Tektronix

Tektronix TSP Toolkit

Software

QUICK START GUIDE



077-1878-04 November 2026

TSP Toolkit Quick Start Guide

Introduction

The Tektronix TSP™ Toolkit is an open-source Microsoft™ Visual Studio Code™ extension that provides support for Tektronix's Test Script Processor (TSP) technology to edit and execute scripts on TSP-enabled Tektronix instruments.

The extension includes language features such as syntax error detection, code navigation, and code-completion suggestions, as well as TSP command-set documentation and hover help.

This guide shows you how to:

- Install the TSP Toolkit extension
- Set up your workspace
- Connect to an instrument
- Configure a project
- Run a TSP script
- Use the Terminal
- Use the debugger
- Use automated TSP script generation
- Use the TriggerFlow feature
- Use the data export feature
- Use the Python interoperability feature
- Download and use TSP example scripts

NOTE: You can download Visual Studio Code from https://code.visualstudio.com/.

Install the TSP Toolkit extension

NOTE: Before installing the extension from the Marketplace, select **Help > Check for Updates** to make sure that you have the most recent version of Visual Studio Code.

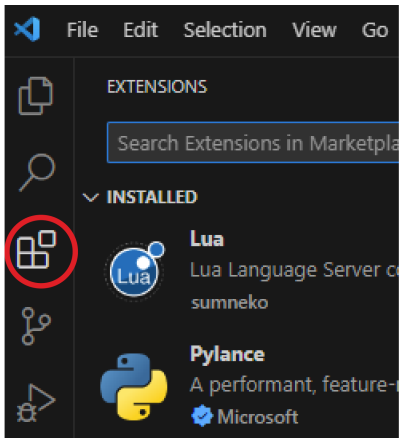
If you are using Microsoft Windows, be sure to also have the latest Visual C++ Redistributable library installed.

Copyright © Tektronix
tek.com

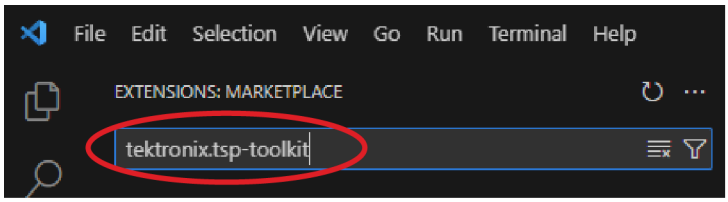
077-1878-04 November 2026

To install the extension from the Visual Studio Code Marketplace:

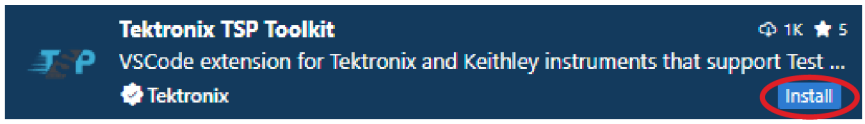
1. Select the extensions icon.



2. Select the search icon, then enter `tektronix.tsp-toolkit` in the search field.



3. Select **Install** under the Tektronix TSP Toolkit.

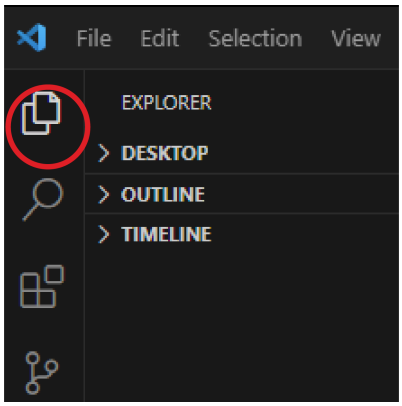


4. The extension installs. Reload the window if you are prompted.

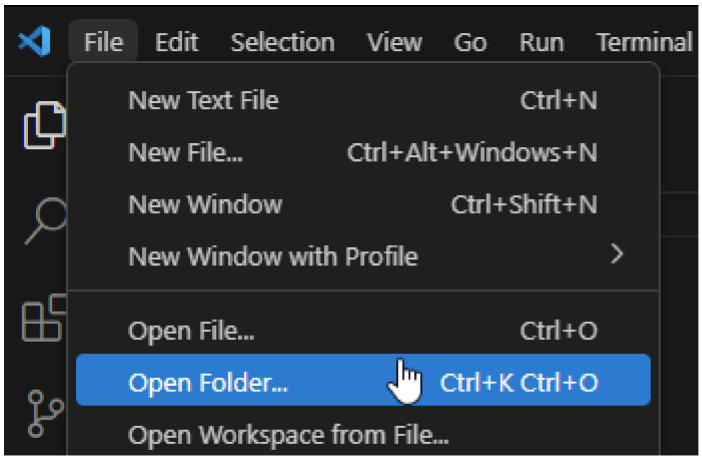
Set up your workspace

To set up your workspace in Visual Studio Code:

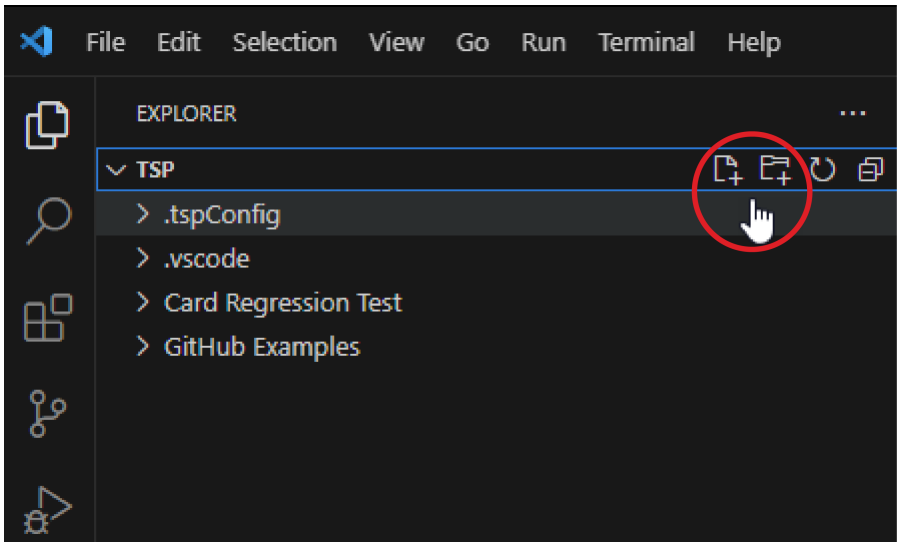
1. Select the explorer icon.



