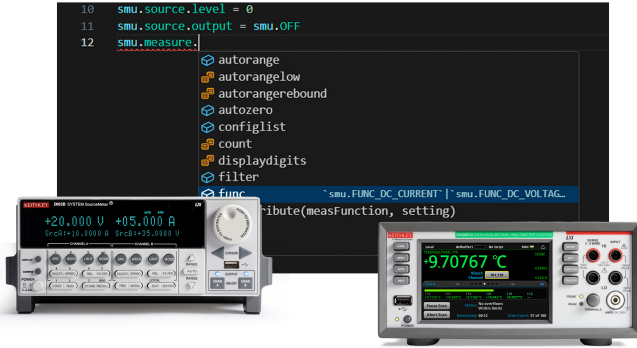


TSP Toolkit

Visual Studio Code Extension



Test Script Processor ([TSP™](#)) technology is a unique instrument automation command set and programming language. TSP-enabled instruments contain an embedded scripting engine that's capable of executing both instrument control commands and basic programming functionality.

The TSP Toolkit is a [Visual Studio Code](#) extension that provides rich support for TSP technology to edit and execute scripts. The extension includes command-set documentation and language features such as syntax error detection and code navigation, as well as code completion suggestions, inline help, and built-in TSP command documentation.

Key features

The TSP Toolkit allows you to:

- Write your code with the assistance of autocompletion and syntax checking
- Access detailed information on individual commands such as definition, accepted parameters and usage examples
- Send commands and interact directly with your instruments through the terminal
- Export and save instrument buffers to a PC
- Debug TSP Scripts
- Discover available instruments on your local network
- Remotely upgrade the instrument firmware
- Open saved factory scripts to view or edit
- Export the output of a TSP script from the terminal to a *.csv or *.txt file
- Leverage all the benefits of TSP scripting more easily than ever

Applications

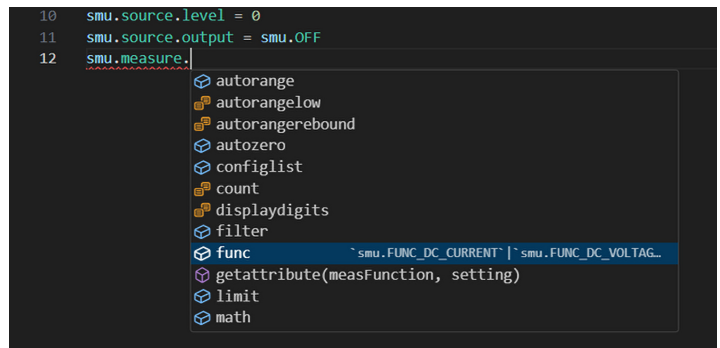
- Test automation via embedded script execution
- Scalable test systems
- Synchronized test system control

A New Era in Test Script Development

The TSP Toolkit is an updated script development environment that includes instrument support for a majority of TSP-enabled instrumentation plus many quality-of-life features that improve the script development experience.

TSP Toolkit features the modern user interface of the Visual Studio Code IDE, complete with syntax highlighting and the increased readability it affords, along with the convenience of the many extensions available on the VSCode marketplace.

The TSP Toolkit extension also features autocompletion alongside in-line and hover help, reducing pesky mistakes from copying/pasting TSP commands improperly and eliminating the need to manually parse through dense reference manuals to confirm proper command usage and syntax.



The autocompletion feature assists with writing code and reducing mistakes

Optimizing Productivity and Increasing Throughput

TSP technology is both an instrument automation command set and programming language. TSP-enabled instruments, like source measure units (SMU), data acquisition (DAQ) systems, and digital multimeters (DMMs), contain an embedded scripting engine that can execute basic programming functionality and instrument control commands.

Demo.tsp

```
1 reset()
2 local volts = smu.FUNC_DC_VOLTAGE
3 local fun
4 local lim
5 local lev
6
7 function setSMU(fun, lim, lev)
8   smu.source.func = fun
9   smu.source.ilevel.level = lim
10  smu.source.level = lev
11  print("2460 (global) smu.source.level: any = 0|20|300|5|6|600
12  print("2460 Discharge the capacitor to 0 V and turn off the output.
13  print("2460 Discharge the device to 0 V and turn off the output.
14 end
15
16 function toggle
17   smu.source.
18   print("2460 Set to pulselevel (amplitude) and save to list
19 end
20   Set to biaslevel and save to list
21
22 function main()
23   setSMU(volt
24   toggleOutput
25   delay(0.5)
26   setSMU(volts, 0.001, 0)
27   toggleOutput(smu.OFF)
28 end
29
30 main()
31 print("done")
32
33
```

2450/29907.htm

smu.source.level

This attribute immediately selects a fixed amplitude for the selected source function.

