



UNI-T UTE9802 Plus Smart Digital Power Meter User Manual

[Home](#) » [UNI-T](#) » UNI-T UTE9802 Plus Smart Digital Power Meter User Manual 



UTE9802 Plus Smart Digital Power Meter User Manual



UTE9802+ Smart Digital Power Meter SCPI Programming Manual

REV 00
2023.2

Contents

- 1 UTE9802 Plus Smart Digital Power Meter
- 2 Chapter 1 SCPI
- 3 Chapter 2 Communication Interface and Setting
- 4 Chapter 3 Instruction
- 5 Chapter 4 Acquire Newest Measurement Data
- 6 Documents / Resources

UTE9802 Plus Smart Digital Power Meter

Copyright

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

Brand Information

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

Software Version

00.00.01

Software upgrade may have some change and add more function, please subscribe UNI-T website to get the new version or contact **UNI-T**.

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.

Chapter 1 SCPI

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.

Instruction Format

Command is consisting of a keyword, separator, parameter domain and end mark. Take the following command as an example.

:VOLTage:RANGe 150

Voltage, Ranged is keyword, “.” and blank is separator, “150” is parameter (some commands have multiple parameters and separated by “,”), the line separator or carriage return after the command is the end mark. For the convenience of description, the following conventions are adopted for subsequent symbols.

- Square Brackets “[]”

The contents in square brackets (command keywords) can be omitted.

- Braces “{ }”

It represents the parameter in command string.

- Angle Braces “< >”

The parameter enclosed in the angle brackets must be a numerical parameter.

- Vertical Bar “|”

It is used to separate multiple parameters.

- End Mark: line separator <LF> 0x0A or carriage return <CR> 0x0D

Parameter Description

The data type of programming parameters include numeric, character and Boolean type. Regardless of the type, it is expressed as an ASCII. For details, see the following table.

Symbol	Meaning	Example
<NR1>	Integer	123, 0123
<NR2>	Fixed Floating point number	123., 12.3, 0.123, .123
<NR3>	Floating point number	123, 12.3, 123E+3
<NRF>	It may be <NR1>, <NR2> or <NR3>	
<Boolean>	Boolean data	0 1 ON OFF

Shorthand Rule

All the commands are case-insensitive. The commands can be all input in uppercase letters or in lowercase letters. For abbreviations, it should enter all the uppercase letters that exist in the command syntax.

Chapter 2 Communication Interface and Setting

The detailed description can refer to “Chapter 6 Communication Setting” and “Chapter 8 Communication Interface” in UTE9802+ User’s Manual.

Chapter 3 Instruction

3.1 Instruction List

Instruction	Function
*IDN?	Query the instrument information.
*RST	Restore to the factory setting.
*STB?	Query status byte register.
*SAV	Save the current setting into nonvolatile memory for next time use.
:HOLD	Turn on or off hold mode.
:MODE	Set the measurement mode of voltage/current.
:VOLTage:RANGe	Set the voltage range.
:VOLTage:AUTO	Turn on or off auto range of voltage.
:CURRent:RANGe	Set the current range.
:CURRent:AUTO	Turn on or off auto range of current.
:Rate	Set update rate.
:Averaging	Set the average switch and average count.
:Mute	Turn on or off mute key.
:ALARm:FLAG?	Query the alarm state.
:ALARm:CURRent:HIGH	Set the upper limit of current alarm.
:ALARm:CURRent:LOW	Set the lower limit of current alarm.
:ALARm:POWer:HIGH	Set the upper limit of power alarm.
:ALARm:POWer:LOW	Set the lower limit of power alarm.
:Alarm: Time	Set alarm delay.
:Update: Count?	Query the current update count.
:MEASure:FREQuency:VOLTage?	Query the measured frequency of voltage.

:MEASure:VOLTage?	Query the currently measured value of voltage.
:MEASure:CURRent?	Query the currently measured value of current.
:MEASure:POWer:ACTive?	Query the currently measured value of power.
:MEASure:PFACtor?	Query the currently measured value of power factor.
:SYSTem:ERRor?	Query error information.
:LOCK	Set the lock key state on the front panel.

