ADT286 Commands Set

1 Commands Instruction

1.1 IEEE488.2 common commands

NO.	Command	Explanation	Parameters	Returning values
1	*CLS	This command eliminates the following registers; Standard event register Querying event register Operating event register Status byte register Error queue	-	-
2	*IDN?	To Query instrument identifies, the returned data is divided into 2 parts a. product sequence number b. software version number	-	Product sequence number and software version number
3	*RST	Main software reset	-	-

1.2 Measurement and configuration command

All raw values refer to the values before calibration.

NO.	Command	Explanation	Parameters	Returning values
1	[MEASure:]MODule:INFormation?	Acquire the information of front panel and junction box	None	N*7 values,N may be 1,2,3,4,5, each message is separated by semicolon, each parameter is separated by commas. Identifier of box :The front panel is 0. The embedded junction box is 1. Then the seris-wound junction boxes are in 2, 3, 4 Box serial number Box type, 0=front panel, 1=temperature box, 2=process box Box hardware version Box software version Total number of box channel Label of box
2	JSON:[MEASure:]MODule:INFormation?	Acquire front panel and junction box information, JSon format	None	Character string which return back to JSon serialization, the original format is List<DIModuleInfo>

NO.	Command	Explanation	Parameters	Returning values
3	[MEASure:]MODule:LABel <index>,<"label">	Set junction box label	2 values, and separated by commas Index, junction box identifier. Label, quotation mark label	None
4	[Measure:]MODule:CONFig? <moduleIndex>	Acquire channel configuration of one junction box	1 value moduleIndex: Junction box identifier. the front panel is 0, the embedded junction box is 1, then the serial-wound junction boxes are in 2.3.4	N channel information, separated by semicolon. 8+M value, separated by commas. Channel name. Enable or not Label Function type Range index Channel delay Automatic range or not Filter M parmeters, M is based on electrical logging type;(voltage ,M=1: high impedence or not; current,M=0: None resistances,M=2: wires, whether to open positive or negative current; RTD / SPRT/ custom RTD, M=6 : wires , sensor name,

NO.	Command	Explanation	Parameters	Returning values
				sensor serial number, sensor Id, whether to open 1.4 times current, compensation interval; thermistors,M=4: wires , sensor name, sensor serial number, sensor Id;TC / standard TC. M=7: Whether the break detection, sensor name, sensor serial number, sensor Id, cold junction type, cold junction fixed value, custom cold junction channel name; current / voltage transmitters,M=4: wires, sensor name, sensor serial number, sensor Id;)
5	JSON:[Measure:]MODule:CONFig? <moduleIndex>	Acquire channel configuration of one junction box, in JSon format	1 value moduleIndex: Junction box identifier. the front panel is 0, the embedded junction box is 1, then the serial-wound junction boxes are in 2.3.4	Character string which return back to JSon serialization, the original format is List<DIFunctionChannelConfig>
6	[MEASure:]MODule:CONFig <moduleIndex>,<"params">	Set the channel configuration of one junction box	2 value, separated by commas moduleIndex: Junction box identifier. the front panel is 0, the embedded	None

NO.	Command	Explanation	Parameters	Returning values
			junction box is 1, then the serial-wound junction boxes are in 2.3.4 “params”： N channel information, separated by semicolon, the parameters of each information are separated by commas. Common parameters: Channel name Enable or not label Function type,0=voltage, 1=current, 2=resistance,3= RTD, 4=thermister, 100=TC, 101=switch, 102=SPRT, 103=voltage transmitter, 104=current transmitter, 105=standard TC, 106=customRTD, 110=standard resistance Range index Channel delay Automatical range or not Filter Extra parameter: Current parameter:none Voltage parameter: high impedence or	

NO.	Command	Explanation	Parameters	Returning values
			not Resistance parameter: wires, positive and negative current RTD / SPRT / custom RTD parameters: sensor name, wires, compensation interval, whether 1.4 times current Thermistor parameters: sensor name, wires, TC / standard TC parameters: sensor name, whether the break couple detection, cold junction type, cold junction fixed value, custom cold junction channel name Switch parameters: switch type Current/voltage transmitter: wires, sensor name;	
7	JSON:[MEASure:]MODule:CONFig <moduleIndex>,<"params">	Set channel configuration of junction box, in JSon format	two parameters, separated by semicolon. moduleIndex: Junction box identifier. the front panel is 0, the embedded junction box is 1, then the serial-wound junction boxes are in 2.3.4 "params"： JSON character string after serialization and the original format is List<DIFunctionChannelConfig >	None

NO.	Command	Explanation	Parameters	Returning values
8	[MEASure:]SCAN:STARt <"params">	Set the configuration and start scanning	One value "params": two parameters, separated by commas. NPLC, sample work Frequency cycle(100,1000,4000) Channel name	None
9	JSON:[MEASure:]SCAN:STARt <"params">	Set the configuration and start scanning in JSON format	One value "params", the JSON character string after serialization, the original format is DIScanInfo	None
10	[MEASure:]SCAN:STARt?	Acquire scanning configuration	None	Two values, separated by commas NPLC Name of current scanning channel
11	JSON:[MEASure:]SCAN:STARt?	Acquire scanning configuration in JSon format	None	Serializable character string, the original type is DIScanInfo
12	[MEASure:]SCAN:STOP	Stop scanning	None	None
13	[MEASure:]SCAN:DATA:Last? [<time>]	Acquire scanning data	One parameter, optional time,1=yyyy:MM:dd HH:mm:ss fff format 2=longformat	The latest data:N specific data, according to N scanning channels, separated by semicolons in quotation marks

NO.	Command	Explanation	Parameters	Returning values
				Each channel data, separated by comma according to the difference length of different data types. Electrical measurement data: Channel name Electrical unit Id Number of electrical measurement data 1 One electrical measurement data electrical measurement data after filter Temperature data: Channel name Electrical unit Id Number of electrical measurement data 1 electrical measurement data electrical measurement data after filter Indication unit Id Number of indication data 1

NO.	Command	Explanation	Parameters	Returning values
				the indication data TC data: Channel name Electrical unit Id Number of electrical measurement data 1 electrical measurement data electrical measurement data after filter Indication unit Id Number of indication data 1 the indication data Cold junction electrical unit Id Cold junction electrical measurement data number 1 cold junction electrical test data Cold junction temperature unit Id Cold junction temperature data number 1 cold junction temperature data

NO.	Command	Explanation	Parameters	Returning values
				Switch data: Based on the TC or RTD data, there is an additional status data for the full switch channel in the box. Voltage/Current Transmitter Data Channel name Electrical unit Id Number of electrical measurement data 1 electrical measurement data of electrical measurement data after filter Input signal unit Id Input signal unit name Number of input signals 1 input signal data
14	JSON:[MEASure:]SCAN:DATA? <count>	Acquire scanning data, JSon format	One parameter the number of scanning data	One value Serialiable JSon character string, the format is List<DIReading>
15	[MEASure:]CHANnel:CONFig?	Acquire channel configuration	One value	8+m values, separated by