Type	TSP-Link accessible	Affected by	Where saved	Default value
Attribute (RW)	Yes	Restore configuration Instrument reset Power cycle Source configuration list Function change	Configuration script Source configuration list	0

sourceLevel

Current: -1.05 A to 1.05 A
Voltage: -210 V to 210 V

function

The source function:

- Current source: smu.FUNC_DC_CURRENT
- Voltage source: smu.FUNC_DC_VOLTAGE

Details

This command sets the output level of the voltage or current source. If the output is on, the new level is sourced immediately.

The sign of the source level dictates the polarity of the source. Positive values generate positive voltage or current from the high terminal of the source relative to the low terminal. Negative

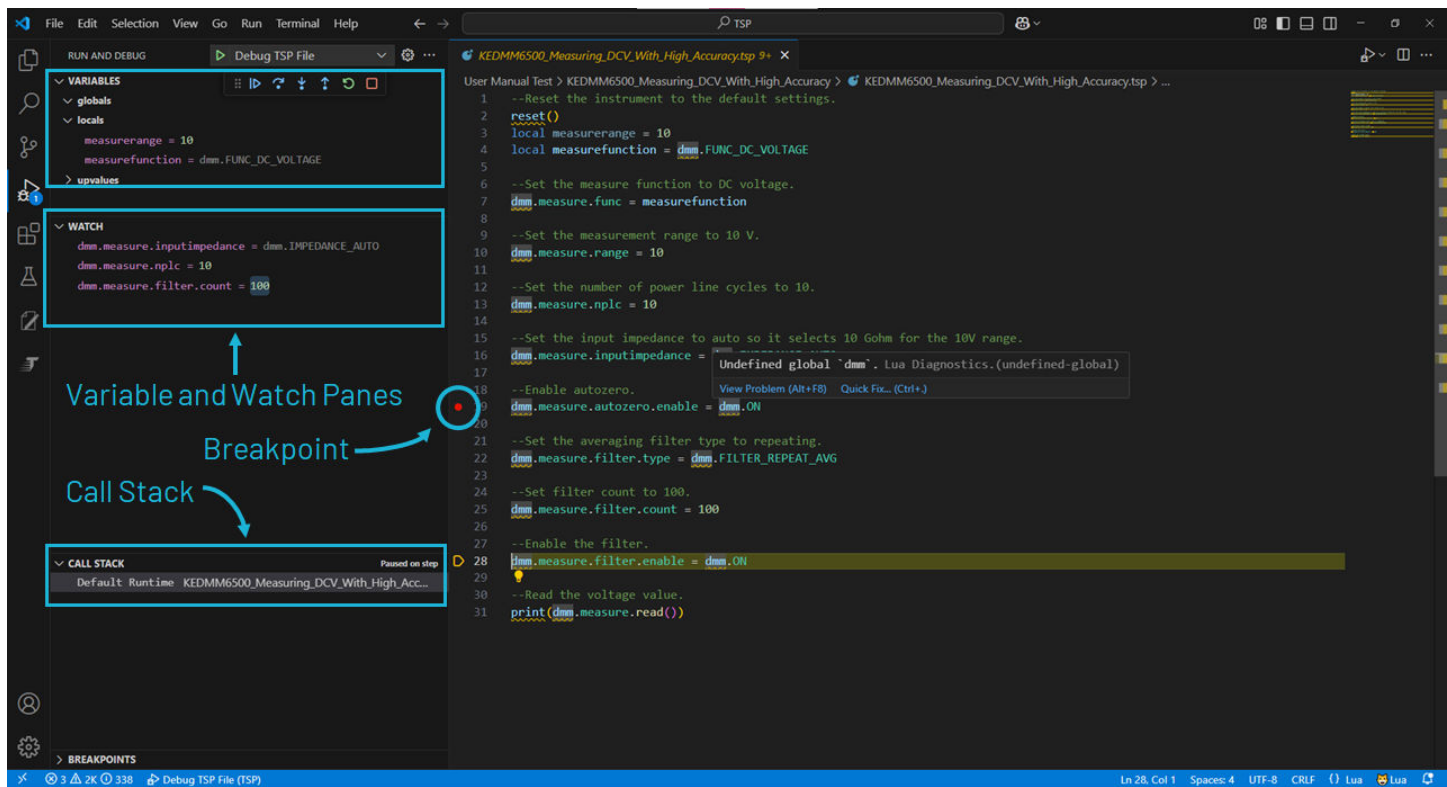
The automated TSP script generation feature allows you to configure a variety of bias and sweep sourcing operations

Advanced Features

On-Instrument Debugger

Gives a user the ability to debug scripts as they run in the instrument environment.

