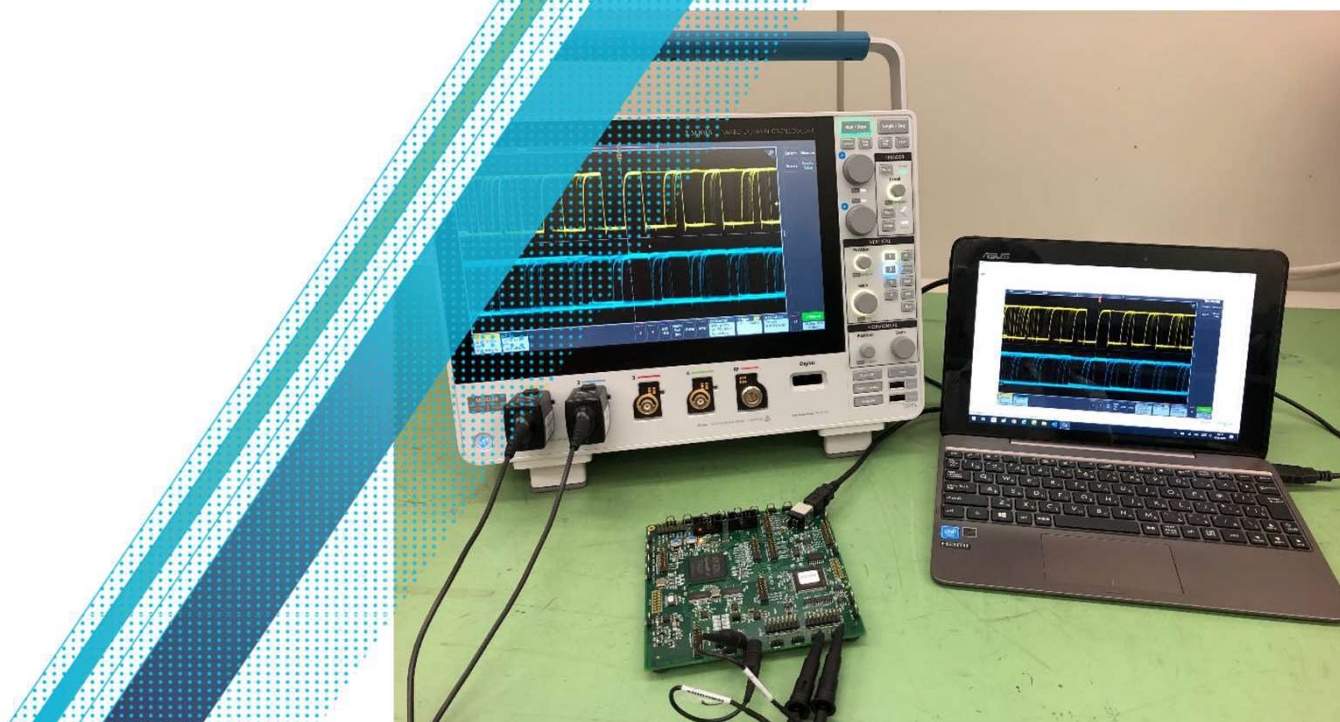


## 基于源表和纳伏表

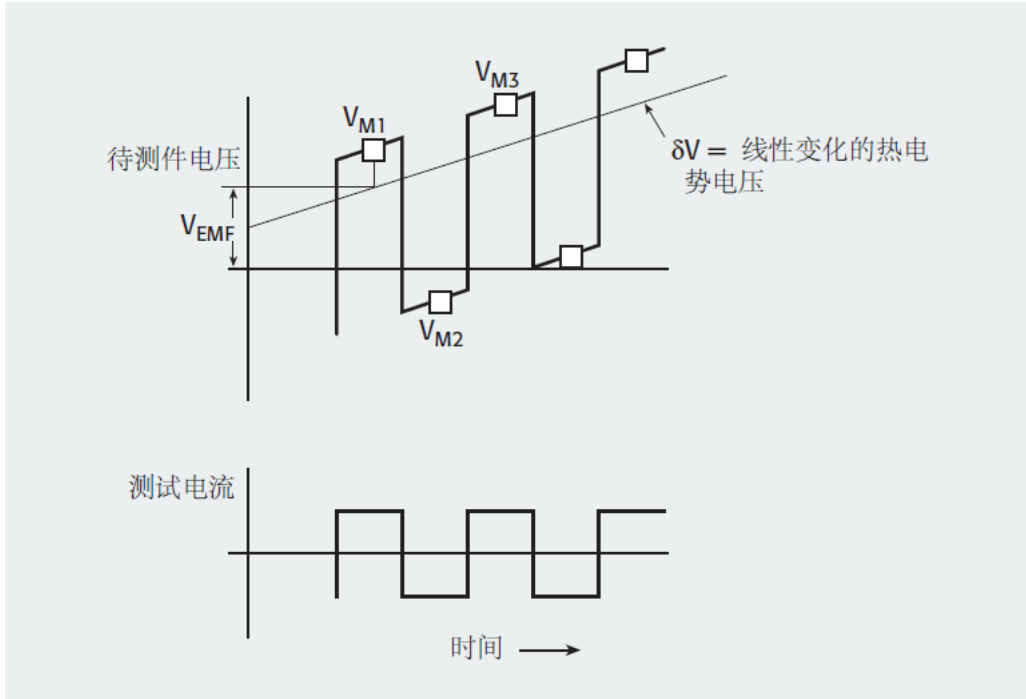
### 使用Delta模式进行超低电阻测量设置方法



# Delta模式介绍

进行超低电阻测试时，热电动势(热电电压)是常见的误差来源之一，热电动势是当电路的不同部分处在不同的温度之下，或者由不同材料制成的导体互相接触时所产生的电压。举例来讲，由实验室温度波动或敏感电路附近的通风引起的测试电路中的温度变化可以产生几微伏的热电动势，进而影响超低电阻的测量结果。

消除热电动势对超低电阻测量影响常见的方法之一是Delta模式，下图和以下计算公式解释了Delta模式的具体内容



每次电流极性改变时，测量电压( $V_{M1}$ ,  $V_{M2}$ ,  $V_{M3}$ 等)。每个电压测量包括恒定的热电压偏移量( $V_{EMF}$ )和线性变化的电压偏移量( $\delta V$ )。热电压漂移可以近似为短时间内的线性函数，因此电压随时间的变化率( $\delta V$ )也可以视为常数。前三种电压测量包括以下电压：

$$V_{M1} = V_1 + V_{EMF}$$

$$V_{M2} = V_2 + V_{EMF} + \delta V$$

$$V_{M3} = V_3 + V_{EMF} + 2 \delta V$$

