Tektronix®

Visual Trigger Model Construction with TSP Toolkit TriggerFlow

APPLICATION NOTE

Introduction

Global semiconductor demand is skyrocketing, driven by the explosion of generative AI and high bandwidth memory (HBM) solutions. This historical boom in the industry requires manufacturers to increase test throughput to meet demand without sacrificing precision or accuracy. Adopting high density, fully automated parallel testing solutions allows high demand production requirements to be met but complicates testing with the need for channel synchronization and scaling in the future.

The Tektronix MP5000 Series Modular Precision Test System is designed to meet parallel testing needs. The MP5103, a high density 1U mainframe, can be equipped with 3 modular source measure units (SMUs) and/or power supply units (PSUs) for up to 6 independent channels. The MP5103 supports Test Script Processor (TSP™), which makes expansion easy through TSP-Link ™ to connect up to 32 mainframes.

Unlocking the full potential of the MP5000 Test System and its on-instrument test script processor has never been easier. Tektronix's very own TSP scripting tool TSP Toolkit is equipped with real-time error checking, intelligent autocompletion, inline help, an on-instrument debugger, and automated script generation, which makes writing TSP scripts to automate the MP5000 a powerful and intuitive experience.

This application note will discuss the latest TSP Toolkit feature: TriggerFlow™, and how it simplifies test synchronization by introducing the ability to generate trigger model code visually via a user interface.

What is a Trigger Model?

The key to precisely timed parallel testing with TSP is the trigger model, which coordinates the actions of each instrument channel. The trigger model on the MP5000 is completely customizable, following a block flow chart style. Users can control the actions and settings of the instrumentation in any order within the trigger model. Various delay and notify blocks create precise timing and handshaking between channels without the need for complex external triggering code.

Trigger Model Blocks

Trigger models are composed of four different block types: Action blocks, Branch blocks, Notify blocks, and Timing blocks.

Action blocks are responsible for the actions the instrument will perform during the test sequence. Branch blocks allow for the creation of loops within the TriggerFlow by branching from one block to another block in the flow. They are used when a model needs to repeat actions or make decisions during testing. Notify blocks can log events or be used alongside Timing blocks to synchronize multiple trigger models.

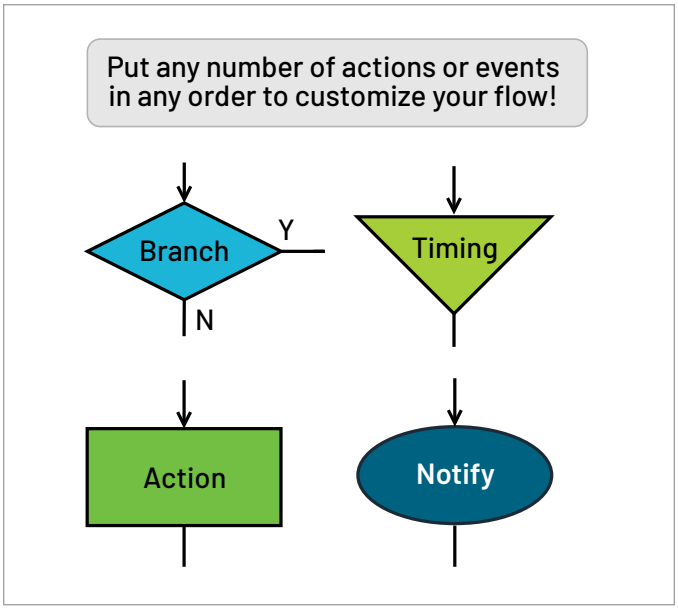

Figure 1: TriggerFlow has 4 block types to customize your test flow

To learn more about the MP5000 Series and trigger models, see Synchronizing Parallel Testing with the MP5000 Series, located on tek.com under MP5000 Series Modular Precision Test System.

Build Trigger Models Visually with TriggerFlow®

The trigger model is the key to precise synchronization, therefore, converting the test procedure is a necessary step. The MP5000 trigger model is designed to simplify this process by emulating a flow chart style. The TSP Toolkit TriggerFlow feature brings this style to life, allowing users to visually build their own flow by dragging trigger model blocks onto an infinite canvas. The blocks are then converted to trigger model TSP commands and provided in an editable TSP script.

