

How do I make measurements?

NOTE: This example assumes you are using a 10kΩ resistor as a device under test (DUT), but you may use any 2-terminal DUT with appropriate source values.

Figure 1-a: Channel A connectors

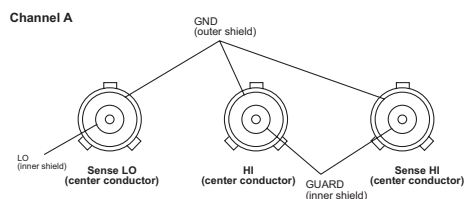


Figure 1-b: Channel B connectors

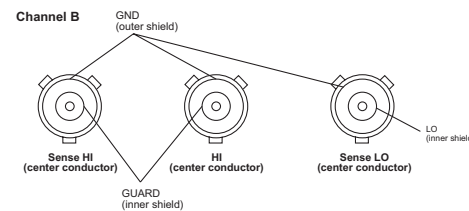
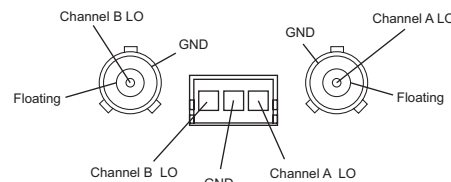


Figure 1-c: Ground module connectors



Step 1: Connect the DUT and reset the instrument to factory default conditions

Connect a 10kΩ resistor to the Keithley Instruments Series 2600 System SourceMeter® Channel A HI and LO terminals, as shown in Figure 2-a or 2-b.

To reset the instrument: Press **MENU > SETUP > RECALL > FACTORY** (press **ENTER** after each selection). To return to normal display, press **EXIT**.

Step 2: Select source and set source level

Perform the following steps to select the voltage source and set the source value to 10V:

1. Press **Channel A SRC** to select the V-Source (indicated by the "V" units). The flashing cursor indicates which value is presently selected for editing.
2. Press the up **RANGE** key, and select the 20V source range. Use the lowest possible source range for best accuracy.
3. Use the **CURSOR** keys to move the cursor to the 10s digit, then press the **rotary knob** to enter the EDIT mode (EDIT annunciator on). The display will output +00.000V.
4. Using either the numeric keys or the **rotary knob**, set the source value to 10.0000V, then press **ENTER** or the **rotary knob**.

Step 3: Set the compliance limit

Perform the following steps to set the current compliance limit value to 10mA:

1. Put the Model 2636 in the single-channel display mode with the **DISPLAY** key (toggle the **DISPLAY** key to see different modes).
2. Press the **LIMIT** key, then press **ENTER** or the **rotary knob**.
3. Move the cursor to the 10s digit, then press the **rotary knob** to enter the EDIT mode.
4. Using the numeric keys or the rotary knob, set the limit value to 10.000mA, then press **ENTER** or the **rotary knob**.

Step 4: Select the measurement function and range

1. Put the Series 2636 in the single-channel display mode with the **DISPLAY** key, then select the current measurement function by pressing **MEAS** or **MODE**.
2. Select the measurement range with the **RANGE** keys. For the purposes of this example, press **AUTO RANGE**, and note that the **AUTO** annunciator turns on. The instrument will automatically select the best range based on the measured value. You can also use manual ranging by pressing the up or down **RANGE** key.

Step 5: Turn the output on

Turn the output on by pressing the appropriate **OUTPUT ON/OFF** key. The ON/OFF indicator LED will light when the channel output is on (NOTE: the safety interlock line on the digital IO connector must be pulled high in order to use 200V range. See Section 10 in the Series 2600 Reference Manual for more information).