2. Select **File**.
3. Select **Open Folder** to select a folder or create a new folder to use as your workspace.



4. In your workspace, use the **New File** and **New Folder** icons to create new TSP files and subfolders.



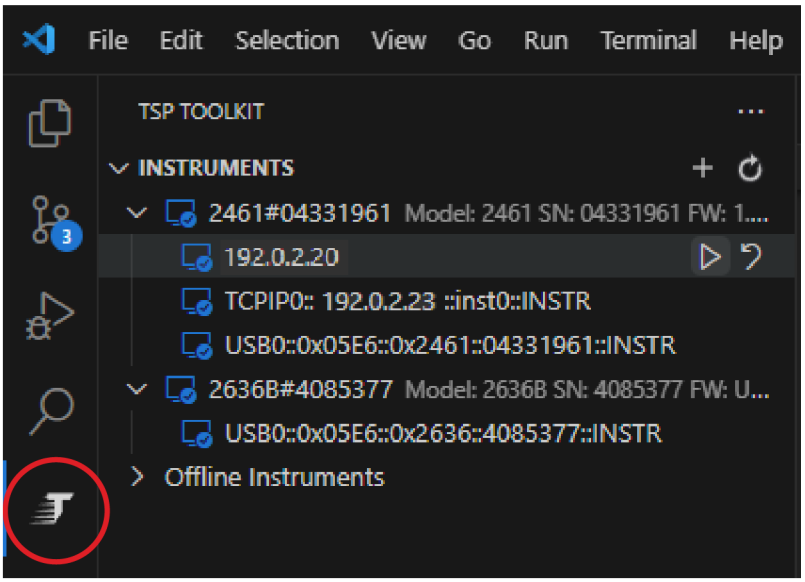
Connect to an instrument

You can connect your TSP-enabled instrument to your computer with a LAN, GPIB, or USB connection. GPIB and USB connections require a VISA driver.

NOTE: If your instrument is operating using secure connections, TSP Toolkit will prompt for a password to be entered before connecting. The security settings for your instrument must be configured before using TSP Toolkit. For more information on configuring your instrument's security settings, visit tek.com for supporting documentation.

To connect to a TSP-enabled instrument:

1. Select the TSP icon on the left of the screen to open the instruments menu.



2. Hover over the instrument you want to connect to, then select **Connect**.



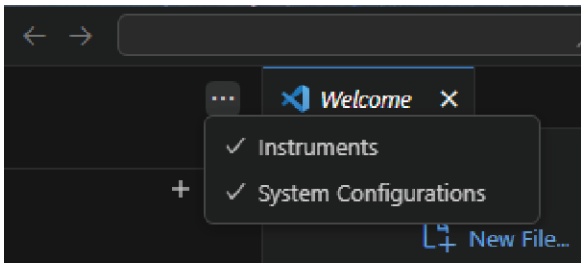
If the connection was successful, a terminal window opens, and the `*IDN?` string is displayed.

Configure a project

You can configure your project to have language features enabled for your TSP instruments and TSP-Link node network.

To configure a project:

1. Select the TSP icon on the left of the screen to open the Instruments menu.
2. Be sure that the System Configurations menu is selected.



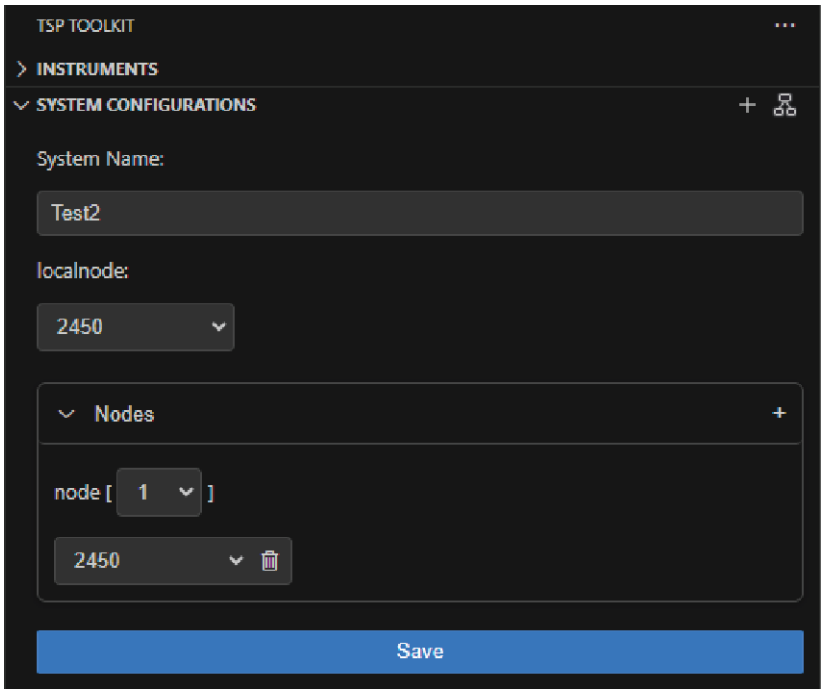
3. Select the instrument tree icon to access the connected instrument and its TSP-Link nodes.