Creating a Trigger Model

To use the TriggerFlow feature, open TSP Toolkit, navigate to the Tools menu, and click the the "+" next to the TriggerFlow option to create a new TriggerFlow project.

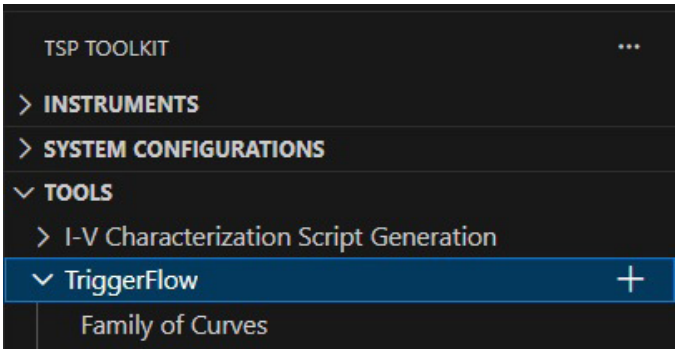

Figure 2: Creating a new TriggerFlow from the TSP Toolkit Tools Menu

When a new TriggerFlow project is created, the TriggerFlow UI will appear with an empty canvas.

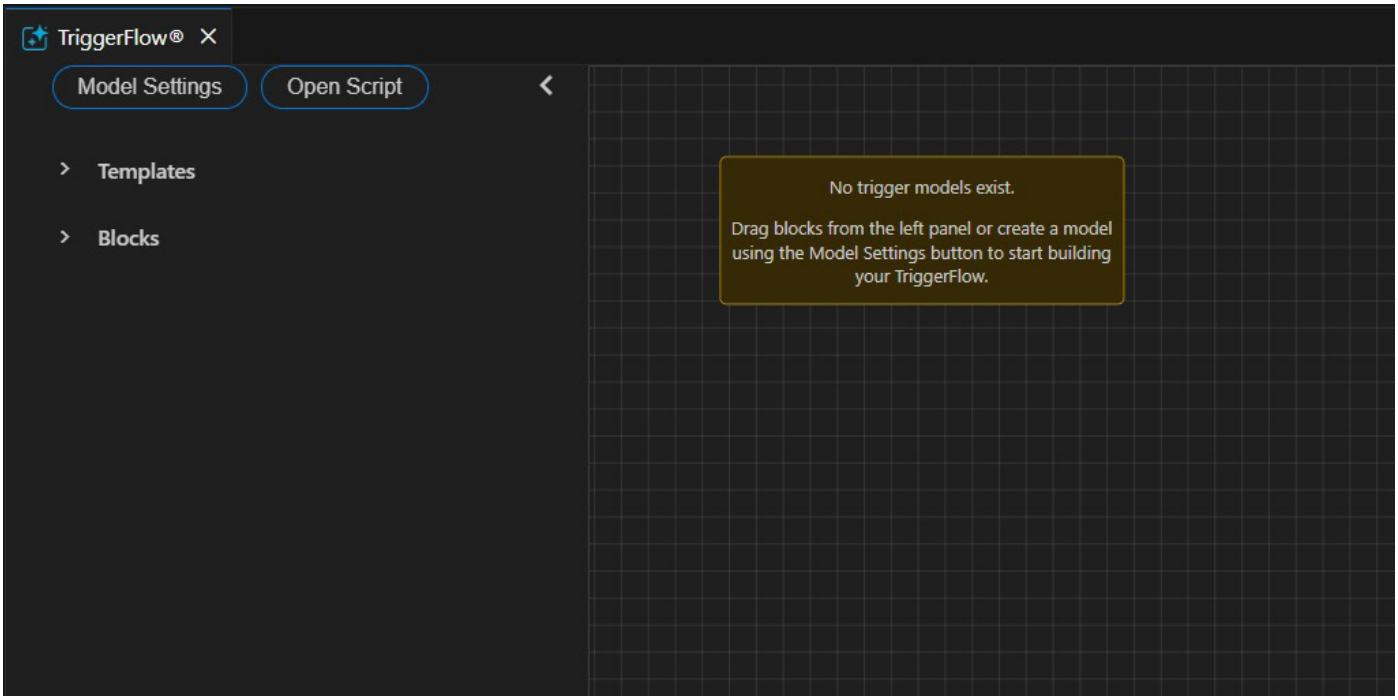

Figure 3: A new TriggerFlow project

To create a new trigger model, drag and drop a block onto the empty canvas or click the “Model Settings” button to add up to six trigger models to the script, one for each channel installed in the MP5000.