其中： $V_{M1}$ 、 $V_{M2}$ 、 $V_{M3}$ 为电压测量值

假定  $V_{M1}$  是在时间= 0 时取的

$V_1$ 、 $V_2$  和  $V_3$  是被测单元由于施加电流而产生的压降

$V_{EMF}$  是  $V_{M1}$  测量时恒定的热电动势偏移量

$\delta V$  是热电动势的变化

通过三种电压测量的数学计算，可以消除热电动势电压偏移 ( $V_{EMF}$ ) 项和热电动势电压变化 ( $\delta V$ ) 项。首先，取前两个电压测量值之差的一半，称之为  $V_A$ ：

$$V_A = \frac{V_{M1} - V_{M2}}{2} = \frac{(V_1 + V_{EMF}) - (V_2 + V_{EMF} + \delta V)}{2} = \frac{(V_1 - V_2)}{2} - \frac{\delta V}{2}$$

然后，取第二次 ( $V_{M2}$ ) 和第三次 ( $V_{M3}$ ) 电压测量差值的一半，将这个项称为  $V_B$ ：

$$V_B = \frac{V_{M3} - V_{M2}}{2} = \frac{(V_3 + V_{EMF} + 2\delta V) - (V_2 + V_{EMF} + \delta V)}{2} = \frac{(V_3 - V_2)}{2} - \frac{\delta V}{2}$$