NO.	Command	Explanation	Parameters	Returning values
	<"channelName">		channelName",channel name, only one	commas Channel name Enable or not Label Function type Range Index, Channel delay Automatical range or not Number of filter m parameters, m depends on the type of electrical measurement:, (voltage, m=1: high impedance or not; Current, m=0: none; resistance,m=2: wires, open positive and negative current or not; RTD / SPRT / custom RTD, m=6: wires, sensor name, sensor serial number, sensor Id, whether to open 1.4 times current, compensation interval; thermistors,m=4: wires, sensor name, sensor serial number, sensor Id;TC /

NO.	Command	Explanation	Parameters	Returning values
				standard TC. m=7: Whether the break detection, sensor name, sensor serial number, sensor Id, cold end type, cold end fixed value, external cold end channel name; current / voltage transmitters,m=4: wires, sensor name, sensor serial number, sensor Id;)
16	[MEASure:]CHANnel:CONFig:JSON? <"chNames">	Acquire channel configuration, format data of JSon	One value "chNames",channel name, supports one or more, separated by commas	One value Serialiable JSon character string, the original format is List<DIFunctionChannelConfig>
17	[MEASure:]CHANnel:CONFig <"chName">,<enable>,<"label">,<elecType>,<range>,<delay>,<autoRange>,<filter>,<"otherParam">	Set channel configuration	9 values, separated by commas "chName", channel name, only supports one Enable or not Label Function type Range Channel delay, Automatical Range or not1=Yes, 0=No the number of filters "otherParam", electrical configuration	None

NO.	Command	Explanation	Parameters	Returning values
			parameters m, m depends on the function type, comma separated:(voltage,m=1: high impedance or not;Current m=0; resistance: wires, whether to open positive and negative current; RTD / SPRT / custom RTD, m=6: wires, sensor name, sensor serial number, sensor Id, whether to open 1.4 times current, compensation interval; thermistors,m=4: wires, sensor name, sensor serial number, sensor Id; TC / standard TC, m=7: whether the break couple detection, sensor name, sensor serial number, sensor Id, cold junction type (0 internal 1 external 2 custom), cold end fixed value, external cold junction channel name; Switch: switch type current / voltage transmitters,m=4: wires, sensor name, sensor serial number, sensor Id;)	
18	JSON:[MEASure:]CHANnel:CONFig <"jsonStr">	Set channel configuration	One value "jsonStr",channel configuration is inJson character string, the original	None

NO.	Command	Explanation	Parameters	Returning values
			format is List< DIFunctionChannelConfig>	
19	JSON:[MEASure:]SCAN:SCONnection:DATA ? <count>	Acquire scanning data of intelligent wiring, JSon format	one value the number of reading data	None
20	[MEASure:]CHANnel:ZERo <enable>	Set zero clearing for channel, only single channel current or voltage measurement is valid	one value enable zero clearing or cancel , 1=ON,0=Off	None
21				

1.3 Calibration commands

NO.	Command	Explanation	Parameters	Returning values
1	CALibration:ElECtricity:SCAN <mode>,<function>,<range>	Start scanning of electrical calibration	4 parameters 1、 mode： 0 = active calibration 2、 function：(only for 0 mode)： 0 - voltage 1- current 2- resistance 3- PRT 4- thermister 3、 range：(only for 0 mode) voltage： 0)100.00000 mV	None

NO.	Command	Explanation	Parameters	Returning values
			1)1.0000000 V 2)10.000000 V 3)50.000000 V current： 0)100.00000 µA 1)1.0000000 mA 2)10.000000 mA 3)100.000000 mA resistance： 0)100.00000 Ω 1)1.0000000 kΩ 2)10.000000 kΩ 3)100.00000 kΩ 4)1.0000000 MΩ 5)10.000000 MΩ 6)100.00000 MΩ PRT 0)100.000000 Ω 1)400.000000 Ω 2)4.00000000 kΩ Thermistor 0)0--10.0000000 kΩ	

NO.	Command	Explanation	Parameters	Returning values
			1)10--100.000000 kΩ 2)0.1--1.00000000 MΩ	
2	CALibration:ElECtricity:SCAN?	Read the original data of electrical logging	None	6 values, separated by commas: Exception code (4 bytes) Mode (no response when ADC calibration) Function Range Status: 1 indicates data is available Data: the double original value when the state is available, otherwise it is empty
3	CALibration:ElECtricity:DATA Manufactor|User,<password>,<channel>,<function>,<range>,<unitID>,<count>,<points>,<values>,<year>,<month>,<day>	write in calibration data	12 values, 1. Manufactor: factory calibration ;User: user calibration; 2. password, corresponding to the manufacturer password or user password 3. channel: channel number 01~02 means REF1 and REF2; 101~110 means the inner box 01A~10A channel,	None

NO.	Command	Explanation	Parameters	Returning values
			111~120 means the inner box 01B~10B channel (only one box); 201~210 means external 1 box 01A~10A channel ,211~220 means external 1 box 01B~10B channel (only one box); The external 2 boxes and the external 3 boxes are similar to the external one box; 4. function: 0 - voltage; 1- current; 2- resistance; 3- Prt; 4- Thermistor 5- Cjc) 5. Range voltage： 0)100.00000 mV 1)1.0000000 V 2)10.000000 V 3)50.000000 V current：	

NO.	Command	Explanation	Parameters	Returning values
			0)100.00000 µA 1)1.0000000 mA 2)10.000000 mA 3)100.000000 mA resistance： 0)100.00000 Ω 1)1.0000000 kΩ 2)10.000000 kΩ 3)100.00000 kΩ 4)1.0000000 MΩ 5)10.000000 MΩ 6)100.00000 MΩ PRT 0)100.000000 Ω 1)400.000000 Ω 2)4.00000000 kΩ Thermistor 0)0--10.0000000 kΩ 1)10--100.0000000 kΩ 2)0.1--1.00000000 MΩ 6.unitI:ID	

NO.	Command	Explanation	Parameters	Returning values
			7.count:the number of points;8.points: Calibration points(character string with quotation mark, separated by commas) 9.values:standard value(character string with quotation mark, separated by commas) 10.year:year 11.month:month 12.day: day	
4	CALibration:ELECtricity:DATA? Manufactor|User,<password>,<channel>,<function>,<range>	Acquire calibration data	5 values 1. Manufactor: factory calibration; User:user calibration; 2. password, corresponding to the manufacturer password or user password 3. channel number 4. function item 5. range,	N*2+5 values, separated by commas: Unit Id count of calibration points list of standard value,N list of calibration points,N year month day
5	CALibration:ELECtricity:CJCenable <enable>	Open and close the cold junction calibration, read the original value after opening, and read the final value after closing	1 value 1=ON 0=OFF	None
6	CALibration:ELECtricity:DATA:CJC? Manufactor|User,<password>,<location>,	Read cold junction calibration data	4 values, separated by commas Manufactor:factory calibration;User:user	8 values, separated by commas