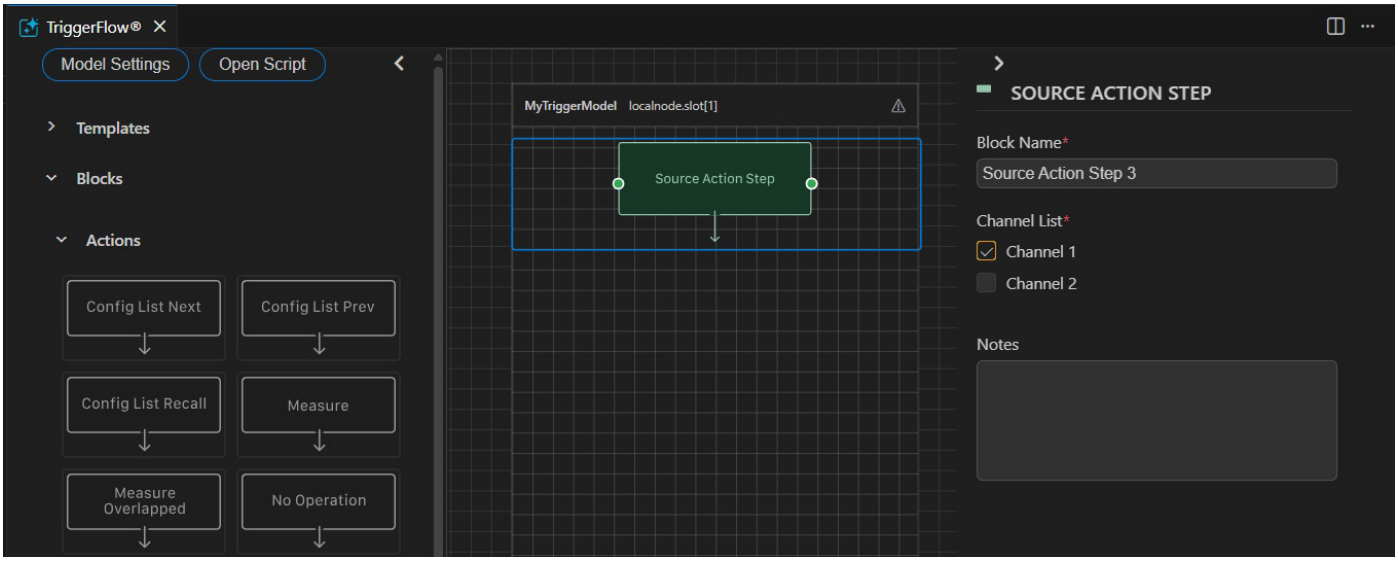

Figure 4: Adding the first block to the canvas creates a new trigger model automatically

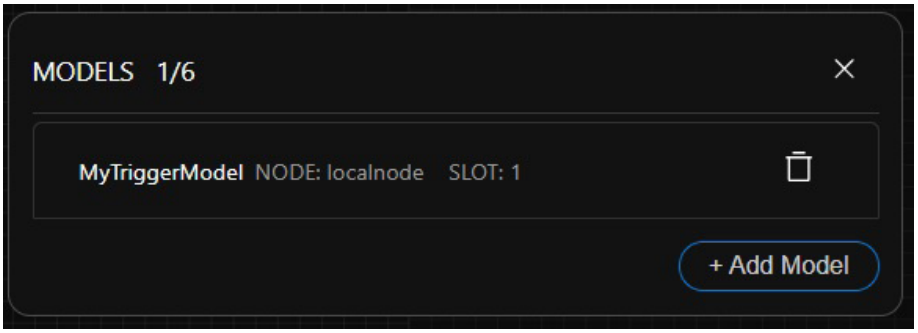

Figure 5: Additional trigger models can be added to the canvas via the Model Settings menu

Continue adding blocks to the canvas until you are satisfied with your test flow.

