SCPI Command Reference

This chapter contains the following information:

- Command parser - rules of the command parser.
- Command syntax - command line writing rules
- Query syntax - writing rules of query command
- Query response - format of the query response
- Command reference

This section provides all SCPI commands used by the AT381x. With these SCPI commands, you can complete control over all functions of the instrument.

1.1 Handshake Protocol

Since the AT381x uses the minimum subset of the RS-232 standard and does not use hardware handshaking signals, the AT381x can use software handshaking in order to reduce possible data loss or data errors in communication. High-level language software engineers should strictly follow the handshake below agreement to make preparation of computer communication software:

- Instrument terminator only accepts ASCII format, command response also returns ASCII code.
- **Command string that sent by host must be ended with LF/CR/LF+CR mark, instrument terminator will begin performing command string only after it receives end mark.**
- Instrument can set command handshake: instrument will return an identification code after it receives command and finishes processing.

If the host cannot accept the data returned by the instrument, you can try to solve it by using the following methods:
1. The software handshake is turned off, please refer to the <SYSTEM CONFIG> page of the instrument to turn it on.
2. The serial port connection is faulty, please check the cable connection.
3. The communication format of the high-level language program on the computer side is incorrect. Try on checking the serial port number, whether the communication format is correct, and the baud rate is the same as the instrument setting.
4. If the instrument is parsing the last command and the host cannot receive the response from the instrument, please try again later.

1.2 Terminator

The AT381x supports 4 terminators:
LF (Hexadecimal：0x0A)
CR(Hexadecimal：0x0D)
CR+LF (Hexadecimal：0x0D 0x0A)

The terminator can be selected in the system configuration page, and the instrument defaults is LF.

Note:

The AT381x allows the command sent by the host without the terminator, but it is recommended to add the terminator at the end of the command. Otherwise, it will cause a timeout wait after each command is received (the command timeout is 10ms~50ms depending on the baud rate).

1.3 Terminator

Host can send a string of command to instrument, instrument terminator will begin parsing after it captures end mark (LF, CR or LF+CR) or after input buffer overflows.

For example:

Legal command string:

AAA:BBB CCC,DDD,EEE<LF>
AAA:BBB CCC,DDD,EEE<CR>
AAA:BBB CCC,DDD,EEE<LF+CR>

The AT381x terminator is responsible for parsing and performing all commands, before programming, users must know about parsing rules.

1.3.1 Terminator Rules

1. Terminator only parses and responds ASCII code's data.
2. **Command string must be ended with terminator, terminator will begin performing command string only after it receives end mark or after** buffer overflows.
3. If command handshake is turned on, every time terminator receives one string, it will promptly return this string to the host, only when host receives this returned string, can it continues sending the next string.
4. After terminator parses error, it will promptly stop parsing, and the current command is canceled.
5. When terminator parses the query command, it will terminate parsing this command string, the latter command string will be ignored.
6. When parsing command string, terminator is case insensitive.
7. Terminator supports command abbreviated form, please refer to the latter chapter regarding abbreviation norms.

1.3.2 Notation Conventions and Definitions

This chapter employs some marks, these marks are not a part of command tree; they are only for better understanding of command string.

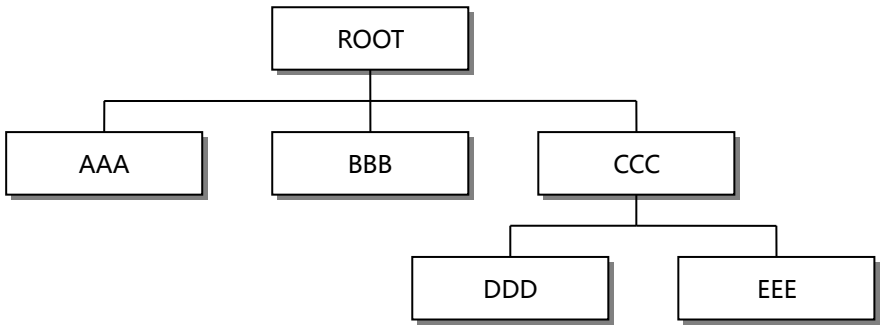
< >	the character in <> means this command's parameter
[ ]	the character in [] means optional command
{}	When there includes several parameter items in {}, means that users can only choose one item from it.
( )	the abbreviated form of parameter is put in ()
Capital letter	Abbreviated form of command.

1.3.3 Command Structure

The SCPI commands are tree structured three levels deep. The highest level commands are called the subsystem

commands in this manual. So the lower level commands are legal only when the subsystem commands have been selected. A colon (:) is used to separate the higher level commands and the lower level commands.

Figure 0-1 Command Tree Structure



Example

ROOT:CCC:DDD ppp

ROOT Subsystem Command

CCC Level 2

DDD Level 3

ppp Parameter

1.4 Header and Parameters

A command tree consists of header and parameters, it uses a space（ASCII: 20H）to separate in the middle.

Example

AAA:BBB 1.234

Header [Parameter]

1.4.1 Header

Headers can be of the long form or the short form. The long form allows easier understanding of the program code and the short form is suitable for writing.

1.4.2 Parameter

- Single command word, no parameter.
  Example: AAA:BBB
- Parameter can be character string form, the abbreviation rules are the same as the rules for command.
  Example: AAA:BBB 1.23
- Parameter can be numeric form
  - *<integer>* integer 123, +123, -123
  - *<float>* floating number
  1. <Fixfloat>: fixed point floating number: 1.23, -1.23
  2. <Scifloat>: scientific notation floating number: 1.23E+4, +1.23e-4
  3. <Mpfloat>: multiplier expressed by floating number: 1.23k, 1.23M, 1.23G,1.23u

Table 0-1 Multiplier Mnemonics

Definition	Mnemonic
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

Multiplier is Case Insensitive, its writing style is different from standard name.

1.4.3 Separator

The AT381x terminator only accepts allowed separators, terminator will occur "Invalid separator (illegal separator)" error if beyond this separator, and these separators include:

: Colon, used for separate command tree, or restart command tree.

Example: AAA:BBB:CCC 123.4

? Question mark, used for query

Example: AAA?

□ Space, used for separate parameter

Example: AAA:BBB□1.234

1.4.4 Error Code

The AT381x temporarily stores the result of the processing of each received command in the buffer. Before the next command is sent, the status can be acquired by the ERR? Command. If the error code function is turned on in [SYSTEM CONFIG] page, the instrument will automatically return the processing result after processing the command.

Table 0-2 SCPI error code

Error code	Description	Explanation
*E00	NO ERROR	No error
*E01	BAD COMMAND	Command error
*E02	PARAMETER ERROR	Parameter error
*E03	MISSING PARAMETER	Missing parameters, With parameter commands, no parameters are provided
*E04	INPUT BUFFER OVERRUN	Receive buffer overflow, the maximum buffer of the AT381x is 1000 bytes
*E05	SYNTAX ERROR	Syntactic error
*E06	INVALID SEPARATOR	Invalid separator

*E07	INVALID MULTIPLIER	Invalid multiplier
*E08	BAD NUMERIC DATA	Value error
*E09	VALUE TOO LONG	The value is too long, the numeric parameter exceeds 20 bytes
*E10	INVALID COMMAND	Invalid command, the command is invalid under certain conditions
*E11	UNKNOWN ERROR	Other unknown errors except the above errors

1.5 Command Reference

- DISPlay SUBSYSTEM
- FUNCtion SUBSYSTEM
- FREQuency SUBSYSTEM
- VOLTage SUBSYSTEM
- APERture SUBSYSTEM
- FETCh SUBSYSTEM
- COMParator SUBSYSTEM
- LIST SUBSYSTEM
- CORRection SUBSYSTEM
- TRIGger SUBSYSTEM
- BIAS SUBSYSTEM
- FILE SUBSYSTEM
- ERRor SUBSYSTEM
- Common commands

  *TRG

  *IDN?

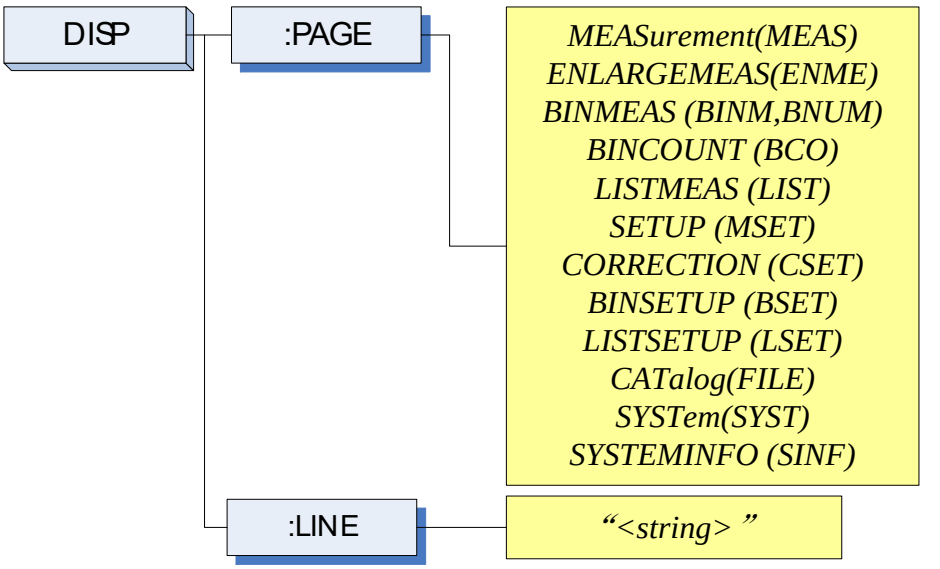
  *SAV

  *RCL

1.6 DISPlay Subsystem

The DISP Subsystem command group sets the display page.

Figure 0-1 DISP Command Tree



1.6.1 DISP:PAGE

The :PAGE command sets the display page.

The :PAGE? Query returns the abbreviated page name currently displayed on the LCD screen.

Command Syntax	**DISP:PAGE <*page name*>**
Parameter	**Where, <page name> is:** MEASurement [or MEAS] Sets display page to MEAS DISPLAY ENLARGE[or ENLA] Sets display page to ENLARGE DISPLAY BINMEAS [or BINM] Sets display page to BIN MEAS BINCOUNT [or BCO] Sets display page to BIN COUNT LISTMEAS [or LIST] Sets display page to LIST MEAS SETUP [or MSET] Sets display page to MEAS SETUP CORRECTION [or CSET] Sets display page to CORRECTION BINSETUP [or BSET] Sets display page to BIN TABLE LISTSETUP [or LSET] Sets display page to LIST TABLE CATalog [or CAT] Sets display page to FILE SYSTem [or SYST] Sets display page to SYSTEM CONFIG SYSTEMINFO [or SINF] Sets display page to SYSTEM INFORMATION
Example	**SEND> DISP:PAGE SYST //Set to the SYSEMT CONFIG**
Query Syntax	**DISP:PAGE?**
Query Response	**<*page name*>**
Example	**SEND> DISP:PAGE?** **RET> SYST**