Step 6: Make measurements

1. Observe the readings on the display (press **TRIG** if necessary to trigger the unit to begin taking readings). For the single-channel display mode, the readings appear on the top line, while source and limit values are on the bottom line. For the 10kΩ resistor under test, typical display values are 1.00000mA, SrcA: +10.0000 V LimA:010.0000mA.
2. Use the **DISPLAY** key to cycle through the various display modes.
3. Press the **MEAS** key several times to display measured voltage, resistance, and power. Typical values for the 10kΩ resistor are: 10.0000V, 10.0000kΩ, and 10.0000mW.

Step 7: Turn output off

When finished making measurements, turn the output off by pressing the **OUTPUT ON/OFF** key. The OUTPUT indicator light will turn off.

Figure 2-a: 2-wire connections (non-floating)

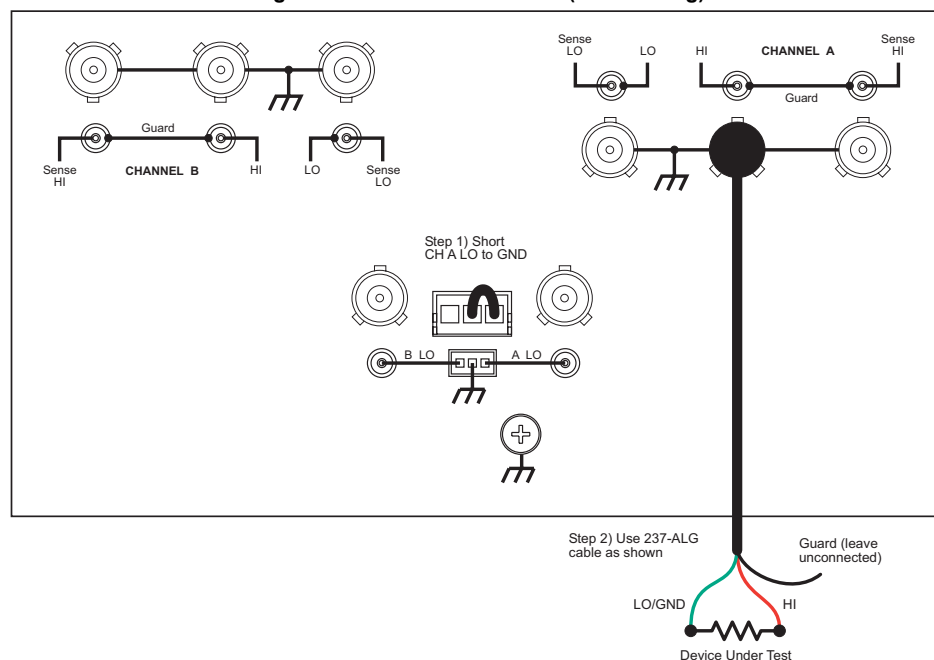
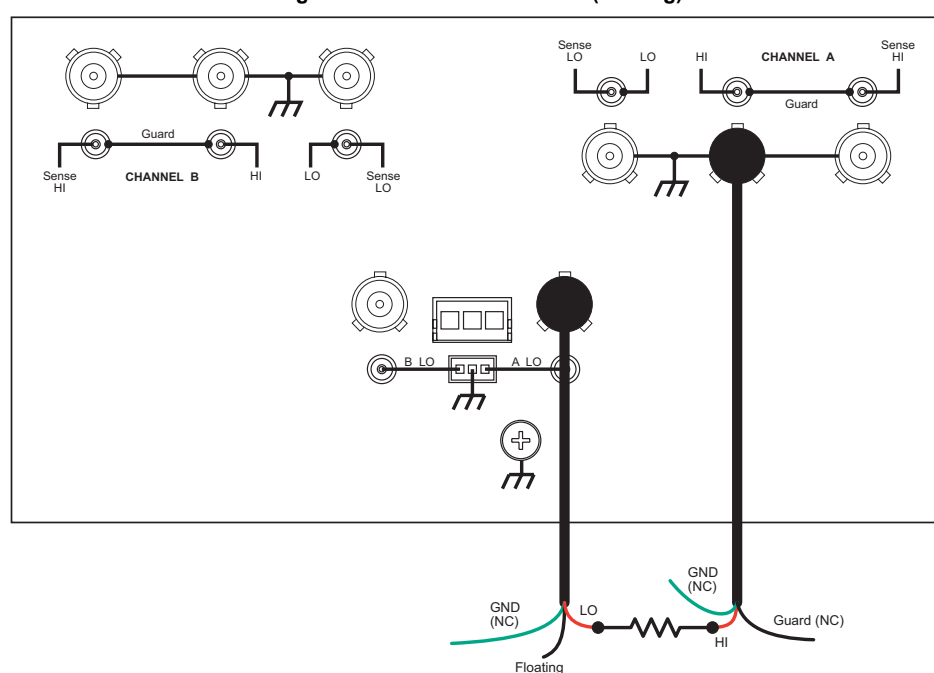