3.2 Instruction Parsing

*IDN?

Function Query the instrument information.

Syntax *IDN?

Example *IDN?

-> UNI-T,UTE9802+,012345678,F1.02

Description

The return format of instrument information is <manufacturer>,<model>,<serial number>,< firmware version>.

***RST**

Function Restore to the factory setting.

Syntax *RST

Example *RST

Description

Except communication configuration parameter instruction type, baud rate, address , other configuration parameter will restore to the factory setting.

***STB?**

Function Query status byte register.

Syntax *STB?

Example *STB? -> 4

Description If returned value is 4, it represents the status byte register set to 00000100; it means the error queue is not empty, which also means an error has been generated.

***SAV**

Function Save the current setting into nonvolatile memory for next time use.

Syntax *SAV

Example *SAV

:HOLD

Function Turn on/off hold mode.

Syntax :HOLD {<Boolean>}

:HOLD?

Example :HOLD OFF

:HOLD? -> 0

:MODE

Function Set the measurement mode of voltage/current.

Syntax :MODE {AC|ACDC|DC}

:MODE?

AC, ACDC = (RMS) DC

Example :MODE ACDC

:MODE? -> ACDC

:VOLTage:RANGe

Function Set the voltage range.

Syntax :VOLTage:RANGe {<Voltage>}

:VOLTage:RANGe?

<Voltage> = 75,150,300,600

Example :VOLTage:RANGe 150

:VOLTage:RANGe? -> 150

:VOLTage:AUTO

Function Turn on or off auto range of voltage.

Syntax :VOLTage:AUTO {<Boolean>}

:VOLTage:AUTO?

Example :VOLTage:AUTO 1

:VOLTage:AUTO? -> 1

:CURRent:RANGe

Function Set the current range.

Syntax :CURRent:RANGe {<Current>}

:CURRent:RANGe?

<Current> = 0.5, 2, 8, 20

Example :CURRent:RANGe 2

:CURRent:RANGe? -> 2

:CURRent:AUTO

Function Turn on or off auto range of current.

Syntax :CURRent:AUTo {<Boolean>}

:CURRent:AUTo?

Example :CURRent:AUTo 1

:CURRent:AUTo? -> 1

:RATE

Function Set update rate.

Syntax :RATE {<Time>}

:RATE?

<Time> = 0.1,0.25,0.5,1,2,5

Example :RATE 0.25

:RATE? -> 0.25

:AVERaging

Function Set the average switch and average count.

Syntax :AVERaging {<Average>}

:AVERaging?

<Average> = OFF,8,16,32,64

·OFF = Average is turned off.

·8,16,32,64 = Average is turned on and it represents the average count.

Example :AVERaging 16

:AVERaging? -> 16

|:MUTe

Function Turn on or off mute key.

Syntax :MUTe {<Boolean>}

:MUTe?

Example :MUTe 1

:MUTe? ->1

:ALARm:FLAG?

Function Query the alarm state.

Syntax :ALARm:FLAG? {<Type>,<State>}

<Type> = CURRENT,POWER

CURRENT = current; POWER = power

<State> = DISABLE, WAITING, RUNNING, OK, LOW, HIGH

·DISABLE = the test is forbidden;

·WAITING = wait for connect to the load;

·RUNNING = testing;

·OK = the test is completed, the test result within the lower and upper limit;

·LOW = the test is completed, the test result is below the lower limit;

·HIGH = the test is completed, the test result is higher than the upper limit.

Example :ALARm:FLAG? CURRENT -> RUNNING

:ALARm:CURRent:HIGH

Function Set the upper limit of current alarm.

Syntax :ALARm:CURRent:HIGH {<NRf>}

:ALARm:CURRent:HIGH?

Example :ALARm:CURRent:HIGH 10.1

:ALARm:CURRent:HIGH? -> 10.1

:ALARm:CURRent:LOW

Function Set the lower limit of current alarm.

Syntax :ALARm:CURRent:LOW {<NRf>}

:ALARm:CURRent:LOW?

