UNI-T®

Instruments.uni-trend.com

Programming Manual

UTR2810+ Series LCR Meter

V1.0

2024-07-19

Warranty and Statement

Copyright

Brand Mark 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.

1. SCPI

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that builds on existing standards such as IEEE 488.1 and IEEE 488.2. It adheres to the floating-point rules defined by the IEEE 754 standard, uses ISO 646 7-bit encoding notation (equivalent to ASCII programming), and incorporates many other standards.
This section introduces the format, symbols, parameters, and abbreviations of the SCPI command.

1.1 Introduction

This chapter introduces the RS232C command of UTR2810+ series in detail. These commands are all compliant with the SCPI.
Each command description contains the following contents:
Command name: The name corresponding to a SCPI command.
Command Syntax: The command format, including all required and optional parameters.
Query Syntax: The query format, including all required and optional parameters.
Query Respond: The returned data format of UTR2810E.

1.2 Notation Conventions and Definitions

Notations and definitions for RS232C commands.
< >: The content in angle brackets indicates the parameter of the command.
[ ]: The content in square brackets indicates an optional command that can be omitted.
{ }: Curly brackets containing several optional parameters indicate that only one parameter can be selected.

The following notations will be used in the command.
<NL>: **Line separator** (decimal 10).
Space: Single ASCII character (decimal 0-9, 11-32). For example, carriage return (decimal 13) or space (decimal 32).

1.3 Command Structure

UTR2810+ series commands are divided into two types: common commands and SCPI commands.

Common command: Defined by IEEE (Institute of Electrical and Electronic Engineers), suitable for all instruments.

SCPI command: Use a tree-like structure with three levels. The highest level is called the subsystem command. Lower- level commands within a subsystem are only valid when the subsystem command is selected. A colon ":" is used to separate high-level command from the low-level command.

Basic Rules of tree-like Structure Commands:

- Case insensitive
  For example, **LIMIT:NOMINAL <value> = limit:nominal <value> = LiMiT:NoMiNaL <value>**
- No spaces around colon (␣ indicates a space)
  For example, Incorrect: **LIMIT␣:␣NOMINAL <value>**
  Correct: **LIMIT:NOMINAL <value>**
- Abbreviated command or full spelled command
  For example, **LIMIT:NOMINAL <value> = LIM:NOM <value>**
- Add a question mark "?" after a command to form a query command.
  For example, **LIMIT:NOMINAL_C ?**

Multiple Commands in One Line:

- Use a semicolon ";" to separate multiple commands at the same level within one subsystem command.
  For example, LIMIT:NOMINAL <value>; BIN <n> <low limit>,<high limit>
- A colon ":" after a semicolon ";" indicating that the following command restarts from the top level of the command tree.
  For example, LIMIT:NOMINAL <value>;:LIMIT:BIN <n> <low limit>,<high limit>

1.4 Command Abbrevations

Every command and character parameter has at least two spelled forms: an abbreviated from and a full spelled from. In some cases, the two forms are totally different. Please follow these rules for abbreviation.

- If the word length is four letters or shorter, the abbreviated form and full spelled are the same.
- If the word length is longer than four letters and the fourth letter is a vowel, the abbreviated form is the first three letters.
  For example, **LIMIT** abbreviates to **LIM**.
  If the word length are longer than four letters and the fourth letter is a consonant, the abbreviated form is the first four letters.
  For example, **RANGE** abbreviates to **RANG**.
  FREQUENCY abbreviates to **FREQ**.

- If a phrase needs to be abbreviated, the full spelled form consists the capital letter of the first word and the entire last word. The abbreviated form is derived from the full spelled form.
  For example, a phrase "Source RESistor",
  Full spelled: **SRESISTOR**
  Abbreviated form: **SRES**

1.5 Header and Parameter

UTR2810+ series control command consists of a command header and parameters. The header can be in full spelled or abbreviated form. The full spelled is used for easy understanding, while the abbreviated form is used to improve the input efficiency. The parameter can be of two types as follows.

- **Character Data and String Data**
  Character data consists of ASCII characters. The abbreviated form is the same as the abbreviated form of command header.
  String data consists of ASCII characters enclosed in double quotes ("").
- **Numeric Data**
  Numeric data can be integer (NR1), fixed point (NR1), or floating point (NR3). The range of numeric data is ±9.9E37.
  Examples for NR1:
  123
  +123
  -123
  Examples for NR2:
  12.3
  +1.234
  -123.4
  Examples for NR3:
  12.3E+5
  123.4E-56

2. SCPI Reference Command

2.1 DISPlay Command

DISPlay:PAGE command is used to set the display mode.