NO.	Command	Explanation	Parameters	Returning values
	<channel>		calibration; password,corresponding to the manufacturer password or user password location,reading corresponding to location infromation,1=embedded , 0=external connection channel,channel number(101means embedded loaction01A channel)	The channel is located in the junction box number (1~10 means A, 11~20 means B) Internal wiring or external wiring, 0 means external, 1 means internal Data type, 0 means user, 1 means manufacturer Whether the data is valid, 1 means valid, 0 means invalid Calibration value year month day
7	CALibration:ELECtricity:DATA:CJC Manufactor|User,<password>,<location>,<channel> ,<offset>,<year>,<month>,<day>	Write in cold junction calibration data	8 values, separated by commas Manufactor: factory calibration; User: user calibration; Password, corresponding to the manufacturer password or user password Location, write the corresponding location data, 1 = internal wiring, 0 = external wiring Channel, channel number (101 indicates the	None

NO.	Command	Explanation	Parameters	Returning values
			embedded position 01A channel) offset value year month day	

1.4 System commands

NO.	Command	Explanation	Parameters	Returning values
1	SYSTem:VERSion? [<module>]	According to parameters, to Query version NO. of different modules, to overlook this parameter, returning back to SCIP version NO followed by system	“APPLication”: Software version No of main program “ElECtricity:FIRMware”: electric measuring board firmware version No.; “ElECtricity:HARDware”: electric measuring board hardware “OS:FIRMware”:system firmware version “OS:HARDware”: system hard ware version "JUNCtion:HARDware": hardware version of all junction box, separated by semicolon; "JUNCtion:FIRMware": firmware version of all junction box, separated by semicolon;	version NO.

NO.	Command	Explanation	Parameters	Returning values
2	SYSTem:ERRor[:NEXT]?	Query the next error item in the error queue and delete the item from the queue. The error queue can store 50 error messages. If there are more than 50, the last one is replaced with -350, "Queue overflow". System power down or *CLS instructions can clear the error queue.	None	wrong information
3	SYSTem:DATE<year>,<month>,<day>	Design the date of system	year month day	None
4	SYSTem:DATE?	Query system date	-	Year ,month ,day
5	SYSTem:TIME<hour>,<minute>,<second >	Design system time	hour minute second	None
6	SYSTem:KLOCk <Boolean>|ON|OFF	Design local lock-out state of system, only to lock out the functional operation of panel	1 ON: system is locked –out 0 OFF: system is unlock	None
7	SYSTem:KLOCk?	Query local lock-out state of system	None	1:lock 0:unlock
8	SYSTem:BEEPer:ALARm <Boolean>|ON|OFF	Design warning tone state	Open or not	None
9	SYSTem:BEEPer:TOUCh <Boolean>|ON|OFF	Design keypad tone state	Open or not	None

NO.	Command	Explanation	Parameters	Returning values
10	SYSTem:COMMunicate:SOCKet:WLAN[:ST ATe] <Boolean>|ON|OFF	Design WIFI state Attention: if the wifi is opened, the serial port of controller will be closed. During the time of opening wifi and connecting wifi, thecommunication with controller is only done through Ethernet	1,ON: open WIFI; 0,OFF: close WIFI	None
11	SYSTem:COMMunicate:SOCKet:WLAN[:ST ATe]?	Query wifi state	None	1: WIFI open ; 0: WIFI close
12	SYSTem:COMMunicate:SOCKet:WLAN:AD DRess<IP address>	Design the IP address of WIFI Before designing the DHCP、IP subset mask and gateway of WIFI, please confirm that the wifi module has been opened and doesn't connect with any hot spots.	P address: character string without quotation, format is <NR1>.<NR1>.<NR1>.<NR1>	None
13	SYSTem:COMMunicate:SOCKet:WLAN:AD DRess?	Query the IP address of WIFI	None	IPaddress
14	SYSTem:COMMunicate:SOCKet:WLAN:MA SK <IP address>	Design subnet mask of wifi Before designing the DHCP、IP subset mask and gateway of WIFI, please confirm that the wifi module has been opened and doesn't connect with any hot spots.	IP address: character string without quotation, format is <NR1>.<NR1>.<NR1>.<NR1>	None

NO.	Command	Explanation	Parameters	Returning values
15	SYSTem:COMMunicate:SOCKet:WLAN:MA SK?	Query subnet mask of WIFI	None	IP address
16	SYSTem:COMMunicate:SOCKet:WLAN:GA Teway <IPaddress>	Design gateway of wifi Before designing the DHCP、IP subset mask and gateway of WIFI, please confirm that the wifi module has been opened and doesn't connect with any hot spots.	IP address: character string without quotation, format is <NR1>.<NR1>.<NR1>.<NR1>	None
17	SYSTem:COMMunicate:SOCKet:WLAN: GA Teway?	Query gateway of wifi	None	IP address
18	SYSTem:COMMunicate:SOCKet:WLAN:MA C?	Query physical address of wifi	None	Physical address
19	SYSTem:COMMunicate:SOCKet:WLAN:DH CP[:STATe] <Boolean>|OFF|ON	Design WIFIDHCP state Before designing the DHCP、IP subset mask and gateway of WIFI, please confirm that the wifi module has been opened and doesn't connect with any hot spots.	1=ON: open DHCP; 0=OFF: close DHCP	None
20	SYSTem:COMMunicate:SOCKet:WLAN:DH CP[:STATe]?	Query WIFIDHCP state	None	1: DHCP open ; 0: DHCP close
21	SYSTem:COMMunicate:SOCKet:WLAN:SSI D? [ALL]	If the parameter is all, the Query will be done and all the Queried SSID names and the ways of encryption will be returned. If the parameter is overlooked,	None	{[“ssid: way of encryption”]}

NO.	Command	Explanation	Parameters	Returning values
		the result will return back to the current connected SSID name and the ways of encryption, if there is no connections or no queried hot spots, please return “		
22	SYSTem:COMMunicate:SOCKet:WLAN:CO NNect <ssid> [,<password>]	Make the wifi connect with the appointed hot spot	1)”ssid hot spot name, the character string with quotation 2) Encryption Mode: encryption Mode, WEP_OFF, WEP_ON, WEP_AUTO,WPA_PSK,WPA_TKIP, WPA2_PSK,WPA2_AES,CCKM_TKIP, WEP_CKIP,WEP_AUTO_CKIP, CCKM_AES,WPA_PSK_AES,WPA_AES, WPA2_PSK_TKIP,WPA2_TKIP, WAPI_PSK,WAPI_CERT; 3) password: the character string with quotation	None
23	SYSTem:COMMunicate:SOCKet:WLAN:C O NNect?	Search the connection state of wifi	None	Successfully, Initialization, SSIDNotFound SSIDNotConfigured, JoinFaile ScaningConfiguredSSID WaitingIPConfiguration ModuleJoinedListeningSocke ts

