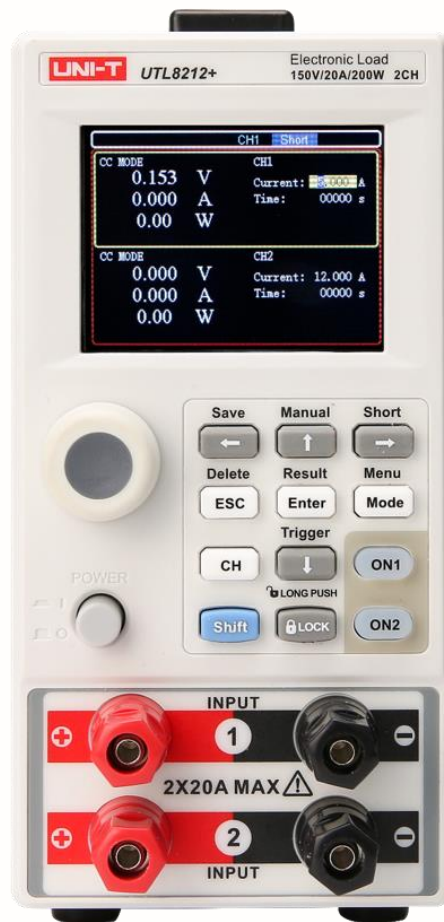# Programming Manual
# UTL8200+ Series DC Electronic Load

# Warranty and Statement

## Copyright
2023 Uni-Trend Technology (China) Co., Ltd.

## Brand Information
UNI-T is the registered trademark of Uni-Trend Technology (China) Co., Ltd.

## Statement
- UNI-T products are protected by patents (including obtained and pending) in China and other countries and regions.
- UNI-T reserves the right to change specifications and prices.
- The information provided in this manual supersedes all previous publications.
- The information provided in this manual is subject to change without notice.
- UNI-T shall not be liable for any errors that may be contained in this manual. For any incidental or consequential damages arising out of the use or the information and deductive functions provided in this manual.
- No part of this manual shall be photocopied, reproduced or adapted without the prior written permission of UNI-T.

## Product Certification
UNI-T has certified that the product conforms to China national product standard and industry product standard as well as ISO9001:2008 standard and ISO14001:2004 standard. UNI-T will go further to certificate product to meet the standard of other member of the international standards organization.

# SCPI Introduction

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

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

# Command String Parse

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

For example

Valid command string:

AAA:BBB CCC;DDD EEE;:FFF

The instrument command parser is responsible for all command parsing and execution, and you must understand its parsing rules before writing a program.

# Command Parse Rule

IE command parser only parses and responds to ASCII data.

SCPI command string must be end with end mark NL(' \n' ASCII 0x0A). The command parser starts to executing when receive the end mark or an input buffer overflow.

If the command handshake is open, the command parser sends the character back to the host immediately after each character is received, and the host can continue to send the next character only after receiving this return character.

The command parser will terminate the parsing immediately after parsing an error, and the current command will be invalidated.

After the command parser parses the query command, it terminates this command string parsing and the subsequent strings are ignored.

The command parser is case-insensitive for parsing command strings.

The command parser supports abbreviated form of command and the detailed see the following section.

# Symbol Stipulation and Definition

This chapter uses some symbols that are not part of the command tree, but only for a better understanding of the command string.

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

[ ]    The text in square brackets indicates the optional command.

{}    When the curly brackets contain several parameter items, it means that only one item can be selected from them.

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

Capital letter    Abbreviated form of the command.

# Command Tree Structure

SCPI commands have a tree-like structure with three level (note: the command parser of this instrument can parse any level), where the highest level is called the subsystem command. SCPI uses a colon (:) to separate high level commands from low level commands.
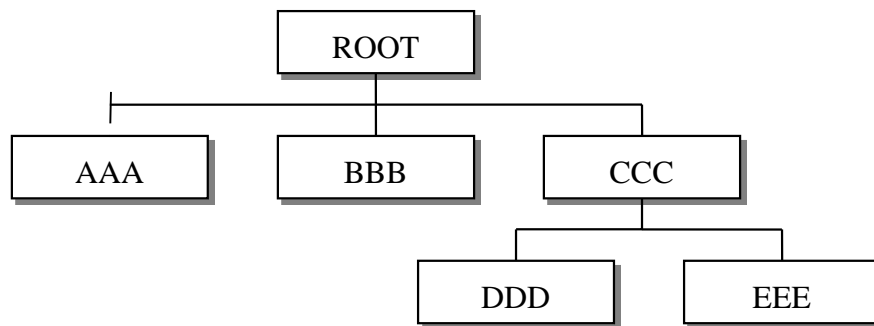


Figure 1-1 Command Tree Structure

For Example

ROOT:CCC:DDD ppp

ROOT                Subsystem command

CCC                Second level

DDD                Third level

ppp                Parameter

# Command and Parameter

A command tree is consist of command and [parameter], use a blank to separate (ASCII: 20H).

