`

参数: %: 百分比值

设置百分比比较 (PER) 的百分比值。

例如: 发送 > `:CALC:LIM:VOLT:PERC 0.3` //设置百分比极限为 -0.3%, 0.3%

查询语法: `:CALCulate:LIMit:VOLT:PERCent?`

查询响应: `<float>`

##.### 不带符号位, 当前数据值对应的上限值

例如: 发送 > `:CALC:LIM:VOLT:PERC?`

接收 > 0.300



注意

:CALCulate:LIMit:VOLTage:PERC 会将电压比较方式强制切换到 PER 方式。
:CALCulate:LIMit: VOLTage:PERC? 指令不会切换比较方式。

1.12.15 :CALCulate:LIMit:ABS 电压绝对偏差比较方式*

(*新设计不建议使用)

此指令不建议使用，使用等价指令 VOLT:LMT:MODE ABS

命令语法: :CALCulate:LIMit:ABS {ON(1),OFF(0)}

参数: ON: 设置为绝对偏差比较方式
OFF: 设置为相对偏差 (百分比) 比较方式。

例如: 发送> :CALC:LIM:ABS ON //设置为 ABS 比较方式

查询语法: :CALCulate:LIMit:ABS?

查询响应: {on,off}
on: 绝对偏差方式
off: 其它比较方式 (可能是 OFF,SEQ,或 PER)

例如: 发送> :CALC:LIM:ABS?
接收> OFF



注意

此指令与 VOLT:LMT:MODE ABS 功能相同。
此指令仅对电压有效。

1.12.16 :CALCulate:STATistics[:STATe] 统计功能

打开或关闭统计功能。

命令语法: CALCulate:STATistic[:STATE] {LOG,STAT}

参数: ON: 打开统计功能
OFF: 统计功能关闭

例如: 发送> CALC:STAT LOG

查询语法: CALCulate:STATistic[:STAT]?

查询响应: {LOG,STAT}

例如: 发送> CALC:STAT LOG;STAT?
接收> LOG



注意

如果当前触发方式是内部，请使用 LOG:START ON 指令执行记录和进行统计运算。
如果当前触发方式为外部时，请使用 TRIG 按键执行记录和统计。

1.12.17 CALCulate:STATistics:RESistance:NUMBer? 电阻统计数量

查询电阻统计数量。

查询语法: CALCulate:STATistic:RESistance:NUMBer?
CALCulate:STATistic:RESistance:NUM?
CALCulate:STATistic:RESistance:NO?

查询响应: <总数 (整数) >, <有效数量 (整数) >

例如: 发送> CALC:STAT:RES:NUM?
接收> 10,8 //总共记录 10 个数值, 8 个数值有效并用于统计



注意

有效数量: 不包括溢出 (OF) 或错误 (FAULT) 的数量, 只要屏幕上能显示出数值即认为数据有效。

1.12.18 CALCulate:STATistics:RESistance:MEAN? 平均值

平均值查询。

查询语法: CALCulate:STATistic:RESistance:MEAN?

查询响应: <平均值 (浮点数) >

例如: 发送> CALC:STAT:RES:MEAN?
接收> +1.2568E-3



注意

$$\text{平均值 } \bar{x} = \frac{\sum x}{n}$$

1.12.19 CALCulate:STATistics:RESistance:MAXimum? 最大值

最大值查询。

查询语法: CALCulate:STATistic:RESistance:MAXimum?

查询响应: <最大值(浮点数)>, <位号>

例如: 发送> CALC:STAT:RES:MAX?
接收> +354.76E+0,2 //位于第2个数据为最大值

1.12.20 CALCulate:STATistics:RESistance:MINimum? 最小值

最小值查询。

查询语法: CALCulate:STATistic:RESistance:MIMimum?

查询响应: <最小值(浮点数)>, <位号>

例如: 发送> CALC:STAT:RES:MIN?
接收> +354.33E+0,7 //第7个数据为最小值

1.12.21 CALCulate:STATistics:RESistance:LIMit? 档计数

查询计数值。

查询语法: CALCulate:STATistic:RESistance:LIMit?
CALCulate:STATistic:RESistance:LMT?

查询响应: <HI 计数>, <OK 计数>, <LO 计数>, <FAULT 计数>

例如: 发送> CALC:STAT:RES:LMT?
接收> 10,0,0,0 //10个数据未上超不合格
发送> CALC:STAT:RES:LIM?
接收> 0,10,0,0 //10个数据合格



注意

查询比较器档计数时, 请确保比较器功能启用, 否则数据将返回 0,0,0,0

1.12.22 CALCulate:STATistics:RESistance:DEVIation? 标准偏差

查询标准偏差值。

查询语法: CALCulate:STATistic:RESistance:DEVIation?

查询响应: <母体标准差 σ_n >, <样本标准差 σ_{n-1} >

例如: 发送> CALC:STAT:RES:DEV?
接收> 0.0016, 0.0017 // $\sigma_n=0.0016$ $s=0.0017$



注意

$$\text{母体标准差: } \sigma_n = \sqrt{\frac{\sum(x-\bar{x})^2}{n}} = \sqrt{\frac{\sum x^2 - n\bar{x}^2}{n}}$$

$$\text{样本标准差: } \sigma_{n-1} = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}} = \sqrt{\frac{\sum x^2 - n\bar{x}^2}{n-1}}$$

1.12.23 CALCulate:STATistics:RESistance:Cp? 工序能力指数

查询工序能力指数 (Cp/Cpk)。

查询语法: CALCulate:STATistic:RESistance:CP?

查询响应: <工序能力指数 (偏差) C_p >, <工序能力指数 (偏移) C_{pk} >

例如: 发送> CALC:STAT:RES:CP?
接收> 99.85, 75.56 // $C_p = 99.85$, $C_{pk} = 75.56$



注意

工序能力指数 (偏差) $Cp = \frac{|Hi-Lo|}{6\sigma_{n-1}}$
 工序能力指数 (偏移) $CpK = \frac{|Hi-Lo|-|Hi+Lo-2\bar{x}|}{6\sigma_{n-1}}$

1.12.24 CALCulate:STATistics:VOLTage:NUMBer? 电压统计数量

查询电阻统计数量。

查询语法: `CALCulate:STATistic:VOLTage:NUMBer?`
`CALCulate:STATistic:VOLTage:NUM?`
`CALCulate:STATistic:VOLTage:NO?`