NO.	Command	Explanation	Parameters	Returning values
24	SYSTem:COMMunicate:SOCKet:WLAN:DIS Connect	Break the wifi connection	None	None
25	SYSTem:COMMunicate:SOCKet:WLAN:DB M?	Query signal strength and dBm value of WIFI	None	DBM value, unit dBm
26	SYSTem:COMMunicate:SOCKet:ETHernet: DHCP?	Acquire DHCP state of Ethernet	None	1=DHCP,0= static status
27	SYSTem:COMMunicate:SOCKet:ETHernet: DHCP <enable>	Design DHCP state of Ethernet	Open or nor enable,1=ON=open, 0=OFF=close	None
28	SYSTem:COMMunicate:SOCKet:ETHernet: ADDRess?	Acquire IP address of Ethernet	None	IPaddress
29	SYSTem:COMMunicate:SOCKet:ETHernet: ADDRess <ip>	Design the IP address of Ethernet under the static state	IP address	None
30	SYSTem:COMMunicate:SOCKet:ETHernet: MASK?	Acquire subnet mask of Ethernet	None	Subnet mask
31	SYSTem:COMMunicate:SOCKet:ETHernet: MASK <mask>	Design subnet mask of Ethernet under the static state	Subnet mask	None
32	SYSTem:COMMunicate:SOCKet:ETHernet: GATeway?	Acquire gateway of Ethernet	None	Gate way
33	SYSTem:COMMunicate:SOCKet:ETHernet: GATeway <gateway>	Design gateway of Ethernet under the static state	gateway	None
34	SYSTem:COMMunicate:SOCKet:ETHernet: PHYSicaladdress?	Read physical Address of Ethernet	None	Physical address
35	SYSTem:REGistry:INITiate [<Boolean>]	initialize registry		None
36	SYSTem:REGistry:DATA<QuoteStr>,<Quot	Write the key value to the registry.	1. Path: quoted string	None

NO.	Command	Explanation	Parameters	Returning values
	eStr>,<QuoteStr>,BINary|DWord|ExpandString|MultiString|QWord|String	BINary is binary data, and each byte is separated by -, for example, binary data 0x11, 0x22, 0xaa, 0xbb, expressed as "11-22-aa-bb"; DWord is a 32-bit integer; ExpandString specifies a NULL-terminated string containing an unexpanded reference to an environment variable (such as %PATH%, which expands when the value is retrieved).MultiString is an array of strings, separating each string with -, and a single string needs to be enclosed in parentheses, for example"(abc)-(123er)-(hello,333)" QWord is a 64-bit integer String is a string	2. The name of the key: a quoted string 3. Key value: quoted string 4. Value type	
37	SYSTem:REGistry:DATA? <QuoteStr>,<QuoteStr>	Read key value from registry	1. Path: quoted string 2. The name of the key: a quoted string	key value
38	SYSTem:REGistry:DELete<QuoteStr>,<Qu oteStr>	Delete key value from registry	1. Path: quoted string 2. The name of the key: a quoted string	None
39	SYSTem:REGistry:SAVE HKEY_LOCAL_MACHINE|HKEY_CLASSES_R OOT|HKEY_CURRENT_USER|HKEY_USERS| ALL	Save registry	Key name	None

NO.	Command	Explanation	Parameters	Returning values
40	SYSTem:PASSword:EDIT <oldPassword>,<newPassword>,<newPasswordRepeat>	Editor the user password	3 values, comma separated, password is only consist of figures. Old password/super administrative New password New password repeat	None
41	SYSTem:PASSword:ENABle:SENSor?	Query that the protection of sensor bank password is opened or not	None	1 value Open or not,1=open, 0=close
42	SYSTem:PASSword:ENABle:SENSor <enable>	Design the protection of sensor bank password	1 value enable,0=close,1=open	None
43	SYSTem:COMMunicate:BLUEtooth[:STATe] ?	Read open and close status of Bluetooth	None	1 value Open or not 1=open,0=close
44	SYSTem:COMMunicate:BLUEtooth[:STATe] <Boolean>|ON|OFF	Design open and close status of Bluetooth	1 value Open or not 1=open,0=close	None
45	SYSTem:COMMunicate:BLUEtooth:NAMe?	Read Bluetooth name	None	Character string: Bluetooth name
46	SYSTem:COMMunicate:BLUEtooth:NAMe <UnquoStr>	Design Bluetooth name	Character string: Bluetooth name	None

1.5 Program Commands

NO.	Command	Explanation	Parameters	Returning values
1	PROGram:RUN	Run the appointed program	1. program name, a quoted string;	-

NO.	Command	Explanation	Parameters	Returning values
	<progname>[,<parameters>]		2. Parameter, quoted string;	
2	PROGram:EXIT [<progname>]	Stop the program. without parameters means Stop program specified by PROGram:RUN	Program name, quoted string	-
3	PROGram:STATe? [<progname>]	Status of interrogator , without parameters means to question the program specified by PROGram:RUN	Program name, quoted string	RUNNING EXITED

1.6 Display commands

NO.	Command	Explanation	Parameters	Returning values
1.	DISPlay:BRIGhtness <type>,<level>	Design brightness	2 values, separated by comma type:Percentage,Value Level:	
2.	DISPlay:BRIGhtness? <type>	Query brightness	One value type:Percentage,Value	
3.	DISPlay:LANGuage?	Acquire languages	None	One value LCID
4.	DIAGnostic:LANGuage <lcid>[,<reboot>]	Set language	two values lcid reboot or not	None
5.	DISPlay:MESSagebox < "Message ">	Display dialog box	one value Message	None

6.	DISPlay:HOME?	Query whether in main interface	None	0 not in main interface,1 in main interface
7.	DISPlay:HOME	Return to the main interface from the current interface(Only support the system settings interface to return)	None	None
8.	DISPlay:THEMe?	Acuqire current theme pattern	None	One value topic name
9.	DISPlay:THEMe:ALLNames?	Acquire names of all current supported themes	None	Several values, separated by commas Theme name 1, theme name 2,…..
10.	DISPlay:THEMe <themeName>	Set system theme(work after restarting)	One value Support theme name	None

1.7 Function module commands

NO.	Commands	Explanation	Parameters	Returning value
1	PATTern:MAIN:PATTerns Dual|SCMM|SConn[,<"otherParams">]	Switch main interface function	2 values, separated by commas Function: Dual=Thermometry bridge SCMM=Multichannel Thermometer, SConn=Smart connection; "otherParams", extra parameters, smart wiring means only test the address of the junction box	None
2	PATTern:SCONn:MATCh	Set the matching conditions of the	2parameters, separated by commas	None

	<paramIndex>[,<” matchStr” >]	intelligent wiring base	parmaIndex, matching the corresponding parameter number, 1 = ChannelInfo1, 2 = ChannelInfo2, 3 = ChannelInfo3, other = close match; "matchStr": A matching string with quotes.	