1.6.2 DISP:LINE

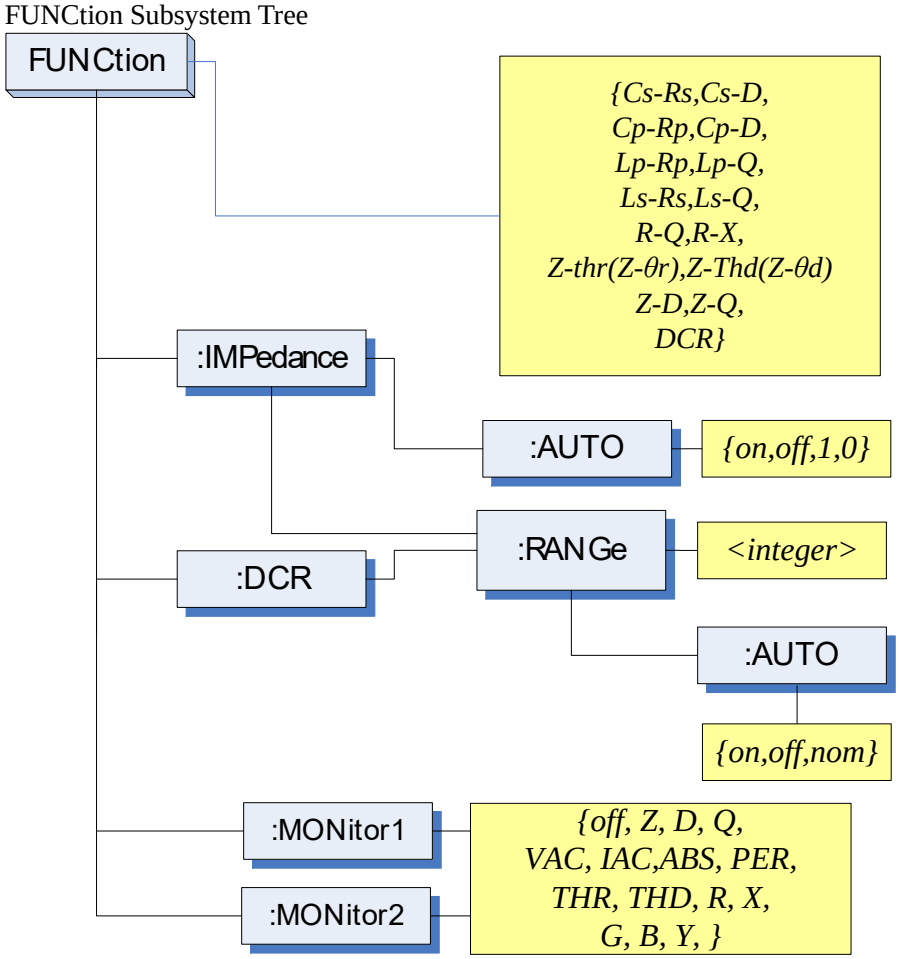
The :LINE command enters an arbitrary comment line of up to 30 ASCII characters in the comment field.

Command Syntax	**DISP:LINE "<*string*>"**
Parameter	Where, <string> is ASCII character string (30 ASCII characters)
Example	**SEND> DISP:LINE "This is a comment."**

1.7 FUNCtion Subsystem

The FUNCtion subsystem command group sets the measurement function, the measurement range, monitors parameter control.

Figure 0-2 FUNCtion Subsystem Tree



1.7.1 FUNCtion

The FUNCtion command sets the measurement function.

Command Syntax	`FUNC <function>`

Parameter	`Where, <function> is:`
	`Cs-Rs, Cs-D, Cp-Rp, Cp-D,`
	`Lp-Rp, Lp-Q, Ls-Rs, Ls-Q,`
	`Rs-Q, Rp-Q, R-X, DCR,`
	`Z-θr(or Z-thr)*1, Z-θd(Z-thd) *1`
	`Z-D,Z-Q`
	`(*1: θ is ASCII Hex 0xE9)`

Example	`SEND> FUNC Cp-D //Set measurement function to Cp-D`
Query Syntax	`FUNC?`
Query Response	`<function>`
Example	`SEND> FUNC?` `RET> Cp-D`

1.7.2 FUNCtion:IMPedance:AUTO

The FUNCtion:IMPedance:AUTO command sets the impedance's LCZ Automatic selection.

Command Syntax	`FUNC:IMPedance:AUTO {ON,OFF, 0,1}`

Example	SEND> FUNC:IMP:AUTO ON
Query Syntax	FUNC:IMPedance:AUTO?
Query Response	{on,off}
Example	SEND> FUNC:IMP:AUTO? RET> off

1.7.3 FUNCtion:IMPedance:RANGe

The FUNCtion:IMPedance:RANGe command sets the impedance's measurement range.

Command Syntax	FUNC:IMPedance:RANGe *<0-8,MIN,MAX>*
Parameter	Where, <0-8,MIN, MAX> is: 0-8, The range number MIN, =Range 0 MAX, =Range 8
Example	SEND> FUNC:IMP:RANG 2 //Set measurement range to [2] 10kΩ
Query Syntax	FUNC:IMPedance:RANGe?
Query Response	*<0-8>*
Example	SEND> FUNC:IMP:RANG? RET> 0

1.7.4 FUNCtion:DCR:RANGe

The FUNCtion:DCR:RANGe command sets the DCR's measurement range.

Command Syntax	FUNC:DCR:RANGe *<0-8,MIN,MAX>*
Parameter	Where, <0-8,MIN, MAX> is: 0-8, The range number MIN, =Range 0 MAX, =Range 8
Example	SEND> FUNC:DCR:RANG 2 //Set DCR range to [2] 10kΩ
Query Syntax	FUNC:DCR:RANGe?
Query Response	*<0-8>*
Example	SEND> FUNC:DCR:RANG? RET> 0

1.7.5 FUNCtion:RANGe:AUTO

The FUNCtion:RANGe:AUTO command sets the auto range to ON or OFF.

Command Syntax	FUNC:RANGe:AUTO *{off(hold),on(auto),NOMinal}*
Parameter	Where, *{off(hold),on(auto),NOMinal}* is: off(or hold): Sets the auto range to off. on(or auto): Sets the auto range to on. NOMinal: See Page 错误!未定义书签。 Section 错误!未找到引用源。 错误!未找到引用源。
Example	SEND> FUNC:RANG:AUTO AUTO //Sets to auto range. SEND> FUNC:RANG:AUTO off //Sets auto range to off.
Query Syntax	FUNC:RANGe:AUTO?
Query Response	*{HOLD,AUTO,NOM}*
Example	SEND> FUNC:RANG:AUTO? RET> auto

1.7.6 FUNCtion:MONitor1 /2

The FUNCtion:MONitor1 and FUNCtion:MONitor2 commands set the two monitor parameter.

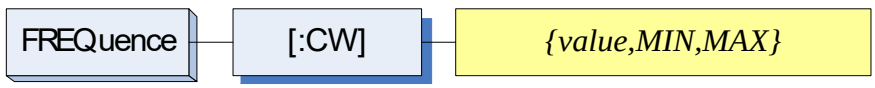
Command Syntax	FUNC:MONitor1 *{off, Z, D, Q, THR, THD, R, X, G, B, Y, ABS, PER VAC, IAC}* FUNC:MONitor2 *{off, Z, D, Q, THR, THD, R, X, G, B, Y, ABS, PER VAC, IAC}*
Parameter	Where, *{off, Z, D, Q, THR, THD, R, X, G, B, Y, ABS, PER VAC, IAC}*

Example	`SEND> FUNC:MON1 Z`
Query Syntax	`FUNC:MON1?` `FUNC:MON2?`
Query Response	*{off, Z, D, Q, THR, THD, R, X, G, B, Y, ABS, PER VAC, IAC}*
Example	`SEND> FUNC:MON1?` `RET> off`

1.8 FREQuency Subsystem

The FREQuency command sets the oscillator frequency. The FREQuency? Query returns the current test frequency setting.

Figure 0-3 FREQ Subsystem Command Tree

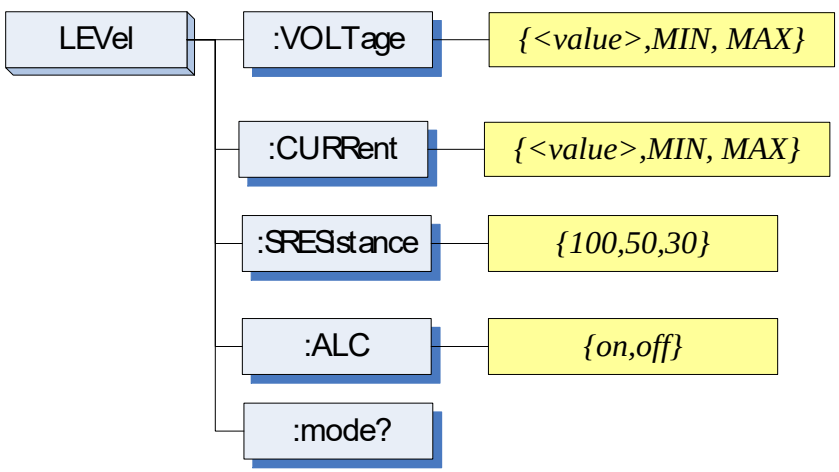


Command Syntax	`FREQ[:CW] {<value>,MIN,MAX}`
Parameter	`Where,` <value> is the numeric data (NR1 integer, NR2 fix float or NR3 floating point). MIN Sets to the minimum value MAXSets to the maximum value
Example	`SEND> FREQ 1K //Set to 1kHz,the Hz cannot be added.`
Query Syntax	`FREQ[:CW]?`
Query Response	`<NR3>` NR3 floating point
Example	`SEND> FREQ?` `RET> 1.000000E+03`
Note	A suffix multiplier (k) can be used with this command. But the suffix unit Hz can't be used. This command CANNOT be used in LIST SWEEP DISPLAY page and CORRECTION page.

1.9 LEVel Subsystem

The Level subsystem sets the oscillator's output voltage/current level and source output Impedance

Figure 0-4 LEVel Subsystem Command Tree



1.9.1 LEVel:VOLTage (=VOLTage[:LEVel])

The LEVel:VOLTage or VOLTage[:LEVel] command sets the oscillator's output voltage level.

Command Syntax	**LEVel:VOLTage {<value>,MIN,MAX}** **or VOLTage:LEVel {<value>,MIN,MAX}**
Parameter	**Where,** <value> is the numeric data (NR1, NR2 or NR3). MIN Sets to the minimum value of voltage MAXSets to the maximum value
Example	**SEND> LEV:VOLT 0.3 //Set to 0.3V,the unit V can be ignored.**
Query Syntax	**LEVel:VOLTage?** **or CURRent:LEVel?**
Query Response	**<NR3>** NR3 floating point
Example	**SEND> VOLT?** **RET> 1.000000e+00**
Note	The suffix unit V can't be ignored This command CANNOT be used in LIST MEAS page and CORRECTION page.

1.9.2 LEVel:CURRent (=CURRent[:LEVel])

The LEVel:CURRent or CURRent[:LEVel] command sets the oscillator's output current level.

Command Syntax	**LEVel:CURRent {<value>,MIN,MAX}** **or CURRent:LEVel {<value>,MIN,MAX}**
Parameter	**Where,** <value> is the numeric data (NR1, NR2 or NR3). MIN Sets to the minimum value of current MAXSets to the maximum value of current
Example	**SEND> LEV:CURR 1m //Set to 1mA, unit V can be ignored.**
Query Syntax	**LEVel:CURRent?** **or CURRent:LEVel?**
Query Response	**<NR3>** NR3 floating point
Example	**SEND> VOLT?** **RET> 1.000000e+00**
Note	The suffix unit V can't be ignored This command CANNOT be used in LIST MEAS page and CORRECTION page.

1.9.3 LEVel:SRESistance (= VOLTage:SRESistance)

The LEVel:SRESistance or VOLTage:SRESistance command sets the source output Impedance.

Command Syntax	**LEVel:SRESistance {30,50,100}** **VOLTage:SRESistance {30,50,100}**
Parameter	**{30,50,100}** **Where,** 30 Sets the output impedance to 30Ω 50 Sets the output impedance to 50Ω 100 Sets the output impedance to 100Ω
Example	**SEND> LEV:SRES 30** //Set to 30Ω, the unit Ω cannot be added.
Query Syntax	**VOLTage:SRES?** **or LEVel:SRES?**
Query Response	**<NR1>** NR1 integer
Example	**SEND> LEV:SRES?** **RET> 30**

Note	The suffix unit Ω can't be used with this command. This command CANNOT be used in LIST SWEEP DISPLAY page and CORRECTION page.

1.9.4 LEVel:ALC (=AMPlitude:ALC)

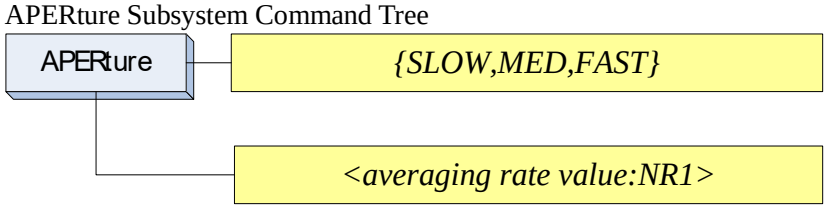
The LEVel:ALC or AMPlitude:ALC command enables the Automatic Level Control (ALC).