4. If you do not have a connected instrument, select the add icon to manually add a new system.



5. Define your test system, then select **Save**.

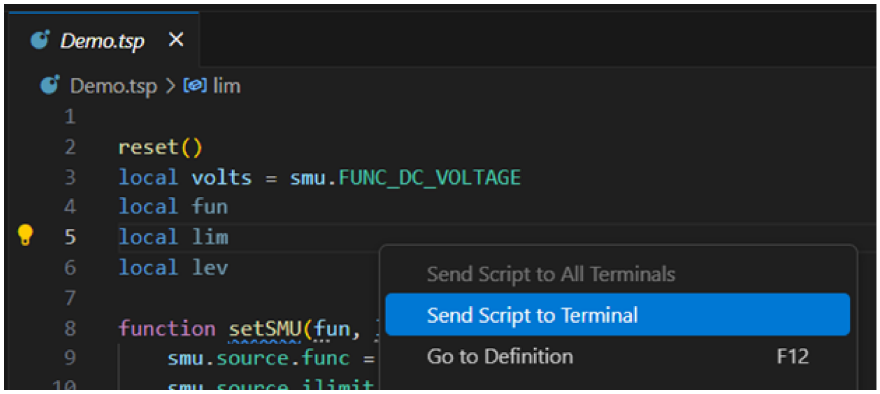


When your project is configured, you are shown relevant code-completion suggestions, signature help, and command documentation for your connected instruments.

Run a TSP script

To run a .TSP script:

1. Open a TSP script in the editor by clicking on it in the workspace or by selecting **File > Open File**.
2. Right-click anywhere within the script editor to display the context menu.
3. Select **Send Script to Terminal** to run the script.

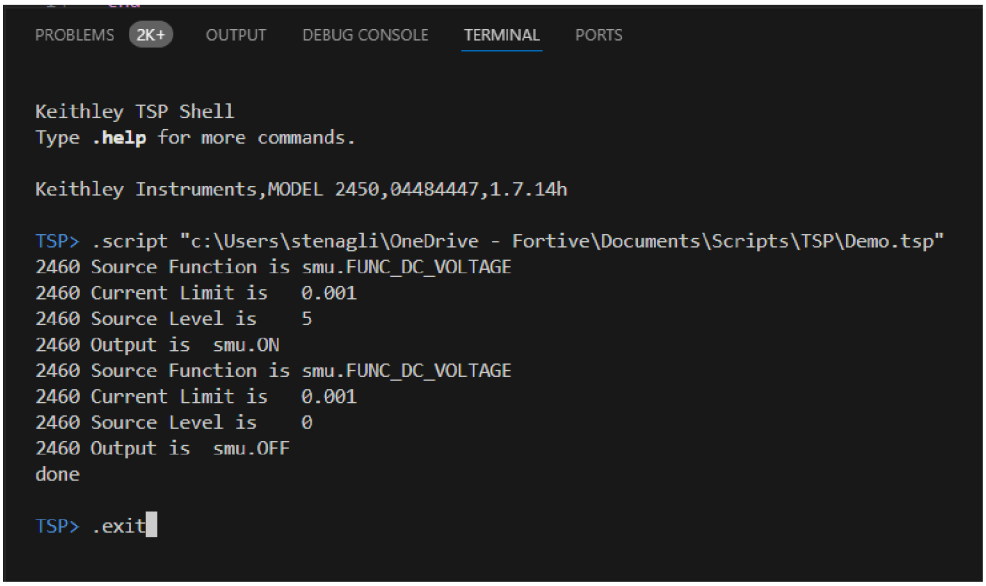


NOTE: When scripts or commands are run from the Terminal, errors are only returned after the requested action completes. No new errors are printed while the operation is in progress.

Using the Terminal

Once you have established a connection with your instrument, the Terminal can be used to send TSP commands and run TSP scripts.

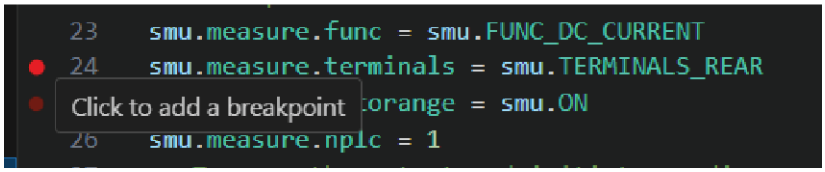
To close the Terminal and disconnect from the instrument, send the `.exit` command.



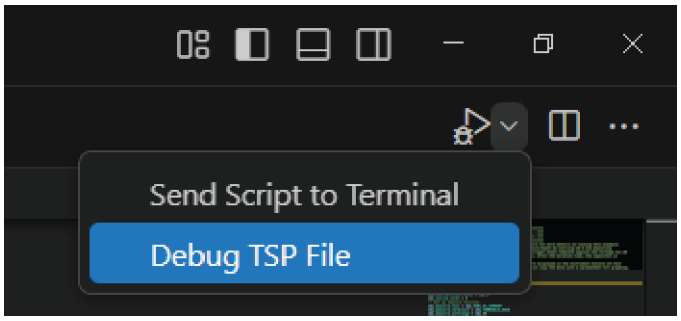
Using the Debugger feature

To debug a TSP script using the TSP Toolkit extension:

1. Add a breakpoint by clicking to the left of a line number.



2. In the upper-right corner of the script editor, select the Run and Debug icon and then select **Debug TSP File**. The script runs until the breakpoint is triggered.