1.8 Unit commands

NO.	Commands	Explanation	Parameters	Returning value
2	UNIT:TEMPerature <unit_ID>|<unit_name>	Design temperature unit of current system	Unit: unit name or unit ID unit_name is the character string with quotation unit_ID is digit	None
3	UNIT:TEMPerature?	Acquire temperature unit of current system	None	2values, comma separated. Name of temperature unit, temperature unit id

1.9 Sensor commands

No.	Commands	Explanation	Parameters	Returning value

1	SENSor:COUNt? <SenorType>	Acquire the quantity of sensor	1 value
			SenorType:
			RTD=10=customized thermal resistance,
			SPRT=3=standard platinum resistance, CVD=2=CVD custom resistance
			NTC=1=the negative temperature coefficient thermistor β coefficient calculation formula
			NTC_SH3=13=the negative temperature coefficient thermistor Steinhart-Hart calculation formula,
			StandardTC=6=standard thermocouple,
			StandardTCB=60 type B standard thermocouple
			StandardTCS=61 type S standard thermocouple
			StandardTCR=64 type R standard thermocouple
			StandardTCPolynom=62 = polynomial standard TC(B，S, R)
			CustomTC=63=custom standard thermocouple
			SensorUUT= All temperature sensor types
			Pressure=110=pressure sensor,
			Humidity=111=humidity sensor,
			CurrentTransmitter=100= current type transmitter,
			VoltageTransmitter=101=voltage type transmitter,
			TransmitterUUT= All transmitter types,
			UUT= all sensor types (excluding standard resistance)
			RS=102=standard resistance
				1 value
				the quantity of custom sensor

2	SENSor:CATalog:HEAD?
	<SensorType>,<offset>,<count>
		Acquire sensor head information	3 values
			SensorType:
			RTD=10=customized thermal resistance,
			SPRT=3=standard platinum resistance, CVD=2=CVD custom resistance
			NTC=1=the negative temperature coefficient thermistor β coefficient calculation formula
			NTC_SH3=13=the negative temperature coefficient thermistor Steinhart-Hart calculation formula,
			StandardTC=6=standard thermocouple,
			StandardTCB=60 type B standard thermocouple
			StandardTCS=61 type S standard thermocouple
			StandardTCR=64 type R standard thermocouple
			StandardTCPolynom=62 = polynomial standard TC(B，S, R)
			CustomTC=63=custom standard thermocouple
			SensorUUT= All temperature sensor types
				Sensor head information
				Temperature sensor: information between groups is separated by”&”, information within one group is separated by “,”
				Each group contains: ID，Name，SN, type of electrical, calibration date, calibration interval

			Pressure=110=pressure sensor，
			Humidity=111=humidity sensor，
			CurrentTransmitter=100= current type transmitter，
			VoltageTransmitter=101=voltage type transmitter，
			TransmitterUUT= All transmitter types，
			UUT= all sensor types (excluding standard resistance)
			RS=102=standard resistance
			
			offset，
			
			count	

3	SENSor:TEMPerature:INFormations? <id>	Acquire the information of single temperature sensor	1 value Sensor id	2 values： Sensor type， “info”: Info reference format, refer to the second parameter of SENSor:TEMPerature:ADD
4	SENSor:TEMPerature:ADD <SensorType >,< "Info">	Add temperature sensor	1、SensorType: RTD=10=customized thermal resistance, SPRT=3=standard platinum resistance, CVD=2=CVD custom resistance NTC=1=the negative temperature coefficient thermistor β coefficient calculation formula NTC_SH3=13=the negative temperature coefficient thermistor Steinhart-Hart calculation formula, StandardTCB=60 type B standard thermocouple StandardTCS=61 type S standard thermocouple StandardTCPolynom=62 = polynomial standard TC(B，S) CustomTC=63=custom standard thermocouple 2、data information“info” **Common information**： Name SN ElectricalUnitId: electrical unit id（refer to：appendix 1，such as mV = 1243） TemperatureUnitId: Temperature Unit id(refer to：appendix 1；such as ℃ =1001) TMin: minimum temperature range TMax: maximum temperature range CalibrateDate: (such as：M/D/Y, 8/31/2020) IndateDays: required calibration interval Note :	

			IsReadOnly: read-only or not **RTD:** BaseSensorName:（refer to：appendix 2） R0: the resistance value at 0 temperature **SPRT:** a_Positive b_Positive c_Positive d_Positive a_ Negative b_ Negative HRType：positive calculation coefficient 0："a6,b6,c6,d,W660_323:(0~961.78)℃", 1:"a7,b7,c7:(0~660.323)℃", 2:"a8,b8:(0~419.527)℃", 3:"a9,b9:(0~231.928)℃", 4:"a10:(0~156.5985)℃ ", 5:"a11:(0~29.7646)℃" LRType：negative calculation coefficient 0："a4,b4:(-189.3442~0.01)℃" 1："a5,b5:(-38.8344~0)℃" 2："None" Rtp：the resistance value at 0 temperature W660_323: aluminum point **CVD:** A B C eMax eMin R0: the resistance value at 0 temperature **StandardTCB：standard type B TC** baseTC base TC type（B） data: data to be modified， For fix point correction: The data corresponds to the mV value of 1100， 1200,1300,1400,1500℃. If null, it will be filled with default value, different data are separated by “&” For coefficient correction: The data corresponds to the correction value of a0, a1 and a2. If null, it will be filled with zero, different data are separated by “&” 4: fix point correction	