Command Syntax	**LEVel:ALC {on,1,off,0}** **AMPlitude:ALC {on,1,off,0}**
Parameter	**{on,1,off,0}** **Where,** on (1) Enable ALC off(0) Turn off the ALC.
Example	**SEND> LEV:ALC on**
Query Syntax	**LEV:ALC?** **or AMP:ALC?**
Query Response	**{on,off}**
Example	**SEND> LEV:ALC?** **RET> off**
Note	This command CANNOT be used in LIST MEAS page , CORRECTION page and DCR mode.

1.10 APERture Subsystem

The **APERture** subsystem command sets the integration time of the ADC and the averaging rate.

Figure 0-5 APERture Subsystem Command Tree



Command Syntax	**APERture {SLOW,MED,FAST}** **APERture <value>** **SPEED(spd) {SLOW,MED,FAST}** **SPEED(spd) <value>**
Parameter	Where, SLOW Set test speed to slow MED Set test speed to medium FAST Set test speed to fast <value> NR1(0 to 256): Averaging rate (0=OFF=1)
Example	**SEND> APER FAST** **SEND> APER 10**
Query Syntax	**APER?**
Query Response	**{SLOW,MED,FAST},<avg value>**
Example	**SEND> APER?** **RET> slow,0**

1.10.1 APERture:RATE?

The **APERture:RATE?** query returns the current integration time.

Query Syntax	**APER:RATE?**
Query Response	**SLOW**

Example	SEND> APER:RATE? RET> slow

1.10.2 APERture:AVG?

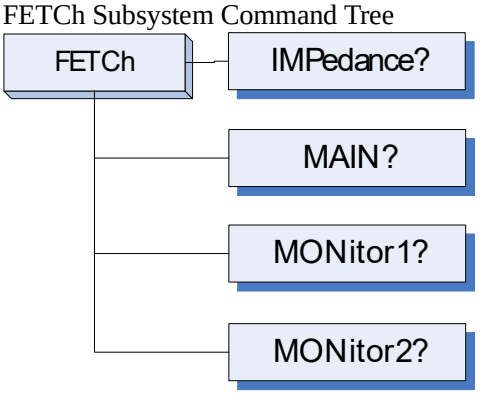
The **APERture:AVG?** query returns the averaging rate settings.

Query Syntax	APER:AVG?
Query Response	<NR1> Integer (0 to 256)
Example	SEND> APER:AVG? RET> 0

1.11 FETCh Subsystem

The FETCh subsystem command group is a sensor-only command which retrieves the measurement data taken by measurement(s) initiated by a trigger, and places the data into the output buffer.

Figure 0-6 FETCh Subsystem Command Tree



1.11.1 FETCh?

The FETCh? query sets the latest measurement data of the primary , secondary parameters and comparator result into the output buffer.

Query Syntax	FETCh?
Query Response	<NR3:primary value>,<NR3:secondary value>,<comparator result>
Example	SEND> FETC? RET> +2.617886e-11,+5.454426e-01,BIN1,AUX-OK,OK RET> +1.23434e+05,OUT ,NG //DCR & Comp on

1.11.2 FETCh:IMPedance?

The FETCh:IMPedance? query sets the latest measurement data of the primary , secondary parameters monitor1 and monitor2 result into the output buffer.

Query Syntax	FETCh?
Query Response	<NR3:primary value>,<NR3:secondary value>,<comparator result>
Example	SEND> FETC? RET> +2.617886e-11,+5.454426e-01,BIN1,AUX-OK,OK RET> +1.23434e+05,BIN1,OK //DCR & Comp on

1.11.3 FETCh:MAIN?

The FETCh:MAIN? query sets the latest measurement data of the primary and secondary parameters

Query Syntax	**FETCh:MAIN?**
Query Response	<NR3:primary value>,<NR3:secondary value>
Example	**SEND> FETC:MAIN?** **RET> +2.021009e-11,+1.644222e-01//LCR** **Primary ,Secondary** **RET> +1.23434e+05//DCR**

1.11.4 FETCh:MONitor1? /2?

The FETCh:MONitor1? and FETCh:MONitor2 set the latest measurement data of the moniter1 and moniter2 parameters into the output buffer.

Query Syntax	**FETCh:MONitor1? and FETCh:MONitor2?**
Query Response	<NR3: moniter1/2 value>
Example	**SEND> FETC:MON1?** **RET> +3.886517e+05** **RET> +0.000000e+00 //0: The monitor 1 is OFF**

1.11.5 FETCh:MONitor?

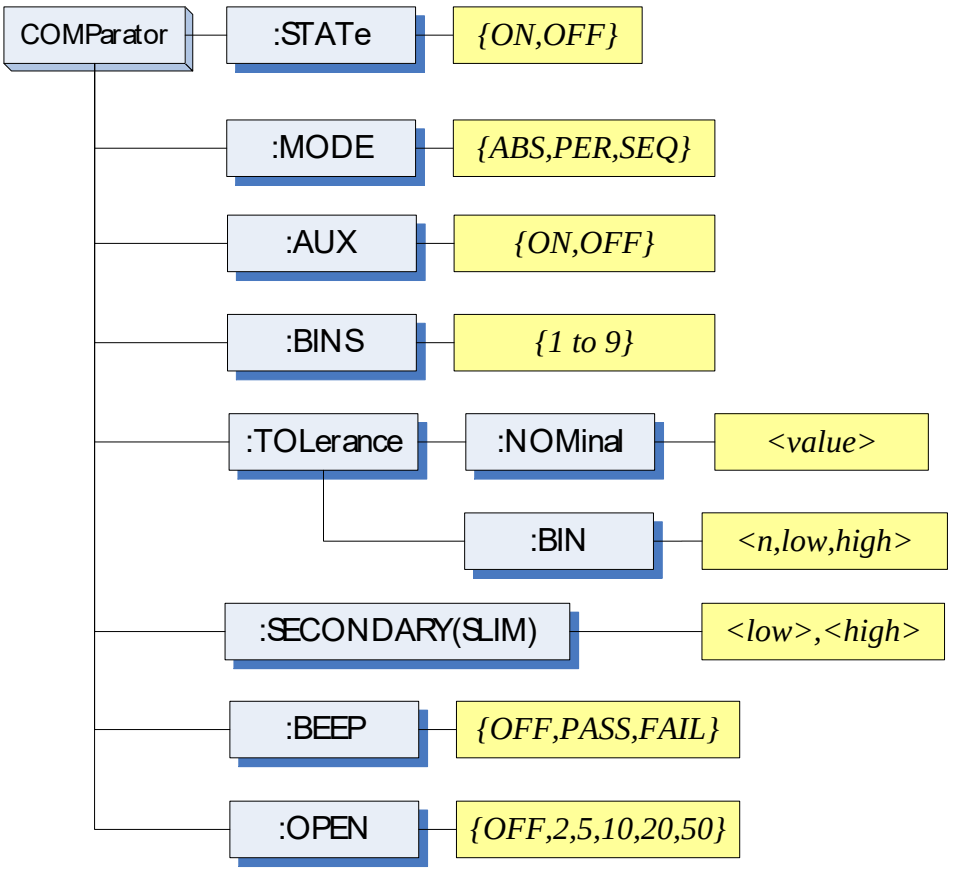
The FETCh:MONitor? set the latest measurement data of the moniter1 and moniter2 parameters into the output buffer.

Query Syntax	**FETCh:MONitor?**
Query Response	<NR3: moniter1/2 value>
Example	**SEND> FETC:MON?** **RET> +3.886517e+05,+0.000000e+00** **(0: The monitor 2 is OFF)**

1.12 COMParator Subsystem

The COMParator subsystem command group sets the comparator function, including its ON/OFF setting, limit mode, and limit values.

Figure 0-7 COMParator Subsystem Command Tree



1.12.1 COMParator:STATe

The COMParator:STATe command sets the comparator function to ON or OFF.

Command Syntax	`COMParator:STATe {ON,OFF,1,0}`
Parameter	`Where,` ON or 1 Sets the comparator to ON OFF or 0 Sets the comparator to OFF
Example	`SEND> COMP:STAT OFF`
Query Syntax	`COMParator:STATe?`
Query Response	`{on,off}`
Example	`SEND> COMP:STAT?` `RET> on`

1.12.2 COMParator:MODE

The :COMParator:MODE command sets the limit mode of the comparator function.

Command Syntax	`COMParator:MODE {ABS,PER,SEQ}`
Parameter	`Where,{ABS,PER,SEQ} is:` ABS Absolute tolerance mode PER Percent tolerance mode SEQ Sequential mode
Example	`SEND> COMP:MODE PER`
Query Syntax	`COMParator:MODE?`
Query Response	`{abs,per,seq}`
Example	`SEND> COMP:MODE?` `RET> abs`

1.12.3 COMParator:AUX

The COMParator:AUX command sets the auxiliary BIN counting function of the comparator to ON or OFF.

Command Syntax	**COMParator:AUX {ON,OFF,1,0}**
Parameter	**Where,{ON,OFF,1,0} is:** ON or 1 Set the AUX BIN to ON OFF or 0 Set the AUX BIN to OFF
Example	**SEND> COMP:AUX OFF**
Query Syntax	**COMParator:AUX?**
Query Response	**{on,off}**
Example	**SEND> COMP:AUX?** **RET> on**

1.12.4 COMParator:BINS

The COMParator:BINS command sets the total number of bins.

Command Syntax	**COMParator:BINS <*value*>**
Parameter	**Where,{value} is:** NR1 (1 to 9)
Example	**SEND> COMP:BINS 3**
Query Syntax	**COMParator:BINS?**
Query Response	**<NR1> (1 to 9)**
Example	**SEND> COMP:BINS?** **RET> 3**

1.12.5 COMParator:TOLerance:NOMinal

The COMParator:TOLerance:NOMinal command sets the nominal value for the tolerance mode of the comparator function.

Command Syntax	**COMParator:TOLerance:NOMinal <*value*>**
Parameter	**Where,<value> is:** NR1, NR2 or NR3 A suffix multiplier can be used with this command. But the suffix unit F/Ω/H can't be used.
Example	**SEND> COMP:TOL:NOM 100N** **SEND> COMP:TOL:NOM 1E-6**
Query Syntax	**COMParator:TOLerance:NOMinal?**
Query Response	**<NR3>**
Example	**SEND> COMP:TOL:NOM?** **RET> 1.000000e-06**

1.12.6 COMParator:TOLerance:BIN

The COMParator:TOLerance:BIN command sets the low/high limit values of each BIN for the comparator function tolerance mode.

Command Syntax	**COMParator:TOLerance:BIN <*n*>,<*low limit*>,<*high limit*>**
Parameter	**Where,<*n*>,<*low limit*>,<*high limit*> is:** n NR1 (1 to 9): Bin number low limit NR1,NR2 or NR3: low limit value high limit NR1,NR2 or NR3: high limit value
Example	**SEND> COMP:TOL:BIN 1,100P,200P** **SEND> COMP:TOL:BIN 2,200E-6,300E-6**

Query Syntax	**COMParator:TOLerance:BIN? <n>**
Parameter	**Where,<*n*> is:** NR1 (1 to 9): Bin number
Query Response	**<NR3:low limit>,<NR3:high limit>**
Example	**SEND> COMP:TOL:BIN? 2** **RET> 1.000000e-06,2.000000E-6**

1.12.7 COMParator:SLIM

The COMParator:SLIM or COMParator:secondary command sets the LOW/HIGH limit values for the secondary parameter.

Command Syntax	**COMParator:SLIM <low *value*>,<*high value*>** **COMParator:secondary <low *value*>,<*high value*>**
Parameter	**Where, <low *value*>,<*high value*> is:** <low value> NR1,NR2 or NR3: low limit value <high value> NR1,NR2 or NR3: high limit value A suffix multiplier can be used with this command.
Example	**SEND> COMP:SLIM 0.0001,0.0010**
Query Syntax	**COMParator:SLIM?** **COMParator:secondary?**
Query Response	**<NR3:low limit>,<NR3:high limit>**
Example	**SEND> COMP:SLIM?** **RET> 1.000000e-04,1.000000e-03**

1.12.8 COMParator:BEEP

The :COMParator:BEEP command sets beep mode of the comparator function.