>       For example    AAA:BBB 1.234

>       Command    [parameter]

# Command

Command words can be in long command format or in abbreviated form. Long format facilitates engineers to better understand the meaning of the command string; abbreviated form is suitable for writing.

# Parameter

Single character command, no parameter

For Example    AAA:BBB

Parameter can be string format and its abbreviated form is also follow the last section " command abbreviated rule" For example AAA:BBB 1.23

Parameter can be numerical value format

<integer>    integer 123, +123, -123

<float>         floating point number

1. <fixfloat>: fixed floating point number: 1.23, -1.23

2. <Scifloat>: floating point number represented by scientific notation: 1.23E+4, +1.23e-4

3. <mpfloat>: floating point number represented by multiplying power: 1.23k, 1.23M, 1.23G, 1.23u represented by 0-1 multiplying power abbreviation

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

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

# Separator

The instrument command parser can only receive allowable separator. Other separator will cause error "Invalid separator".

The allowable separator is as follows.

**; Semicolon** is for separating two commands.

    For Example    AAA:BBB 100.0 ; CCC:DDD

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

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

**? Question mark** is for querying.

    For ExampleAAA ?

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

    For Example    AAA:BBB□1.234

# Error Code

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

# Command Reference

All commands is explained by the subsystem command order.

SOURce          Setup subsystem

SYSTem          System subsystem

MEASure     Measurement subsystem

CHANnel          Channel subsystem


Common Command

*IDN?           Instrument query subsystem

*RST           Reset the instrument state

# SYSTem Subsystem

| | | |
|---|---|---|
| SYSTem | :ERRor[:NEXT]? | |
| | :ERRor:COUNt? | |
| | :VERSion? | |
| | :BEEPer[:STATe] | |
| | :BEEPer[:STATe]? | |

Figure 1-2 SYSTem Subsystem Tree

* The parameter is set by SYSTem subsystem will not saved in the instrument.

  It should be reset when reboot the instrument.

### SYSTem:ERRor[:NEXT]?

SYSTem:ERRor[:NEXT]?   Query the error message

Query Syntax          SYSTem:ERRor[:NEXT]?        // Query the error message

Query Response          Error message

### SYSTem:ERRor:COUNt?

SYSTem:ERRor:COUNt?    Query the count of error message.

| | |
|---|---|
| Query Syntax | SYSTem:ERRor:COUNt? // Query the count of error message |
| Query Response | The count of error message |

### SYSTem:VERSion?

SYSTem:VERSion?    Query the version of SCPI system

| | |
|---|---|
| Query Syntax | SYSTem:VERSion? // Query the version of SCPI system |
| Query Response | SCPI system version |

### SYSTem:BEEPer:STATe

SYSTem:BEEPer:STATe    Enable or disable the beeper

| | | |
|---|---|---|
| Command Syntax | SYSTem:BEEPer:STATe<bool> | |
| Parameter | 0|1 OFF|ON | |
| For Example | SYST:BEEP:STAT ON | // Enable the beeper |
| Query Syntax | SYSTem:BEEPer:STATe? | // Query the state of beeper |
| Return Parameter | 0|1 | |

### [SOURce:]INPut

[SOURce:]INPut    Turn on or turn off the input

| | | |
|---|---|---|
| Command Syntax | [SOURce;]INPut[:STATe]<bool> | |
| Parameter | 0|1 OFF|ON | |
| For Example | INP 1 | // Turn on the input |
| Query Syntax | INPut[:STATe]? | // Query the state of the input |
| Return Parameter | 0|1 | |

### [SOURce:]INPut:SHORt

[SOURce:]INPut:SHORt    Turn on or turn off the short circuit

| | |
|---|---|
| Command Syntax | [SOURce;]INPut[:STATe]<bool> |
| Parameter | 0|1 OFF|ON |

| For Example | INP:SHOR 1 | // Turn on the short circuit |
|---|---|---|
| Query Syntax | INPut:SHORt? | // Query the state of the short circuit |
| Return Parameter | 0\|1 | |