查询响应: <总数 (整数)>, <有效数量 (整数)>

例如: 发送 > CALC:STAT:VOLT:NUM?
 接收 > 10, 10 // 总共记录 10 个数值, 10 个数值有效并用于统计



注意

有效数量: 不包括溢出 (OF) 或错误 (FAULT) 的数量, 只要屏幕上能显示出数值即认为数据有效。

1.12.25 CALCulate:STATistics:VOLTage:MEAN? 平均值

平均值查询。

查询语法: `CALCulate:STATistic:VOLTage:MEAN?`

查询响应: <平均值 (浮点数)>

例如: 发送 > CALC:STAT:VOLT:MEAN?
 接收 > +3.70601E+0



注意

平均值 $\bar{x} = \frac{\sum x}{n}$

1.12.26 CALCulate:STATistics:VOLTage:MAXimum? 最大值

最大值查询。

查询语法: `CALCulate:STATistic:VOLTage:MAXimum?`

查询响应: <最大值 (浮点数)>, <位号>

例如: 发送 > CALC:STAT:VOLT:MAX?
 接收 > +3.70890E+0, 4 // 位于第 4 个数据为最大值 3.70890V

1.12.27 CALCulate:STATistics:VOLTage:MINimum? 最小值

最小值查询。

查询语法: `CALCulate:STATistic:VOLTage:MIMimum?`

查询响应: <最小值 (浮点数)>, <位号>

例如: 发送 > CALC:STAT:VOLT:MIN?
 接收 > +3.70566E+0, 4 // 第 4 个数据为最小值 3.70566V

1.12.28 CALCulate:STATistics:VOLTage:LIMit? 档计数

查询计数值。

查询语法: `CALCulate:STATistic:VOLTage:LIMit?`
`CALCulate:STATistic:VOLTage:LMT?`

查询响应: <HI 计数>, <OK 计数>, <LO 计数>, <FAULT 计数>

例如: 发送 > CALC:STAT:VOLT:LMT?
 接收 > 0, 10, 0, 0 // 10 个数据合格



注意

查询比较器档计数时, 请确保比较器功能启用, 否则数据将返回 0, 0, 0, 0

1.12.29 CALCulate:STATistics:VOLTage:DEVIation?

查询标准偏差值。

查询语法: **CALCulate:STATistic:VOLTage:DEVIation?**

查询响应: <母体标准差 σ_n >, <样本标准差 σ_{n-1} >

例如: 发送> CALC:STAT:VOLT:DEV?
接收> 0.0002, 0.0002 // $\sigma_n=0.0002$ $s=0.0002$



注意

$$\text{母体标准差: } \sigma_n = \sqrt{\frac{\sum(x-\bar{x})^2}{n}} = \sqrt{\frac{\sum x^2 - n\bar{x}^2}{n}}$$

$$\text{样本标准差: } \sigma_{n-1} = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}} = \sqrt{\frac{\sum x^2 - n\bar{x}^2}{n-1}}$$

1.12.30 CALCulate:STATistics:VOLTage:Cp?

查询工序能力指数 (Cp/Cpk)。

查询语法: **CALCulate:STATistic:VOLTage:CP?**

查询响应: <工序能力指数 (偏差) C_p >, <工序能力指数 (偏移) C_{pk} >

例如: 发送> CALC:STAT:VOLT:CP?
接收> 72.110 , 8.6692 // $C_p = 72.110$, $C_{pk} = 8.6692$



注意

$$\text{工序能力指数 (偏差) } C_p = \frac{|Hi-Lo|}{6\sigma_{n-1}}$$

$$\text{工序能力指数 (偏移) } C_{pk} = \frac{|Hi-Lo|-|Hi+Lo-2\bar{x}|}{6\sigma_{n-1}}$$

1.13 LOGger (MEMory) 子系统

LOGger (MEMory) 子系统用来设置和读取数据缓冲区数据。

图 1-13

LOGger (MEMory) 子系统树

LOGger MEMory	[: STATE]	<LOG, STAT>
	: START	<ON (1), OFF (0)>
	: SIZE	{<1~10000>, max}
	: COUNT?	<0~10000>
	: DATA?	<1~10000>

1.13.1 LOGger[:STATE] 或 MEMory[:STATE] 数据记录和统计模式

LOGger:STATE 或 MEMory:STATE 可以设置数据记录 (LOG) 或数据统计 (STAT) 模式。

命令语法: **LOGger[:STATE] {LOG, STAT}**
MEMory[:STATE] {LOG, STAT}

参数: LOG: 数据记录模式
STAT: 数据统计模式

例如: 发送> LOG:STAT STAT

查询语法: **LOGger[:STATE]?**
MEMory[:STATE]?


查询响应: {LOG, STAT}

例如: 发送> LOG:STAT?
接收> LOG
发送> LOG?
接收> LOG

1.13.2 LOGger:START 或 MEMory:START 数据记录启动

LOGger:START 或 MEMory:START 启动/停止数据记录。

命令语法: **LOGger:START {ON (1), OFF (0)}**

	MEMory:START {ON(1),OFF(0)}
参数:	ON: 开始记录 OFF: 停止记录
例如:	发送> LOG:START ON
查询语法:	LOGger:START? MEMory:START?
查询响应:	{on,off}
例如:	发送> LOG:START? 接收> OFF
 注意	此指令只在[数据记录]功能打开后有效。请确认<系统配置>页[数据记录]状态。 如果当前页面不在<测量>页,此指令会自动切换到<测量>页面。

1.13.3 LOGger:SIZE 或 MEMory:SIZE 数据记录缓冲区大小设置

命令语法:	LOGger:SIZE {<1~10000>,max} MEMory:SIZE {<1~10000>,max}
参数:	<1~10000>: 整数,小于 1 的数被强制为 1 max: 缓冲区将设置为 10000
例如:	发送> LOG:SIZE max 发送> MEM:SIZE 1000
查询语法:	LOGger:SIZE? MEMory:STATE?
查询响应:	{1~10000}
例如:	发送> LOG:SIZE 100;SIZE? 接收> 100 发送> MEM:SIZE? 接收> 200

1.13.4 LOGger:COUNt? 或 MEMory:COUNt? 数据缓冲区已记录的总数

查询语法:	LOGger:COUNt? MEMory:COUNt?
查询响应:	{0~10000} 0,代表缓冲区空
例如:	发送> LOG:COUN? 接收> 10 发送> MEM:COUN? 接收> 0

1.13.5 LOGger:DATA? 或 MEMory:DATA? 数据缓冲区数据

查询语法 1:	LOGger:DATA? <1~10000> MEMory:DATA? <1~10000>
参数:	<1~10000>: 整数,大于缓冲区已记录的总数或小于 1 的数,返回 0.
例如:	发送> LOG:DATA? 1 接收> <input type="text"/> 1,+123.45E-03,+12.3456E+00
查询语法 2:	LOGger:DATA? MEMory:DATA?
查询响应:	<COUNT>; <input type="text"/> 1,<RESISTANCE1>,<VOLTAGE1>; <input type="text"/> 2,<RESISTANCE2>,<VOLTAGE2>; ...
例如:	发送> LOG:DATA? 接收> 3; <input type="text"/> 1,+123.45E-03,+12.3456E+00; <input type="text"/> 2,+123.44E-03,+12.3455E+00; <input type="text"/> 3,+123.45E-03,+12.3456E+00;



注意

1. 实际返回值没有换行符，此处增加换行为了书写方便。
2. 总位数为 5 位，不足使用空格填充。
3. 数据记录较多时，返回数据时间较长。
4. 指定的索引超过缓冲区大小时，数据返回 0。

1.14 SYSTem 子系统

SYSTem 子系统用来设置与系统相关的参数。

SYSTem 子系统设置的数据将不会保存在仪器内部。

图 1-14

SYSTem 子系统树

SYSTem	:LANGUage	{ ENGLISH , CHINESE , EN , CN }
	:TIME	<YEAR> , <MONTH> , <DAY> , <HOUR> , <MINUTE> , <SECOND>
	:KEYLock (KLOC)	{ ON (1) , OFF (0) }
	:BEEP	[: STATE] { ON (1) , OFF (0) }
	:SHAKEHAND (SHAK)	{ ON (1) , OFF (0) }
	:HEADer*	{ ON (1) , OFF (0) }
	:CODE	{ ON (1) , OFF (0) }
	:CURRent	{ CONTInuous , PULSe }
	:CALibration	(NO PARAMETER)
		:AUTO { ON (1) , OFF (0) }
	:DATAout*	{ OFF (0) , ON (1) }
	:RESult	{ FETCh , AUTO }
	:BACKup*	

(*新设计不建议使用)

1.14.1 SYSTem:LANGUage 系统语言

仪器语言设置。

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 }

1.14.2 SYSTem:TIME 系统时间设置

命令语法: SYSTem:TIME <YEAR> , <MONTH> , <DAY> , <HOUR> , <MINUTE> , <SECOND>

例如: 发送> SYST:TIME 2016,12,30,11,18,31 //2016-12-30 11:18:31

查询语法: SYSTem:TIME?