DISPlay:PAGE? command is used to query and return the display mode of measured results.

Command Syntax	**DISPlay:PAGE {BINSetup,MEASurement,SYSsetup}**
Parameter	BINSetup: Sorting setup page MEASurement: Measurement display page SYSsetup: System setup page
Query Syntax	**DISPlay:PAGE BINSetup**
Query Respond	**{DIRect,PERcent,ABSolute}, <NL>**

2.2 FUNCtion Subsystem Command

FUNCtion Subsystem Tree

FUNCtion		**{L_Q,C_D,R_X,Z_RAD,G_B,Y_R,L_r}**
	:IMPedance:AUTO	**{ON,OFF}**
	:IMPedance:RANGe	**{3,10,30,100,300,1k,3k,10k,30k,100k,300k }**

FUNCtion Command

FUNCtion command is used to set the type for Parameter A (Primary Parameter) and Parameter B (Secondary Parameter).

FUNCtion? command is used to query and return the measured parameters. The primary and secondary parameter can both be selected, with a total of 42 types of measurement modes.

Several modes are listed below as examples.

Command Syntax	**FUNCtion {L_Q,C_D,R_X,Z_RAD,G_B,Y_R,L_r}**
Parameter	L_Q: Inductance _ Quality Factor C_D: Capacitance _ Dissipation R_X: Resistance _ Reactance Z_RAD: Impedance _ Rad (Arc) G_B: Conductance _ Susceptance Y_R: Admittance _ Resistance L_r: Inductance _ Phase Angle
Query Syntax	FUNCtion?
Query Respond	**{L_Q,C_D,R_X,Z_RAD,G_B,Y_R,L_r}, <NL>**

FUNCtion:IMPedance:AUTO Command

FUNCtion:IMPedance:AUTO command is used to set the automatic range of the parameter to ON or OFF.
FUNCtion:IMPedance:AUTO? command is used to query and return the range state.

Command Syntax	**FUNCtion:IMPedance:AUTO {ON,OFF}**
Parameter	ON: Automatic range is enabled.
	OFF: Automatic range is disabled.
Query Syntax	**FUNCtion:IMPedance:AUTO?**
Query Respond	**{ON,OFF}, <NL>**

FUNCtion:IMPedance:RANGe Command

FUNCtion:IMPedance:RANGe command is used to set the range number.
FUNCtion:IMPedance:RANGe? command is used to query and return the range number.

Command Syntax	**FUNCtion:IMPedance:RANGe {3,10,30,100,300,1k,3k,10k,30k,100k,300k}**
Parameter	3: 3 Ω
	10: 10 Ω
	30: 30 Ω
	100: 100 Ω
	300: 300 Ω
	1k: 1 kΩ
	3k: 3 kΩ
	10k: 10 kΩ
	30k: 30 kΩ
	100k: 100 kΩ
	300k: 300 kΩ
Query Syntax	**FUNCtion:IMPedance:RANGe?**
Query Respond	**{3,10,30,100,300,1k,3k,10k,30k,100k,300k}, <NL>**

2.3 FREQuency Command

FREQuency command is used to set the frequency for the test signal source.
FREQuency? command is used to query and return the frequency of the test signal source.

Command Syntax	**FREQuency {100,120,1k,10k}**
Parameter	100: Sets the test frequency to 100 Hz.
	120: Sets the test frequency to 120 Hz.
	1k: Sets the test frequency to 1 kHz
	10k: Sets the test frequency to 10 kHz.
Query Syntax	**FREQuency?**
Query Respond	**{100,120,1k,10k}, <NL>**

2.4 LEVel Subsystem Command

LEVel Subsystem Tree

LEVel	**:VOLTage**	**{1.0V,0.3V,0.1V}**
	:SRESistance	**{30,100}**

LEVel:VOLTage Command

LEVel:VOLTage command is used to set the output voltage for the test signal source.
LEVel:VOLTage? command is used to query and return the output voltage of the test signal source.