## [SOURce:]FUNCtion

| [SOURce:]MODE | 2 command equivalences |
|---|---|

[SOURce:]MODE          Select the input mode of the load, 2 command equivalences

| Command Syntax | [SOURce:]FUNCtion <function> |
|---|---|
| | [SOURce:]MODE <function> |
| Parameter | Operation mode |
| CURRent | Constant current mode |
| VOLTage | Constant voltage mode |
| POWer | Constant power mode |
| RESistance | Constant resistance mode |
| DYNamic | Dynamic mode |
| BATtery | Battery mode |
| LIST | List mode |
| For Example | MODE RES                         // Set to constant resistance mode |
| Query Syntax | [SOURce:]FUNCtion?   [SOURce:]MODE? |
| | // Query the input mode of the load |
| Return Parameter | <CRD> |

## [SOURce:]CURRent:RANGe

[SOURce:]CURRent:RANGe    Set the range for the current

| Command Syntax | [SOURce;]CURRent:RANGe<NRF+> |
|---|---|
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | A |
| Reset Value | MAXimum range |
| For Example | CURR:RANGE MIN            // Set the current range to MIN |
| | CURR:RANGE 25              // Set the current range to MAX |
| Query Syntax | [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? |
| | // Query the size of the current |
| Return Parameter | <NR2> |

［SOURce］:CURRent:SLEW

| ［SOURce］:CURRent:SLEW | Set the rising and falling rate for the same current |
|---|---|
| Command Syntax | ［SOURce］:CURRent:SLEW[:BOTH]<NRF+> |
| Parameter | MIN~MAX \|MINimum\|MAXimum |
| Unit | A/us |
| Reset Value | 1 |
| For Example | CURR:SLEW 1     // Set the rising and falling rate for the current to 1 A/us |
| | CURR:SLEW 0.4, 0.8      // Set the rising rate 0.4 A/us, falling rate 0.8 A/us |
| Query Syntax | ［SOURce］:CURRent:SLEW? |
| | // Query the rising and falling rate of the current |
| Return Parameter | <NR2> |

［SOURce］:CURRent:SLEW:RISE

| ［SOURce］:CURRent:SLEW:RISE | Set the rising rate for the current |
|---|---|
| Command Syntax | ［SOURce］:CURRent:SLEW:RISE<NRF+> |
| Parameter | MIN~MAX \|MINimum\|MAXimum |
| Unit | A/us |
| Reset Value | 1 |
| For Example | CURR:SLEW:RISE 1                    // Set the rising rate for the current to 1 A/us |
| Query Syntax | ［SOURce］:CURRent:SLEW:RISE?      // Query the rising rate of the current |
| Return Parameter | <NR2> |

［SOURce］:CURRent:SLEW:FALL

| ［SOURce］:CURRent:SLEW:FALL | Set the falling rate for the current |
|---|---|
| Command Syntax | ［SOURce］:CURRent:SLEW:FALL <NRF+> |
| Parameter | MIN~MAX \|MINimum\|MAXimum |
| Unit | A/us |
| Reset Value | 1 |
| For Example | CURR:SLEW:FALL 1                    // Set the falling rate for the current to 1 A/us |
| Query Syntax | ［SOURce］:CURRent:SLEW:FALL?     // Query the falling rate of the current |
| Return Parameter | <NR2> |

［SOURce］:VOLTage:SLEW

| ［SOURce］:VOLTage:SLEW | Set the rising and falling rate for the voltage |
|---|---|

| Command Syntax | [SOURce]:VOLTage:SLEW[:BOTH]<NRF+> |
|---|---|
| Parameter | MIN~MAX|MINimum|MAXimum |
| Unit | V/ms |
| For Example | VOLT:SLEW 0.3      // Set the rising and falling rate for the voltage to 0.3 V/ms |
| Query Syntax | [SOURce]:VOLTage:SLEW?<br>// Query the rising and falling rate of the voltage |
| Return Parameter | <NR2> |

## [SOURce]:CURRent:PROTection

[SOURce]:CURRent:PROTection    Set the protective value for the current

| Command Syntax | [SOURce]:CURRent:PROTection[:LEVel]<NRF+> |
|---|---|
| Parameter | 0~MAX|MINimum|MAXimum |
| Unit | A |
| Reset Value | MAXimum |
| For Example | CURR:PROT 3            // Set the protective value for the current to 3A |
| Query Syntax | [SOURce]:CURRent:PROTection[:LEVel]?<br>// Query the protective value of the current |
| Return Parameter | <NR2> |