			5. coefficient correction **StandardTCS：standard type S TC** baseTC base TC type（S） paramType: srandard tc parameter type 0：A_B_C 1：Zn_Al_Cu 2：Zn_Sb_Cu Param1：1 st parameter Param2：2 nd parameter Param3：3 rd parameter **StandardTCR：standard type R TC** baseTC base TC type（R） paramType: srandard tc parameter type 5. coefficient correction data: data to be modified，corresponds to the correction value of a0, a1 and a2. If null, it will be filled with zero, different data are separated by “&” **StandardTCPolynom：standard TC polynomial** baseTC base TC type（B,S,R） TtoE set of temperature to electrical parameter EtoT set of electrical to temperature parameter Note：the structure of TtoE and EtoT：The two parameters can be a set of multiple parameter groups. Each group of data is separated by “&”, and the data in the group is separated by “|”; if it is null, it should use the default parameters corresponding to baseTC. Each group of data contains: TRangelow: low limit of temperature range, TrangeHigh: high limit of temperature range, a0,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10 InversePolynomialEnable 0: disable. EtoT uses polynomial calculation 1: enable, EtoT uses inverse polynomial calculation **CustomTC: custom TC** baseTC base TC type（T,E,J,K,N） paramType：parameter type 0 offset，1 polynomial offsetOrOrder： paramType=0: offset value paramType=1: highest power (1,2,3) sensorParam: corrected data，can be null effective when paramType=1；the quantity of the	

			groups is offsetOrOrder+1，separeted by ”&”; each group contains standard value and readout value，separeted by ”| ” (standard value cannot be same) **NTC：Thermistor** Rn Tn β_NTC **NTC_SH3：negative temperature coefficient thermistor** powerOfPolynomial (3~6) PolynomialA: coefficients of the Polynomial, separeted by“&”，the quanity of coefficients= PowerOfPolynomial+1 (A0&A1&A2&A3&A4&A5&A6)	
5	SENSor: TEMPerature: EDIT <id>,< "Info">	Edit sensor, use detailed information	Parameter 1：Id of the sensor needs edited Parameter 2：data information“info” Refer to the second parameter of SENSor:TEMPerature:ADD	
6	SENSor: TEMPerature:DELete <”ids”>	Delate sensor	Sensor ids, separated by comma in the quote mark	None

NO.	Commands	Explanation	Parameters	Returning value
2	SENSor:CATalog? <SensorType>,<offset>,<count>	Acquire sensor head information	3 values Sensor type: RTD=10=custom thermal resistance, SPRT=3=standard platinum resistance, CVD=2= The Callendar – van Dusen coefficients, NTC=1= the negative temperature coefficient thermistor β coefficient calculation formula, NTC_SH2=12=the negative temperature coefficient thermistor Steinhart-Hart calculation formula, StandardTC=6=standard thermocouple, SensorUUT=All temperature sensor types, Pressure=110=pressure transmitter, Humidity=111=humidity transmitter, CurrentTransmitter=100=current type transmitter, VoltageTransmitter=101=Voltage Transmitter, TransmitterUUT=All transmitter types, UUT=all sensor types (excluding standard resistance) RS=102=standard resistance Starting position offset, Count	3values, comma separated ClassName,the real thing is List<SensorHeader> Base64 character data CRC16 check code

NO.	Commands	Explanation	Parameters	Returning value
4	SENSor:INFormations? <id>	Acquire the information of single sensor	One value Sensor id	3 values, separated by commas ClassName , actually is TemperatureSensorInfo or TransducerInfoor or StandardResInfo Base64 character data CRC16 check code

5	SENSor:SETSensorinfo:ADD
	<SensorType >,< "Info">
		Add new sensor	2values
			Sensor type: RTD=10=custom thermal resistance,
			SPRT=3=standard platinum resistance,
			CVD=2= The Callendar – van Dusen coefficients
			NTC=1= the negative temperature coefficient thermistor β coefficient calculation formula,
			NTC_SH2=12=the negative temperature coefficient thermistor
			Steinhart-Hart calculation formula,
			StandardTC=6=standard thermocouple,
			Pressure=110=pressure transmitter,
			Humidity=111=humidity transmitter,
			RS=102=standard resistance "Info" is Base64 character data
				None

NO.	Commands	Explanation	Parameters	Returning value

6	SENSor:Delete <”ids”>	Delate sensor	2values, SensorUUT=Temperature Sensor
			TransmitterUUT=Pressure/Humidity Transmitter
			RS = standard resistance
			Sensor ids, separated by commas within quotes
				None

7	SENSor:Query? <”condition”>	Query sensor	2values, separated by commas:
			SensorUUT=Temperature Sensor
			TransmitterUUT=Pressure/Humidity Transmitter
			RS = standard resistance
			Sensor Query condition "condition",
			Base64 character data
				3values,separatedby commas
				ClassName, actually List< SensorHeaderInfo >
				Base64 character data
				CRC16 check code

Appendix 1: SCPI unit id list

Unit Id	Unit
2000	Text unit
32767	Empty unit
1211	mA
1212	µA
1209	A
1240	V
1243	mV
1281	Ω

1284	kΩ
1283	MΩ
1000	K
1001	℃
1002	℉
1003	°R
999	°Re
1005	°
1342	%
1133	kPa
1130	Pa
1131	GPa
1132	MPa
1134	mPa
1135	μPa

1136	hPa
1137	bar
1138	mbar
1139	torr
1140	atm
1141	psi
1142	psia
1143	psig
1144	gf/cm^2
1145	kgf/cm^2
1147	inH2O@4°C
1148	inH2O@68°F
1150	mmH2O@4°C
1151	mmH2O@20°C
1153	ftH2O@4°C
1154	ftH2O@68°F

1156	inHg@0°C
1158	mmHg@0°C
2001	mtorr
2002	lb/ft^2
2003	tsi
2004	psf
2005	inH2O@60°F
2006	ftH2O@60°F
2007	cmH2O@4°C
2008	mH2O@4°C
2009	cmHg@0°C
2010	mHg@0°C
2011	kgf/m^2

Appendix 2: default industrial sensor

Sensor type	Sensor name(used in command)
R100	100Ω
R400	400Ω
R4k	4kΩ
Pt100-385	Pt100(385)
Pt10-385	Pt10(385)
Pt50-385	Pt50(385)
Pt200-385	Pt200(385)
Pt400-385	Pt400(385)
Pt1000-385	Pt1000(385)
Pt25-385	Pt25(385)
Pt100-3916	Pt100(3916)
Pt100-3926	Pt100(3926)
Pt100-391	Pt100(391)
Cu100-428	Cu100(428)
Cu50-428	Cu50(428)

Cu10-427	Cu10(427)
Ni100-617	Ni100(617)
Ni100-617	Ni100(618)
Ni120-672	Ni120(672)
Ni1000	Ni1000
TC-S	S
TC-R	R
TC-B	B
TC-K	K
TC-N	N
TC-E	E
TC-J	J
TC-T	T
TC-C	C
TC-D	D
TC-G	G

TC-L	L
TC-U	U
TC-LR	LR
TC-A	A
mV	mV

Appendix 3: Error Definition

NO.	Error code	Error description	Explain
1	0	No error	No error
Command error			
2	120	Commandparameter error	Command parameter error
3	-108	Parameter not allowed	Too many parameters,or no parameters in the command withparameters
4	-109	Missing parameter	Missing parameter
5	-110	Command header error	The command header is error
6	-114	Header suffix out of range	Command header suffix overrange
7	-123	Numeric overflow	Digital spillover,the absolute exponential value of a number greater than43
8	-151	Invalid string data	Invalid string, such as quotation mark mismatch
9	-171	Invalid expression	Invalid expressions, such as parentheses mismatch