$V_A$  和  $V_B$  都受到热电动势漂移的影响，但对  $V_A$  和  $V_B$  的影响大小相等，方向相反。最终的电压读数是  $V_A$  和  $V_B$  的平均值，计算公式如下：

$$V_{Final} = \frac{V_A + V_B}{2} = \frac{(V_1 + V_3 - 2V_2)}{4}$$

注意，在最后的电压计算中， $V_{EMF}$  和  $\delta V$  这两项都被抵消了。

在Delta方法中，每个数据点是三个电压读数的移动平均值。这种对电压测量值的额外平均意味着，Delta法得到的数据比使用电流反转法计算时得到的数据具有更低的噪声，即使这两组数据是在同一时间段内获得的。Delta方法的成功取决于热漂移的线性近似，这必须在短时间内进行。成功地补偿热电动势电压的变化，意味着测量周期时间必须快于待测件的热时间常数。因此，要使得Delta方法成功，必须使用较快的电流源和纳伏表。

Keithley 6220系列电流源和2182A纳伏表是最常见的使用Delta模式进行超低电阻测量的配置，但是6220系列电流源的电流输出最大是100mA，相对比较小，如果需要进行更大的电流，或者没有6220系列电流源，可以使用Keithley源测量单元(源表)替代6220系列电流源。

## 正文

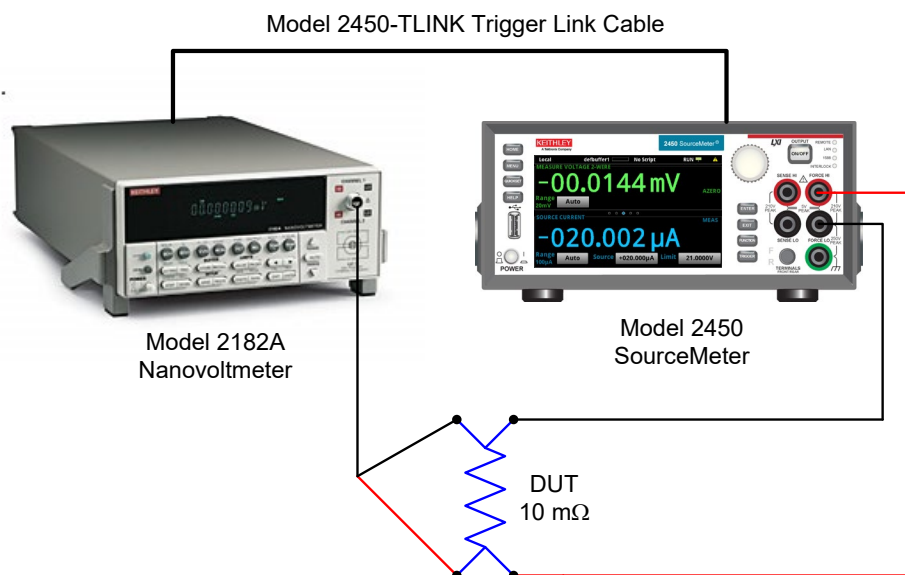
本文主要介绍以2450系列触摸屏源表为例介绍和2182A纳伏表进行Delta测试的设置方法，此设置方法同样适用于2460源表和2461源表，具体内容如下：

- 硬件连接
- 源表设置
- 纳伏表设置
- 软件程控

# 1. 硬件连接

## 1) 仪器间通信

仪器间触发通信，使用2450-TLINK触发线，连接2450源表的DIGITAL I/O 接口和2182A的TRIGGER LINK接口



## 2) 待测件连接

可以使用2182A标配线缆和2450前面板香蕉插头线的红色和黑色夹子连接待测件，具体方式，可参考上图。

# 2. 纳伏表设置

- 1) 点击RESER选择FACT, 将纳伏表恢复默认设置
- 2) 点击RATE调整PLC设置，多次点击RATE知道显示5PLC
- 3) 依次点击SHIFT和V1/V2键，选择Delta模式
- 4) 点击EXT TRIG键选择外部触发模式，设置成功后，仪器会进入等待外部触发模式，面板显示会变成一行破折号-----

