UNI-T

UT3550 Battery Tester Programming Manual

(SCPI)-REV.1.0

Feb, 2023

UNI-TREND TECHNOLOGY (China) Co., Ltd.

Warranty and Statement

Copyright

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**.

1. Remote Communication

This instrument uses Type-C interface (standard configuration) to communicate with computer to fulfill all the function. Via the standard SCPI command, user can program other collection system to suit your own needs.

1.1 SCPI Language

The instrument uses SCPI-Standard Commands for Programmable Instruments. It also called TMSL-Test and Measurement System Language, which is made by Agilent Technologies IEEE488.2. It has been widely used by manufacturers of test equipment.

1.2 FUNCtion Subsystem

FUNCtion	:RANGe	{Range name, max, min}	
		:MODE	{AUTO,HOLD,NOMinal}
	:RATE	{SLOW,MED,FAST, ULTRA}	

1.2.1 FUNCtion:RANGe

FUNC:RANG use to set range mode and range name.

Command syntax:	**FUNCtion:RANGe {<Range name>,min,max}**
Parameter:	<Range name> includes **0~4** **min** the minimum range=**0** **max** the maximum range=**4**
For example:	send> **FUNC:RANG 3**<NL> //switch to 3 range (300mΩ)
Query syntax:	**FUNC:RANG?**
Query response:	**Range name0~4**
For example:	send> **FUNC:RANGE?** <NL> return > **5**<NL>

1.2.2 FUNCtion:RANGe:MODE

FUNC:RANG:MODE use to switch range mode.

Command syntax:	**FUNCtion:RANGe:MODE {AUTO,HOLD,NOMinal}**
For example:	send> **FUNC:RANG:MODE NOM**<NL> //switch to nominal range
Query syntax:	**FUNC:RANG:MODE?**
Query response:	**{AUTO,HOLD,NOM}**

1.2.3 FUNCtion:TIME

FUNC:TIME use to set timing sampling.

Command syntax:	**FUNCtion:TIME <time (s)>**
For example:	send> **FUNC:TIME 5**<NL> // set it as 5s
Query syntax:	**FUNC:TIME?**
Query response:	**5**

1.3 COMParator Subsystem

Use COMParator subsystem set the parameter of comparator, it will saved in system to use when the instrument is power on.

COMP subsystem is use to set the parameter of comparator.

Figure 1-1 COMParator Subsystem Tree

COMParator	**:BEEP**	**{OFF,GD,NG}**	
	:RMODe	**{OFF,SEQ,PER,ABS}**	
	:VMODe	**{OFF,SEQ,PER,ABS}**	
	:TOLerance	**RNOMinal**	**<float>**
		VNOMinal	**<float>**
		RLIMIT(RLMT)	**<LOWER,UPPER>**
		VLIMIT(VLMT)	**<LOWER,UPPER>**

1.3.1 RES: LMT:RMODe

RES: LMT:RMODe use to set the mode of resistance comparator.

Command syntax:	RES: LMT:**RMODEe {ABS,PER,SEQ}**
Parameter:	**{OFF,ABS,PER,SEQ}** ABS absolute value sorting mode PER percentage sorting mode SEQ sorting mode
For example:	send> RES: LMT:**RMOD SEQ**<NL> // turn on comaparator and set SEQ sorting mode
Query syntax:	RES: LMT: **RMOD?**
Query response:	**{ABS,PER,SEQ}**

1.3.2 VOLT:LMT:VMODe

VOLT: LMT:VMODe use to set the mode of voltage comparator.

Command syntax:	VOLT: LMT:**VMODEe {OFF,ABS,PER,SEQ}**
Parameter:	**{ABS,PER,SEQ}** ABS absolute value sorting mode PER percentage sorting mode SEQ sorting mode
For example:	send> VOLT: LMT:**VMOD SEQ**<NL> //

	turn on comaparator and set SEQ sorting mode
Query syntax:	VOLT: LMT: **VMOD?**
Query response:	**{ABS,PER,SEQ}**

1.3.3 COMParator:BEEP

COMP:BEEP use to enable beep function.

Command syntax:	**COMParator:BEEP {OFF,OK,NG}**
For example:	send> **COMP:BEEP OK**<NL> //qualified beep
Query syntax:	**COMP:BEEP?**
Query response:	**{OFF,OK,NG}**

1.3.4 RES: LMT:NOMinal

RES: LMT:NOM use to set the nominal value of resistance.