- **Variable Pane:** Monitor the status of declared variables and edit values dynamically
- **Watch Pane:** Monitor the status of selected variables, commands, tables, and expressions dynamically as the debugger steps through the code
- **Call Stack:** View the function or procedure calls that are currently on the stack
- **Compatible with all currently-supported instrumentation**



The debugger feature allows you to monitor variables, watch expressions and view the call stack

Automated TSP Script Generation

Perform current vs. voltage (I-V) test on a variety of materials, such as two-terminal and multi-terminal semiconductor devices, solar cells, and more. You can configure each SMU or PSU for a variety of bias and sweep sourcing operations, then run, edit, and leverage the resulting TSP code.

- **Interactive GUI:** Quickly create I-V characterization routines by entering test parameters in a user interface and generate TSP scripts with a single mouse click
- **Waveform Previews:** Verify sourcing behavior prior to generating a script to eliminate guesswork
- **Compatible with MP5000 Series SMUs and PSUs**



The automated TSP script generation feature allows you to configure a variety of bias and sweep sourcing operations

TriggerFlow for Building Trigger Models

The trigger model of the MP5000 can be represented through a graphical display within TSP Toolkit called TriggerFlow™. The TriggerFlow™ UI allows complex trigger models to be configured visually by simply arranging the operation blocks that the instrument executes.

- **Interactive GUI:** Drag and drop trigger model blocks to build complex trigger flows in an infinite canvas
- **Templates:** Pre-configured groups of trigger model blocks that perform common operations such as pulse sweeps can be added to a TriggerFlow
- **Help Text:** Comments within the generated test script define which additional TSP commands are needed to create a functional trigger model
- **Compatible with MP5000 Series SMUs and PSUs**

The screenshot displays the TSP Toolkit Visual Studio Code Extension interface. On the left, the 'TriggerFlow' panel shows a 'Model Settings' tab and an 'Open Script' button. Below these are 'Blocks' and 'Actions' categories. The 'Actions' category is expanded, showing various trigger model blocks like 'Config List Next', 'Measure', 'Log Event', 'Constant Delay', 'Source Action Step', 'Source Action Bias', 'Source Action Skip', 'Source Action Set', 'Source Output', 'Reset Branch Counter', 'Measure Overlapped', 'No Operation', 'Source Action Step', and 'Source Output'. The main canvas shows a graphical flowchart for a trigger model named 'Drain'. The flowchart starts with a 'Source Action Step' block, followed by a 'Log Event' block, a 'Constant Delay' block, a 'Measure' block, another 'Constant Delay' block, a 'Log Event' block, and finally a 'Loop Counter' block. The 'Loop Counter' block has a 'Y' (Yes) path that loops back to the 'Source Action Step' block and an 'N' (No) path that exits the loop. On the right, the '2ChannelCurves.tsp' script is shown, which contains the generated TSP code for the trigger model. The script includes comments and code for initializing the trigger model, adding blocks, and initiating the trigger model.

```

27 for i = 1, CH do
47 end
48
49 -- If your trigger model requires TSP-Linked instruments, be sure to initialize TSP-Link h
50 -- tsplink.initialize()
51
52 -- Initialize and configure any reading buffers into which you wish to put your readings
53 drainSMU.trigger.source.linearv(0, 10, 5)
54 drainSMU.trigger.measure.iv(drainSMU.defbuffer1, drainSMU.defbuffer2)
55
56
57 -- BEGIN GENERATED TRIGGER MODEL --
58 -- The lines between "BEGIN GENERATED TRIGGER MODEL" and "END GENERATED TRIGGER MODEL"
59 -- are machine generated and will be replaced when the trigger model is edited in the
60 -- TriggerFlow UI in TSP Toolkit. Any manual changes between these two comments will
61 -- be lost.
62
63
64 -- Drain --
65 slot[1].trigger.model.delete("Drain")
66 slot[1].trigger.model.create("Drain")
67 slot[1].trigger.model.addblock.source.action.step("Drain", "Source Action Step", { 1 })
68 slot[1].trigger.model.addblock.logevent("Drain", "Log Event", slot[1].trigger.model.LOG_INI
69 slot[1].trigger.model.addblock.delay.constant("Drain", "Constant Delay", 1 )
70
71 slot[1].trigger.model.addblock.measure("Drain", "Measure 8", { 1 })
72 slot[1].trigger.model.addblock.delay.constant("Drain", "Constant Delay 1", 1 )
73
74 slot[1].trigger.model.addblock.logevent("Drain", "Log Event 3", slot[1].trigger.model.LOG_2M01, "Log
75 slot[1].trigger.model.addblock.branch.counter("Drain", "Loop Counter", "Source Action Step", 10)
76 -- IMPORTANT: Don't forget to start your trigger model by
77 -- uncommenting the following line (after you complete the setup)
78 -- slot[1].trigger.model.initiate("Drain")
79
80 -- do not remove the line below
81 -- END GENERATED TRIGGER MODEL --
82 -- Process and/or print your data as needed after this point.
83 -- e.g. printbuffer(startIndex, endIndex, buffername)
84 slot[1].trigger.model.initiate("Drain")