## [SOURce]:POWer:PROTection

[SOURce]:POWer:PROTection    Set the protective value for the power

| Command Syntax | [SOURce]:POWer:PROTection[:LEVel]<NRF+> |
|---|---|
| Parameter | 0~MAX|MINimum|MAXimum |
| Unit | W |
| Reset Value | MAXimum range |
| For Example | CURR:PROT 100            // Set the protective value for the power to 100 W |
| Query Syntax | [SOURce]:POWer:PROTection[:LEVel]?<br>// Query the protective value of the power |
| Return Parameter | <NR2> |

## [SOURce]:VOLTage[:LEVel]:ON

[SOURce]:VOLTage[:LEVel]:ON    Set voltage value of the start load (Von)

| Command Syntax | [SOURce]:VOLTage[:LEVel]:ON <NRF+> |
|---|---|
| Parameter | 0~MAX|MINimum|MAXimum |

| Unit | V |
|---|---|
| Reset Value | 1 |
| For Example | VOLT:ON 3          // Set voltage value of the start load to 3V |
| Query Syntax | [SOURce]:VOLTage[:LEVel]:ON? |
|  | // Query voltage value of the start load |
| Return Parameter | <NR2> |

## [SOURce]:VOLTage[:LEVel]:OFF

[SOURce]:VOLTage[:LEVel]:OFF    Set voltage value of the start unload (Voff)

| Command Syntax | [SOURce]:VOLTage[:LEVel]:OFF <NRF+> |
|---|---|
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | V |
| Reset Value | 0.5 |
| For Example | VOLT:ON 2            // Set voltage value of the start unload to 2V |
| Query Syntax | [SOURce]:VOLTage[:LEVel]:OFF? |
|  | // Query voltage value of the start unload |
| Return Parameter | <NR2> |

## [SOURce]:CURRent

[SOURce]:CURRent      Set the current in CC mode

| Command Syntax | [SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]<NRF+> |
|---|---|
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | A |
| Reset Value | MINimum |
| For Example | CURR 5                  // Set the current to 5A in CC mode |
| Query Syntax | [SOURce]:CURRent[:LEVel][:IMMediate][:AMPLitude]? |
|  | // Query the current in CC mode |
| Return Parameter | <NR2> |

## [SOURce]:VOLTage

[SOURce]:VOLTage      Set the voltage in CV mode

| Command Syntax | [SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]<NRF+> |
|---|---|
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | V |
| Reset Value | MAXimum |

| For Example | VOLT 5                                    // Set the voltage to 5V in CV mode |
|---|---|
| Query Syntax | [SOURce]:VOLTage[:LEVel][:IMMediate][:AMPLitude]? |
|  | // Query the voltage in CV mode |
| Return Parameter | <NR2> |

## [SOURce]:RESistance

| [SOURce]:RESistance | Set the resistance in CR mode |
|---|---|
| Command Syntax | [SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude]<NRF+> |
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | ohm |
| Reset Value | MAXimum |
| For Example | RES 5                          // Set the resistance to 5ohm in CR mode |
| Query Syntax | [SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude]? |
|  | // Query the resistance in CR mode |
| Return Parameter | <NR2> |

## [SOURce]:POWer

| [SOURce]:RESistance | Set the power in CP mode |
|---|---|
| Command Syntax | [SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]<NRF+> |
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | W |
| Reset Value | MINimum |
| For Example | POW 10                          // Set the power t0 10W in CP mode |
| Query Syntax | [SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]? |
|  | // Query the power in CP mode |
| Return Parameter | <NR2> |

## [SOURce]:DYNamic:LOW

| [SOURce]:DYNamic:IA | 2 command equivalences |
|---|---|

| [SOURce]:DYNamic:LOW | Set the load current (IA) in dynamic mode |
|---|---|
| Command Syntax | [SOURce]:DYNamic:LOW[:LEVel]<NRF+> |
| Command Syntax | [SOURce]:DYNamic:IA[:LEVel<NRF+> |
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | A |

| Reset Value | 0 |
|---|---|
| For Example | DYN:LOW 10 or DYN:IA 10 |
| | // Set the load current (IA) to 10A in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:LOW[:LEVel]? |
| | // Query the low level of load current (IA) in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:IA[:LEVel? |
| | // Query the low level of load current (IA) in dynamic mode |
| Return Parameter | <NR2> |

[SOURce]:DYNamic:LOW:DWELl

| [SOURce]:DYNamic:TA | 2 command equivalences |
|---|---|

| [SOURce]:DYNamic:LOW:DWELl | Set the duration (TA) for the load current in dynamic mode |
|---|---|
| Command Syntax | [SOURce]:DYNamic:LOW:DWELl<NRF+> |
| Command Syntax | [SOURce]:DYNamic:TA:DWELl<NRF+> |
| Parameter | MIN~MAX \|MINimum\|MAXimum |
| Unit | mS |
| Reset Value | 0.1 |
| For Example | DYN:LOW:DWELL 10 or DYN:TA:DWELL 10 |
| | // Set the duration (TA) to 0.01S for the load current in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:LOW:DWELl? |
| | // Query the duration (TA) of the load current in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:TA:DWELl? |
| | // Query the duration (TA) of the load current in dynamic mode |
| Return Parameter | <NR2> |