Command Syntax	**COMParator:BEEP {OFF,PASS,FAIL}**
Parameter	**Where,** OFF Turns the beeper off. PASS Sounds a beep when the test is passed (BIN1~BIN9). FAIL Sounds a beep when the test is failed (OUT).
Example	**SEND> COMP:BEEP PASS**
Query Syntax	**COMParator:BEEP?**
Query Response	**{OFF,PASS,FAIL}**
Example	**SEND> COMP:BEEP?** **RET> OFF**

1.12.9 COMParator:OPEN

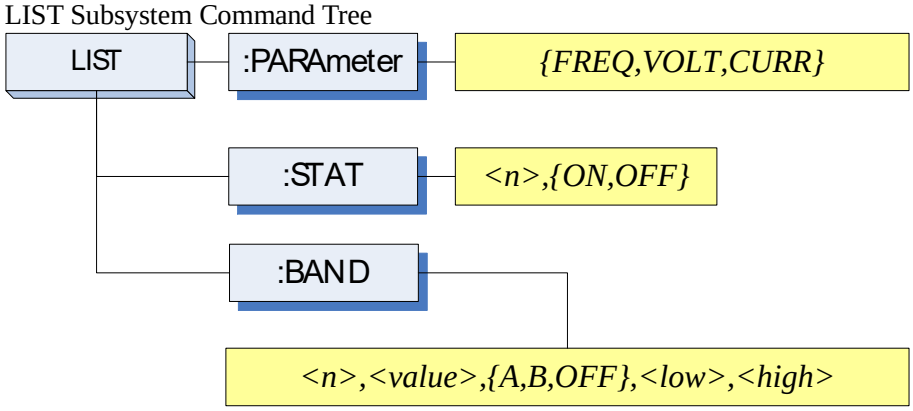
The :COMParator:OPEN command selects the open condition for main parameter.

Command Syntax	**COMParator:OPEN {OFF,2,5,10,20,50}**
Parameter	**Where,** OFF Turns the beeper off. 2,5,10,20,50 The percent range value
Example	**SEND> COMP:OPEN 2**
Query Syntax	**COMParator:OPEN?**
Query Response	**{OFF,2,5,10,20,50}**
Example	**SEND> COMP:OPEN?** **RET> OFF**

1.13 LIST Subsystem

The LIST or SWEEP Subsystem command group sets the List Sweep measurement function, including the sweep point setting and limit values for the limit function.

Figure 0-8 LIST Subsystem Command Tree



1.13.1 LIST:PARAmeter

The LIST:PARAmeter command sets the list sweep parameter.

Command Syntax	`LIST:PARAmeter {FREQ,VOLT,CURR}`
Parameter	`Where, {FREQ,LEVEL} is:` FREQ Sets the sweep parameter to frequency VOLT Sets the sweep parameter to voltage level CURR Sets the sweep parameter to current level
Example	`SEND> LIST:PARA VOLT`
Query Syntax	`LIST:PARAmeter?`
Query Response	`{FREQ,VOLT,CURR}`
Example	`SEND> LIST:PARA?` `RET> FREQ`

1.13.2 LIST:STAT

The LIST:STAT command turns on/off the specified sweep point.

Command Syntax	`LIST:STAT <n>,{ON,OFF,1,0}`
Parameter	`Where,<n> is:` n NR1(1 to 10): List sweep point ON or 1 Set this point to ON OFF or 0 Set this point to OFF
Example	`SEND> LIST:STAT 1,ON`
Query Syntax	`LIST:STAT? <n>`
Parameter	`Where,<n> is:` n NR1(1 to 10): List sweep point
Query Response	`{on,off}`
Example	`SEND> LIST:STAT? 1` `RET> on`

1.13.3 LIST:BAND

The LIST:BAND command sets the List Sweep point value, limit mode and low/high limit values.

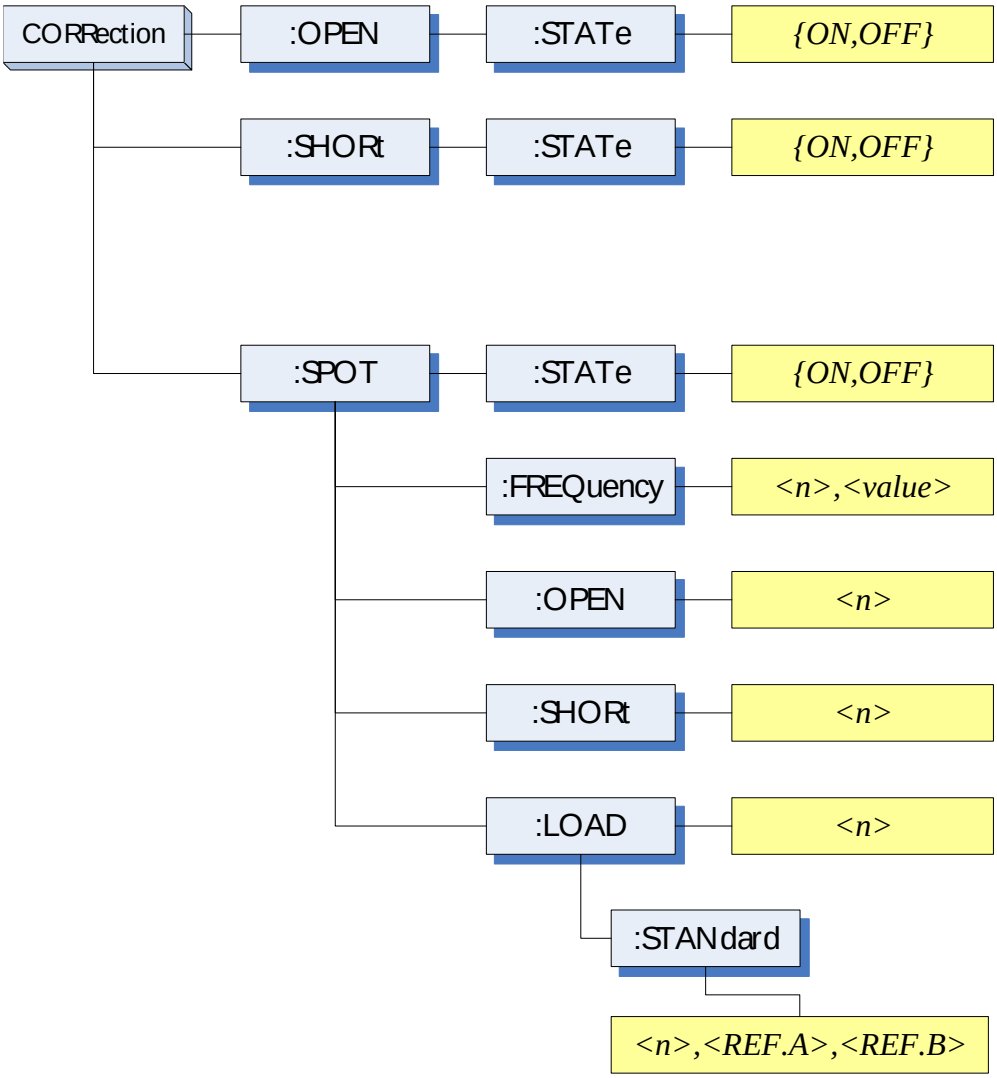
Command Syntax	`LIST:BAND <n>,<point value>,{A,B,OFF},<low>,<high>`
Parameter	`Where, <n>,<point value>,{A,B,OFF},<low>,<high> is:` n NR1(1 to 10): List sweep point <point value> sweep point value (frequency value or signal level voltage value) A, Uses the primary parameter as the limit parameter. B, Uses the secondary parameter as the limit parameter. OFF Turn off the List Sweep's comparator function <low> NR1,NR2 or NR3: low limit value <high> NR1,NR2 or NR3: high limit value Note: The suffix multipliers can be used with this command. But the suffix units CANNOT be added.
Example	`SEND> LIST:BAND 1,1k,A,1n,2n` `SEND> LIST:BAND 2,10k,A,1E-9,2E-9`
Query Syntax	`LIST:BAND? <n>`
Parameter	`Where,<n> is:` n NR1(1 to 10): List sweep point
Query Response	`{on,off},<point value>,{A,B,-},<NR3:low>,<NR4:high>`
Example	`SEND> LIST:BAND? 1` `RET> on,1.00000e+03,A,1.000000E-9,2.000000E-9`

1.14 CORRection Subsystem

The CORRection subsystem command group sets the correction function, including the OPEN, SHORT and LOAD correction settings.

NOTE: The CORRection subsystem CANNOT work in <LIST MEAS> page.

Figure 0-9 CORRection Subsystem Command Tree



1.14.1 CORRection:OPEN

The CORRection:OPEN command execute all presetted OPEN correction data measurement points.

Command Syntax	`CORRection:OPEN`
Example	`SEND> CORRection:OPEN`

1.14.1.1 CORRection:OPEN:STATe

The CORRection:OPEN:STATe command sets the OPEN correction function to ON or OFF.

Command Syntax	`CORRection:OPEN:STATe {ON,OFF,1,0}`
Parameter	`Where, {ON,OFF,1,0} is:` ON, 1 When the function is ON OFF,0 When the function is OFF
Example	`SEND> CORR:OPEN:STATe ON` `RET> open`
Query Syntax	`CORRection:OPEN:STATe?`
Query Response	`{on,off}`
Example	`SEND> CORR:OPEN:STATe?` `RET> on`

1.14.2 CORRection:SHORt

The CORRection:SHORt command execute all presetted SHORT correction data measurement points.

Command Syntax	`CORRection:SHORt`
Example	`SEND>  CORRection:SHOR` `RET>  short`

1.14.2.1 CORRection:SHORt:STATe

The CORRection:SHORt:STATe command sets the SHORT correction function to ON or OFF.

Command Syntax	`CORRection:SHORt:STATe {ON,OFF,1,0}`
Parameter	`Where, {ON,OFF,1,0} is:` ON, 1 When the function is ON OFF,0 When the function is OFF
Example	`SEND> CORR:SHOR:STATe ON`
Query Syntax	`CORRection:SHOR:STATe?`
Query Response	`{on,off}`
Example	`SEND>  CORR:SHOR:STATe?` `RET>  on`

1.14.3 CORRection:SPOT:FREQuency

The CORRection:SPOT:FREQuency command sets the frequency point for the specified frequency point correction.

Command Syntax	`CORRection:SPOT:FREQuendy <value>`

Parameter	`Where, <value> is:`
	`value  NR1,NR2 or NR3:Frequecny value.`
	A suffix multiplier can be used with this command. But the unit "Hz" cannot be added.

Example	`SEND> CORR:SPOT:FREQ 1k` `SEND> CORR:SPOT:FREQ 10k`
Query Syntax	`CORRection:SPOT:FREQuency?`
Query Response	`<NR3>`
Example	`SEND>  CORR:SPOT:FREQ?` `RET>  1.000000e+03`

1.14.4 CORRection:SPOT:OPEN

This command executes the OPEN correction data measure for the specified frequency correction.

Command Syntax	`CORRection:SPOT:OPEN`
Example	`SEND> CORR:SPOT:OPEN`

1.14.5 CORRection:SPOT:SHORt

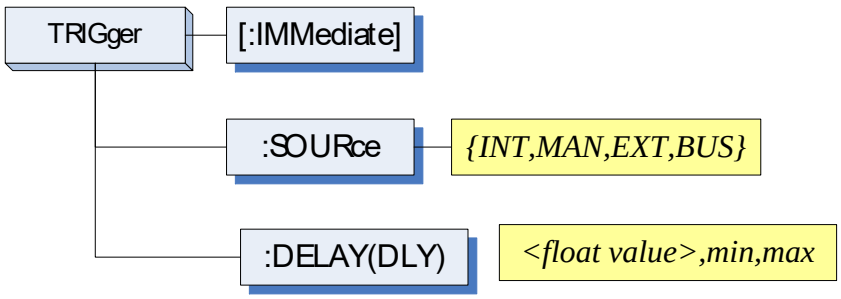
This command executes the SHORT correction data measure for the specified frequency correction.