查询响应: <YEAR>-<MONTH>-<DAY> <HOUR>:<MINUTE>:<SECOND>

例如: 发送> SYST:TIME?

接收> 2016-12-30 11:18:31

1.14.3 SYSTem:KEYLock 或 SYSTem:KLOCK 键盘锁设置

命令语法: SYSTem:KEYLock { ON , OFF , 0 , 1 }
SYSTem:KLOCK { ON , OFF , 0 , 1 }

例如: 发送> SYST:KEYL OFF //键盘解锁

查询语法: SYSTem:KEYLock?
SYSTem:KLOCK?

查询响应: { on , off }

1.14.4 SYSTem:CODE 错误码返回

SYSTem:CODE 开启后，允许在每次接收到指令后都返回错误码。

对应的错误码如下：

错误码	说明
*E00	No error
*E01	Bad command
*E02	Parameter error
*E03	Missing parameter
*E04	buffer overrun
*E05	Syntax error
*E06	Invalid separator
*E07	Invalid multiplier
*E08	Numeric data error
*E09	Value too long
*E10	Invalid command
*E11	Unknow error

如果错误码功能关闭，主机可以通过发送 ERR?指令获取错误码。

命令语法: `SYSTem:CODE {ON,OFF,0,1}`

例如: 发送> SYST:CODE ON

查询语法: `SYSTem:CODE?`

查询响应: `{on,off}`

1.14.5 SYSTem:BEEPer 按键音

此指令不影响比较器讯响。

命令语法: `SYSTem:BEEPer {OFF,ON,0,1}`

参数:
{OFF,ON,0,1}
OFF/0: 按键音关闭
ON/1: 按键音关闭

例如: 发送> SYST:BEEP OFF

查询语法: `SYSTem:BEEPer?`

查询响应: `{OFF,ON}`

1.14.6 SYSTem:SHAKhand 或 SYSTem:HEADer 通讯握手指令 (数据头返回)

通讯握手开启后，仪器会将接收到的指令原样返回给主机，之后再返回数据。

命令语法: `SYSTem:SHAKhand {ON,OFF,0,1}`
`SYSTem:HEADer {ON,OFF,0,1}`

例如: 发送> SYST:SHAK ON
发送> SYST:HEAD ON

查询语法: `SYSTem:SHAKhand?`
`SYSTem:HEADer?`

查询响应: `{on,off}`

1.14.7 SYSTem:CURREnt 电流输出设置

设置仪器电流输出方式。在多台仪器同时使用时，使用电流脉冲输出方式有助于防止多台仪器间互相串扰，单台仪器工作时，请使用持续输出方式。

命令语法: **SYSTem:CURRent {CONTInuous,PULSe}**

参数: **CONTInuous**: 电流持续输出

PULSe: 电流仅在测量时输出

例如: 发送> **SYST:CURR PULS**

查询语法: **SYSTem:CURRent?**

查询响应: **{continuous,pulse}**

1.14.8 SYSTem:CALibration 仪器自校准功能

执行一次自校准。

命令语法: **SYSTem:CALibration**

例如: 发送> **SYST:CAL**

注意! 一次自校准将花费 40ms 时间, 此指令发送后至少延时 40ms 才可以发生下一条指令。
在测量速度为慢速时, 此指令将被忽略。

1.14.9 SYSTem:CALibration:AUTO 仪器自校准开关

设置仪器自校准开关。

命令语法: **SYSTem:CALibration:AUTO {ON,OFF,1,0}**

参数: **ON/1**: 仪器将每隔 30 分钟自校准一次。

OFF/0: 仪器自校准关闭。

例如: 发送> **SYST:CAL:AUTO ON**

查询语法: **SYSTem:CALibration:AUTO?**

查询响应: **{on,off}**

1.14.10 SYSTem:RESult 测试结果发送

SYSTem:RESult 可以设置数据发送方式: 自动或是 FETCH 指令。

命令语法: **SYSTem:RESult {FETCH,AUTO}**

参数: **{FETCH,AUTO}**

FETCH: 数据需要通过指令 `fetch?` 才能返回到主机, 仪器被动发送。

AUTO: 数据在每次测试完成后, 自动发送测试结果给主机, 仪器主动发送。

例如: 发送> **SYST:RES AUTO //设置为自动发送**

查询语法: **SYST:RESult?**

查询响应: **{FETCH,AUTO}**

1.14.11 SYSTem:DATAout 测试结果发送*

(*新设计不建议使用)

SYSTem:DATAout 与 SYSTem:RESult 都可以设置数据发送方式: 自动或是 FETCH 指令, 但参数有所不同。

命令语法: **SYSTem:DATAout {OFF,ON,0,1}**

参数: **{OFF,ON,0,1}**

OFF/0: 数据需要通过指令 `fetch?` 才能返回到主机, 仪器被动发送。

ON/1: 数据在每次测试完成后, 自动发送测试结果给主机, 仪器主动发送。

例如: 发送> **SYST:DATA ON //设置为自动发送**

查询语法: **SYST:DATAout?**

查询响应: **{OFF,ON}**

1.14.12 SYSTem:BACKup 保存测量参数到当前文件里*

(*新设计不建议使用)

请参考 FILE:SAVE 指令。

命令语法: `SYSTem:BACKup`

例如: 发送> `SYST:BACKUP`

1.14.13 SYSTem:RESet 出厂设置

此指令将复位所有设置为出厂设置。此指令不会影响校准数据。

命令语法: `SYSTem:RESet`

例如: 发送> `SYST:RESET` //蜂鸣器鸣叫 2 声

1.15 TRIGger 子系统

图 1-15 TRIGger 子系统树

TRIGger	[:IMMEDIATE]	
	:SOURce	{ INT ,EXT }
	:DELay	<1ms~10s>
	:STATe	{ ON(1) ,OFF(0) }
TRG (*TRG)		

TRIGger 用来设置触发源和产生一次触发。

1.15.1 TRIGger[:IMMEDIATE]

TRIG[:IMM] 在触发源设置为 BUS 时, 产生一次触发, 但不会返回触发测试的数据。如果要返回数据需要使用 TRG 指令。

命令语法: `TRIGger[:IMMEDIATE]`

例如: 发送> `TRIG` //仪器测试一次后停止

1.15.2 TRIGger:SOURce

TRIG:SOUR 用来设置触发源。

命令语法: `TRIGger:SOURce {INT,EXT}`

例如: 发送> `TRIG:SOUR EXT` //设置为外部触发模式。

查询语法: `TRIGger:SOURce?`

查询响应: `<INT,EXT>`

1.15.3 TRIGger:DELay

TRIG:DELay 用来设置触发延时定时器数值。

命令语法: `TRIGger:DELay <1ms-10.000s>`

例如: 发送> `TRIG:DEL 10m` //设置 10ms, 如果触发延时功能未启用, 该指令会首先将其打开

查询语法: `TRIG:DELay?`

查询响应: `<0.001-10.000>` //单位秒

1.15.4 TRIGger:DELay:STATe

TRIG:DELay:STATe 用来打开/关闭触发延时功能。

命令语法: `TRIGger:DELay:STATe {ON(1) ,OFF(0) }`

例如: 发送> TRIG:DEL:STAT ON //触发延时功能开启

查询语法: TRIG:DElay:STATe?