Command Syntax	**LEVel:VOLTage {1.0V,0.3V,0.1V}**
Parameter	1.0V: Sets the output voltage of the signal source to 1.0 V. 0.3V: Sets the output voltage of the signal source to 0.3 V. 0.1V: Sets the output voltage of the signal source to 0.1 V.
Query Syntax	**LEVel:VOLTage?**
Query Respond	**{1.0V,0.3V,0.1V}, <NL>**

LEVel:SRESistance Command

LEVel:SRESistance command is used to set the output resistance for the test signal source.
LEVel:SRESistance? command is used to query and return the output resistance of the test signal source.

Command Syntax	**LEVel:SRESistance {30,100}**
Parameter	30: Sets the output resistance of the signal source to 30 Ω. 100: Sets the output resistance of the signal source to 100 Ω.
Query Syntax	**LEVel:SRESistance?**
Query Respond	**{30,100}, <NL>**

2.5 SPEED Command

SPEED command is used to set the test speed.
SPEED? command is used to query and return the test speed settings.

Command Syntax	**SPEED {SLOW,MEDium,FAST}**
Parameter	SLOW: Slow speed, approx. 3 meas/sec. MEDium: Medium speed, approx. 6.25 meas/sec. FAST: Fast speed, approx. 20 meas/sec.
Query Syntax	**SPEED?**
Query Respond	**{SLOW,MEDIUM,FAST}, <NL>**

2.6 MODE Command

MODE command is used to set the test equivalent mode.

MODE? command is used to query and return the test equivalent mode.

Command Syntax	**MODE {SER,PAR}**
Parameter	SER: Series equivalent mode
	PAR: Parallel equivalent mode
Query Syntax	**MODE?**
Query Respond	**{SER,PAR}, <NL>**

2.7 CORRection Subsystem Command

CORRection Subsystem Tree

CORRection	**:OPEN:STATe**	**{ON,OFF}**
	:SHORt:STATe	**{ON,OFF}**
	:OPEN	
	:SHORt	

CORRection:OPEN:STATe Command

CORRection:OPEN:STATe command is used to set the open-circuit correction to ON or OFF.

CORRection:OPEN:STATe? command is used to query and return the switch state of open-circuit correction.

Command Syntax	CORRection:OPEN:STATe {ON,OFF}
Parameter	ON: Open-circuit correction is enabled.
	OFF: Open-circuit correction is disabled.
Query Syntax	CORRection:OPEN:STATe?
Query Respond	{ON,OFF} <NL>

CORRection:SHORt:STATe Command

CORRection:SHORt:STATe command is used to set the short-circuit correction to ON or OFF.

CORRection:SHORt:STATe?command is used to query and return the switch state of short-circuit correction.

Command Syntax	CORRection:SHORt:STATe {ON,OFF}
Parameter	ON: Short-circuit correction is enabled.
	OFF: Short-circuit correction is disabled.
Query Syntax	CORRection:SHORt:STATe?
Query Respond	{ON,OFF} <NL>

CORRection:OPEN Command

CORRection:OPEN command is used to execute open-circuit correction. Open-circuit correction can only be executed correctly when the correction function is ON.

Command Syntax **CORRection:OPEN**

Query Respond Open-circuit correction is executed correctly.

CORRection:SHORt Command

CORRection:SHORt command is used to execute short-circuit correction. Short-circuit correction can only be executed correctly when the correction function is ON.

Command Syntax **CORRection:SHORt**

Query Respond Short-circuit zeroing is executed correctly.

2.8 TRIGger Command

TRIGger Subsystem Tree

TRIGger	**:SOURce**	**{INT,BUS,MAN,EXT}**
***TRG(TRIGger)**	The instrument will be tested once and return the test results.	

TRIGger:SOURce Command

TRIGger:SOURce command is used to set the trigger mode.

TRIGger:SOURce? command is used to query and return the trigger mode.

Command Syntax **TRIGger:SOURce {INT,BUS,MAN,EXT}**

Parameter INTernal: Sets the trigger mode to internal trigger mode.
EXTernal: Sets the trigger mode to external trigger mode.
IMMediate: Trigger a measurement immediately.

Query Syntax **TRIGger:SOURce?**

Query Respond **{INT,BUS,MAN,EXT} <NL>**

TRIGger Command

TRIGger (*TRG) command is used to generate a measurement.

Command Syntax TRIGger or (*TRG)

Query Respond TRIGger start

2.9 FETCh? Command

FETCh? command is used to query and return the latest measured results of the primary and secondary parameters.