3. When the breakpoint is triggered, use the debugger controls to step through the script.

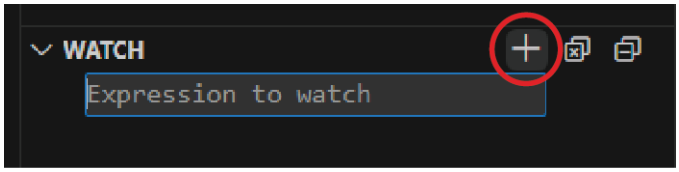


4. You can navigate to the Debug view at any time by selecting Run and Debug.

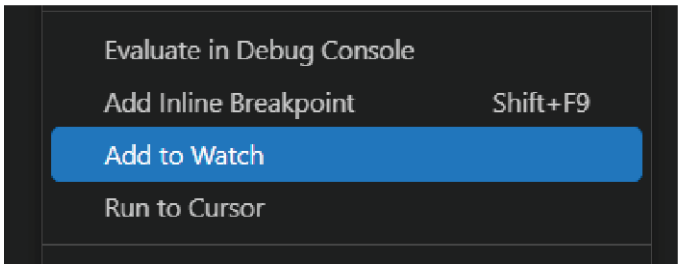


To add expressions to the Watch menu:

1. Select the add icon in the top right corner.



2. Enter the variable, command, or expression you wish to monitor, then press **Enter**.
3. You can also highlight the expression within the script, then right-click and select **Add to Watch**.



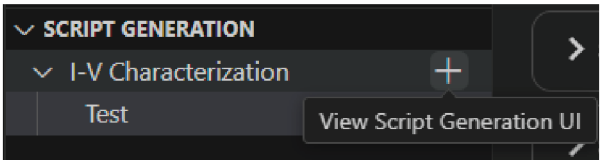
Using the automated TSP script generation feature

NOTE: This feature is only compatible with the Tektronix MP5000 Series test system PSU and SMU modules.

Automated TSP script generation is not compatible with cloud-based workspaces in Microsoft OneDrive™. If you are using OneDrive, be sure that your TSP Toolkit workspace is saved to your local computer.

To create a TSP script using the Script Generation UI:

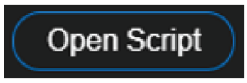
1. From the Explorer menu, select **Script Generation**.
2. Select + to view the TSP Toolkit Script Generation user interface.



3. Use the UI controls to configure the source and measure settings. You can use the waveform previews to verify that the settings produce the desired sourcing behavior.



4. When you are finished adjusting parameters, select **Open Script**.



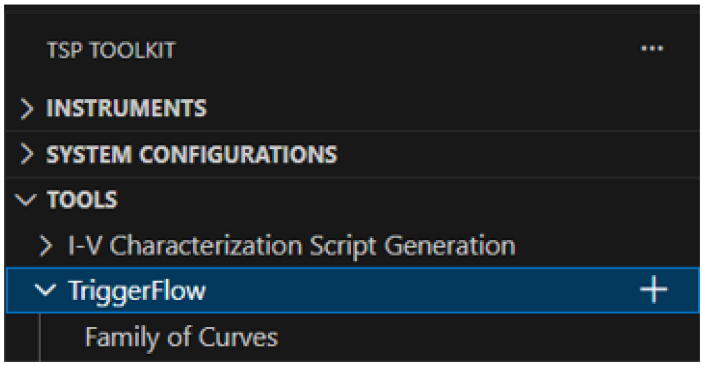
5. The generated script is displayed in a new editor tab. You can run, save, or edit the script.

Using the TriggerFlow feature

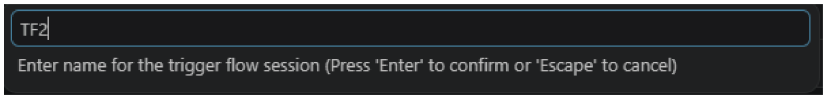
The MP5000 Series trigger model can be represented with a graphical display in TSP Toolkit called TriggerFlow™. This interactive tool lets you configure complex trigger models by arranging operation blocks for the instrument to execute instead of creating a script through conventional programming. You can add up to six trigger models to your script.

To create a trigger model with TriggerFlow:

1. From the **Tools** menu, navigate to TriggerFlow, then select **+** to begin a new session.

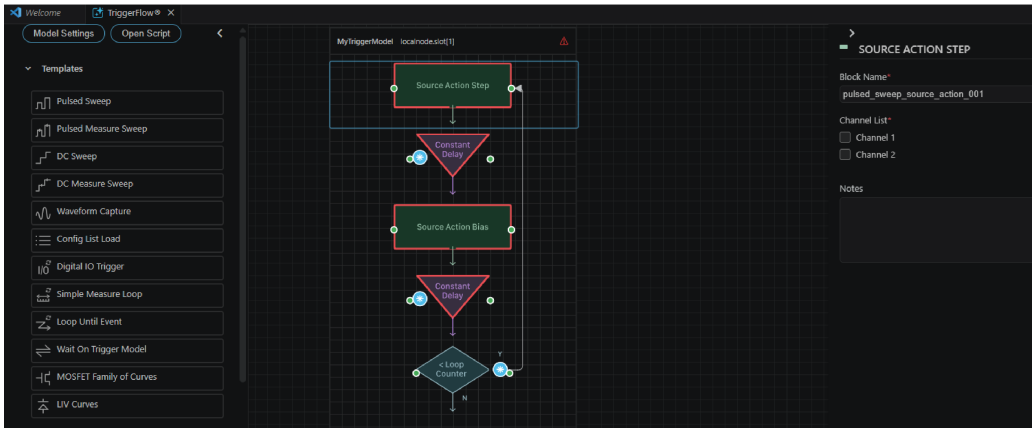


2. From the search field, enter a name for the session, then press **<Enter>**.



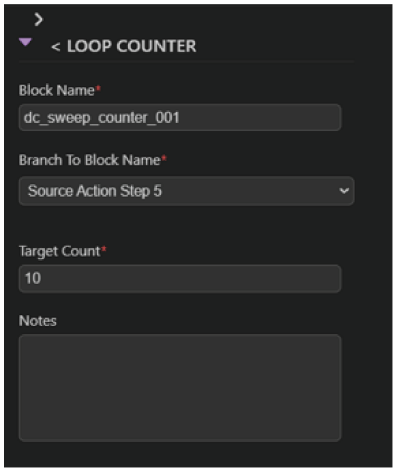
3. Open the **Templates** menu to add a predefined trigger model onto the canvas. You can drag and drop or select individual action blocks from the **Blocks > Actions** menu.