Command Syntax	`CORRection:SPOT:SHORt`
Example	`SEND> CORR:SPOT:SHOR`

1.15 TRIGger Subsystem

The TRIGger subsystem command group is used to enable a measurement or a sweep measurement, and to set the trigger mode.

Figure 0-10 TRIGger Subsystem Command Tree



1.15.1 TRIGger[:IMMediate]

The TRIGger:IMMediate command causes the trigger to execute a measurement or a sweep measurement, regardless of the trigger state.

Command Syntax	`TRIGger[:IMMediate]`
Example	`SEND> TRIG`
Note	`This command can be ONLY used in BUS trigger mode.`

1.15.2 TRIGger:SOURce

The TRIGger:SOURce command sets the trigger mode.

Command Syntax	`TRIGger:SOURce {INT,MAN,EXT,BUS}`
Parameter	Where, {INT,MAN,EXT,BUS} is INT Internal Trigger Mode MAN Manual Trigger Mode EXT External Trigger Mode BUS BUS Trigger Mode
Example	`SEND> TRIG:SOUR BUS`
Query Syntax	`TRIGger:SOURce?`
Query Response	`{INT,MAN,EXT,BUS}`
Example	`SEND> TRIG:SOUR?` `RET> INT`

1.15.3 TRIGger:DELAY

The TRIGger:DELAY command sets the trigger delay time.

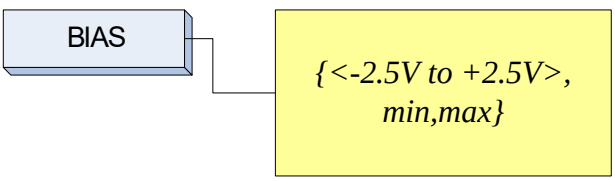
Command Syntax	`TRIGger:DELAY {<float>,min,max}` `TRIGger:DLY {<float>,min,max}`
Parameter	Where, is float value: from 1ms to 60.00s min: =0ms max: =60.000s
Example	`SEND> TRIG:DLY 1` //1.000s
Query Syntax	`TRIGger:DELAY?` `TRIGger:DLY?`

Query Response	{0.000s~60.00s}
Example	SEND> TRIG:DLY? RET> 1.000s

1.16 BIAS Subsystem

The BIAS subsystem command group sets the DC BIAS switch to ON or OFF, and sets the DC bias voltage value.

Figure 0-11 BIAS Subsystem Command Tree

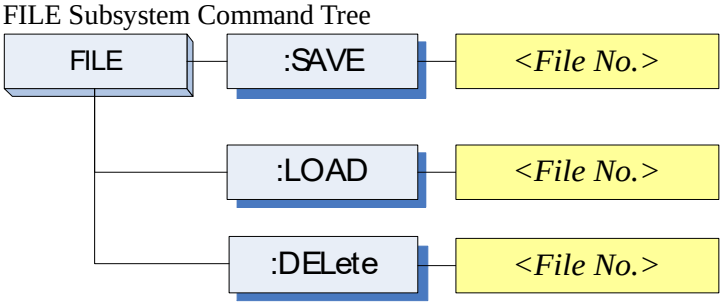


Command Syntax	BIAS {OFF,<-2.5 to +2.5V,min,max}
Example	SEND> BIAS OFF SEND> BIAS 2
Query Syntax	BIAS?
Query Response	<-2.50V~+2.50V>
Example	SEND> BIAS? RET> OFF

1.17 FILE Subsystem

The FILE subsystem command group executes the file operation.

Figure 0-12 FILE Subsystem Command Tree



1.17.1 FILE?

The FILE? query returns the file number used by system.

Query Syntax	FILE?
Query Response	<NR1(0 TO 9):File number>
Example	SEND> FILE? RET> 0

1.17.2 FILE:SAVE

The FILE:SAVE command saves all user settings into current used file.

Command Syntax	FILE:SAVE
Example	SEND> FILE:SAVE

The FILE:SAVE <n> command saves all user settings into specified file.

Command Syntax	**FILE:SAVE <*File No.*>**
Parameter	**Where, <File No.> is:** NR1 (0 to 9)
Example	**SEND> FILE:SAVE 0**

1.17.3 FILE:LOAD

The FILE:LOAD command recalls all user settings from current used file.

Command Syntax	**FILE:LOAD**
Example	**SEND> FILE:LOAD**

The FILE:LOAD <n> command recalls all user settings from specified file.

Command Syntax	**FILE:LOAD <*File No.*>**
Parameter	**Where, <File No.> is:** NR1 (0 to 9)
Example	**SEND> FILE:LOAD 0**

1.17.4 FILE:DELete

Command Syntax	**FILE:DELete <*File No.*>**
Parameter	**Where, <File No.> is:** NR1 (0 to 9)
Example	**SEND> FILE:DELete**

1.18 ERRor Subsystem

1.18.1 ERRor?

The ERRor? retrieves last error information.

Query Syntax	**ERRor?**
Query Response	**Error string**
Example	**SEND> ERR?** **RET> no error.**

1.19 SYSTEM Subsystem

1.19.1 SYSTem:SHAKehand

The SYSTem:SHAKehand command feeds back the sent commands.

Command Syntax	**SYSTem:SHAKehand {on,off}**
Example	**SEND> SYST:SHAK ON**
Query Syntax	**SYSTem:SHAKehand?**
Query Response	**{on,off}**
Example	**SEND> SYST:SHAK?** **RET> OFF**

1.19.2 SYSTem:CODE

The SYSTem:CODE command feeds back error code for each sent command.

Command Syntax	**SYSTem:CODE {on,off}**

Example	`SEND> SYST:CODE ON`
Query Syntax	`SYSTem:CODE?`
Query Response	`{on,off}`
Example	`SEND> SYST:CODE?` `RET> OFF`

1.19.3 SYSTem:KEYLock

SYSTem:KEYLock command unlocks the keypad.

Command Syntax	`SYST:KEYLOCK OFF` `or UNLOCK(UNLK)`
Example	`SEND> UNLOCK`

1.19.4 SYSTem:RESult

SYSTem:RESult command selects the test results send mode.

Command Syntax	`SYSTem:RESult {fetch,auto}`
Example	`SEND> SYST:RES fetch`
Parameter	`Where,` `fetch` The test results will be sent back by command "fetch?" `auto` The results will be sent back by one trig.
Query Syntax	`SYSTem:RESult?`
Query Response	`{FETCH,AUTO}`
Example	`SEND> SYST:RES?` `RET> fetch`

1.20 Common Commands

1.20.1 *IDN?

The *IDN? query returns the instrument ID.

Query Syntax	`IDN? Or *IDN?`
Query Response	`<manufacturer>,<model>,<serial no.>,<fireware>`

1.20.2 *TRG

The *TRG command (trigger command) performs the same function as the Group Execute Trigger command.

Command Syntax	`*TRG`
Query Response	`<primary value>,<secondary value>,<comparator result>`
Example	`SEND> *TRG` `RET> +5.566785e-11,+7.253470e-01,OUT`
Note	This command can be used in BUS trigger mode. *TRG = TRIG;:FETC?

1.20.3 *SAV

*SAV = FILE:SAVE

The *SAV command saves all user settings into current used file.

Command Syntax	`*SAV`
Example	`SEND> *SAV`

1.20.4 *RCL

*RCL = FILE:LOAD

The *RCL command recalls all user settings from current used file.

Command Syntax	**`*RCL`**
Example	**`SEND> *RCL`**

2.Modbus (RTU) Protocol

This chapter include the following information:

- Data format – About the Modbus communication format.
- Function
- Variable Area
- Function Code

2.1 Data Format

We follow the Modbus (RTU) communication protocol, the instrument will respond to commands of the host computer and return the standard response frame.

Reference:

You can contact our sales department to get the communication test tool, which has Modbus communication debugging method. It contains CRC-16 calculator and floating point numbers into Modbus floating point format.

2.1.1 Command Frame

Figure 2-1 Modbus Command Frame

Table 2-1 Command Frame Description

	At least 3.5 character time squelch interval is requested
Station address	1 byte Modbus can support 00~0x63 stations Designated as 00 when unified broadcast In instruments that do not have an optional RS485, the default station address is 0x01
Function code	1 byte 0x03: Read multiple registers 0x04: =03H, not used 0x06: Write to a single register, can use 10H instead 0x08: Echo test (used only for debugging) 0x10: Write to multiple registers
Data	Specify register address, quantity, and content
CRC-16	2 bytes, low in front Cyclic Redundancy Check Calculate all data from the station address to the end of the data to

	get the CRC16 check code
	At least 3.5 character time squelch interval is requested

2.1.2 CRC-16 Calculation Method

1. Set the initial value of the CRC-16 register to 0xFFFF.
2. Perform XOR calculation fo CRC-16 register and information of the first byte, then it return the result to the CRC register.
3. Fill in the MSB with 0 and use the CRC register to shift to the right by one bit.
4. If the bit moved from the LSB is "0", repeat step (3) (process the next shift). If the bit moved from the LSB is "1", will perform XOR calculation for the CRC register and 0xA001, then the result is returned to the CRC register.
5. Repeat steps (3) and (4) until you move 8 bits.
6. If the information processing has not been completed, perform XOR calculation for CRC register and information of the next Byte, and return to the CRC register, repeating from step (3).
7. Append the result of the calculation (the value of the CRC register) from the low byte to the message.

The following is a CRC calculation function for a VB language.

```
Function CRC16(data() As Byte) As Byte()
    Dim CRC16Lo As Byte, CRC16Hi As Byte   'CRC register
    Dim CL As Byte, CH As Byte       ' Polynomial code &HA001
    Dim SaveHi As Byte, SaveLo As Byte
    Dim i As Integer
    Dim flag As Integer
    CRC16Lo = &HFF
    CRC16Hi = &HFF
    CL = &H1
    CH = &HA0
    For i = 0 To UBound(data)
        CRC16Lo = CRC16Lo Xor data(i) 'Each data is XORed with the CRC register
        For flag = 0 To 7
            SaveHi = CRC16Hi
            SaveLo = CRC16Lo
            CRC16Hi = CRC16Hi \ 2      ' High bit right shift one bit
            CRC16Lo = CRC16Lo \ 2      ' Low bit right shift one bit
            If ((SaveHi And &H1) = &H1) Then ' If the last bit of the high byte
is 1
                CRC16Lo = CRC16Lo Or &H80   'Then the low byte is shifted to the
right and then supplement 1 at front
            End If             ' Otherwise automatically fill 0
            If ((SaveLo And &H1) = &H1) Then ' If the LSB is 1, then XOR with
the polynomial code
                CRC16Hi = CRC16Hi Xor CH
                CRC16Lo = CRC16Lo Xor CL
            End If
        Next flag
    Next i
    Dim ReturnData(1) As Byte
    ReturnData(0) = CRC16Hi       'CRC high bit
    ReturnData(1) = CRC16Lo       'CRC low bit
    CRC16 = ReturnData
    End Function
```

Reference: My company's "Communication Test Tool", which has Modbus communication debugging method. It contains the CRC-16 calculator.

Calculate the CRC-16 data to be appended to the end of the command frame, for example: 1234H:

Figure 2-2 Modbus append CRC-16 value

2.1.3 Response Frame

Unless it is a command broadcast by 00H station address, the other station address will return a response frame by the AT381x.