[SOURce]:DYNamic:HIGH

| [SOURce]:DYNamic:IB | 2 command equivalences |
|---|---|

| [SOURce]:DYNamic:HIGH | Set the load current (IB) in dynamic mode |
|---|---|
| Command Syntax | [SOURce]:DYNamic:HIGH[:LEVel]<NRF+> |
| Command Syntax | [SOURce]:DYNamic:IB[:LEVel]<NRF+> |
| Parameter | 0~MAX \|MINimum\|MAXimum |
| Unit | A |
| Reset Value | 0 |
| For Example | DYN:HIGH 1 or DYN:IB 1 |

|  | // Set the load current (IB) to 1A in dynamic mode |
| --- | --- |
| Query Syntax | [SOURce]:DYNamic:HIGH[:LEVel]? |
|  | // Query the high level of the load current (IB) in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:IB[:LEVel]? |
|  | // Query the high level of the load current (IB) in dynamic mode |
| Return Parameter | <NR2> |

## [SOURce]:DYNamic: HIGH:DWELl

| [SOURce]:DYNamic:TB | 2 command equivalences |
| --- | --- |

[SOURce]:DYNamic: HIGH:DWELl    Set the duration for the load current (IB) in dynamic mode

| Command Syntax | [SOURce]:DYNamic: HIGH:DWELl<NRF+> |
| --- | --- |
| Command Syntax | [SOURce]:DYNamic:IB:DWELl<NRF+> |
| Parameter | MIN~MAX |MINimum|MAXimum |
| Unit | mS |
| Reset Value | 0.1 |
| For Example | DYN: HIGH:DWELL 10 or DYN:TB:DWELL 10 |
|  | // Set the duration of the load current (IB) to 10S in dynamic mode |
| Query Syntax | [SOURce]:DYNamic: HIGH:DWELl? |
|  | // Query the duration of the load current (IB) in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:TB:DWELl? |
|  | // Query the duration of the load current (IB) in dynamic mode |
| Return Parameter | <NR2> |

## [SOURce]:DYNamic:SLEW

[SOURce]:DYNamic:SLEW    Set the current slop for the dynamic mode

| Command Syntax | [SOURce]:DYNamic:SLEW<NRF+> |
| --- | --- |
| Parameter | MIN~MAX |MINimum|MAXimum |
| Unit | A/us |
| Reset Value | MAX |
| For Example | DYN:SLEW 1          // Set the current slope to 1 A/us for the dynamic mode |
|  | DYN:SLEW 0.3, 0.5   //Set rising slope to 0.3A/us, and falling slope to 0.5A/us for   the dynamic mode |
| Query Syntax | [SOURce]:DYNamic:SLEW? |
|  | // Query the current slop of the dynamic mode |
| Return Parameter | <NR2> |

## [SOURce]:DYNamic:SLEW:RISE

[SOURce]:DYNamic:SLEW:RISE    Set the rising rate for the current in dynamic mode

| | |
|---|---|
| Command Syntax | [SOURce]:DYNamic:SLEW:RISE <NRF+> |
| Parameter | MIN~MAX \|MINimum\|MAXimum |
| Unit | A/us |
| Reset Value | MAX |
| For Example | DYN:SLEW:RISE 1<br>// Set the rising rate of the current to 1 A/us in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:SLEW:RISE?<br>// Query the rising rate of the current in dynamic mode |
| Return Parameter | <NR2> |

## [SOURce]:DYNamic:SLEW:FALL

[SOURce]:DYNamic:SLEW:FALL    Set the falling rate for the current in dynamic mode

| | |
|---|---|
| Command Syntax | [SOURce]:DYNamic:SLEW:FALL <NRF+> |
| Parameter | MIN~MAX \|MINimum\|MAXimum |
| Unit | A/us |
| Reset Value | MAX |
| For Example | DYN:SLEW:FALL 1<br>// Set the falling rate of the current to 1 A/us in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:SLEW:FALL?<br>// Query the falling rate of the current in dynamic mode |
| Return Parameter | <NR2> |