Query Syntax	**FETCh?**
Query Respond	**<primary>,<secondary> <NL>**

2.10 COMParator Subsystem Command

COMParator Subsystem Tree

COMParator	**:STATAe**	**{ON,OFF}**
	:STATBe	**{ON,OFF}**
	:MODE	**{ABS,PER,SEQ}**
	:TOLerance:NOMinal	**<value>**
	:TOLerance:BIN<n>	**<low limit>,<high limit>**
	:SEQuence:BIN	**<value>,<value>,<value>,<value>**
	:SLIMit	**<value>,<value>**

COMParator:STATAe Command

COMParator:STATAe command is used to set the comparator function to ON or OFF for the primary parameter.

COMParator:STATAe? command is used to query and return the comparator state of the primary parameter.

Command Syntax	**COMParator:STATAe {ON,OFF}**
Parameter	ON: The comparator function of the primary parameter is enabled. OFF: The comparator function of the primary parameter is disabled.
Query Syntax	**COMParator:STATAe?**
Query Respond	**{ON,OFF}, <NL>**

COMParator:STATBe Command

COMParator:STATBe command is used to set the comparator function to ON or OFF for the secondary parameter.

COMParator:STATBe? command is used to query and return the comparator state of the secondary parameter.

Command Syntax	**COMParator:STATBe {ON,OFF}**
Parameter	ON: The comparator function of the secondary parameter is enabled. OFF: The comparator function of the secondary parameter is disabled.
Query Syntax	**COMParator:STATBe?**

Query Respond	**{ON,OFF}, <NL>**

COMParator:MODE Command

COMParator:MODE command is used to set the comparator mode.
COMParator:MODE? command is used to query and return the comparator mode.

Command Syntax	**COMParator:MODE {ABS,PER,SEQ}**
Parameter	ABS: Absolute deviation mode PER: Percentage deviation mode SEQ: Sequence mode
Query Syntax	**COMParator:MODE?**
Query Respond	**{ABS,PER,SEQ}, <NL>**

COMParator:TOLerance:NOMinal Command

COMParator:TOLerance:NOMinal command is used to set the nominal value. The comparator uses the nominal value to calculate the absolute deviation and percentage deviation.
COMParator:TOLerance:NOMinal? command is used to query and return the nominal value.

Command Syntax	**COMParator:TOLerance:NOMinal <value>**
Parameter	<value> represents the nominal value in NR1, NR2, or NR3 formats.
Query Syntax	**COMParator:TOLerance:NOMinal?**
Query Respond	**<NR3> <NL>**

COMParator:TOLerance:BIN<n> Command

COMParator:TOLerance:BIN<n> command is used to set the upper and lower limit for BIN<n>.
COMParator:TOLerance:BIN<n>? command is used to query and return the upper and lower limit of BIN<n>.

Command Syntax	**COMParator:TOLerance:**BIN<n> <low limit>,<high limit>
Parameter	<n> represents 1 to 3 (NR1), BIN number. <low limit> represents the nominal value in NR1, NR2, or NR3 formats. <high limit> represents the nominal value in NR1, NR2, or NR3 formats.
Query Syntax	**COMParator:TOLerance:BIN<n>?**
Query Respond	**<NR3>,<NR3> <NL>**

COMParator:SEQuence:BIN Command

COMParator:SEQuence:BIN command is used to set the limit value for the sequence mode.
COMParator:SEQuence:BIN?command is used to query and return the limit value of the sequence mode.

Command Syntax	**COMParator:SEQuence:BIN <value>,<value>,<value>,<value>**
Parameter	<value> represents the nominal value in NR1, NR2, or NR3 formats.
Query Syntax	**COMParator:SEQuence:BIN?**
Query Respond	**<NR3>,<NR3>,<NR3>,<NR3> <NL>**

COMParator:SLIMit Command

COMParator:SLIMit command is used to set the limit value for the secondary parameter.
COMParator:SLIMit?command is used to query and return the limit value of the secondary parameter.

Command Syntax	**COMParator:SLIMit <value>,<value>**
Parameter	<value> represents the nominal value in NR1, NR2, or NR3 formats.
Query Syntax	**COMParator:SLIMit?**
Query Respond	**<NR3>,<NR3> <NL>**

2.10 *IDN? Command

*IDN? command is used to query and return the instrument version number.

Query Syntax	**IDN?**
Query Respond	**< Manufacturer>,<MODEL>,<Revision><NL>**
For Example	***IDN?<NL>** // Query the instrument version number.
Query Respond	**UNIT,UTR2810E+, CDB2024140001,REVA2.7<NL>**