Fiture 2-3 Normal response frame

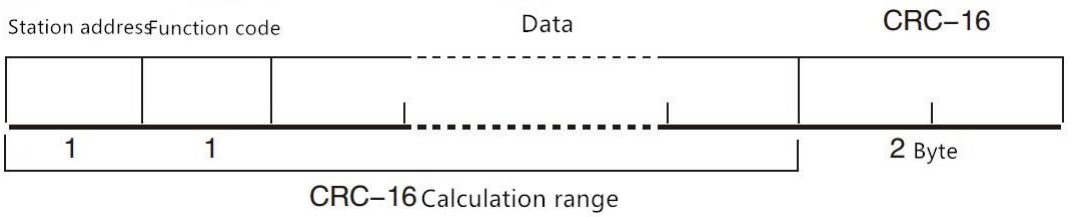


Figure 2-4 Abnormal response frame

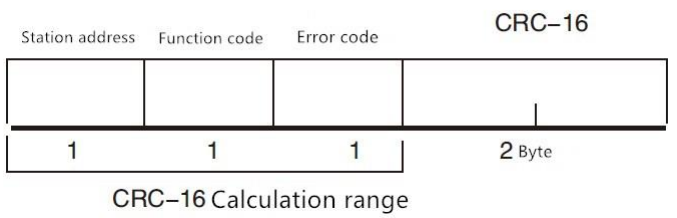


Table 2-2 Description for abnormal response frame

Station address	1 byte Returned from station address as it is
Function code	1 byte The function code of the command frame logical OR (OR) on BIT7 (0x80), for example: 0x03 OR 0x80 = 0x83
Error code	Exception code: 0x01 function code error (function code is not supported) 0x02 register error (register does not exist) 0x03 data error 0x04 execution error
CRC-16	2 bytes, low in front Cyclic Redundancy Check Calculate all data from the station address to the end of the data to get the CRC16 check code

2.1.4 No Response

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

1. Station address error
2. Transmission error
3. CRC-16 error
4. The number of digits is incorrect. For example, the function code 0x03 must have a total

digit of 8, and the received digits are less than 8 or greater than 8 bytes.

5. When the station address is 0x00, it represents the broadcast address, the AT381x does not respond.

2.1.5 Error Code

Table 2-3 Description for Error Code

Error code	Name	Description	priority
0x01	Function code error	Function code does not exist	1
0x02	Register error	Register code does not exist	2
0x03	Data error	The number of registers or the number of bytes is incorrect	3
0x04	Execution error	The data is illegal, and the written data is not allowed.	4

2.2 Function Code

The instrument only supports the following function codes, other function codes will respond to the error frame.

Table 2-4 Function code

Function code	Name	Description
0x03	Read multiple registers	Read multiple consecutive register data
0x04	Same as 0x03	Please use 0x03 instead
0x08	Echo test	The received data is returned as it was.
0x10	Write to multiple registers	Write to multiple consecutive registers

2.3 Register

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

Data：

The instrument supports the following values:

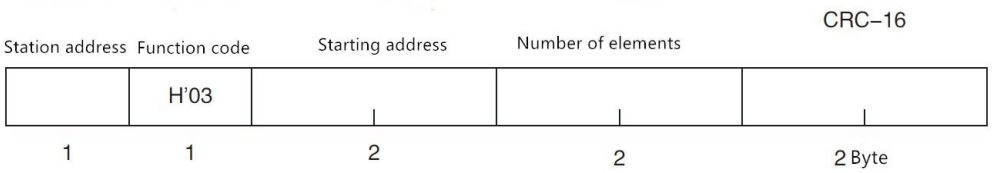
1. 1 register, double-byte (16-bit) integer, for example: 0x64 → 00 64
2. 2 registers, four-byte (32-bit) integer, for example: 0x12345678 → 12 34 56 78
3. 2 registers, 4 bytes (32 bits) single precision floating point number, 3.14 → 40 48 F5 C3

Reference: 

My company's "Communication Test Tool", which has Modbus communication debugging method. It contains floating-point converter..

2.4 Read Multiple Registers

Figure 2-5 Read Multiple Registers（0x03）

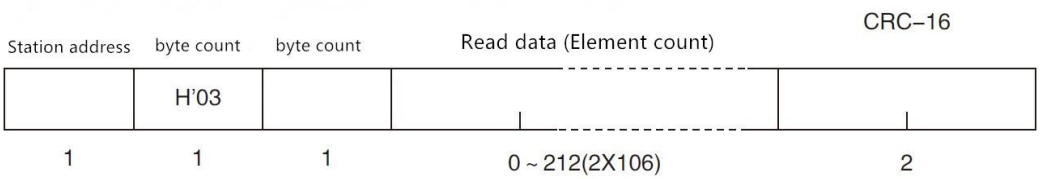


The function code for reading multiple registers is 0x03.

Table 2-5 Read Multiple Registers

Name	Name	Description
	Station address	When there is no RS485 address specified, the default is 01.
0x03	Function code	
	Starting address	Register start address, please refer to the Modbus command set
	Number of read registers 0001~006A（106）	The number of registers read continuously. Please refer to the Modbus command set to ensure that these register addresses arevalid, otherwise an error frame will be returned.
CRC-16	Check code	

Figure 2-6 Read Multiple Registers（0x03）Response Frame



Name	Name	Description
	Station address	Returned as it was.
0x03 Or 0x83	Function code	No abnormality: 0x03 Error code: 0x83
	Number of bytes	= number of registers x 2 For example: 1 register returns 02
	Data	Data Read
CRC-16	Check code	

2.5 Writing to Multiple Registers

Figure 2-7 Writing to multiple registers（0x10）

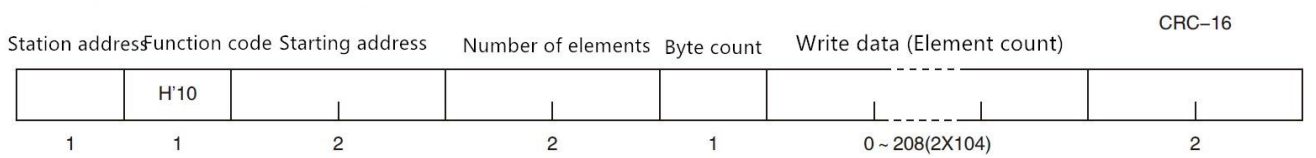
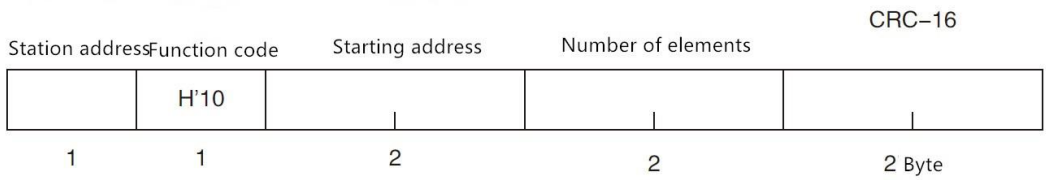


Table 2-6 Writing to multiple registers

Name	Name	Description
	Station address	When there is no RS485 address specified, the default is 01.
0x10	Function code	
	Starting address	Register start address, please refer to the Modbus

		command set
	Number of write registers 0001~0068（104）	The number of registers read continuously. Please refer to the Modbus command set to ensure that these register addresses arevalid, otherwise an error frame will be returned.
	Number of bytes	= number of registers x 2
CRC-16	Data	

Figure 2-8 Write to Multiple Registers（0x03）Response Frame



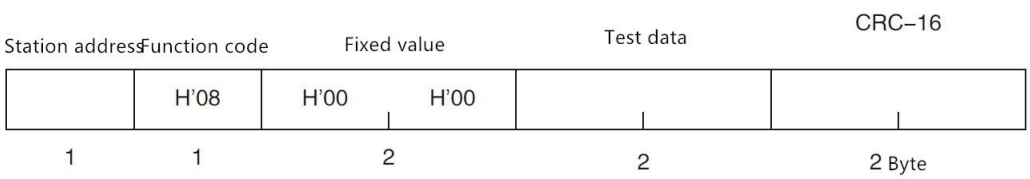
Name	Name	Description
	Station address	Returned as it was.
0x10 Or 0x90	Function code	No abnormality: 0x10 Error code: 0x90
	Starting address	
	Number of write registers 0001~0068（104）	
	CRC-16 Check code	

2.6 Echo Test

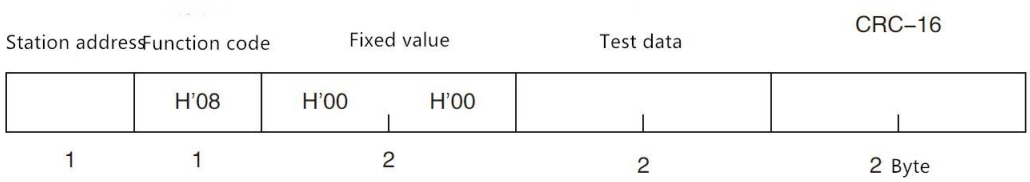
echo test function code 0x08, used for debug Modbus。

Figure 2-9 echo test（0x08）

Instruction frame



Response frame



Name	Name	Description
	Station address	Returned as it was.
0x08	Function code	
	Fixed value	00 00

	Test data	Any value: for example 12 34
	CRC-16 Check code	

Example:

Assume that the test data is 0x1234:

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

3.Modbus（RTU）Command

This chapter contains the following information:

- Register overview
- Detailed operation

Reference: Be sure to contact our sales department to obtain the communication test tool, which has Modbus communication debugging method. It contains CRC-16 calculator and floating point numbers converted into Modbus floating point format.

Note: Unless special stated, the values of the command and response frames in the following instructions are all hexadecimal data.

3.1 Register Overview

The following is a list of all register addresses used by the AT381x. Any address not in the table will return error code 0x02.

Table 3-1 Register Overview

Register address	Name	Value	Description
2000-2001	Read the primary parameter measurement result	4-byte floating point number	Read-only register, data occupies 2 registers
2002-2003	Read the secondary parameter measurement result	4-byte floating point number	Read-only register, data occupies 2 registers
2004	Get the comparator result	2 Byte integer	Read-only register, data occupies 1 register
0000	Read instrument version number	4 Byte ASCII	Read-only register, data occupies 2 registers
3000	Function register	Data function 0000 Cs-Rs 0001 Cs-D 0002 Cp-Rp 0003 Cp-D 0004 Lp-Rp 0005 Lp-Q 0006 Ls-Rs 0007 Ls-Q 0008 Rs-Q	Read and write registers, 2-byte integer

		0009 Rp-Q 000A R-X 000B DCR 000C Z-r 000D Z-d 000E Z-D 000F Z-Q	
3001	LCR range No.	0000~0008	Read and write registers, 2-byte integer
3002	Range mode	0000：Manual 0001：Auto 0002：Nominal	Read and write registers, 2-byte integer
3003	Test speed	0000：Slow 0001：Medium1 0002：Medium2 0003：Fast	Read and write registers, 2-byte integer
3004	Average number of times	0000：Invalid 0001~0x0100 (1~256)	Read and write registers, 2-byte integer
3005	Trigger mode	0000：Internal 0001：Manual 0002：External 0003：Remote	Read and write registers, 2-byte integer
3006-3007	Test frequency	4-byte floating point number	Read and write registers, data occupies 2 registers
3008-3009	Test level	4-byte floating point number	Read and write registers, data occupies 2 registers
300A	DCR range No.	0000~0008	Read and write registers, 2-byte integer
300C	Recall at startup	0000：File 0 0001：Current file	Read and write registers, 2-byte integer
300D	Auto save	0000：Disable 0001：Allow	Read and write registers, 2-byte integer
300E	System language	0000：English 0001：Chinese	Read and write registers, 2-byte integer
3010-3011	Test current	4-byte floating point number	Read and write registers, data occupies 2 registers
3012-3013	DC bias	4-byte floating point number	Read and write registers, data occupies 2 registers
3100	Comparator status	0000：Comparator off 0001：Comparator on	Read and write registers, 2-byte integer
3101	Comparator mode	0000：ABS	Read and write registers, 2-byte