## [SOURce]:DYNamic:MODE

[SOURce]:DYNamic:MODE    Set the operation mode for the dynamic mode

| | |
|---|---|
| Command Syntax | [SOURce]:DYNamic:MODE <mode> |
| Parameter | CONTinuous \| PULSe \| TOGGle |
| Reset Value | CONTinuous |
| For Example | DYN:MODE PULS<br>// Set the operation mode to pulse mode in dynamic mode |
| Query Syntax | [SOURce]:DYNamic:MODE?<br>// Query the operation mode of the dynamic mode |
| Return Parameter | <CRD> |

## [SOURce]:DYNamic:REPeat

| [SOURce]:DYNamic:REPeat | Se the repeat times for the dynamic mode |
|---|---|
| Command Syntax | [SOURce]:DYNamic:REPeat |
| Parameter | 0~99999 \| MINimum \| MAXimum \| LOOP |
| For Example | DYNamic:REPEAT 10 |
| | // Set the repeat times to 10 for the dynamic mode |
| Query Syntax | [SOURce]:DYNamic:REPeat? |
| | // Query the repeat times of the dynamic mode |
| Return Parameter | <NR2> |

## [SOURce]:BATtery:MODE

| [SOURce]:BATtery:MODE | Set the operation mode in battery mode |
|---|---|
| Command Syntax | [SOURce]:BATtery:MODE <mode> |
| Parameter | CURRent \| RESistance \| POWer |
| Reset Value | CURRent |
| For Example | BATery:MODE CURRent |
| | // Set the operation mode to CC mode in battery mode |
| Query Syntax | [SOURce]:BATtery:MODE? |
| | // Query the operation mode in battery mode |
| Return Parameter | <CRD> |

## [SOURce]:BATtery:CURRent

| [SOURce]:BATtery:CURRent | Set the load size of CC mode in battery mode |
|---|---|
| Command Syntax | [SOURce]:BATtery:CURRent |
| Parameter | 0.01~20 \| MINimum \| MAXimum |
| Unit | A |
| Reset Value | 1 |
| For Example | BATtery:CURRent 3 |
| | // Set the load size of CC mode to 3A in battery mode |
| Query Syntax | [SOURce]:BATtery:CURRent? |
| | // Query the load size of CC mode in battery mode |
| Return Parameter | <NR2> |

## [SOURce]:BATtery:POWer

| [SOURce]:BATtery:POWer | Set the load size of CP mode in battery mode |
|---|---|
| Command Syntax | [SOURce]:BATtery:POWer |
| Parameter | 0.1~400 \| MINimum \| MAXimum |
| Unit | W |
| Reset Value | 1 |
| For Example | BATtery: POWer 3 <br> // Set the load size of CP mode to 3W in battery mode |
| Query Syntax | [SOURce]:BATtery:POWer? <br> // Query the load size of CP mode in battery mode |
| Return Parameter | <NR2> |

## [SOURce]:BATtery:RESistance

| [SOURce]:BATtery:RESistance | Set the load size of CR mode in battery mode |
|---|---|
| Command Syntax | [SOURce]:BATtery:RESistance |
| Parameter | 0.05~7500 \| MINimum \| MAXimum |
| Unit | Ohm |
| Reset Value | 1 |
| For Example | BATtery:RESistance 3 <br> // Set the load size of CR mode to 3ohm in battery mode |
| Query Syntax | [SOURce]:BATtery:RESistance? <br> // Query the load size of CR mode in battery mode |
| Return Parameter | <NR2> |

## [SOURce]:BATtery[:VOLTage]:Unloade

| [SOURce]:BATtery[:VOLTage]:Unloade | Set the cut-off size of voltage in battery mode |
|---|---|
| Command Syntax | [SOURce]:BATtery[:VOLTage]:Unloade |
| Parameter | 0.01~150 \| MINimum \| MAXimum |
| Unit | V |
| Reset Value | 1 |
| For Example | BATtery:Unloade 3       // Set the cut-off size of voltage to 3V in battery mode |
| Query Syntax | [SOURce]:BATtery[:VOLTage]:Unloade? <br> // Query the cut-off size of voltage in battery mode |
| Return Parameter | <NR2> |

## [SOURce]:BATtery:CAPAcity?

[SOURce]:BATtery:CAPAcity?          Query the battery capacity

| | |
|---|---|
| Command Syntax | [SOURce]:BATtery:CAPAcity? |
| Parameter | |
| Unit | Ah/Wh          // The unit is Ah in CC/CR mode. The unit is Wh in CP mode |
| Reset Value | 0 |
| For Example | |
| Query Syntax | [SOURce]:BATtery:CAPAcity?          // Query the battery capacity |
| Return Parameter | <NR2> |