Figure 2-b: 2-wire connections (floating)



Special considerations for Models 2635 and 2636

The Models 2635 and 2636 current measurement delays and filter are turned on by default at power-up, or after `smuX.reset()` is called. See Section 8 of the Series 2600 Reference Manual for more details.

How do I use the remote interface?

Step 1: Connect to the interface

- GPIB: Use a shielded IEEE-488 cable such as the Keithley Instruments Model 7007
- RS-232: Use a shielded 9-pin RS-232 cable like the Keithley Instruments Model 7009-5

Step 2: Select the interface

1. Press **MENU** and then choose **GPIB** for the IEEE-488 interface, **RS-232** for the serial interface, or **AUTO** to have the instrument automatically select the interface.

Step 3: Configure the interface

GPIB interface configuration

Set the primary address using the following procedure. The SourceMeter's primary address must be the same as that specified in your program, or the two devices will not be able to communicate.

1. Press **MENU > GPIB > ADDRESS** (press **ENTER** after each selection).
2. Use the rotary wheel to set the **GPIB** address (0 to 30) and then press **ENTER**.
3. Press **EXIT** to return to the main menu.

Remote control of the Series 2600

Series 2600 SourceMeters are powered by TSP (Test Script Processor) technology. As a result, the Series 2600 is capable of running code written in a scripting language called "Lua." This document will refer to Lua as "Test Script Language (TSL)," or simply, "test scripts."

Software tools

There are two free software tools available for the Series 2600:

- Lab Tracer 2 example software: For fast, easy startup curve tracing applications without programming
- Test Script Builder (TSB): Script-writing tool for programming custom test scripts

To download this software, go to <http://www.keithley.com/support> and type in "2600" in the Model Number field, then choose **software** from pull-down menu and click **search**.

How do I begin programming?


Test scripts can be written using any text editor (such as Notepad or PSPad) and sent to the Series 2600 via GPIB/RS-232 writes. Test scripts can also be written directly in the GPIB/RS-232 write commands of your chosen programming environment. The remaining discussion in this document will assume that TSB is used.

Step 1: Download and install TSB

From the above URL, click **Download Test Script Builder Software**, then **Install Test Script Builder Suite Software** to save the installation file to your local disk.

To install, double-click the **.exe** file and follow the prompts. There will be multiple steps and components in this installation; install all components.

Step 2: Establish communication

1. Click on the **Open Instrument** icon on the instrument console. 
2. Choose the desired instrument from the dialog box and click **OK**.

For example:

- **GPIB0::26::INSTR** selects GPIB interface board 0 with address 26.
- **ASRL1::INSTR** selects the COM1 serial port of the PC.

Once communication is established with the instrument, you will see "TSP>" output on the instrument console.

Step 3: Create a new project

1. Click **File > New > Project > TSP Project > Next**.
2. Give the project a name that begins with a letter (e.g., "K12600sweep"), then select **Use default** and click **Finish**. The newly-created project will appear in the project view with "main.tsp" in it. You may wish to rename this file to something more descriptive (right click on **main.tsp** and rename). Follow the remaining steps to complete your own test script project.