   Connect Branch blocks to other blocks by selecting a green dot on the block and dragging to a green dot on the target block.

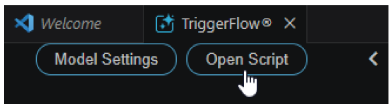


4. Select a block to edit its parameters.

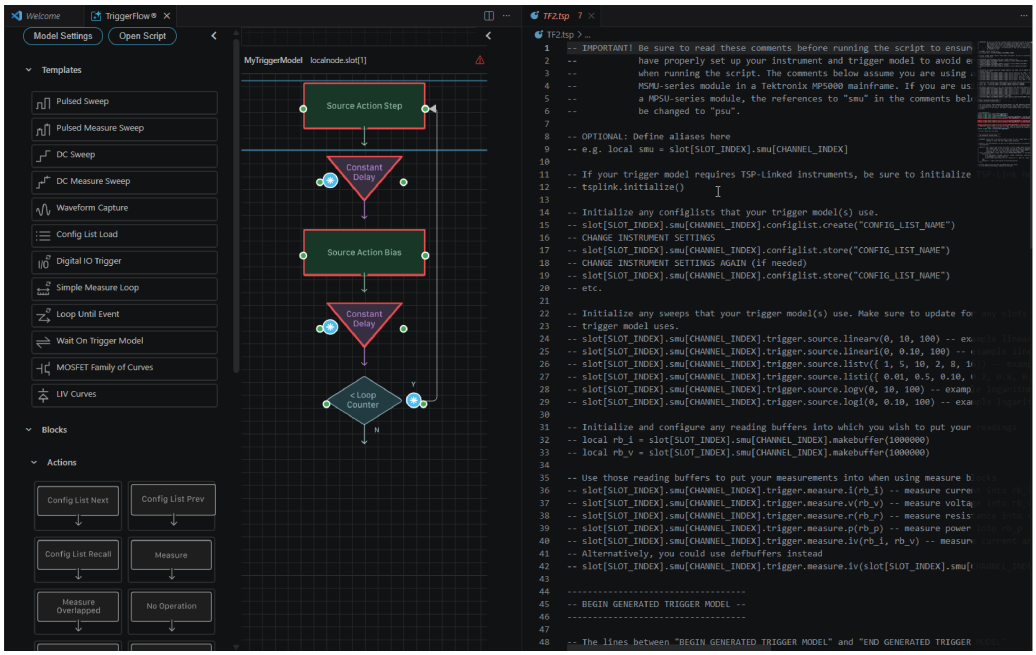
   Blocks with red borders require additional parameters set after adding them to the canvas. Some action blocks may not have adjustable parameters.



5. Select **Open Script** to generate the script code.



   The script code is displayed

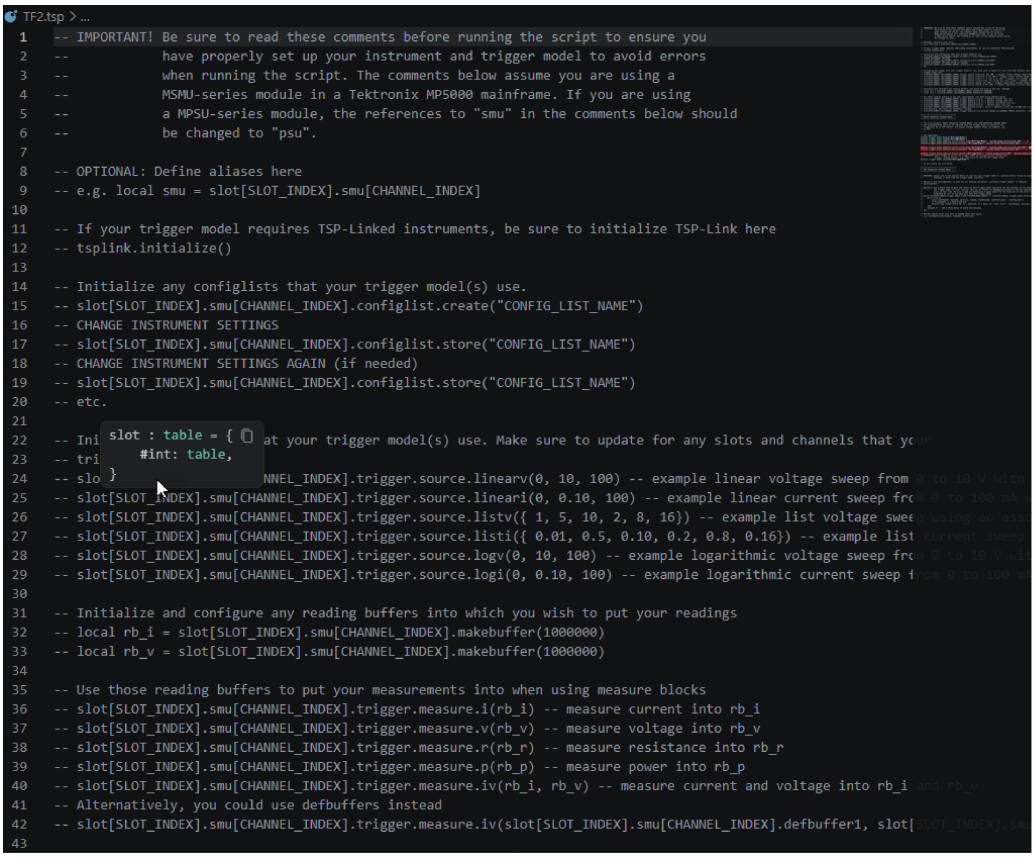


Completing the script

While the TriggerFlow feature automatically generates trigger model code, additional configuration code must be added and other tasks must be completed before a TSP script will run:

- Source and measure settings must be configured
- Configuration lists and sweeps must be created
- TSP-Link must be initialized and nodes configured
- Reading buffers must be initialized and configured

Commented text within the generated code includes detailed descriptions and examples to help complete these tasks. You can edit these notes and remove commenting tags as necessary.