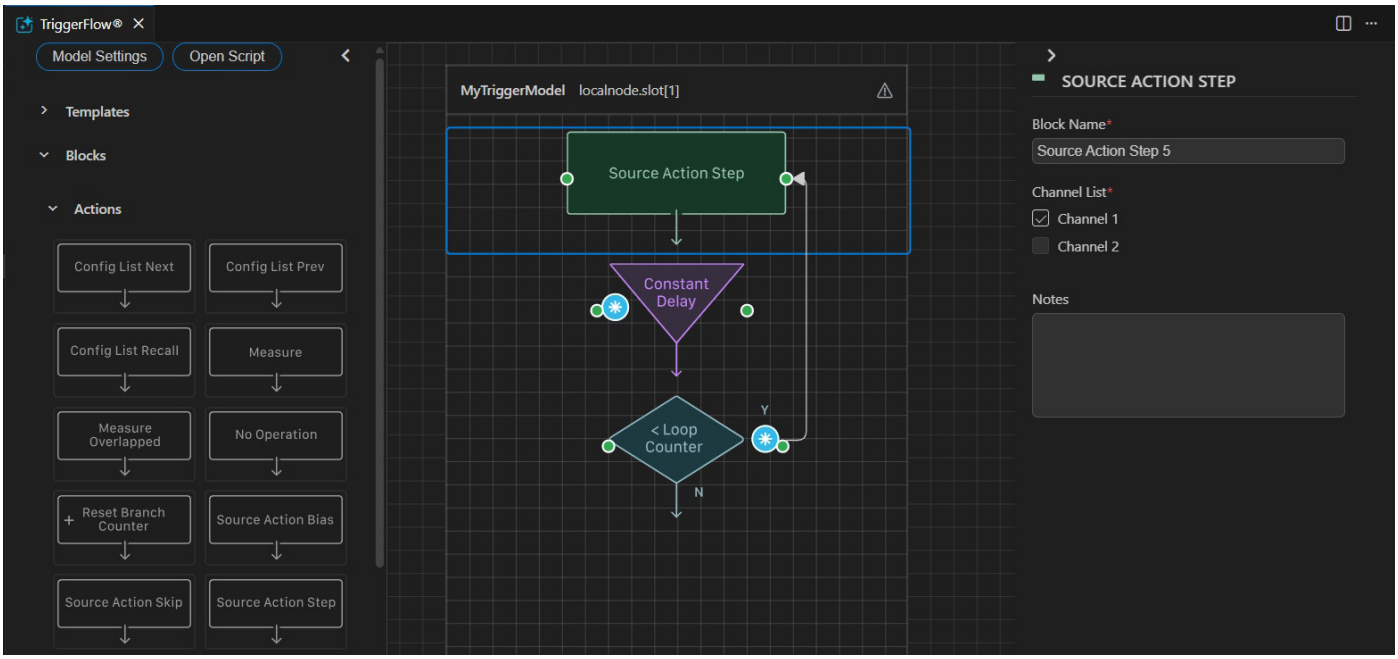

Figure 6: A simple DC sweep created using three blocks

Clicking on a block within the canvas opens the settings pane on the right. This is where each block's parameters can be adjusted. Any required parameters are denoted with a red asterisk.

For Branch blocks, a line can be drawn between the Branch block and the block that is being branched to by clicking the green dots on the block and dragging to the next green dot. Alternatively, the branch to block name can also be specified within the Branch block's parameters.