Step 4: Write the scripts in a new project

You can write the test scripts in the .tsp file of your project. In order to write the test scripts, you need two pieces of information:

- Specific commands to control the Series 2600; these commands are called "Instrument Command Libraries" or "ICL"
- General test script language (Lua) syntax information (such as loop controls, math libraries, etc.)

For your convenience, you can find this information in Test Script Builder under the TSP Help View tab. Additionally, you can find relevant information in the following resources:

- Section 2, "TSP Programming Fundamentals" in the Series 2600 Reference Manual
- Comprehensive Lua documentation at <http://www.lua.org/>

The following is an example of how a voltage sweep can be performed with on a Series 2600 SourceMeter. Notice that the script is organized in a function called "SwpVMeasI" with several parameters:

```
function SwpVMeasI(smu, startv, stopv, pts, ilimit, nplc, srcdelay)
    local stepv = (stopv-startv)/(pts-1)
    local vsrc
    smu.reset()
    smu.source.func = smu.OUTPUT_DCVOLTS
    smu.measure.autozero = smu.AUTOZERO_ONCE
    smu.source.limiti = ilimit
    smu.measure.nplc = nplc
    --configure the buffer
    smu.nvbuffer1.clear()
    smu.nvbuffer1.appendmode = 1
    smu.nvbuffer1.collecttimestamps = 1
    smu.source.output = smu.OUTPUT_ON
    vsrc = startv
    for i=1, pts do
        smu.source.levelv = vsrc
        delay(srcdelay)
        smu.measure.i(smu.nvbuffer1)
        vsrc = vsrc + stepv
    end
    smu.source.output = smu.OUTPUT_OFF
    print("test complete")
end
```

end

NOTE: See ICL help for SMU functions and attributes in this script.

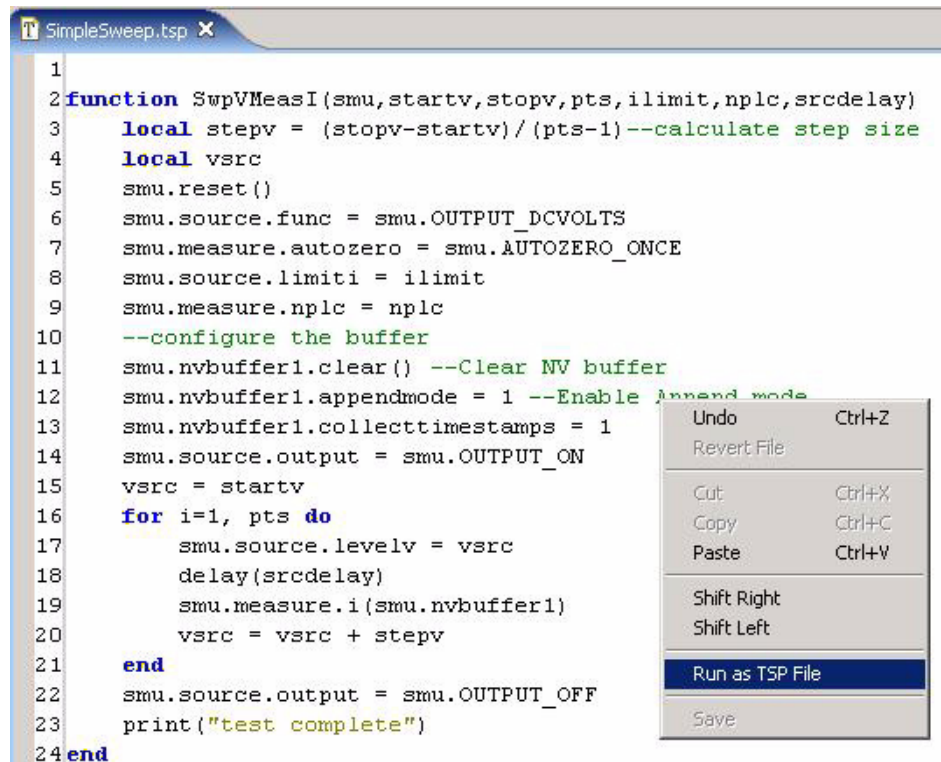
Step 5: Load and run the scripts

The quickest way to load and run the script on the instrument is to right-click in the window containing the content of your .tsp file and choose **Run as TSP File**, as shown on Figure 3.