# 3. 源表设置

- 1) 依次点击MENU ->Manage -> System Reset将仪器恢复默认设置，然后点击Home键回到主界面
- 2) 点击FUNCTION键，选择Source Current and Measure Voltage功能
- 3) 依次点击MENU -> Measure Settings, 将NPLCs设为0.01
- 4) 在仪器主机面，将SOURCE设置到需要的测试电流i
- 5) 依次点击MENU -> Source下面的: Config List -> Select Config List ->Create New, 名称delta然后确认
- 6) 点击add settings
- 7) 点击HOME回到主界面，将电流值设置为-i，依次MENU -> Source下面的: Config List,

点击Add Settings,选择Append,然后确认,点击HOME返回主界面

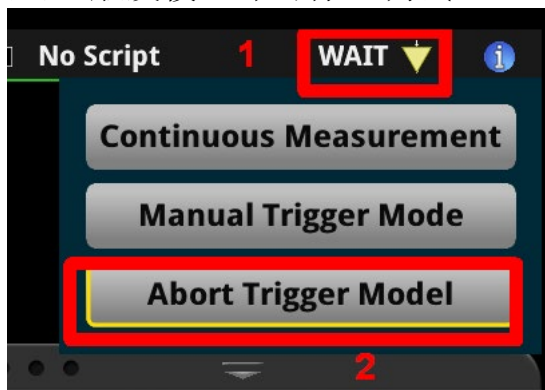
- 8) 依次点击MENU -> Trigger下面的Templates,选择LogicTrigger,点击HOME返回主界面
- 9) 依次点击MENU -> Trigger下面的Configure,点击Buffer Clear,点击Insert After,选择Action-Config List Recall->Source List->delta,添加第一个source值
- 10) 点击刚才添加的Config Recall,点击Insert After,选择Action->Source Output
- 11) 点击Measure/Digitize,点击Insert After,选择Action->Config List Next->Source List->delta,添加第二个source值
- 12) 如果需要无限重复测试,可以点击Loop Counter,点击Insert After->Branch->Always,branch back to Block 选择Wait
- 13) 点击MENU->Scripts下面的Save Setup->Create,输入名称delta script,以保存设置

## 4.开始测试

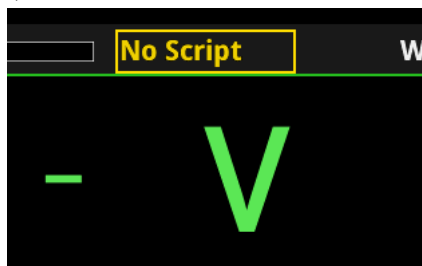
- 1) 点击源表的Trigger键
- 2) 点击2182的TRIG键,2182显示的是平均电压,电源除以设置的电流值即可得到电压值

## 5.停止测试

- 1) 点击下图红框位置,确认终止触发模型即可停止测试



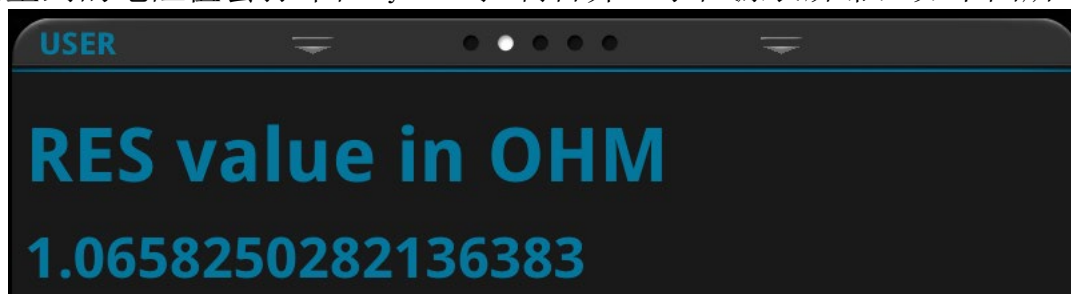
- 2) 出现开始测试: 点击下图中No Script,选择之前保存的delta script确定,点击源表Trigger键和2182 TRIG键即可



## 6. 软件程控

源表的USB、GPIB和LAN口，2182A有RS-232和GPIB口可以跟上位机通信，文末附有基于PyVisa通信的简单程例，具体环境配置可参考Python程控入门指南