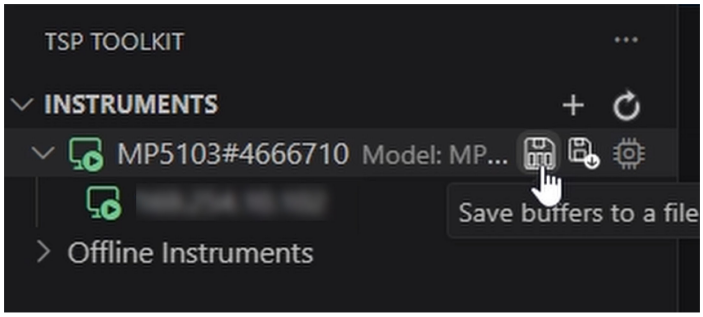


Using the data export feature

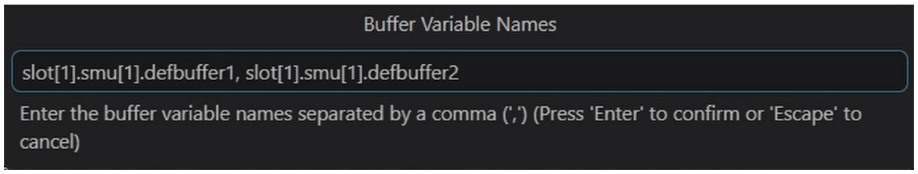
The data export feature allows you to export and save instrument buffers as well as record and store commands and output. You can save the data to your PC using either the `*.csv` or `*.txt` file format.

To save instrument buffers to your PC:

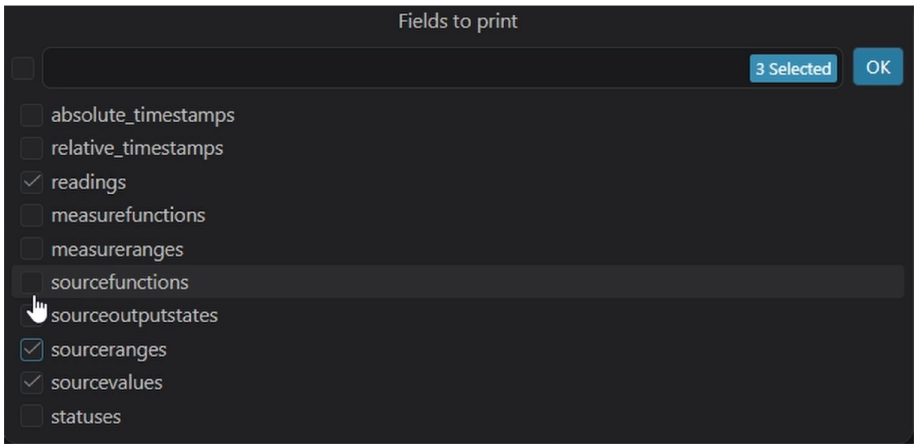
1. From the Instruments menu, navigate to your connected instrument and select **Save buffers to a file**.



2. Enter the variable buffer names separated by a comma, then press **<Enter>** to continue.



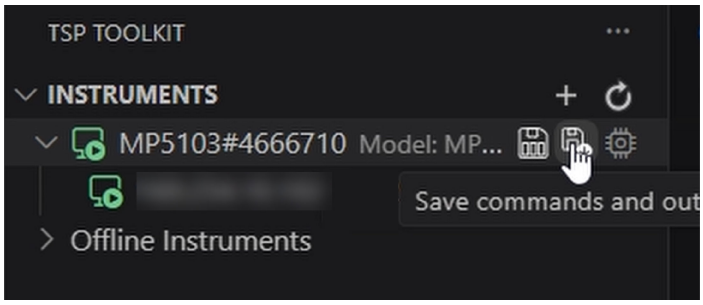
3. Specify a string separator for the data fields, then press **<Enter>** to continue.
4. Select the fields to include in the export, then select **OK**.



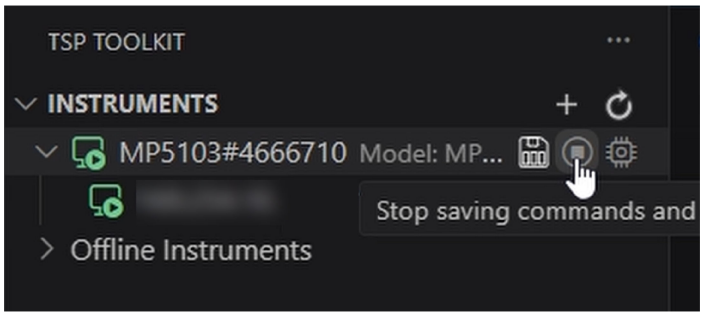
5. Enter a file name with a `*.csv` or `*.txt` extension, then select **Save**.

To record instrument commands and output and save it to your PC:

1. From the Instruments menu, navigate to your connected instrument and select **Save commands and output**.



2. Enter a file name with a `*.csv` or `*.txt` extension, then select **Save**. The terminal session will begin recording.
3. When you are finished recording, select **Stop saving commands and output**.