NO.	Error code	Error description	Explain
		Execution error	
10	-200	Execution error	Execution error
11	-221	Settings conflict	Setting Conflicts
12	-222	Data out of range	Parameter values exceed the valid range of instructions
13	-223	Too much data	Too much data to process
14	-224	Illegal parameter value	Illegal parameter values
15	-230	Data corrupt or stale	The data is invalid, or is reading the data, and no valid data has been obtained.
16	-240	Hardware error	Hardware failure
17	-256	File name not found	No filename found
18	-282	Illegal program name	Illegal procedure name
19	220	Measure error	Measurement error
20	221	Failed to set meaure function	Failure to switch measurement items
21	222	Failed to read measure value	Failed to read measurements
22	223		
23	224		
24	240	Control error	Control error
25	241		
26	242		
27	243		
28	260	Calibration error	Calibration error
29	261	Calibration secured	The equipment is in calibration protection state and cannot perform calibration.
30	262	Invalid calibration secure code	Invalid Calibration Password
31	263	Missing calibration value	This error occurs when setting the calibration value without setting the calibration point in current/voltage calibration.

Serial NO.	Error code	Error description	Explain
32	264	Missing calibration data	This error occurs when the calibration point is set continuously without setting the calibration value.
33	265	Failed to set calibration function	Setting Calibration Item Failed
34	266	Calibration data is not enough	When saving calibration data, this error occurs if the calibration data does notreach three points.
35	271	Setion_name_not_found	No paragraph name found
36	272	Key_name_not_found	No key name found
37	291	Update secured	The equipment is upgraded and protected and cannot be upgraded.
38	292	Invalid update secure code	Invalid upgrade password
39	293	Not found the service pack	No upgrade package found
40	294	The service pack unavailable	Upgrade package unavailable
41	295	App Update not found	Can't find AppUpdate.exe
Equipment-related errors			
42	-310	System error	System error
43	-311	Memory error	Memory error
44	-350	Queue overflow	Queue overflow
45	-360	Communication error	Communication error
46	301	Internal module is not connected	Unconnected internal module
47	302	External module is not connected	Unconnected External Modules
48	303	Supply module is not connected	Unconnected positive pressure module
49	304	Vacuum module is not connected	Unconnected negative pressure module
50	361	Open WLAN Failed	Failed to open WIFI
51	362	Set WLAN address mode failed	Failed to set WIFI address mode
52	363	Set WLAN address failed	Failed to set WIFI address

NO.	Error code	Error description	Explain
53	364	Communication port to WIFI module is not open	Communication port with WIFI module is not open
54	365	WLAN is not connected	WIFI not connected

Appendix 4: State report

4.1Status byte register

Register of status bytes shows the information of other state registers. Its value is unlocked, so if an event register is done with zero cleaning, the corresponding bits of Register of state bytes will also be done with zero cleaning.

Table 4-1 Definition of status byte register places

Bytes	Decimalism value	Definition	Explanation
0	1	Unused	Always 0
1	2	Unused	Always 0
2	4	Error queue	error queue is not empty
3	8	Question data	Many Bits set 1 or one of question data register(corresponding places of enabling register must work)
4	16	Unused	0 always 0
5	32	Standard event	Many Bits set 1 or one of Standard event register(corresponding places of enabling register must work)
6	64	Service request	Many Bits set 1 or 1 bit except this bit (corresponding places of enabling register must work)
7	128	Operation state	Many Bits set 1 or one of Standard event register(corresponding places of enabling register must work)

4.2Standard event register

Standard event register shows the following events: power on, grammatical error of command, command execution error, the error of self-testing or calibration, or a *OPC command have been executed. The bits are defined as follows:

Table 4-3 standard event register bit definition

Bits	Decimalism value	Definition	Explanation
0	1	Finished operation	Before *OPC command, the other command are all executed
1	2	Unused	always 0
2	4	Unused	always 0
3	8	Instrument error	The error of self-testing , calibration or overloading
4	16	Execution error	Execution error occurred
5	32	Commands erroe	Commands grammatical error occurred
6	64	Unused	
7	128	Power on	Power on and off operation occurred

4.3 Question data register

Question data register shows the information of testing results, for example: outrange and so on. Its bit definition is as follows:

Table 4-3 Definition of question data register place

Bytes	Decimalism value	Definition	Explanation
0	1	Voltage overload	Voltage overrange
1	2	Current overload	Current overrange
2	4	Unused	Always 0
3	8	Unused	Always 0

4	16	Unused	Always 0
5	32	Unused	Always 0
6	64	Unused	Always 0
7	128	Unused	Always 0
8	256	Unused	Always 0
9	512	Pressure overload	Pressure overrange
10	1024	Unused	Always 0
11	2048	Unused	Always 0
12	4096	Unused	Always 0
13	8192	Unused	Always 0
14	16384	Unused	Always 0
15	32768	Unused	Always 0

4.4 Operation status register

The operational status register provides information on the normal operation of the device. Its bits are defined as follows:

Table 4-4 operation status register bit definition

Bits	**Decimalism value**	**Definition**	**Explanation**
0	1	Unused	Always 0

1	2	Unused	Always 0
2	4	Unused	Always 0
3	8	Unused	Always 0
4	16	Measuring	Device is initiative to make pressure measurement
5	32	Unused	Always 0
6	64	Unused	Always 0
7	128	Unused	Always 0
8	256	Unused	Always 0
9	512	Unused	Always 0
10	1024	Unused	Always 0
11	2048	Unused	Always 0
12	4096	Unused	Always 0
13	8192	Unused	Always 0
14	16384	Unused	Always 0
15	32768	Unused	Always 0

Appendix 5: ADT286 Programming Commands Illustration

1. Read information of front panel and scanners

   Command format: [MEASure:]MODule:INFormation?

   Example: **MODule:INFormation?**

   Returned value example：0,,0,,,2,;1,6851019T10005,1,TAU-M1 V01.00.00.00,TAU-M1 V01.05,20,

   Note: N*7 values,N may be 1,2,3,4,5, each message is separated by semicolon, each parameter is separated by commas.

2. Read channel configuration of one scanner

   Command format: [Measure:]MODule:CONFig? <moduleIndex>

   Example: **MODule:CONFig? 0**

   Returned value example：REF1,1,,3,0,0,1,1,4,Pt25(385),,,0,0;REF2,0,,4,0,0,1,1,2,Auto Range,,;

   Note: (1) It can get the channel name, as below:

   - x means module No., the embedded module=1, then the serial-wound modules are in 2.3.4
   - Front panel: REF1 and REF2
   - TS module: CHx-01A~10A   CHx-01B~10B,
   - PS module: CHx-01~10

   (2) Description:

   - moduleIndex: Module identifier. The front panel is 0, the embedded module is 1, then the serial-wound modules are in 2.3.4, depending on the connection.
   - Read the front panel channel configuration, REF1 is for industrial RTD Pt25(385), REF2 is for thermistor.