## MEASure:VOLTage?

MEASure:VOLTage?          Read the average value of voltage

| | |
|---|---|
| Command Syntax | MEASure[:SCALar]:VOLTage[:DC]? |
| For Example | MEAS:VOLT?                    // Query the average value of voltage |
| Query Syntax | MEASure[:SCALar]:VOLTage[:DC]?<br>// Query the average value of voltage |
| Return Parameter | <NR2> |

## MEASure:CURRent?

MEASure:CURRent?          Read the average value of current

| | |
|---|---|
| Command Syntax | MEASure[:SCALar]:CURRent[:DC]? |
| For Example | MEAS:CURR?                    // Query the average value of current |
| Query Syntax | MEASure[:SCALar]:CURRent[:DC]?<br>// Query the average value of current |
| Return Parameter | <NR2> |

## MEASure:POWer?

MEASure:POWer?          Read the average value of power

| | |
|---|---|
| Command Syntax | MEASure[:SCALar]:POWer[:DC]? |
| For Example | MEAS:POWer?                    // Query the average value of power |
| Query Syntax | MEASure[:SCALar]:POWer[:DC]?    // Query the average value of power |
| Return Parameter | <NR2> |

## MEASure:RESistance?

MEASure:RESistance?    Read the equivalent resistance

| Command Syntax | MEASure[:SCALar]:RESistance[:DC]? |
|---|---|
| For Example | MEAS:RESistance?                // Query the equivalent resistance |
| Query Syntax | MEASure[:SCALar]:RESistance[:DC]? |
| | // Query the equivalent resistance |
| Return Parameter | <NR2> |

## MEASure:REAL?

MEASure:REAL?    Query the real-time measured data set

| Command Syntax | MEASure[:SCALar]:REAL[:TIME][:DC]? |
|---|---|
| For Example | MEAS:REAL?              // Query the real-time measured data set |
| Query Syntax | MEASure[:SCALar]:REAL[:TIME][:DC]? |
| | // Query the real-time measured data set |
| Return Parameter | Voltage, current, power, resistance <NR2> |

## [SOURce]:LIST:GROUP

[SOURce]:LIST:GROUP    Set the group number for the list mode

| Command Syntax | [SOURce]:LIST:GROUP |
|---|---|
| Parameter | 0~60 | MINimum | MAXimum |
| For Example | LIST:GROUP 3          // Set the group number to 3 for the list mode |
| Query Syntax | [SOURce]:LIST:GROUP? |
| | // Query the group number of the list mode |
| Return Parameter | <NR2> |

## [SOURce]:LIST:MODE

[SOURce]:LIST:MODE    Set the operation mode in list mode

| Command Syntax | [SOURce]:LIST:MODE |
|---|---|
| Parameter | CONTinuous | TRIGger | TRIGger EX| CONTinuousEX |
| For Example | LIST:MODE CONTinuous |
| | // Set the operation mode to continuous in list mode |
| Query Syntax | [SOURce]:LIST:MODE? |
| | // Query the operation mode in list mode |
| Return Parameter | <CDR> |

## [SOURce]:LIST:STEP

[SOURce]:LIST:STEP          Set the step for the list mode

| | |
|---|---|
| Command Syntax | [SOURce]:LIST:STEP |
| Parameter | 1~16 | MINimum | MAXimum |
| For Example | LIST:STEP 3                    // Set the step to 3 for list mode |
| Query Syntax | [SOURce]:LIST:STEP?        // Query the step of list mode |
| Return Parameter | <NR2> |

## [SOURce]:LIST:REPEAT

[SOURce]:LIST:REPEAT     Set the repeat times for list mode

| | |
|---|---|
| Command Syntax | [SOURce]:LIST:REPEAT |
| Parameter | 0~99999 | MINimum | MAXimum |
| For Example | LIST:REPEAT 10                    // Set the repeat times to 10 for list mode |
| Query Syntax | [SOURce]:LIST:REPEAT?          // Query the repeat times of list mode |
| Return Parameter | <NR2> |

## [SOURce]:LIST:PARAMeter:ITEM

[SOURce]:LIST:PARAMeter:ITEM      Set each parameter in list

| | |
|---|---|
| Command Syntax | [SOURce]:LIST:PARAMeter:ITEM |
| Parameter | 1~16,0~40,200~99999 ,| MINimum | MAXimum     OFF|OF |
| | CURRent | VOLTage | RESistance | POWer | OPEN | SHORT |
| For Example | LIST:PARAMeter:ITEM 0,CURR,2.0,1000,OFF,1.0, 2.0 |
| | // Set the serial number to 1, mode to CC, constant value to 2.0, timing to1000ms, check is off, the lower limit is 1.0, the upper limit is 2.0 for the list mode |
| Query Syntax | [SOURce]:LIST:PARAMeter:ITEM? |
| | // Query each parameter of list in list mode |
| Return Parameter | <> |