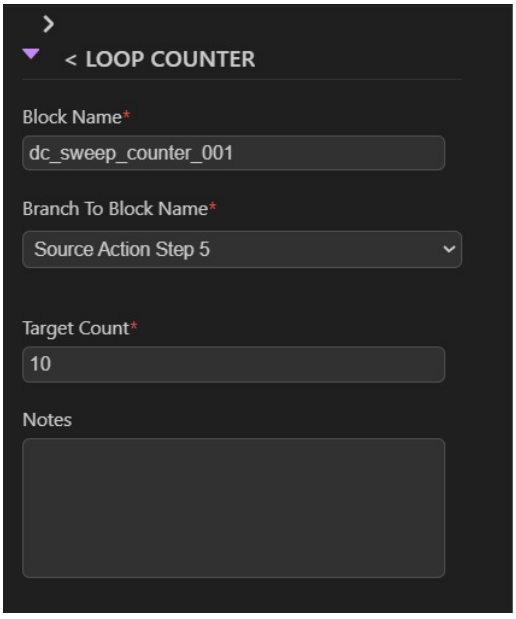

Figure 7: Clicking a block opens the settings where parameters can be edited

To view the resulting TSP script, click the "Open Script" button.

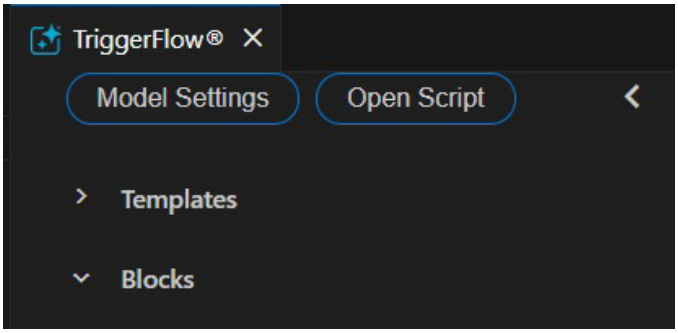

Figure 8: The Open Script button will open the generated TSP script in a new tab

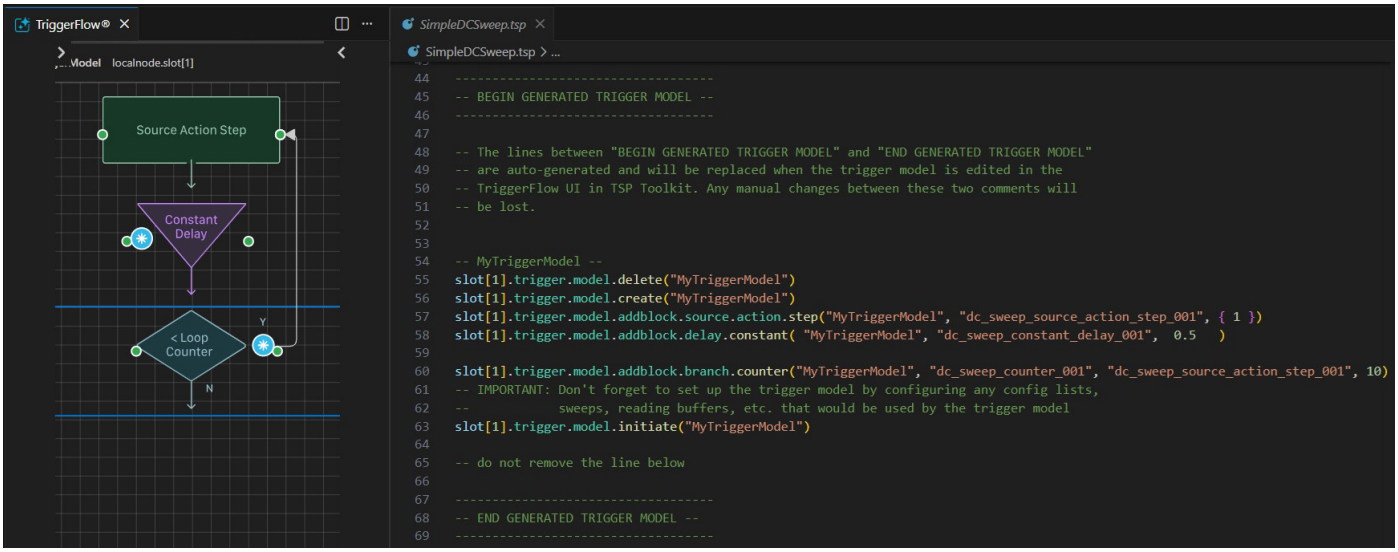

Figure 9: A simple DC sweep and the generated trigger model code

Using Templates

The TSP Toolkit TriggerFlow feature also includes templates for frequently used trigger model configurations that can be added to the canvas or appended to an existing model the same way a single block can be added.