查询响应: <on,off>

1.15.5 TRG

TRG 在触发源设置为 EXT 时, 产生一次触发, 并返回触发测试的数据。

命令语法: TRG

例如: 发送> TRG //仪器测试一次, 并返回完整测试数据

返回> __21.993E+0, _3.70088E+0, OK, HI, FAIL, RPER: +2.18930e+04
//电阻值, 电压值, 电阻档, 电压档, 总合格, 监视名称及数值

注意! 如果当前页面不在<测量>页或<全屏显示>页, 该指令会首先切换到<测量>页, 再进行触发及返回数据。

1.16 FETCh 和 READ 子系统

图 1-16-A FETCh? 子系统树

FETCh?	NO PARAMETER
	: FULL

图 1-16-B READ? 子系统树

READ?	NO PARAMETER
	: FULL

FETCh 和 READ 子系统类似, FETCh 是返回的上一次测量数据, 而 READ 是返回最新一次测量数据, 因此 READ 会等待一次完整测量周期后才返回数据, 在慢速测量时执行效率稍差。

1.16.1 FETCh? 或 READ? 获取测量数据

FETCh? 用来获取测试数据。使用该指令前, 需要将<系统配置>页面下的【结果发送】字段设置为[FETCH]。

FETCh? 指令和 READ? 将返回主测试数据。

查询语法: FETCh?

查询响应: 根据测量参数:

RV: <电阻>, <电压>

R: <电阻>

V: <电压>

例如: 发送> FETC?

返回> □□22.005E+0, □3.69943E+0 //电阻值, 电压值

注意! 如果当前页面不在<测量>页或<全屏显示>页, 该指令会首先切换到<测量>页, 再返回数据。

1.16.2 FETCh:FULL? 或 READ:FULL? 获取完整测量数据

FETCh:FULL? 或 READ:FULL? 用来获取完整测试数据, 包括测量数据、比较结果和监视数据。使用该指令前, 需要将<系统配置>页面下的【结果发送】字段设置为[FETCH]。

查询语法: FETCh:FULL?

查询响应: <电阻>, <电压>, <电阻 HI/OK/LO>, <电压 HI, OK, LO>, <PASS/FAIL/WIRE/OPEN>

例如: 发送> FETC:FULL?

发送> READ:FULL?

返回> □□22.005E+0, □3.69943E+0, --, --, □□□□/-- 代表比较器未启用

返回> □□21.990E+0, □3.70120E+0, OK, HI, FAIL //电阻值, 电压值, 电阻档, 电压档, 总合格

返回> □□21.993E+0,□3.70088E+0,OK,HI,FAIL,RPER:+2.18930e+04
 //电阻值,电压值,电阻档,电压档,总合格,监视名称及数值

注意! 如果当前页面不在<测量>页或<全屏显示>页,该指令会首先切换到<测量>页,再返回数据。

1.17 CORRection 子系统

CORRection 子系统用来执行一次短路清零校准。

图 1-17 SYSTem 子系统树

CORRection	:SHORT
------------	--------

1.17.1 CORRection:SHORT

查询语法: CORRection:SHORT

例如: 发送> CORRection:SHOR
 返回> Short Clear Zero Start..
 返回> PASS

注: 在发送命令前,请务必短路测试端。

1.18 FILE(MMEM) 子系统

FILE(MMEM) 子系统用来管理文件,可以用来保存用户参数到内部闪存中,或读取闪存文件到系统里。

图 1-18 FILE(MMEM) 子系统树

FILE MMEM	:SAVE	<无参数>或<文件号 0-9>
	:LOAD	<无参数>或<文件号 0-9>
	:DELeTe	<文件号 0-9>

1.18.1 FILE:SAVE 保存文件

FILE:SAVE 可以保存当前设置到当前文件或指定的文件中。

命令语法: FILE:SAVE
 FILE:SAVE <File No. 0-9>

例如: 发送> FILE:SAVE //保存到当前文件中
 发送> FILE:SAVE 1 //保存到文件 1 中

1.18.2 FILE:LOAD 读取文件

FILE:LOAD 可以读取文件数据到系统中。

命令语法: FILE:LOAD
 FILE:LOAD <File No. 0-9>

例如: 发送> FILE:LOAD //读取当前文件数据到系统中
 发送> FILE:SAVE 1 //读取文件 1 的数据到系统中

1.18.3 FILE:DELeTe 删除指定文件

FILE:DELeTe 可以删除指定文件的数据。

命令语法: FILE:SAVE
 FILE:SAVE <File No. 0-9>

例如: 发送> FILE:SAVE //保存到当前文件中
 发送> FILE:SAVE 1 //保存到到文件 1 中

注: 删除当前文件不会影响系统的参数

1.19 IDN? 子系统

IDN?子系统用来返回仪器的版本号。



收到 IDN? 指令，仪器蜂鸣器会鸣叫一下，提示收到数据，并且结果已返回。
通常在调试通讯时，使用此指令进行联机测试。

查询语法: **IDN? 或 *IDN?**

查询响应: **<MODEL>, <SN>, <Revision>**

型号, 序列号, 仪器版本

例如:

发送> IDN?.

返回> UT35XX, UT35XXXXXXXXXX, REV XXXX //仪器蜂鸣器会鸣叫一下

1.20 ERRor 子系统

错误子系统用来获取最近一次发生错误的信息

查询语法:	ERRor?
查询响应:	Error string
例如:	发送> ERR? 返回> no error.