测试结束，测量到的电阻值会打印在Python控制台并显示在源表屏幕，如下图所示



## 附录

### 附录一:

```
#-----  
# File : KI_2460_2182A_Delta_Mode_SCPI.py  
#  
# Purpose: Example using 2460 in SCPI mode and 2182A for triggered delta mode,  
# 2450-TLINK cable that connects pin1 to pin1 and pin2 to pin2 on the DB9 of 2460 and  
# TLINK of 2182A  
#  
# Author: LHT  
#  
# Created Date : 3/3/2021  
#  
# Development Environment: Python 3.7, PyVisa 1.11, NI-VISA 20, PyCharm, Windows  
# 10  
#  
# Compatible Interfaces: USB, Ethernet, GPIB  
#  
# Compatible Instruments:2450, 2460, 2461  
#  
# The following example is provided "AS IS" with no support or warranty.  
#-----  
"""*****  
This script is designed to be simple and easy to follow. Due to  
its simplicity it does not include anything in the way of parameter and  
error checking. Passing invalid values to any of the functions are  
very likely to create errors. Note: that no error generated should  
cause damage to the instrument. However, inappropriate use could cause  
damage to any device under test.  
  
The user of this script uses it at their own risk!  
*****"""  
  
import pyvisa #pip install pyvisa  
import time  
import numpy as np  
  
smu_visa_address = 'USB0::0x05E6::0x2460::04064306::INSTR'  
voltmeter_visa_address = 'GPIB0::7::INSTR'
```

```

rm = pyvisa.ResourceManager("C:\\Windows\\System32\\visa64.dll")
smu = rm.open_resource(smu_visa_address)
voltmeter = rm.open_resource(voltmeter_visa_address)

# setup 2182A
voltmeter_cmds = ["*CLS",
                  ":SYSTEM:PRESET",
                  ":STATUS:PRESET",
                  ":TRACE:CLEAR",
                  ":SENSE:FUNC \\ 'VOLT\\'",
                  ":SYST:FAZ OFF",
                  ":SYST:AZERO:STATE OFF",
                  ":SYST:LSYNC OFF",
                  ":SENSE:VOLT:CHAN1:DFIL:STATE OFF",
                  "SENSE:VOLT:DC:NPLC 5",
                  ":SENSE:VOLT:DC:RANG:AUTO ON",
                  ":TRIGGER:DELAY 0",
                  ":TRIGGER:SOURCE EXTERNAL",
                  ":TRIG:COUN 3",
                  ":SAMP:COUN 1",
                  ":TRACE:POINTS 3",
                  ":TRACE:FEED SENS",
                  ":TRACE:FEED:CONT NEXT",
                  ":STAT:MEAS:ENAB 512",
                  "*SRE 1"]

for cmd in voltmeter_cmds:
    voltmeter.write(cmd)
time.sleep(1)

# SMU setup
test_current = 7
v_limit = .1
smu_cmds = ["*RST",
            ":SYST:CLE", # clear event log
            ":SOUR:CURR:READ:BACK ON",
            ":SOUR:FUNC CURR",
            ":SOUR:CURR:RANG:AUTO OFF", # source current with fixed range
            f":SOUR:CURR:RANG {test_current}", # will coerce to proper source range
            f":SOUR:CURR:VLIM {v_limit}", # compliance limit
            ":SENS:FUNC \\ 'VOLT\\'", # measure voltage with fixed range
            ":SENS:VOLT:RANG:AUTO OFF",