Command syntax:	RES: LMT:**NOM <float>**
For example:	send> RES: LMT:**NOM 1m** // nominal value set to 1mΩ send> RES: LMT:**NOM 1E-3** // nominal value set to 1mΩ send> RES: LMT:**NOM 1000** // nominal value set to 1mΩ
Query syntax:	RES: LMT:**NOM?**
Query response:	**<scifloat>**
For example:	send> RES: LMT:**NOM?** <NL> return > **1.0000E-03**<NL> // nominal value set to 1mΩ

1.3.5 VOLT: LMT:NOMinal

VOLT:LMT:NOM use to set the nominal value of voltage.

Command syntax:	VOLT: LMT:**NOM <float>**
For example:	send> VOLT: LMT:**NOM 1.23** // nominal value set to 1.23V send> VOLT: LMT:**NOM 50** // nominal value set to 50V
Query syntax:	VOLT: LMT:**NOM?**
Query response:	**<scifloat>**
For example:	send> VOLT: LMT: **NOM?** <NL> return > **1.0000E+00**<NL> // nominal value set to 1V

1.4 TRIGger Subsystem

Figure 1-2 TRIGger Subsystem Tree

TRIGger	**[:IMMediate]**	
	:SOURce	**{INT,MAN,EXT,BUS}**

TRG	

TRIGger use to set trigger source and produce a once trigger.

1.4.1 TRIGger[:IMMediate]

When TRIG[:IMM] set at BUS, it will produce a once trigger, but it will not return trigger data.

Use TRG command to return data.

Command syntax:	**TRIGger[IMMediate]**
For example:	send> **TRIG**<NL> //stop when test once

1.4.2 TRIGger:SOURce

TRIG:SOUR use to set trigger source.

Command syntax:	**TRIGger:SOURce {INT,MAN,EXT,BUS}**
For example:	send> **TRIG:SOUR BUS**<NL> // set it as BUS mode
Query syntax:	**TRIG:SOUR?**
Query response:	**<INT,MAN,EXT,BUS>**

1.4.3 TRIG

When TRIG set at BUS mode, it will produce a once trigger and return the data of trigger test.

Command syntax:	**TRIG**

For example:	send> **TRIG**<NL>
	//the instrument make a one test and return test data
	return > **+9.9651e+01,in,+0.0000e+00,ng** <NL>
	//resistance value, resistance position, voltage value, voltage position

1.5 FETCh? Subsystem

FETCh? use to acquire test data. Before use this command, it should set 【Result Send】field to 【FETCH】in the <SYSTEM CONIFG> page.

FETCh? Command will return test data.

Figure 1-3 FETCh? Subsystem Tree

FETCh?	

Query syntax:	**FETCh?**
Query response:	**<scifloat>,{BIN 00,BIN 09}** **BIN 00** presents unqulified
For example:	send> **FETCH?** <NL> return > **+9.9651e+01,in,+0.0000e+00,ng** <NL> //resistance value, resistance position, voltage value, voltage position

1.6 CORR Subsystem

CORR subsystem use to execute a once short circuit zeroed calibration.

Figure 1-4 CORR Subsystem Tree

CORR	:SHOR	

1.6.1 CORR:SHOR

Query syntax:	CORR:SHOR
For example:	send> CORR:SHOR<NL> return > Short Clear Zero Start. <NL> return > PASS<NL>
Note:	Befor send command，please do short circuit of test terminal.

1.7 SYS Subsystem

SYS subsystem use to set the parameter of system.

Setting data by SYS subsystem will not saved in the instrument.

Figure 1-5 SYS Subsystem Tree

SYS	:LANGuage	{ENGLISH,CHINESE,EN,CN}
	:SENDmode	{FETCH,AUTO}

1.7.1 SYS:LANGuage

The lanague setting of the instrument.

Command syntax:	SYS:LANGuage {ENGLISH,CHINESE,EN,CN}
For example:	send> SYS:LANG EN<NL> //set to English display
Query syntax:	SYS:LANG?
Query response:	{ENGLISH,CHINESE}

1.8 IDN? Subsystem

Figure 1-6 IDN? Subsystem Tree

IDN?	

IDN? subsystem use to return the version number of the instrument.

Query syntax:	IDN?
Query response:	<MODEL>,<Revision>,<SN>,< Manufacturer>
For example:	send> IDN? <NL> return > AT526,REV C1.0,0000000,Applent Instruments<NL>