These templates range from common utilities such as loading configuration lists to full applications such as a MOSFET family of curves. When using a template, remember to go into the settings and edit the parameters for each block for your device.

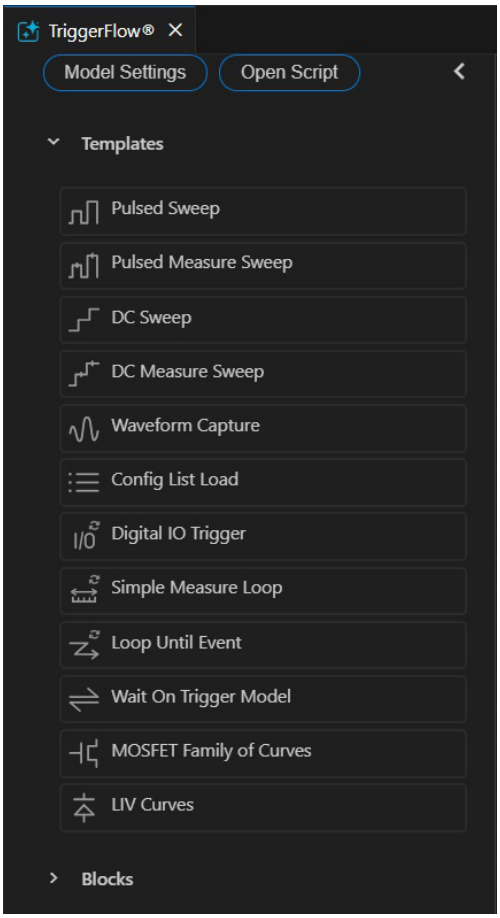

Figure 10: Available built-in TriggerFlow templates

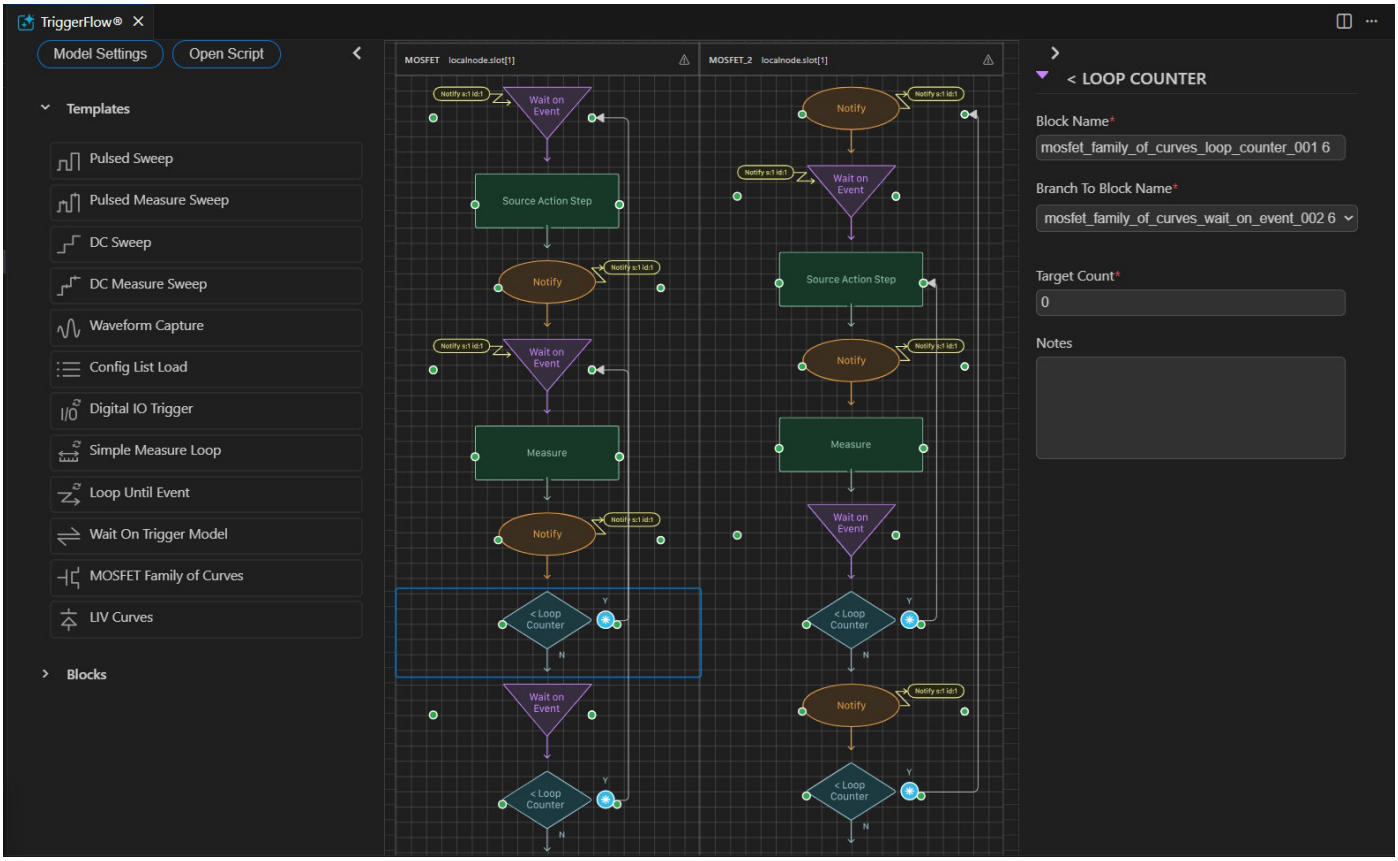

Figure 11: The MOSFET family of curves template automatically creates two new trigger models

Completing the TSP Script

While the TriggerFlow feature automatically generates the trigger model code based on the selections in the user interface, to finish the script, some additional configuration code needs to be added. To assist with this, helper comments which include detailed descriptions and examples are included in the generated script. Many of these examples can be uncommented and edited directly to save time.