		0001：PER 0002：SEQ	integer
3102	Secondary parameter comparison on/off	0000: Secondary parameter comparison off 0001 : Secondary parameter comparison on	Read and write registers, 2-byte integer
3103	Bin count	0001~0009 Bin count 1~9bin	Read and write registers, 2-byte integer
3104	Beep	0000：off 0001：GD beep 0002：NG beep	Read and write registers, 2-byte integer
310A	Primary parameter nominal value	4-byte floating point number	Read and write registers, data occupies 2 registers
310C	Secondary parameter lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
310E	Secondary parameter higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3110	Primary parameter BIN1 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3112	Primary parameter BIN1 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3114	Primary parameter BIN2 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3116	Primary parameter BIN2 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3118	Primary parameter BIN3 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
311A	Primary parameter BIN3 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
311C	Primary parameter BIN4 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
311E	Primary parameter BIN4 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3120	Primary parameter BIN5 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3122	Primary parameter BIN5 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3124	Primary parameter BIN6 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3126	Primary parameter BIN6 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3128	Primary parameter BIN7 lower	4-byte floating point	Read and write registers, data

	limit value	number	occupies 2 registers
312A	Primary parameter BIN7 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
312C	Primary parameter BIN8 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
312E	Primary parameter BIN8 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3130	Primary parameter BIN9 lower limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
3132	Primary parameter BIN9 higher limit value	4-byte floating point number	Read and write registers, data occupies 2 registers
4000	Save settings to current file	Fixed value：0001	Write-only register, data 2 bytes
4008	Read current file data	Fixed value：0001	Write-only register, data 2 bytes
4010	Save settings to the specified file	0000~0009	Write-only register, data 2 bytes
4018	Read specified file data	0000~0009	Write-only register, data 2 bytes
5000	Open circuit full frequency clear register Read correction status	Write a fixed value：0001 Read： 0001 in correcting 0000 correction success FFFF correction fail	Read and write registers, data occupies 1 register Once the correction function is executed, Modbus will disable the execution of the write command and only allow the register to be read.
5008	Short circuit full frequency clear register Read correction status	Write a fixed value：0001 Read： 0001 in correcting 0000 correction success FFFF correction fail	Read and write registers, data occupies 1 register Once the correction function is executed, Modbus will disable the execution of the write command and only allow the register to be read.
5010	Point frequency 1 setting	Frequency value Floating point number 0： Point frequency off 4-byte floating point number	Read and write registers, data occupies 2 registers
5012	Point frequency 2 setting	Frequency value Floating point number 0： Point frequency off 4-byte floating point number	Read and write registers, data occupies 2 registers
5014	Point frequency 3 setting	Frequency value Floating point number 0： oint frequency off 4-byte floating point	Read and write registers, data occupies 2 registers

		number	
5020	Point frequency open circuit correction	0001：Point frequency 1 0002：Point frequency 2 0003：Point frequency 3 Read： 0000 correction success FFFF correction fail	Read and write registers, data occupies 1 register
5028	Point frequency short circuit correction	0001：Point frequency 1 0002：Point frequency 2 0003：Point frequency 3 Read： 0000 correction success FFFF correction fail	Read and write registers, data occupies 1 register

3.2 Fetch Measurement Data

3.2.1 Fetch measurement data [2000H-2003H]

Register 2000~2003 is used for fetching measurement data of the AT381x

Command：

1	2	3	4	5	6	7	8
01	03	2000		0002		CRC-16	
passive station	read	register		number of registers		check code	

Response

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

- **Obtain the primary parameter measurement result：**

Send:

1	2	3	4	5	6	7	8
01	03	20	00	00	02	CF	CB
passive station	read	register		number of registers		check code	

Response:

1	2	3	4	5	6	7	8	9
01	03	04	4E	6E	6B	28	A3	E8
01	03	byte	single precision floating point number				CRC-16	

Among them, B4~B6 is measurement data：4E6E6B28 stands for 1E9（Low position in front）

- **Fetch the secondary parameter measurement result：**

Send:

1	2	3	4	5	6	7	8

01	03	20	02	00	02	6E	0B
passive station	read	register		number of registers		check code	

Response:

1	2	3	4	5	6	7	8	9
01	03	04	50	15	02	F9	3B	D5
01	03	byte	single precision floating point number				CRC-16	

Among them, B4~B6 is measurement data：501502F9 stands for 1E10（Low position in front）

3.2.2 Fetch Comparator Results [2004H]

Register 2004 records voltage and resistance comparator results

16-bit storage domain:

Among them：

BIT8	secondary parameter bin	1：NG
BIT7	total GD bin	1：primary and secondary parameter GD，0：Total NG
BIT3~BIT0	GD bin	1~9：GD　0：NG

Send:

1	2	3	4	5	6	7	8
01	**03**	**20**	**04**	**00**	**01**	**CE**	**0B**
passive station	read	register		number of registers		check code	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**E0**	**E5**

Example：

Data 01 81

BIT8:　1=secondary parameter NG

BIT7:　1= Total NG

BIT3-BIT0：　1=primary parameter GD bin BIN1

Data 00 01

BIT8：　0=secondary parameter GD

BIT7：　0=Total GD

BIT3-BIT0：　1= primary parameter GD bin BIN1

Fetch primary and secondary measurement values and comparator results【2000~2004】

Send:

1	2	3	4	5	6	7	8
01	**03**	**20**	**00**	**00**	**05**	**8E**	**09**
passive station	read	register		number of registers		check code	

Response:

01 03 0A **44 79 D4 B1** **37 D6 9D C2** **00 81** C6 24

3.3 Parameter Setting

3.3.1 Function Register [3000H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**00**	**00**	**01**	**02**	**00**	**00**	**96**	**53**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**00**	**00**	**01**	**AF**	**09**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**00**	**00**	**01**	**8B**	**0A**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**08**	**B9**	**82**
		byte	data		CRC	

Data values:

Data	Function
0000	Cs-Rs
0001	Cs-D
0002	Cp-Rp
0003	Cp-D
0004	Lp-Rp
0005	Lp-Q
0006	Ls-Rs
0007	Ls-Q
0008	Rs-Q
0009	Rp-Q
000A	R-X
000B	DCR
000C	Z-r
000D	Z-d
000E	Z-D
000F	Z-Q

3.3.2 LCR Range Register [3001H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**01**	**00**	**01**	**02**	**00**	**01**	**56**	**42**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**01**	**00**	**01**	**5F**	**09**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**01**	**00**	**01**	**DA**	**CA**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	Range 0	100kΩ
0001	Range 1	30kΩ
0002	Range 2	10kΩ
0003	Range 3	3kΩ
0004	Range 4	1kΩ
0005	Range 5	300Ω
0006	Range 6	100Ω
0007	Range 7	30Ω
0008	Range 8	10Ω

3.3.3 Range Mode Register [3002H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**02**	**00**	**01**	**02**	**00**	**01**	**56**	**71**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**02**	**00**	**01**	**AF**	**09**
		register		number of registers		CRC	

- Read

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

	read	register	number of registers	CRC

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	Manual range	
0001	Auto range	
0002	Nominal	Select range based on nominal value

3.3.4 Measurement Speed Register [3003H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**03**	**00**	**01**	**02**	**00**	**01**	**57**	**A0**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**03**	**00**	**01**	**FE**	**C9**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**03**	**00**	**01**	**7B**	**0A**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	Slow speed	
0001	Invalid	This value is reserved
0002	Medium speed	
0003	Fast speed	

3.3.5 Averaging Factor Register [3004H]

- Write

1	2	3	4	5	6	7	8	9	10	11

01	10	30	04	00	01	02	00	02	16	16
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**04**	**00**	**01**	**4F**	**08**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**04**	**00**	**01**	**CA**	**CB**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**02**	**39**	**85**
		byte	data		CRC	

Data values:

Data	Function	Description
0001~0100	average value 0~256	average value 0=average value 1

3.3.6 Trigger Mode Register [3005H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**05**	**00**	**01**	**02**	**00**	**02**	**17**	**C7**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**05**	**00**	**01**	**1E**	**C8**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**05**	**00**	**01**	**9B**	**0B**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**02**	**39**	**85**
		byte	data		CRC	

Data values:

Data	Function	Description

0000	Internal trigger	
0001	Manual trigger	Use Trigger key
0002	External trigger	Handler trigger
0003	Remote trigger	SCPI trigger

3.3.7 Measurement Frequency Register [3006H-3007H]

- Write [1kHz: 1000 = 44 7A 00 00]

1	2	3	4	5	6	7	8~11	12	13
01	**10**	**30**	**06**	**00**	**02**	**04**	**44 7A 00 00**	**12**	**AD**
	write	register		number of registers		byte	data	CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**06**	**00**	**02**	**AE**	**C9**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**06**	**00**	**02**	**CF**	**1A**
	read	register		number of registers		CRC	

Response:

1	2	3	4~7	8	9
01	**03**	**04**	**44 7A 00 00**	**CF**	**1A**
		byte	data	CRC	

3.3.8 Level Register [3008H-3009H]

- Write [1V: 1.00 = 3F 80 00 00]

1	2	3	4	5	6	7	8~11	12	13
01	**10**	**30**	**08**	**00**	**02**	**04**	**3F 80 00 00**	**EB**	**B4**
	write	register		number of registers		byte	data	CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**08**	**00**	**02**	**FF**	**09**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**08**	**00**	**02**	**7A**	**CA**
	read	register		number of registers		CRC	

Response:

1	2	3	4~7	8	9

01	03	04	3F 80 00 00	F7	CF
		byte	data	CRC	

Note: The following situations will return an error

Test Condition	Description
<CORRECTION> page	Error in operation
<List Sweep> page	Error in operation
Level is current state	Error in operation
Function：DCR	Error in operation

3.3.9 DCR Range Register [300AH]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**0A**	**00**	**01**	**02**	**00**	**01**	**97**	**3A**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**0A**	**00**	**01**	**2E**	**CB**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**0A**	**00**	**01**	**AB**	**08**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**04**	**B9**	**87**
		byte	data		CRC	

Data values：

Data	Function	Description
0000	Range 0	100kΩ
0001	Range 1	30kΩ
0002	Range 2	10kΩ
0003	Range 3	3kΩ
0004	Range 4	1kΩ
0005	Range 5	300Ω
0006	Range 6	100Ω
0007	Range 7	30Ω
0008	Range 8	3Ω

3.3.10 Startup File Recall Register [300CH]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**0C**	**00**	**01**	**02**	**00**	**01**	**FA**	**C8**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**0C**	**00**	**01**	**CE**	**CA**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**0C**	**00**	**01**	**4B**	**09**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	File 0	Default setting
0001	Current file	

3.3.11 Auto Save [300DH]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**0D**	**00**	**01**	**02**	**00**	**01**	**56**	**8E**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**0D**	**00**	**01**	**9F**	**0A**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**0D**	**00**	**01**	**79**	**84**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	OFF	Default setting
0001	ON	

3.3.12 System Language Setting [300EH]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**30**	**0E**	**00**	**01**	**02**	**00**	**01**	**56**	**BD**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**0E**	**00**	**01**	**6F**	**0A**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**0E**	**00**	**01**	**EA**	**C9**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	English	
0001	Chinese	

3.3.13 Test Current Register [3010H-3011H]

- Write [1mA: 0.001 = 3A 83 12 6F]

1	2	3	4	5	6	7	8~11	12	13
01	**10**	**30**	**10**	**00**	**02**	**04**	**3A 83 12 6F**	**17**	**1E**
	write	register		number of registers		byte	Data (1mA)	CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**10**	**00**	**02**	**4F**	**0D**
		register		number of registers		CRC	

Note: The following situations will return an error

Test Condition	Description
<CORRECTION> page	Error in operation

<LIST SWEEP> page	Error in operation
Function：DCR	Error in operation
Level value exceeds specification value	Data error

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**10**	**00**	**02**	**CA**	**CE**
	read	register		number of registers		CRC	

Response:

1	2	3	4~7	8	9
01	**03**	**04**	**3A 83 12 6F**	**4B**	**8F**
		byte	data 1mA	CRC	

Note: The following situations will return an error

Test Condition	Description
<CORRECTION> page	Error in operation
<LIST SWEP> page	Error in operation
Level is voltage state	Error in operation
Function：DCR	Error in operation

Note: The current level is not current mode, or an operation error is returned when the measurement parameter is DCR.

3.3.14 DC Bias Register [3012H-3013H]

- Write 1V: 3F 80 00 00

1	2	3	4	5	6	7	8~11	12	13
01	**10**	**30**	**12**	**00**	**02**	**04**	**3F 80 00 00**	**2A**	**87**
	write	register		number of registers		byte	data (1V)	CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**30**	**12**	**00**	**02**	**EE**	**CD**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**30**	**12**	**00**	**02**	**6B**	**0E**
	read	register		number of registers		CRC	

Response:

1	2	3	4~7	8	9
01	**03**	**04**	**00 00 00 00**	**FA**	**33**
		byte	data	CRC	

3.4 Comparator Setting

The comparator parameter register address starts at 3100.