3. Set the channel configuration of one scanner

Command format：[MEASure:]MODule:CONFig <moduleIndex>,<”params”>

Example: **MODule:CONFig 0,"REF1,1,,3,0,0,1,1,4,Pt25(385),,,0,0;REF2,1,,4,0,0,1,1,2,Auto Range,,;"**

Note: You can refer to the returned valve in command No 2 as the second parameter of this command.

4. Read scanning configuration

Command format: [MEASure:]SCAN:STARt?

Example: **SCAN:STARt?**

Returned value example：1000,REF1

Note: In this example, set the sampling frequency cycle as 1000 (can be 100,1000,4000), channel name is REF1.

5. Set the configuration and start scanning

Command format: [MEASure:]SCAN:STARt <”params”>

Example: **SCAN:STARt "1000,REF1"**

Returned value example：None

6. Read scanning data, most recent one.

Command format: [MEASure:]SCAN:DATA:Last? [<time>]

Example: **SCAN:DATA:Last?**

Returned value example："REF1,1281,1,28.258167,28.258167,1001,1,33.512077;"

7. Read channel configuration

Command format: [MEASure:]CHANnel:CONFig? <”channelName”>

Example: **CHANnel:CONFig? "REF1"**

Returned value example：REF1,1,,3,0,0,1,1,4,Pt25(385),,,0,0

8. Set channel configuration

Command format: [MEASure:]CHANnel:CONFig

<”chName”>,<enable>,<”label”>,<elecType>,<range>,<delay>,<autoRange>,<filter>,<”otherParam”>

Example: **CHANnel:CONFig "REF1",1,"",3,0,0,1,1,"4,Pt25(385),,,0,0"**

9. Set multi-channel configuration and start scanning

Command format: [MEASure:]SCAN:MULT:STARt <Numeric>,<"List">

Example: **SCAN:MULT:STARt 1000,"REF1,CH1-01A,CH1-02A"**

Returned value example：None

10. Stop scanning

Command format: [MEASure:]SCAN:STOP

Example: **SCAN:STOP**

Returned value example：None

11. Read channel configuration

Command format: [Measure:]CHANnel:CONFig? <"channelName">

Example: **CHANnel:CONFig? "REF1"**

Returned value example：REF1,1,,3,1,0,1,1,4,Pt100(385),,,0,0

Note:

Returned value: 8+m values, separated by commas. All channels have at least 8 returned values, “m” depends on the channel type.

- The fourth parameter is the channel type, 0=voltage, 1=current, 2=resistance, 3= industrial RTD, 4=thermistor, 100= industrial TC, 101=switch, 102=SPRT, 103=voltage transmitter, 104=current transmitter, 105=standard TC, 106=custom TC, 110=standard resistance.
- In this example, the channel type is 3, which indicates industrial RTD. For industrial TC, m=6, so it has totally 8+6=14 parameters. The ninth parameter is the wiring, tenth parameter is sensor name.
- For other parameters, please refer to the commands set document.

12. Set channel configuration

Command format: [MEASure:] CHANnel:CONFig

<”chName”>,<enable>,<”label”>,<elecType>,<range>,<delay>,<autoRange>,<filter>,<”otherParam”>

Example: **CHANnel:CONFig "REF1",1,"",3,1,0,1,1,"4,Pt100(385),,,0,0"**

Note: You can refer to the returned values of command No.7, add double quote mark for first and third parameters, and combine the 9th to last parameters into one parameter with double quotes. Use this as the parameter for this commands.

13. Create/Add temperature sensor

Command format: SENSor:TEMPerature:ADD <SensorType >,< "Info">

Example: **SENSor:TEMPerature:ADD** SPRT,"SPTESensor,Sensor123,1281,1001,-189.344192504883,961.780029296875,11/2/2021,365,,False,-0.000579849,-3.60548E-5,-3.45108E-5,0,-0.000520946,-0.000296057,1,1,100.0131,0"

Note: <SensorType >: the sensor type string, please refer to commands set document.

< "Info">: the unused parameters can be set to 0, select positive and negative temperature parameters and decide whether a, b, c and d are used for calculation.

14. Read the quantity of sensor

Command format: SENSor:COUNt? <SenorType>

Example: **SENSor:COUNt? SPRT**

Returned value example：2

15. Read sensor head information

Command format: SENSor:CATalog:HEAD? <SensorType>,<offset>,<count>

Example: **SENSor:CATalog:HEAD? SPRT,0,10**

Returned value example：9d1b93d3-1aac-4f5b-8e62-a0853ef7c471,KPTESTSENSOR,KP12345667,SPRT,11/2/202112:00 AM,365&b364ea60-5da4-463e-8976-808af704c11e,SPTESensor,Sensor123,SPRT,11/2/202112:00 AM,365

Note: The sensor ID will be used when viewing and editing sensor information.

If the quantity of sensor types that need to be returned is greater than the quantity that exists in the devices, only the head information of the existing sensors will be returned.

Each group of the sensor head information are separated by commas.

16. Read the information of single temperature sensor

Command format: SENSor:TEMPerature:INFormations? <id>

Example: **SENSor:TEMPerature:INFormations? b364ea60-5da4-463e-8976-808af704c11e**

Returned value example：SPRT,"SPTESensor,Sensor123,1281,1001,-189.344192504883,961.780029296875,11/2/2021,365,,False,-0.000579849,-3.60548E-05,-3.45108E-05,0,-0.000520946,-0.000296057,1,1,100.0131,0"

17. Edit/Change the information of existing temperature sensors

Command format: SENSor:TEMPerature:EDIT <id>,< "Info">

Example: **SENSor:TEMPerature:EDIT** b364ea60-5da4-463e-8976-808af704c11e,"SPTESensor,Sensor111,1281,1001,-189.344192504883,961.780029296875,11/2/2021,365,,False,-0.000579849,-3.60548E-5,-3.45108E-5,0,-0.000520946,-0.000296057,1,1,100.0131,0"

Note: In this example, it changed the S/N of sensor b364ea60-5da4-463e-8976-808af704c11e, changed from Sensor123 to Sonsor111.

18. Delete the information of temperature sensor

Command format: SENSor:TEMPerature:DELete <"ids">

Example: **SENSor:TEMPerature:DELete "9d1b93d3-1aac-4f5b-8e62-a0853ef7c471,b364ea60-5da4-463e-8976-808af704c11e"**

Note: In this example, it deletes multiple sensors at the same time. Different sensors are separated by “,” without Space.