Example :ALARm:CURRent:LOW 1.1

:ALARm:CURRent:LOW? -> 1.1

:ALARm:POWer:HIGH

Function Set the upper limit of power alarm.

Syntax :ALARm:POWer:HIGH {<NRf>}

:ALARm:POWer:HIGH?

Example :ALARm:POWer:HIGH 1000.1

:ALARm:POWer:HIGH? -> 1000.1

:ALARm:POWer:LOW

Function Set the lower limit of power alarm.

Syntax :ALARm:POWer:LOW {<NRf>}

:ALARm:POWer:LOW?

Example :ALARm:POWer:LOW 10.1

:ALARm:POWer:LOW -> 10.1

:ALARm:TIME

Function Set alarm delay.

Syntax :ALARm:TIME {<NRf>}

:ALARm:TIME?

Example :ALARm:TIME 20.2

:ALARm:TIME? -> 20.2

:UPDAte:COUNt?

Function Query the current update count.

Syntax :UPDAte:COUNt?

Example :UPDAte:COUNt? -> 763

Description Each time the data is updated, the number of updates will be increased by one. By detecting the difference in the number of updates before and after, it can determine whether the data update event occurs, so as to obtain the latest updated data.

:MEASure:FREQuency:VOLTage

Function Query the measured frequency of voltage.

Syntax :MEASure:FREQuency:VOLTage?

Example :MEASure:FREQuency:VOLTage? -> 50.00

:MEASure:VOLTage?

Function Query the currently measured value of voltage.

Syntax :MEASure:VOLTage?

Example :MEASure:VOLTage? -> 110.36

:MEASure:CURREnt?

Function Query the currently measured value of current.

Syntax :MEASure:CURREnt?

Example :MEASure:CURREnt? -> 10.23

:MEASure:POWer:ACTive?

Function Query the currently measured value of power.

Syntax :MEASure:POWer:ACTive?

Example :MEASure:POWer:ACTive? -> 30.5

:MEASure:PFACTOR?

Function Query the currently measured value of power factor.

Syntax :MEASure:PFACTOR?

Example :MEASure:PFACTOR? -> 0.519

:SYSTem:ERRor?

Function Query the last error code and information.

Syntax :SYSTem:ERRor?

Example :SYSTem:ERRor? -> -113,"Undefined header"

Description If there is no error, then it return 0,"No error"

:LOCK

Function Set the lock key state on the front panel.

Syntax :LOCK {<Boolean>}

:LOCK?

Example :LOCK 1

:LOCK? -> 1

Chapter 4 Acquire Newest Measurement Data

The measurement data will in breaks when in auto range or UTE9802+ is reconfigured, the acquired data is "nan" via ":MEASure:####:###?". If user want to acquire the newest measurement data, it need to exit break state and then to acquire the data. By detecting the difference in the number of updates before and after, it can determine whether the data update event occurs, so as to obtain the latest updated data. The specific method as follows.

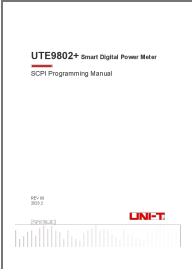

:UPDAte:COUNt? -> 101

:UPDAte:COUNt? -> 101

...
:UPDAtE:COUnT? -> 102 # data update event occurs
:MEASure:VOLTage? -> 110.36
:MEASure:CURRent? -> 10.23
...



Documents / Resources

	<p>UNI-T UTE9802 Plus Smart Digital Power Meter [pdf] User Manual UTE9800 Plus, UTE9802 Plus, UTE9802 Plus Smart Digital Power Meter, UTE9802 Plus, Smart Digital Power Meter, Digital Power Meter, Power Meter, Meter</p>
	<p>UNI-T UTE9802 plus Smart Digital Power Meter [pdf] User Manual UTE9800, UTE9802, UTE9802 plus Smart Digital Power Meter, UTE9802 plus, UTE9802 plus Power Meter, Smart Digital Power Meter, Power Meter, Smart Meter, Power Meter</p>