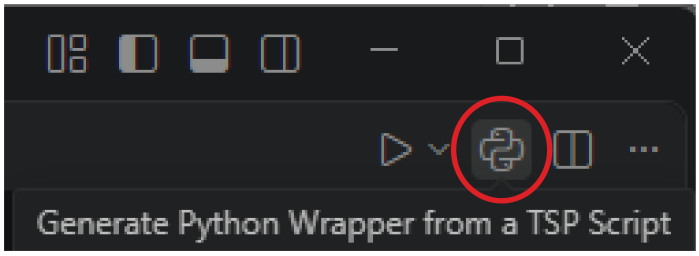


Using the Python interoperability feature

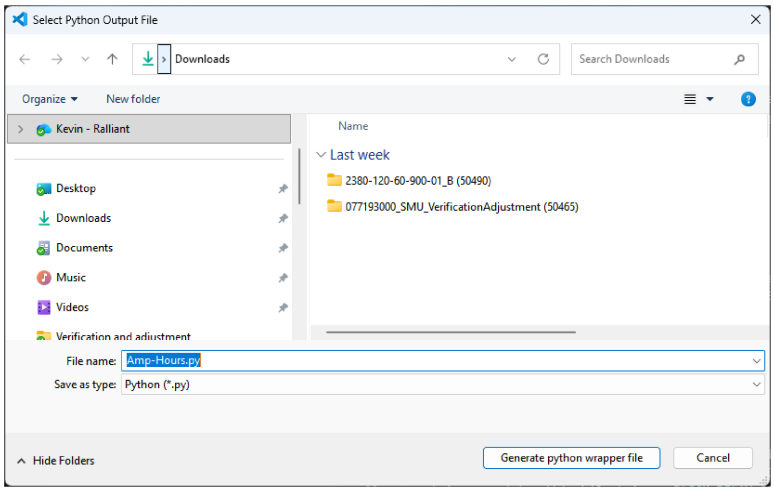
TSP Toolkit can convert a *.tsp script into a Python wrapper class. The wrapper loads the same *.tsp script onto the instrument and exposes its public functions and supported global variables to your Python application. You can use all of the timing and synchronization benefits of TSP on-instrument scripting without needing to overhaul an existing Python code base.

To generate a Python wrapper for a *.tsp script:

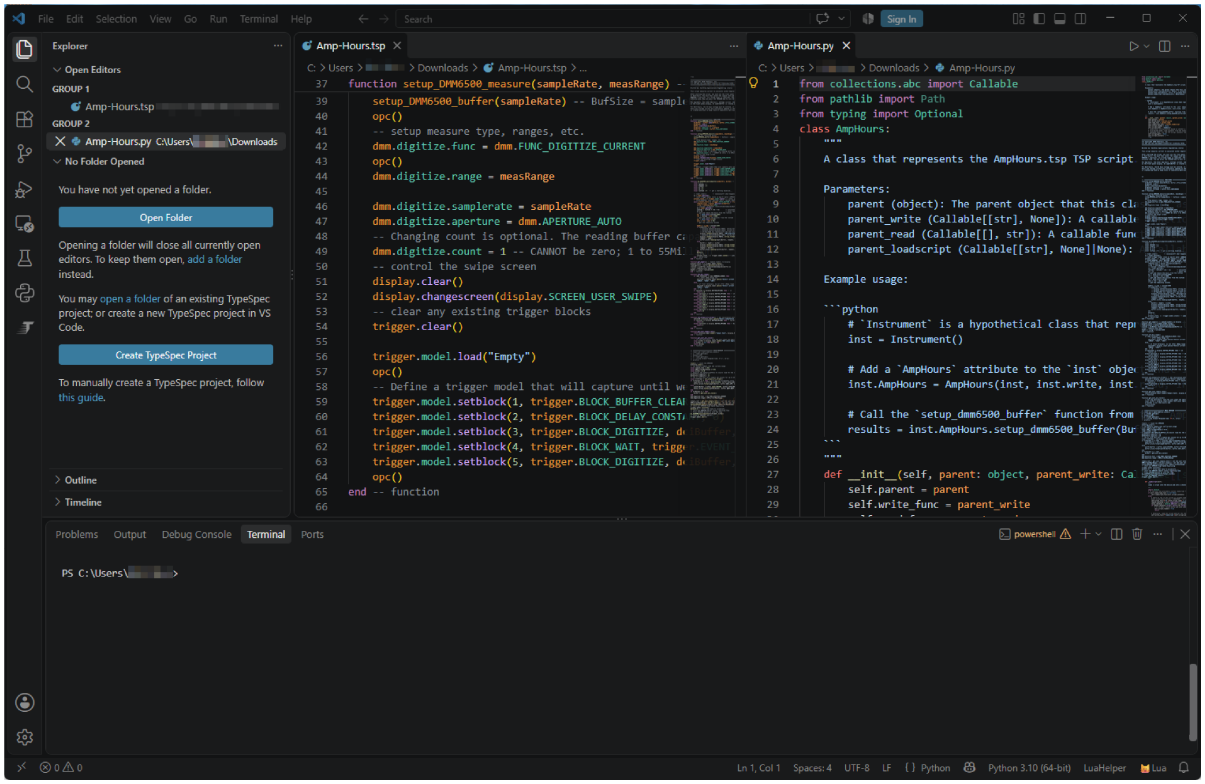
1. Open a *.tsp script.
2. Select the **Generate Python Wrapper from a TSP Script** icon.



3. Select a name for the Python output file, then select **Generate python wrapper file**.



The converted *.py script is displayed next to the *.tsp script. The TSP functions are wrapped into Python methods that can be used or modified within the *.py script.