## [SOURce]: LIST:TEST:RESUlts?

[SOURce]:LIST:TEST:RESUlts?    Return the test result of list mode

| | |
|---|---|
| Command Syntax | [SOURce]: LIST:TEST:RESUlts? |
| Parameter | 1~16 or no parameter (return all test result) |

| For Example | LIST:TEST:RESUlts? 1<br><br>or<br><br>LIST:TEST:RESUlts? |
|---|---|
| Query Syntax | [SOURce]: LIST:TEST:RESUlts?<br>// Query each parameter of the list in the list mode |
| Return Result | 0,CURR,1.00,OFF,0.00,0.00,PASS;1,CURR,3.00,OFF,0.00,0.00,PASS;2,<br>CURR,4.00,OFF,0.00,0.00,PASS;3,VOLT,5.00,OFF,0.00,0.00,PASS; |

## [SOURce]: LIST:TEST[:STATe]?

[SOURce]:LIST:TEST[:STATe]?       Return the test state of list mode

| Command Syntax | [SOURce]: LIST:TEST[:STATe]? |
|---|---|
| Parameter | 1~16 or no parameter |
| For Example | LIST:TEST? 1 or LIST:TEST? |
| Query Syntax | [SOURce]: LIST:TEST?              // Query the test state of list mode |
| Return Result | PASS |

# CHANnel[:LOAD]   Channel Control Subsystem(only for dual-channel)

| Command Syntax | CHANnel[:LOAD] |
|---|---|
| Parameter | 1~2 or CH1~CH2 ALL (channel) |
| For Example | CHANnel 1 or CHANnel CH2 |
| Query Syntax | CHANnel[:LOAD]?<br>// Query the channel that receive the insturction |
| Return Result | 2 |

CHANnel[:LOAD]        Set the channel to receive the insturction

## CHANnel:SHORtcut[:COMMand]

CHANnel:SHORtcut[:COMMand]   Set the shortcut for channel address mode

| Command Syntax | CHANnel:SHORtcut[:COMMand] |
|---|---|
| Parameter | 0~1 or ON/OFF( power on is off by default) |
| For Example | CHANnel:SHORtcut:COMMand ON |

|  | // Turn on the shortcut of channel address mode |
|--|--|
|  | MODE CH2,RES                            // Set CH2 to CR mode |
|  | RES CH2,500                            // Set the resistance of CH2 to 500Ω |
|  | INP 2,1                                // CH2 load |
|  | CHANnel:SHORtcut:COMMand OFF |
|  | // Turn off the shortcut of channel address mode |
|  | CHANnel 1 MODE CURR             // Set CH1 to CC mode |
|  | CURR 2.5                            // Set the resistance of CH1 to 2.5A |
|  | INP 1                                // CH1 load |
| Query Syntax | CHANnel[:LOAD]?          // Query the shortcut of channel address mode |
| Return Result | 1 |

# ADDR    Device Address Subsystem

ADDR<addr>::[cmd]     Set the device address for receving the instruction

# *RST Subsystem

System soft rest (restore to the original state)

| Command Syntax | *RST |
|--|--|
| For Example | Send >*RST <NL> |
|  | No return > |
| Command Syntax | ADDR <addr>::[cmd] |
| Parameter | Addr: 1~255 device address number, cmd: SCPI control command |
| For Example | ADDR 1:: MEASure:REAL? |
|  | // Request the real-time measured data set of N0.1 load device |
|  | or |
|  | ADDR 200:: MEASure:REAL? |
|  | // Request the real-time measured data set of N0.200 load device |
| Remarks | This command is effective in the control application scenario of multiple load devices in RS485 networking. Device address is used to specific the device number. The address instruction system can be ignored if it is a single device. |

# *IDN? Subsystem

| *IDN?subsystem | Return the version number of the instrument |
| --- | --- |
| Command Syntax | *IDN? |
| Query Response | < Manufacturer>,<MODEL>,<SN>,<Revision>, |
| For Example | Send > *IDN? <NL><br>No return > UNI-TREND,UTL8211+,CDLB123060048,V1.68<NL> |

# ERRor Subsystem

| ERRor subsystem | Query the last erro message |
| --- | --- |
| Query Syntax | ERRor? |
| Query Response | Error string |
| For Example | Send > ERR?<NL><br>Return > no error.<NL> |

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