对应的错误码如下:

错误码	说明
*E00	No error
*E01	Bad command
*E02	Parameter error
*E03	Missing parameter
*E04	buffer overrun
*E05	Syntax error
*E06	Invalid separator
*E07	Invalid multiplier
*E08	Numeric data error
*E09	Value too long
*E10	Invalid command
*E11	Unknow error

1.21 SAV 子系统

SAV 子系统用来保存所有修改的设置到仪器内部磁盘。

查询语法:	SAV
查询响应:	OK
例如:	发送> SAV 返回> OK

2. Modbus (RTU) 通讯协议

本章包括以下几方面的内容：

- 数据格式——了解 Modbus 通讯格式。
- 功能
- 变量区域
- 功能码

2.1 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。

2.1.1 指令帧

图 2-1-1 Modbus 指令帧



表 2-1-1

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 在未选配 RS485 选件的仪器里，默认的从站地址为 0x01
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 Cyclic Redundancy 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

2.1.2 CRC-16 计算方法

1. 将 CRC-16 寄存器的初始值设为 0xFFFF。
2. 对 CRC-16 寄存器和信息的第 1 个字节数据进行 XOR 运算，并将计算结果返回 CRC 寄存器。
3. 用 0 填入 MSB，同时使 CRC 寄存器右移 1 位。
4. 从 LSB 移动的位如果为“0”，则重复执行步骤(3)(处理下 1 个移位)。从 LSB 移动的位如果为“1”，则对 CRC 寄存器和 0xA001 进行 XOR 运算，并将结果返回 CRC 寄存器。
5. 重复执行步骤(3) 和(4)，直到移动 8 位。
6. 如果信息处理尚未结束，则对 CRC 寄存器和信息的下 1 个字节进行 XOR 运算，并返回 CRC 寄存器，从第(3)步起重复执行。
7. 将计算的结果(CRC 寄存器的值) 从低位字节附加到信息上。

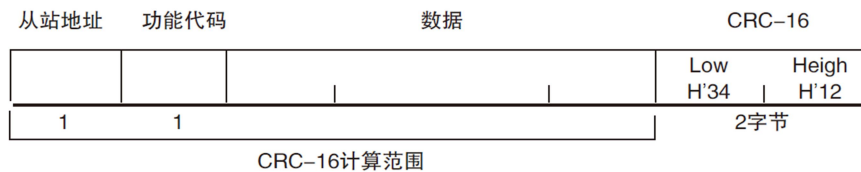
以下是一段 C 语言的 CRC 计算函数：

```
unsigned short CRC16_MODBUS(unsigned char *puchMsg, unsigned int usDataLen)
{
    int i, j; /*循环变量*/
    unsigned short usRegCRC = 0xFFFF; /*用于保存 CRC 值*/
    for(i=0; i<usDataLen; i++) /*循环处理传输缓冲区消息*/
    {
        usRegCRC ^= *puchMsg++; /*异或算法得到 CRC 值*/
        for(j=0; j<8; j++) /*循环处理每个 bit 位*/
        {
            if(usRegCRC&0x0001)
            {
                usRegCRC = (usRegCRC>>1)^0xA001;
            }
            else
            {
                usRegCRC >>=1;
            }
        }
    }
    return (usRegCRC);
}
```

计算出 CRC-16 数据需要附加到指令帧末尾，例如：1234H：

图 1-1-2

Modbus 附加 CRC-16 值



2.1.3 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 2-1-3-1

正常响应帧

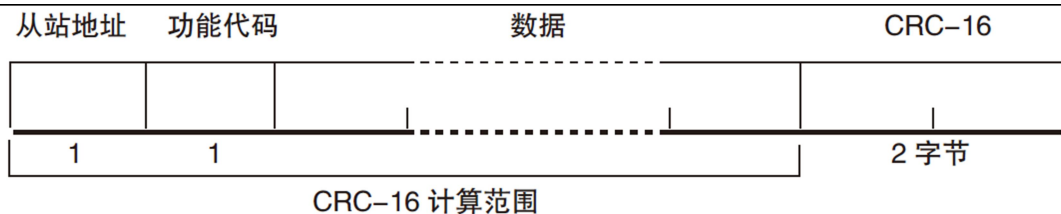


图 2-1-3-2

异常响应帧

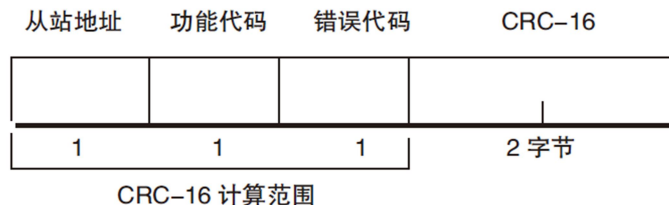


表 2-1-3

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 Cyclic Redundancy 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

2.1.4 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。

2.1.5 错误码

表 2-1-5

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

2.2 功能码

仪器仅支持以下几个功能码，其它功能码，将响应错误帧。

表 2-2

功能码

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
0x04	与 0x03 相同	请用 0x03 代替
0x08	回波测试	接收到的数据原样返回
0x10	写入多个寄存器	写入多个连续寄存器

2.3 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：

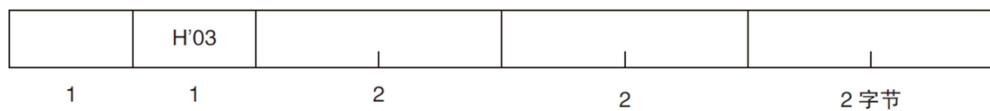
1. 1 个寄存器，双字节（16 位）整数，例如：0x64 → 00 64
2. 2 个寄存器，四字节（32 位）整数，例如：0x12345678 → 12 34 56 78
3. 2 个寄存器，四字节（32 位）单精度浮点数，3.14 → 40 48 F5 C3

2.4 读出多个寄存器

图 2-4-1

读出多个寄存器（0x03）

从站地址 功能代码 读出开始地址 元素数量 CRC-16



读出多个寄存器的功能码是 0x03.

表 2-4

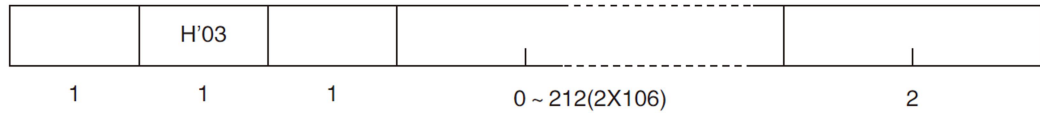
读出多个寄存器

名称	名称	说明
	从站地址	没有指定 RS485 地址时，默认为 01
0x03	功能码	
	起始地址	寄存器起始地址，请参考 Modbus 指令集
	读取寄存器数量 0001~006A (106)	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。
CRC-16	校验码	

图 2-4-2

读出多个寄存器（0x03）响应帧

从站地址 功能代码 字节计数 读出数据(元素数量部分) CRC-16



名称	名称	说明
	从站地址	原样返回
0x03 或 0x83	功能码	无异常: 0x03 错误码: 0x83
	字节数	=寄存器数量 x 2 例如: 1 个寄存器返回 02
	数据	读取的数据
CRC-16	校验码	

2.5 写入多个寄存器

图 2-5-1 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



表 2-5-1 写入多个寄存器

名称	名称	说明
	从站地址	没有指定 RS485 地址时, 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 x 2
CRC-16	校验码	

图 2-5-2 写入多个寄存器 (0x03) 响应帧

从站地址 功能代码 写入开始地址 元素数量 CRC-16



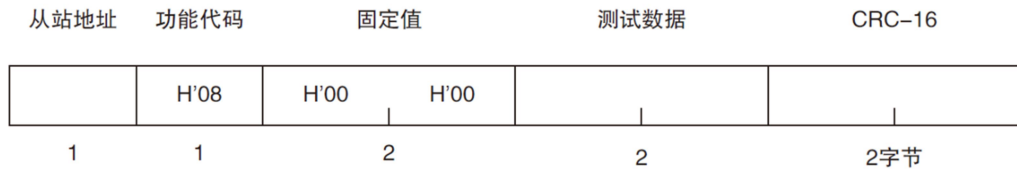
名称	名称	说明
	从站地址	原样返回
0x10 或 0x90	功能码	无异常: 0x10 错误码: 0x90
	起始地址	
	寄存器数量	
	CRC-16 校验码	