```



```

f':SENS:VOLT:RANG {v_limit}",
":SENS:VOLT:NPLC 0.01",
":SOUR:CONF:LIST:CREATE \'delta\'", # build a 4 point config list of positive
current,
f':SOUR:CURR:LEV {test_current}", # neg current, positive current, zero current
":SOUR:CONF:LIST:STOR \'delta\'",
f':SOUR:CURR:LEV -{test_current}",
":SOUR:CONF:LIST:STOR \'delta\'",
f':SOUR:CURR:LEV {test_current}",
":SOUR:CONF:LIST:STOR \'delta\'",
":SOUR:CURR:LEV 0.0",
":SOUR:CONF:LIST:STOR \'delta\'",
":DIG:LINE1:MODE TRIG, IN", # digital IO config
":DIG:LINE2:MODE TRIG, OUT",
":TRIG:DIG2:OUT:STIM NOTIFY1",
# ":TRIG:DIG2:OUT:PULSEWIDTH 1e-3" for the scope
":TRIGger:BLOCK:BUFFer:CLear 1", # SMU trigger model setup
":TRIGger:BLOCK:SOURce:STATe 2, ON",
":TRIGger:BLOCK:CONFig:RECall 3, \'delta\', 1",
":TRIG:BLOC:DEL:CONS 4, 0.01",
":TRIGger:BLOCK:MEASure 5",
":TRIG:BLOC:NOP 6", # no operation...just a place holder
":TRIG:BLOC:NOT 7, 1", # output trigger to 2182A...time to measure
":TRIG:BLOC:WAIT 8, DIG1, OR, DISP",
":TRIG:BLOC:CONF:NEXT 9, \'delta\'",
":TRIG:BLOC:DEL:CONS 10, 0.001", # 1msec delay
":TRIGger:BLOCK:MEASure 11",
":TRIG:BLOC:BRAN:COUNTER 12, 3, 7", # repeat 3 times, go to block number 7
":TRIGger:BLOCK:CONFig:RECall 13, \'delta\', 4", # recall 0 Amps value from
config list
":TRIG:BLOC:DEL:CONS 14, 0.05", # seeing glitch as output turns off...due to high
Z state
":TRIGger:BLOCK:SOURce:STATe 15, OFF",
":TRIG:BLOC:BRAN:ALWAYS 16, 0"]

```

```
for cmd in smu_cmds:
```

```
    smu.write(cmd)
```

```
time.sleep(1)
```

```
status = 0
```

```
loop_cnt = 0
```

```
status = voltmeter.read_stb()
```

```

debug = 0

# start the 2460
smu.write(":INIT")
# polling loop for SRQ AND loopCnt too big
# loopCnt too big = expected SRQ did not happen within N loops and some problem
maybe exists
# test for bit6 in STB set to logic 1 (SRQ has occurred)
while ((status and 64) != 64) and (loop_cnt <= 500):
    status = voltmeter.read_stb()
    time.sleep(0.2)
    if debug: print(f"status: {status}", f"loopCnt: {loop_cnt}")
    loop_cnt += 1

# time.sleep(1)
# read 2182A data
volt_data_all = voltmeter.query(":TRACE:DATA?")

volt_array = volt_data_all.split(',')
if debug:
    print(volt_data_all)
    print(volt_array)

# get the source read back (measured) current values from SMU
measured_current_all = smu.query(":TRAC:DATA? 1,3, 'defbuffer1', SOUR")
current_array = measured_current_all.split(',')
if debug:
    print(measured_current_all)
    print(current_array)
v1 = float(volt_array[0])
v2 = float(volt_array[1])
v3 = float(volt_array[2])
i1 = float(current_array[0])
i2 = float(current_array[1])
i3 = float(current_array[2])
# calculate the ohm readings
Res = (v1 - 2 * v2 + v3) / (i1 - 2 * i2 + i3)
#for i in range(0,3):
#    res = float(volt_data_array[i]) / float(measured_current_array[i])
#    Res.append(res)
print(f"Resistance value: {Res} Ω ")
#i += 1

```

```
res_value = str(Res)
smu.write("DISP:CLE")
smu.write("DISP:SCR SWIPE_USER")
smu.write("DISP:USER1:TEXT \'RES value in OHM\'")
smu.write(f"DISP:USER2:TEXT \'{res_value}\'")
# close the session
smu.clear()
smu.close()
voltmeter.clear()
voltmeter.close()
```

**最近更新: 2021-3-5 LHT**

顺颂商祺！

## 泰克亚太区技术支持部门

发送邮件给泰克，为您提供全面可靠的测试测量解决方案，原来解决问题可以这么简单快速，来信：

获取在线技术支持 [400@tektronix.com](mailto:400@tektronix.com)

获取产品报价 [426@tektronix.com](mailto:426@tektronix.com)

维修、延保、校准 [418@tektronix.com](mailto:418@tektronix.com)

网站下载问题反馈 [401@tektronix.com](mailto:401@tektronix.com)