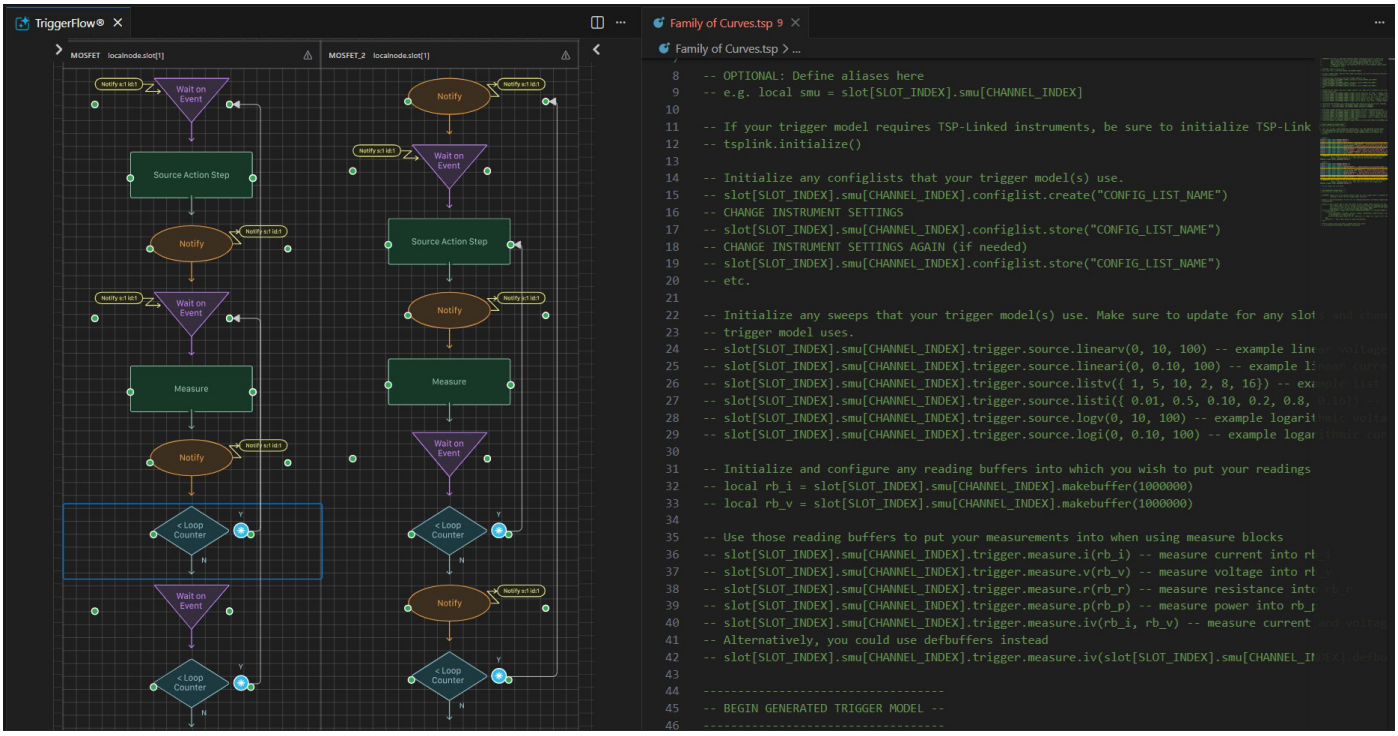

Figure 12: The generated script featuring extensive helper comments and examples for test configuration

These tasks must be completed to have operational code:

1. Source and measure settings must be configured
2. Any config lists, sweeps, or TSP-Link nodes must be initialized
3. Reading buffers must be initialized and configured for trigger model use

Example: MOSFET Family of Curves

I-V characterization is a common test and one of the best ways to ensure that a MOSFET is functioning properly and meets specifications. A MOSFET family of curves allows for the comparison of the IV characterization at the drain terminal when the gate terminal is exposed to a series of voltages or currents.

Test Setup