```

Configure a trigger model graphically with TriggerFlow

Supported Instruments

Source Measure Units	Digital Multimeters	Data Acquisition Systems	Power Supplies
MP5000 SMU Series Modular Precision Test System Source Measure Units	DMM7510 7.5-Digit Graphical Sampling Meter	DAQ6510 6.5-Digit Data Acquisition and Logging Multimeter System	MP5000 PSU Series Modular Precision Test System Power Supply Units
2400 Graphical Touchscreen Series	DMM6500 6.5-Digit Multimeter with Graphical Touchscreen	3700A System Switch / Multimeter	
2600B Single and Dual Channel Systems			
2650 Series for High Power			

Install the latest version

[Microsoft Visual Studio Code Marketplace](#)

Learn More about TSP Toolkit

Visit the [TSP Toolkit product page](#) for the latest information about TSP Toolkit.

For questions, please contact your local Tektronix sales office.

Recommended System Requirements and Prerequisites

- 1.6 GHz or faster processor
- 1 GB of RAM
- Windows 10 and 11 (64-bit)
- Linux (Debian): Ubuntu Desktop 24.04, Debian 13
- Linux (Red Hat): Red Hat Enterprise Linux 10, Fedora 40
- Instrument communication interfaces: USB and LAN
 - VISA drivers are supported for instrument connectivity and required for USB connections
 - LAN communications are supported via raw sockets or via VXI-11 and HiSLIP using VISA drivers.

Contact Information:

Australia 1 800 709 465
Austria* 00800 2255 4835
Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium* 00800 2255 4835
Brazil +55 (11) 3530-8901
Canada 1 800 833 9200
Central East Europe / Baltics +41 52 675 3777
Central Europe / Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France* 00800 2255 4835
Germany* 00800 2255 4835
Hong Kong 400 820 5835
India 000 800 650 1835
Indonesia 007 803 601 5249
Italy 00800 2255 4835
Japan 81(3) 6714 3086
Luxembourg +41 52 675 3777
Malaysia 1 800 22 55835
Mexico, Central/South America and Caribbean 52 (55) 88 69 35 25
Middle East, Asia, and North Africa +41 52 675 3777
The Netherlands* 00800 2255 4835
New Zealand 0800 800 238
Norway 800 16098
People's Republic of China 400 820 5835
Philippines 1 800 1601 0077
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea +82 2 565 1455
Russia / CIS +7 (495) 6647564
Singapore 800 6011 473
South Africa +41 52 675 3777
Spain* 00800 2255 4835



CalPlus GmbH

Zentrale Berlin
Heerstraße 32 • 14052 Berlin
Tel.: 030 214982-0 • Fax: 030 214982-50
office@calplus.de • www.calplus.de

CalPlus GmbH

Niederlassung ScopeShop
Normannenweg 30 • 20537 Hamburg
Tel.: 040 3039595-0 • Fax: 040 3039595-50
scopeshop@calplus.de • www.calplus.de

Find more valuable resources at TEK.COM

Tektronix®

Copyright © Tektronix. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.