At this point the scripts are loaded and compiled in the instrument. But the test is not yet performed.

For additional information on using TSB and running scripts from the non-volatile memory of the instrument, refer to Sections 2 and 3 of Series 2600 User's Manual.

Figure 3: Loading and running the script



```
1
2 function SwpVMeasI(smu,startv,stopv,pts,ilimit,nplc,srcdelay)
3   local stepv = (stopv-startv)/(pts-1)--calculate step size
4   local vsrc
5   smu.reset()
6   smu.source.func = smu.OUTPUT_DCVOLTS
7   smu.measure.autozero = smu.AUTOZERO_ONCE
8   smu.source.limiti = ilimit
9   smu.measure.nplc = nplc
10  --configure the buffer
11  smu.nvbuffer1.clear() --Clear NV buffer
12  smu.nvbuffer1.appendmode = 1 --Enable Append mode
13  smu.nvbuffer1.collecttimestamps = 1
14  smu.source.output = smu.OUTPUT_ON
15  vsrc = startv
16  for i=1, pts do
17    smu.source.levelv = vsrc
18    delay(srcdelay)
19    smu.measure.i(smu.nvbuffer1)
20    vsrc = vsrc + stepv
21  end
22  smu.source.output = smu.OUTPUT_OFF
23  print("test complete")
24 end
```

Step 6: Execute the test

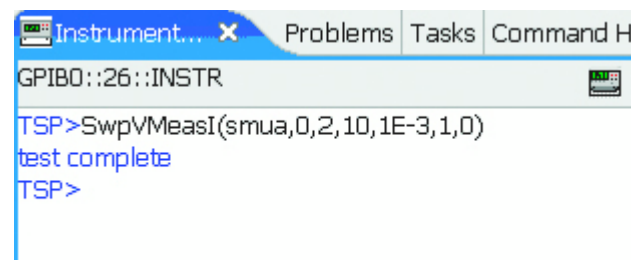
Assuming we want to perform a 0 to 2V, 10-point sweep while measuring current on smua with 1mA compliance current, set up the function as follows:

```
SwpVMeasI(smua,0,2,10,1E-3,1,0)
```

Note that this function also sets NPCL to 1 and the source delay to 0 seconds.

To execute the test, send this function call at the TSP> prompt on the instrument console (as shown on Figure 4). Note that the script prints "test complete" at the end of test.

Figure 4: Executing the test with a function call



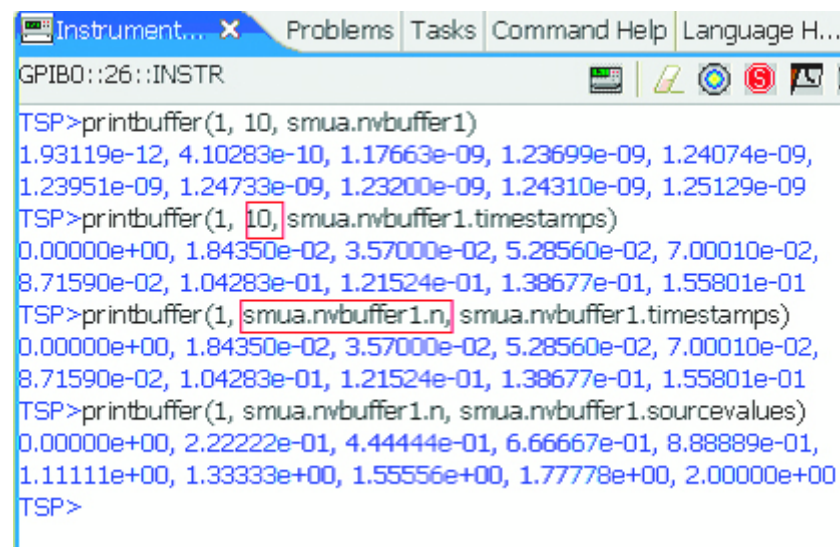
```
GPIBO::26::INSTR
TSP>SwpVMeasI(smua,0,2,10,1E-3,1,0)
test complete
TSP>
```