2.6 回波测试

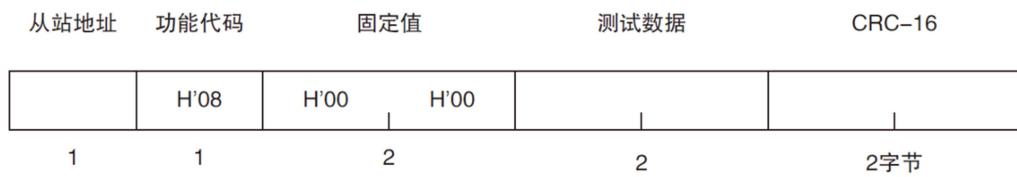
回波测试功能码 0x08, 用于调试 Modbus。
回波测试 (0x08)

图 2-6

指令帧



响应帧



名称	名称	说明
	从站地址	原样返回
0x08	功能码	
	固定值	00 00
	测试数据	任意数值: 例如 12 34
	CRC-16 校验码	

例如:

假定测试数据为 0x1234:

指令:

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

响应:

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

3. Modbus (RTU) 指令集

本章包括以下几方面的内容:

- 寄存器地址



注意: 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。

3.1 寄存器总览

以下列出了仪器使用的所有寄存器地址, 任何不在表中的地址将返回错误码 0x02.

表 3-1 寄存器总览

寄存器地址	名称	数值	说明
2000	读取电阻测量结果	4 字节浮点数	只读寄存器, 数据占用 2 个寄存器
2002	读取电压测量结果	4 字节浮点数	只读寄存器, 数据占用 2 个寄存器
2004	获取比较器结果	2 字节整数	只读寄存器, 数据占用 1 个寄存器
0000	读取仪器版本号	4 字节 ASCII	只读寄存器, 数据占用 2 个寄存器
3000	功能寄存器	0000: R-V 0001: R 0002: V	读写寄存器, 2 字节整数
3001	电阻量程	0000~0006	读写寄存器, 2 字节整数
3002	电压量程	0000~0002	读写寄存器, 2 字节整数
3003	电阻量程方式	0000: 量程自动 0001: 量程手动 0002: 标称量程	读写寄存器, 2 字节整数
3004	电压量程方式	0000: 量程自动 0001: 量程手动 0002: 标称量程	读写寄存器, 2 字节整数
3005	测试速度	0000: 慢速 0001: 中速 0002: 快速 0003: 高速	读写寄存器, 2 字节整数
3006	平均次数	0000: 无效 0001~0x0100 (1~256)	读写寄存器, 2 字节整数
3007	触发方式	0000: 内部	读写寄存器, 2 字节整数

		0001: 外部	
3008	触发延时	0000: 关闭 0001~0x2710 (1~10000)	读写寄存器, 2 字节整数 单位 ms
3009	触发沿	0000: 上升沿触发 0001: 下降沿触发	读写寄存器, 2 字节整数
300A	自校准	0000: 关闭 0001: 开启	读写寄存器, 2 字节整数
300B	电流模式	0000: 连续 0001: 脉冲	读写寄存器, 2 字节整数
300C	文件开机调用	0000: 文件 0 0001: 当前文件	读写寄存器, 2 字节整数
300D	自动保存	0000: 禁止 0001: 允许	读写寄存器, 2 字节整数
300E	系统语言	0000: 英语 0001: 简体中文	读写寄存器, 2 字节整数
3100	电阻比较器状态	0000: 比较器关闭 0001: 比较器打开	读写寄存器, 2 字节整数
3101	电压比较器状态	0000: 比较器关闭 0001: 比较器打开	读写寄存器, 2 字节整数
3102	电阻比较器方式	0000: SEQ 0001: PER 0002: ABS	读写寄存器, 2 字节整数
3103	电压比较器方式	0000: SEQ 0001: PER 0002: ABS	读写寄存器, 2 字节整数
3104	讯响	0000: 关闭 0001: 合格讯响 0002: 不合格讯响	读写寄存器, 2 字节整数
3110	电阻标称值	4 字节浮点数	读写寄存器, 数据占用 2 个寄存器
3112	电压标称值	4 字节浮点数	读写寄存器, 数据占用 2 个寄存器
3114	电阻下限值	4 字节浮点数	读写寄存器, 数据占用 2 个寄存器
3116	电阻上限值	4 字节浮点数	读写寄存器, 数据占用 2 个寄存器
3184	电压下限值	4 字节浮点数	读写寄存器, 数据占用 2 个寄存器
3186	电压上限值	4 字节浮点数	读写寄存器, 数据占用 2 个寄存器
4000	保存设置到当前文件	固定值: 0001	只写寄存器, 数据 2 字节
4008	读取当前文件数据	固定值: 0001	只写寄存器, 数据 2 字节
4010	保存设置到指定文件	0000~0009	只写寄存器, 数据 2 字节
4018	读取指定文件数据	0000~0009	只写寄存器, 数据 2 字节
5000	执行清零寄存器 读取清零状态	写入固定值: 0001 读取:	读写寄存器, 数据占用 1 个寄存器 一旦执行了清零功能, Modbus 将禁

		0001 正在清零 0000 清零成功 FFFF 清零失败	止执行写入指令, 仅允许读取寄存器。
--	--	-------------------------------------	--------------------

3.2 获取测量数据

3.2.1 获取测量结果

寄存器 2000~2003 用来获取仪器测量数据。

指令:

1	2	3	4	5	6	7	8
01	03	2000		0002		CRC-16	
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	字节	单精度浮点数				CRC-16	

- 获取电阻测量结果:

发送:

1	2	3	4	5	6	7	8
01	03	20	00	00	02	CF	CB
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	4E	6E	6B	28	A3	E8
01	03	字节	单精度浮点数				CRC-16	

其中 B4~B6 为测量数据: 4E6E6B28 代表 1E9 (低位在前)

- 获取电压测量结果

发送:

1	2	3	4	5	6	7	8
01	03	20	02	00	02	6E	0B
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	50	15	02	F9	3B	D5
01	03	字节	单精度浮点数				CRC-16	

其中 B4~B6 为测量数据: 501502F9 代表 1E10 (低位在前)

- 获取电阻和电压测量结果

● 发送:

1	2	3	4	5	6	7	8
01	03	20	02	00	04	4F	C9

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	10	11	12	13
01	03	08	3F	B1	69	A8	41	0C	2A	56	54	08

电阻: B4~B7: 3FB169A8 = 1.3860

电压: B8~B11: 410C2A56 = 8.7603

3.2.2 获取比较器结果【2004】

寄存器 2004 记录了电压和电阻的比较器结果

例如: 2203

16 位存储域:

其中: BIT15~BIT12 代表电压档 0: VOK 1: VLO 2: VHI
 BIT11~BIT8 代表电阻档 0: ROK 1: RLO 2: RHI
 BIT7~BIT4 无效
 BIT3~BIT0 总合格档 0: RVOK 3: RVNG

发送:

1	2	3	4	5	6	7	8
01	03	21	04	00	01	CE	0B
从站	读	寄存器	寄存器数量	校验码			

响应:

1	2	3	4	5	6	7
01	03	02	22	03	E0	E5

3.3 参数设置

3.3.1 功能寄存器【3000】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	00	00	01	02	00	00	96	53
	写	寄存器	寄存器数量	字节	数据	CRC				