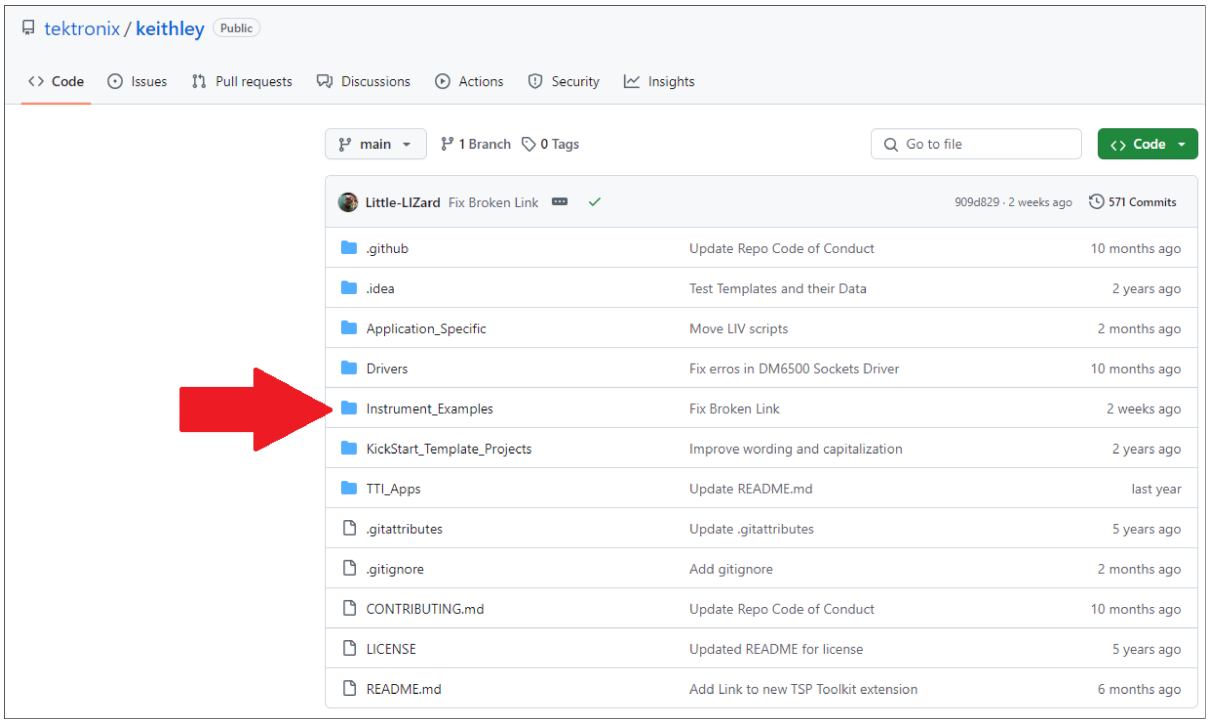


Downloading and using TSP example scripts

Example TSP Scripts are available for download on the [TSP GitHub Repository](#).

To download and use a script:

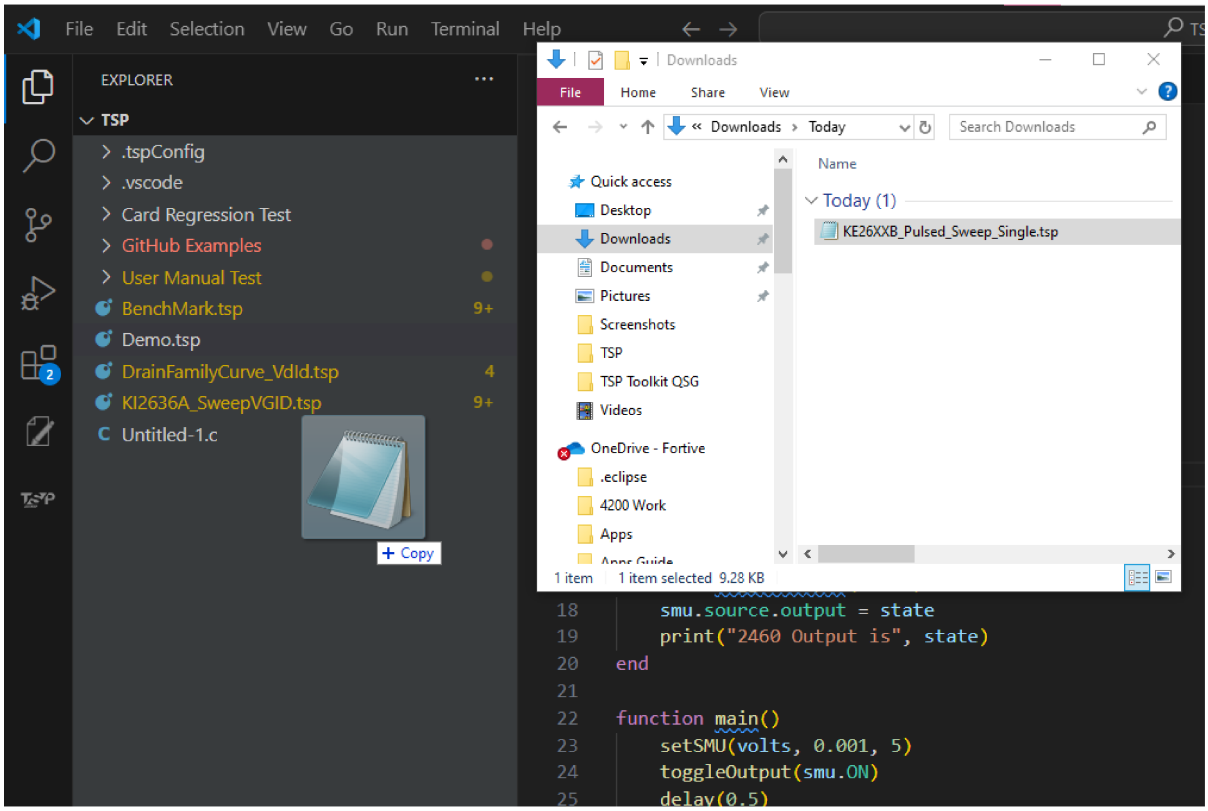
1. Select either the **Application_Specific** or **Instrument_Examples** folder to find TSP scripts.



2. Navigate the folders to find example scripts organized by instrument and application.
3. You can click on a script to view the code within GitHub. Select the download icon to copy the script to your computer.



4. When the download is finished, copy the script to your TSP Toolkit Workspace file location.



Additional resources and tutorials

Application note: How to write scripts for TSP

Tektronix TSP GitHub script example repository

TSP page on Tek.com

TSP Toolkit feature walkthrough video

TSP Toolkit product page

TSP video series