3.4.1 Comparator Status Register [3100H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**31**	**00**	**00**	**01**	**02**	**00**	**01**	**47**	**53**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**31**	**00**	**00**	**01**	**0F**	**35**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**31**	**00**	**00**	**01**	**8A**	**F6**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	Comparator off	Default setting
0001	Comparator on	

3.4.2 Comparator Mode Register [3101H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**31**	**01**	**00**	**01**	**02**	**00**	**01**	**46**	**82**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**31**	**01**	**00**	**01**	**5E**	**F5**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**31**	**01**	**00**	**01**	**DB**	**36**
	read	register		number of		CRC	

			registers	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	ABS compare	Absolute deviation comparison
0001	PER compare	Percent deviation comparison
0002	SEQ compare	Sequential comparison

3.4.3 Secondary Comparator (AUX) ON/OFF Register [3102H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**31**	**02**	**00**	**01**	**02**	**00**	**01**	**46**	**B1**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**31**	**02**	**00**	**01**	**AE**	**F5**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**31**	**02**	**00**	**01**	**2B**	**36**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	Secondary parameter comparison off	AUX OFF
0001	Secondary parameter comparison on	AUX ON

3.4.4 Comparison Total Bins Register [3103H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**31**	**03**	**00**	**01**	**02**	**00**	**01**	**47**	**B1**

	write	register	number of registers	byte	data	CRC

Response:

1	2	3	4	5	6	7	8
01	**10**	**31**	**03**	**00**	**01**	**FF**	**35**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**31**	**03**	**00**	**01**	**7A**	**F6**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	Invalid	
0001~0009	Bin count	

3.4.5 Beep Register [3104H]

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**31**	**04**	**00**	**01**	**02**	**00**	**01**	**46**	**D7**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**31**	**04**	**00**	**01**	**4E**	**F4**
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	**03**	**31**	**04**	**00**	**01**	**CB**	**37**
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7
01	**03**	**02**	**00**	**01**	**79**	**84**
		byte	data		CRC	

Data values:

Data	Function	Description
0000	OFF	Turned off

0001	PASS	OK beep
0002	FAIL	NG beep

3.4.6 Nominal Value Register [310AH-310BH]

The primary parameter nominal value uses 2 registers, 310A and 310B. Please note that reading 310B alone is invalid.

- Write

100E-9 (Single precision floating point number：33D6BF95)

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	31	0A	00	02	04	33	D6	BF	95	74	A2
	write	register		number of registers		byte	data				CRC	

Response:

1	2	3	4	5	6	7	8
01	10	31	0A	00	02	6F	36
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	03	31	0A	00	02	EA	F5
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7	8	9
01	03	04	33	D6	BF	95	A4	D0
		byte	data 100E-9				CRC	

3.4.7 Secondary Parameter Limit Value Register [310CH-310FH]

The secondary parameter limit value starts from 310C, the lower limit uses 2 registers, and the upper limit uses 2 registers, total of 4 registers.

The lower and upper limits can be set separately, or simultaneously.

- Write

Lower limit：0.001, Upper limit：0.01

1	2	3~4	5	6	7	8~11	12~15	16~17
01	10	310C	00	02	04	3A 83 12 6F	3C 23 D7 0A	21 AE
						Lower limit	Upper limit	

Response:

1	2	3	4	5	6	7	8
01	10	31	0C	00	04	0F	35

- read

1	2	3	4	5	6	7	8
01	03	31	0C	00	04	8A	F6

Response:

1	2	3~4	5~8	9~12	13~14

01	03	31 14	3A 83 12 6F	3C 23 D7 0A	51 61
			Lower limit	Upper limit	

3.4.8 Primary Parameter Limit Value Register [3110H-3133H]

There are total of 9 bins for primary parameter, the limit value starts from 3110, the lower limit of each bin uses 2 registers, and the upper limit uses 2 registers, total of 4 registers.
The lower and upper limits can be set separately, or simultaneously.

- Write

BIN1
Lower limit：-10,　Upper limit：10

1	2	3~4	5	6	7	8~11	12~15	16~17
01	10	3110	00	02	04	C1 20 00 00	41 20 00 00	CD5C
						Lower limit	Upper limit	

Response:

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

- Read

1	2	3	4	5	6	7	8
01	03	31	10	00	04	4B	30

Response:

1	2	3~4	5~8	9~12	13~14
01	03	31 10	C1 20 00 00	41 20 00 00	6C7F
			Lower limit	Upper limit	

3.5 File Operation

Since the AT381x settings are stored in the file, after the Modbus command is set, the data cannot be stored in the internal FlashRom in real time, which will cause the register data before the next power-on to be restored to the original file value.
Users can store all set values in the current or specified file with the file manipulation registers. Meanwhile, the specified file data can also be recalled into the setup register.

3.5.1 Save to Current File [4000H]

Send a value of 0001 to 4000 registers, the AT381x will perform a file write operation, and all settings will be saved to the current file.
This register cannot be read.

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**40**	**00**	**00**	**01**	**02**	**00**	**01**	**26**	**54**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**40**	**00**	**00**	**01**	**14**	**09**

		register	number of registers	CRC

Data values:

Data	Function	Description
0001	Allow to operate	Fixed value

3.5.2 Save to Specified File [4008H]

Send the file number to the 4008 register, the AT381x will perform the file write operation, all the settings will be saved to the specified file, and the specified file will be used as the current file of the system.

This register cannot be read.

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**40**	**08**	**00**	**01**	**02**	**00**	**09**	**26**	**DA**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**40**	**00**	**00**	**01**	**95**	**CB**
		register		number of registers		CRC	

Data values:

Data	Function	Description
0000~0009	File 0~9	

3.5.3 Reloading the Current File [4010H]

The fixed value 0001 to 4010 registers are sent and the AT381x loads the current file data into the system.

This register cannot be read.

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**40**	**10**	**00**	**01**	**02**	**00**	**01**	**24**	**C4**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**40**	**10**	**00**	**01**	**15**	**CC**
		register		number of registers		CRC	

Data values:

Data	Function	Description
0001	Fixed value	

3.5.4 Load the Specified File [4018H]

Send the file number to the 4018 register, the AT381x will load the settings of the specified file

into the system, and the specified file will be used as the current file of the system.

This register cannot be read.

- Write

1	2	3	4	5	6	7	8	9	10	11
01	**10**	**40**	**18**	**00**	**01**	**02**	**00**	**00**	**E4**	**4C**
	write	register		number of registers		byte	data		CRC	

Response:

1	2	3	4	5	6	7	8
01	**10**	**40**	**18**	**00**	**01**	**94**	**0E**
		register		number of registers		CRC	

Error response:

The file is empty and the AT381x will respond with an error code: 04

1	2	3	4	5
01	**90**	**04**	**4D**	**C3**
		Error code	**CRC**	

Data values:

Data	Function	Description
0000~0009	File 0~9	

3.6 Correction

3.6.1 Full-frequency Open Circuit Correction [5000H]

Writing 0001 to Register 5000 will cause the AT381x to begin an open-circuit full-frequency correction.

Since the correction process takes a few seconds, during this time, any write operation will be ignored and only the read operation will be open. After the correction is completed, the write command is normally opened.

Correction status can be obtained by reading the 5000 register during correction execution or after correction:

0000 Correction success

FFFF Correction fail

0001 In correction

- Write

Please write a fixed value to the 5000 register: 00 01

Send: 01 10 5000 0001 02 0001 3795

Response: 01 10 5000 0001 10C9

- Read

During the execution of correction, can determine whether correction is completed by reading the register data.

Send: 01 03 5000 0001 950A

Response: 01 03 02 FFFF B9F4

Returns FFFF, indicating that the correction failed

Note:

When performing correction, try not to read the correction status frequently. Continuous interrupts may cause the instrument to fail to perform correction.

Since the correction time is fixed, it is recommended that after the correction command is issued, after the host waits for the correction time to elapse, it will get the correction result.

3.6.2 Full-frequency Short-circuit Correction [5008H]

Writing 0001 to Register 5008 will cause the AT381x to begin an open-circuit full-frequency correction.

Since the correction process takes a few seconds, during this time, any write operation will be ignored and only the read operation will be open. After the correction is completed, the write command is normally opened.

Correction status can be obtained by reading the 5008 register during correction execution or after correction:

0000	Correction success
FFFF	Correction fail
0001	In correction

- Write

Please write a fixed value to the 5008 register:: 00 01

Send：01 10 5008 0001 02 0001 36DD

Response：01 10 5008 0001 910B

- Read

During the execution of correction, can determine whether correction is completed by reading the register data.

Send：01 03 5008 0001 950A

Response：01 03 02 FFFF B9F4

Returns FFFF, indicating that the correction failed

Note:

When performing correction, try not to read the correction status frequently. Continuous interrupts may cause the instrument to fail to perform correction.

Since the correction time is fixed, it is recommended that after the correction command is issued, after the host waits for the correction time to elapse, it will get the correction result.

3.6.3 Spot frequency Correction Setting [5010H-5015H]

The AT381x point frequency can be set by the register [point frequency 1:5010~5011], [point frequency 2:5012~5013] and [point frequency 3:5014~5015], where set to 00000000 means the point frequency is off.

- Write

Point frequency 1：1kHz (Single precision floating point number：44 7A 00 00)

1	2	3	4	5	6	7	8	9	10	11	12	13
01	10	50	10	00	02	04	44	7A	00	00	3B	89
	write	register		number of registers		byte	data				CRC	

Response：

1	2	3	4	5	6	7	8
01	10	50	10	00	02	51	0D
		register		number of registers		CRC	

- Read

1	2	3	4	5	6	7	8
01	03	50	10	00	02	CF	1A
	read	register		number of registers		CRC	

Response：

1	2	3	4	5	6	7	8	9
01	03	04	44	7A	00	00	CF	1A
		byte	data 1000				CRC	

3.6.4 Spot Frequency Open Circuit Correction [5020H]

By writing the point frequency 0001~0003 to the register 5020, the AT381x will begin to perform an open-circuit point frequency correction.

After clearing is complete, you can get the clear status by reading the 5008 register:

0000 Correction success

FFFF Correction fail

- Write

Please write the point frequency to the 5020 register: 0001~0003, and execute the corresponding point frequency open circuit correction.

Send: 01 10 5020 0001 02 0001 30F5

Response: 01 10 5020 0001 1103

- Read

  During the execution of correction, can determine whether correction is completed by reading the register data.

Send：01 03 5020 0001 94C0

Response：01 03 02 0000 B844

Returns FFFF, indicating that the correction failed

Note:
It takes time to perform correction of the point frequency. Please delay the time in the software to return the data.

3.6.5 Spot Frequency Short-Circuit Correction [5028H]

By writing the point frequency 0001~0003 to the register 5028, the AT381x will begin to perform an open-circuit point frequency correction.

After clearing is complete, you can get the clear status by reading the 5028 register:
0000 Correction success
FFFF Correction fail

- Write

Please write the point frequency to the 5028 register: 0001~0003, and execute the corresponding point frequency short-circuit correction.
Send: 01 10 5028 0001 02 0001 31BD
Response: 01 10 5028 0001 90C1

- Read

  During the execution of correction, can determine whether correction is completed by reading the register data.

Send：01 03 5028 0001 1502
Response：01 03 02 FFFF B9F4

Returns FFFF, indicating that the correction failed

Note:
It takes time to perform correction of the point frequency. Please delay the time in the software to return the data.

3.7 System Setup

3.7.1 Instrument version number [0000H]

Read-only register, register [0000] ~ [0003] return the version number of the instrument:

- Read

1	2	3	4	5	6	7	8
01	03	00	10	00	02	4A	6D
	read	register		number of registers		CRC	

Response:

1	2	3	4	5	6	7	8	9
01	03	04	43	37	30	30	4A	6D
		byte	data ASCII: C700				CRC	

The version number is ASCII value: for example, 43 37 30 30 = C700