响应:

1	2	3	4	5	6	7	8
01	10	30	00	00	01	AF	09
		寄存器	寄存器数量	CRC			

● 读取

1	2	3	4	5	6	7	8
01	03	30	00	00	01	8B	0A
	读	寄存器	寄存器数量	CRC			

响应:

1	2	3	4	5	6	7
01	03	02	00	00	B8	44
		字节	数据	CRC		

其中数据值:

数据	功能	说明
0000	R-V	电阻电压同时测量
0001	R	仅电阻测量和显示
0002	V	仅电压测量和显示

3.3.2 电阻量程寄存器【3001】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	01	00	01	02	00	01	56	42
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	01	00	01	5F	09
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	01	00	01	DA	CA
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	量程 0	3mΩ
0001	量程 1	30mΩ
0002	量程 2	300mΩ
0003	量程 3	3Ω
0004	量程 4	30Ω
0005	量程 5	300Ω
0006	量程 6	3kΩ

3.3.3 电压量程寄存器【3002】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	02	00	01	02	00	02	16	70
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	02	00	01	AF	09
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	02	00	01	DA	CA

	读	寄存器	寄存器数量	CRC
--	---	-----	-------	-----

响应:

1	2	3	4	5	6	7
01	03	02	00	02	39	85
		字节	数据	CRC		

其中数据值:

数据	功能	说明
0000	量程 0	8V
0001	量程 1	80V
0002	量程 2	400V

3.3.4 电阻量程方式寄存器【3003】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	03	00	01	02	00	01	57	A0
	写	寄存器	寄存器数量	字节	数据	CRC				

响应:

1	2	3	4	5	6	7	8
01	10	30	03	00	01	FE	C9
		寄存器	寄存器数量	CRC			

● 读取

1	2	3	4	5	6	7	8
01	03	30	03	00	01	7B	0A
	读	寄存器	寄存器数量	CRC			

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据	CRC		

其中数据值:

数据	功能	说明
0000	自动量程	
0001	保持量程	
0002	标称量程	根据标称值选择量程

3.3.5 电压量程方式寄存器【3004】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	04	00	01	02	00	01	56	17
	写	寄存器	寄存器数量	字节	数据	CRC				

响应:

1	2	3	4	5	6	7	8
01	10	30	04	00	01	4F	08
		寄存器	寄存器数量	CRC			

● 读取

1	2	3	4	5	6	7	8
01	03	30	04	00	01	CA	CB
	读	寄存器	寄存器数量	CRC			

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据	CRC		

其中数据值:

数据	功能	说明
0000	自动量程	
0001	保持量程	
0002	标称量程	根据标称值选择量程

3.3.6 测试速度寄存器【3005】

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	05	00	01	02	00	01	56	71
	写	寄存器	寄存器数量	字节	数据	CRC				

响应:

1	2	3	4	5	6	7	8
01	10	30	05	00	01	1E	C8
		寄存器	寄存器数量	CRC			

- 读取

1	2	3	4	5	6	7	8
01	03	30	05	00	01	9B	0B
	读	寄存器	寄存器数量	CRC			

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据	CRC		

其中数据值:

数据	功能	说明
0000	慢速	
0001	中速	
0002	快速	
0003	高速	

3.3.7 平均次数寄存器【3006】

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	06	00	01	02	00	01	57	F5
	写	寄存器	寄存器数量	字节	数据	CRC				

响应:

1	2	3	4	5	6	7	8
01	10	30	06	00	01	EE	C8
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	06	00	01	6B	0B
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0001~0100	平均值 0~256	平均值 0=平均值 1

3.3.8 触发方式寄存器【3007】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	07	00	01	02	00	01	56	24
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	06	00	01	BF	08
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	07	00	01	3A	CB
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	内部触发	
0001	外部触发	使用 Trigger 键、Handler Trig 或远程触发

3.3.9 触发延时寄存器【3008】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	08	00	01	02	00	0A	56	24
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	08	00	01	BF	08
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	08	00	01	0A	C8
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	00	B8	44
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	延时关闭	
0001~2710	1~0x2710 (10000)	十进制 1ms~10000ms 单位 ms

3.3.10 触发沿寄存器【3009】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	09	00	01	02	00	00	96	CA
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	09	00	01	DE	CB
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	09	00	01	5B	08
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	00	B8	44
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	上升沿触发	缺省设置
0001	下降沿触发	

3.3.11 自校准开关寄存器【300A】

● 写入

1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

01	10	30	0A	00	01	02	00	01	2E	CB
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	0A	00	01	2E	CB
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	0A	00	01	AB	08
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	自校准关闭	
0001	自校准打开	缺省设置

3.3.12 电流模式寄存器【300B】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	0B	00	01	02	00	00	97	28
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	0B	00	01	7F	0B
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	0B	00	01	FA	C8
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	00	B8	44
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	连续电流输出	缺省设置
0001	脉冲电流输出	

3.3.13 文件开机调用寄存器【300C】

● 写入

1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	----	----

01	10	30	0C	00	01	02	00	01	FA	C8
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	0C	00	01	CE	CA
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	0C	00	01	4B	09
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	文件 0	缺省设置
0001	当前文件	

3.3.14 自动保存【300D】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	30	0D	00	01	02	00	01	56	8E
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	30	0D	00	01	9F	0A
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	30	0D	00	01	79	84
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	关闭	缺省设置
0001	打开	

3.4 比较器设置

比较器参数寄存器地址从 3100 开始。

3.4.1 电阻比较器状态寄存器【3100】

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	31	00	00	01	02	00	01	47	53
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	31	00	00	01	0F	35
		寄存器		寄存器数量		CRC	

- 读取

1	2	3	4	5	6	7	8
01	03	31	00	00	01	8A	F6
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	电阻比较器关闭	缺省设置
0001	电阻比较器打开	

3.4.2 电压比较器状态寄存器【3101】

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	31	01	00	01	02	00	01	46	82
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	31	01	00	01	5E	F5
		寄存器		寄存器数量		CRC	

- 读取

1	2	3	4	5	6	7	8
01	03	31	01	00	01	DB	36
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	电压比较器关闭	缺省设置
0001	电压比较器打开	

3.4.3 电阻比较器方式寄存器【3102】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	31	02	00	01	02	00	01	46	B1
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	31	02	00	01	AE	F5
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	31	02	00	01	2B	36
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	SEQ	直读值
0001	PER	相对偏差比较
0002	ABS	绝对偏差比较

3.4.4 电阻比较器方式寄存器【3103】

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	31	03	00	01	02	00	01	47	B1
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	31	03	00	01	FF	35
		寄存器		寄存器数量		CRC	

● 读取

1	2	3	4	5	6	7	8
01	03	31	03	00	01	7A	F6
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84

		字节	数据	CRC
--	--	----	----	-----

其中数据值:

数据	功能	说明
0000	SEQ	直读值
0001	PER	相对偏差比较
0002	ABS	绝对偏差比较

3.4.5 讯响寄存器【3104】

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	31	04	00	01	02	00	01	46	D7
	写	寄存器		寄存器数量		字节	数据		CRC	

响应:

1	2	3	4	5	6	7	8
01	10	31	04	00	01	4E	F4
		寄存器		寄存器数量		CRC	

- 读取

1	2	3	4	5	6	7	8
01	03	31	04	00	01	CB	37
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7
01	03	02	00	01	79	84
		字节	数据		CRC	

其中数据值:

数据	功能	说明
0000	OFF	关闭
0001	PASS	合格讯响
0002	FAIL	不合格讯响

3.4.6 电阻标称值寄存器【3110】

电阻标称值使用 2 个寄存器, 3110 和 3111。注意! 单独读取 3110 无效。

- 写入

100E-3 (单精度浮点数: 0x3DCCCCD)

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	31	10	00	02	04	3D	CC	CC	CD	F2	34
	写	寄存器		寄存器数量		字节	数据				CRC	

响应:

1	2	3	4	5	6	7	8
01	10	31	10	00	02	4E	F1
		寄存器		寄存器数量		CRC	

- 读取

1	2	3	4	5	6	7	8
01	03	31	10	00	02	CB	32

	读	寄存器	寄存器数量	CRC				
响应:								
1	2	3	4	5	6	7	8	9
01	03	04	3D	CC	CC	CD	A3	35
		字节	数据 100E-3				CRC	

3.4.7 电压标称值寄存器【3112】

电阻标称值使用 2 个寄存器，3110 和 3111。注意！单独读取 3110 无效。

- 写入

3.6000 (单精度浮点数: 0x40666666)

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	31	12	00	02	04	40	66	66	66	74	BE
	写	寄存器		寄存器数量		字节	数据				CRC	

响应:

1	2	3	4	5	6	7	8
01	10	31	12	00	02	EF	31
		寄存器		寄存器数量		CRC	

- 读取

1	2	3	4	5	6	7	8
01	03	31	12	00	02	6A	F2
	读	寄存器		寄存器数量		CRC	

响应:

1	2	3	4	5	6	7	8	9
01	03	04	40	66	66	66	A4	66
		字节	数据				CRC	

3.4.8 电阻极限值寄存器【3114-3117】

电阻极限值从 3114 开始，下限使用 2 个寄存器，上限使用 2 个寄存器，总共 4 个寄存器。

下限和上限可以分别设置，也可以同时设置。

- 写入

下限: 1E-3, 上限: 10E-3

1	2	3~4	5	6	7	8~11	12~15	16~17
01	10	3114	00	02	04	3A 83 12 6F	3C 23 D7 0A	01 8E
						下限	上限	

响应:

1	2	3	4	5	6	7	8
01	10	31	14	00	04	8F	32

- 读取

1	2	3	4	5	6	7	8
01	03	31	14	00	04	0A	F1

响应

1	2	3~4	5~8	9~12	13~14
01	03	31 14	3A 83 12 6F	3C 23 D7 0A	51 61

			下限	上限	
--	--	--	----	----	--

3.4.9 电压极限值寄存器【3184-3187】

电阻极限值从 3114 开始，下限使用 2 个寄存器，上限使用 2 个寄存器，总共 4 个寄存器。
 下限和上限可以分别设置，也可以同时设置。

● 写入

下限：3.0000, 上限：4.0000

1	2	3~4	5	6	7	8~11	12~15	16~17
01	10	3184	00	02	04	40 40 00 00	40 80 00 00	57 66
						下限	上限	

响应：

1	2	3	4	5	6	7	8
01	10	31	84	00	04	8F	1F

● 读取

1	2	3	4	5	6	7	8
01	03	31	84	00	04	0A	DC

响应

1	2	3~4	5~8	9~12	13~14
01	03	31 14	40 40 00 00	40 80 00 00	51 61
			下限	上限	

3.5 文件操作

由于仪器设置存储在文件里，因此所有的 Modbus 指令设置后，数据无法实时存储在内部 FlashRom 中，会导致下次上电开机之前的寄存器数据恢复成原文件的数值。

用户可以同文件操作寄存器来将所有设置值存储到当前或指定的文件中。同时，也可以调用指定的文件数据到设置寄存器中。

3.5.1 保存到当前文件【4000】

发送数值 0001 到 4000 寄存器，仪器将执行文件写入操作，所有设置将全部保存到当前文件中。
 此寄存器无法读出。

● 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	40	00	00	01	02	00	01	26	54
	写	寄存器		寄存器数量		字节	数据		CRC	

响应：

1	2	3	4	5	6	7	8
01	10	40	00	00	01	14	09
		寄存器		寄存器数量		CRC	

其中数据值：

数据	功能	说明
0001	允许操作	固定值

3.5.2 保存到指定文件【4008】

发送文件号到 4008 寄存器，仪器将执行文件写入操作，所有设置将全部保存到指定文件中，同时指定的文件将作为系统当前文件使用。

此寄存器无法读出。

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	40	08	00	01	02	00	09	26	DA
	写	寄存器		寄存器数量		字节	数据		CRC	

响应：

1	2	3	4	5	6	7	8
01	10	40	00	00	01	95	CB
		寄存器		寄存器数量		CRC	

其中数据值：

数据	功能	说明
0000~0009	文件 0~9	

3.5.3 重新载入当前文件【4010】

发送固定值 0001 到 4010 寄存器，仪器将当前文件数据载入到系统中。

此寄存器无法读出。

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	40	10	00	01	02	00	01	24	C4
	写	寄存器		寄存器数量		字节	数据		CRC	

响应：

1	2	3	4	5	6	7	8
01	10	40	10	00	01	15	CC
		寄存器		寄存器数量		CRC	

其中数据值：

数据	功能	说明
0001	固定值	

3.5.4 载入指定文件【4018】

发送文件号到 4018 寄存器，仪器将载入指定文件的设置到系统中，同时指定的文件将作为系统当前文件使用。

此寄存器无法读出。

- 写入

1	2	3	4	5	6	7	8	9	10	11
01	10	40	18	00	01	02	00	00	E4	4C
	写	寄存器		寄存器数量		字节	数据		CRC	

响应：

1	2	3	4	5	6	7	8
01	10	40	18	00	01	94	0E
		寄存器		寄存器数量		CRC	

错响应:

文件为空, 仪器将响应错误码: 04

1	2	3	4	5
01	90	04	4D	C3
		错误码	CRC	

其中数据值:

数据	功能	说明
0000~0009	文件 0~9	

3.6 清零【5000】

向寄存器 5000 写入 0001, 仪器将开始执行短路清零操作。

执行清零前, 务必将测试线短路, 否则清零将失败。

由于清零过程需要几秒钟时间, 这期间, 任何写入操作将被忽略, 仅开放读操作。清零完成后, 写入指令正常开放。

清零执行期间或清零完成后, 都可以通过读取 5000 寄存器获取清零状态:

0000 清零成功

FFFF 清零失败

0001 正在清零

- 写入

请给 5000 寄存器写入固定值: 00 01

发送: 01 10 5000 0001 02 0001 3795

响应: 01 10 5000 0001 10C9

- 读取

执行清零期间, 可以通过读取寄存器数据来确定是否清零完成

发送: 01 03 5000 0001 950A

响应: 01 03 02 FFFF B9F4

返回 FFFF, 代表清零失败

注意:

在清零时, 尽量不要频繁读取清零状态, 连续的中断容易引起仪器清零失败。

由于清零时间固定, 建议发出清零指令后, 主机强制等待清零时间过后, 再来获取清零结果。