At this point, it may not be obvious where your current measurements are. If you look at the script (see Figure 3), you will notice that the ICL command `smu.measure.i(smu.nvbuffer1)` puts current readings into the "nvbuffer1" of the selected SMU. In this example, the SMU is smua.

These buffers can be printed using the `printbuffer(start_index, end_index, buffer_element)` command.

Figure 5 illustrates printing the buffer readings, timestamps, and source values. Notice that the `smua.nvbuffer1.n` returns current size of the buffer, which may be useful for `end_index` if you don't know the size of the buffer.

Figure 5: Printing the buffer



```
GPIBO::26::INSTR
TSP>printbuffer(1, 10, smua.nvbuffer1)
1.93119e-12, 4.10283e-10, 1.17663e-09, 1.23699e-09, 1.24074e-09,
1.23951e-09, 1.24733e-09, 1.23200e-09, 1.24310e-09, 1.25129e-09
TSP>printbuffer(1, 10, smua.nvbuffer1.timestamps)
0.00000e+00, 1.84350e-02, 3.57000e-02, 5.28560e-02, 7.00010e-02,
8.71590e-02, 1.04283e-01, 1.21524e-01, 1.38677e-01, 1.55801e-01
TSP>printbuffer(1, smua.nvbuffer1.n, smua.nvbuffer1.timestamps)
0.00000e+00, 1.84350e-02, 3.57000e-02, 5.28560e-02, 7.00010e-02,
8.71590e-02, 1.04283e-01, 1.21524e-01, 1.38677e-01, 1.55801e-01
TSP>printbuffer(1, smua.nvbuffer1.n, smua.nvbuffer1.sourcevalues)
0.00000e+00, 2.22222e-01, 4.44444e-01, 6.66667e-01, 8.88889e-01,
1.11111e+00, 1.33333e+00, 1.55556e+00, 1.77778e+00, 2.00000e+00
TSP>
```

Final notes on programming

TSP technology brings about a new architecture in instrument software. There are two distinct parts to this architecture:

- Test scripts for controlling test routines, including complex decision-making
- PC-based development software such as National Instruments LabView™, Microsoft® Visual Basic®, and Microsoft® Visual C++® to provide a user interface and data display

Although sending line-by-line ICL commands and using the Series 2600 like other traditional instruments is still possible, you will get the highest test speed and test routine flexibility by embedding tasks such as test routine control and decision-making in the test scripts. This will allow the instrument, not the PC, to perform these tasks. PC-based software can then interact with test scripts by calling appropriate functions with user-selected parameters and retrieving data for display on the PC.

It is recommended that you use TSB to write and debug your test scripts. Once the scripts are complete and embedded in the instrument, use PC-based development software to make function calls and print and retrieve data. Simple I/O read/write (either GPIB/RS-232 or VISA) is all you need to interface your software with your test scripts embedded in the instrument.

How to get help

If you need assistance or more detailed information:

- Refer to the User's and Reference Manuals on the Product Information CD-Rom that came with your Series 2600.
- Consult the Keithley Instruments support website at www.keithley.com/support.
- Contact your local Keithley Instruments representative or call 1-888-KEITHLEY.

For more specific technical support:

- Email our Applications Engineers at applications@keithley.com.