To generate a MOSFET family of curves using TSP Toolkit, all that is required is a single 2 channel SMU module installed in slot 1 of an MP5103 mainframe. A MOSFET is then connected to the SMU using a test fixture and triaxial cables as shown:

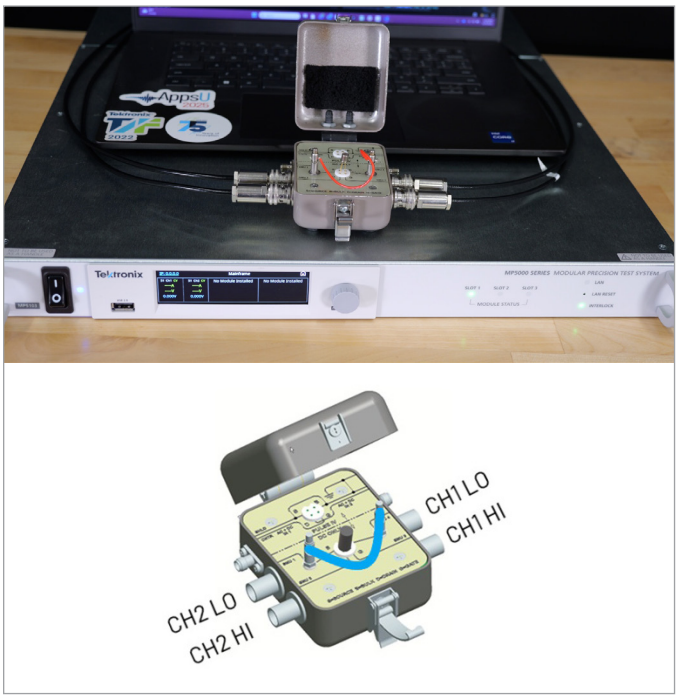

Figure 13: An SD210 N-Channel MOSFET installed in an 8101-PIV Test Fixture connected to an MP5000 SMU Module via Triaxial Cables

Using the MOSFET Family of Curves template as a starting point, add a Wait block with a generator event at the start of each trigger model to ensure that they start at the same time. The action blocks for the first trigger model, MOSFET (the gate trigger model) are set to SMU channel 1 and the action blocks for MOSFET_2 (the drain trigger model) are set to SMU channel 2.

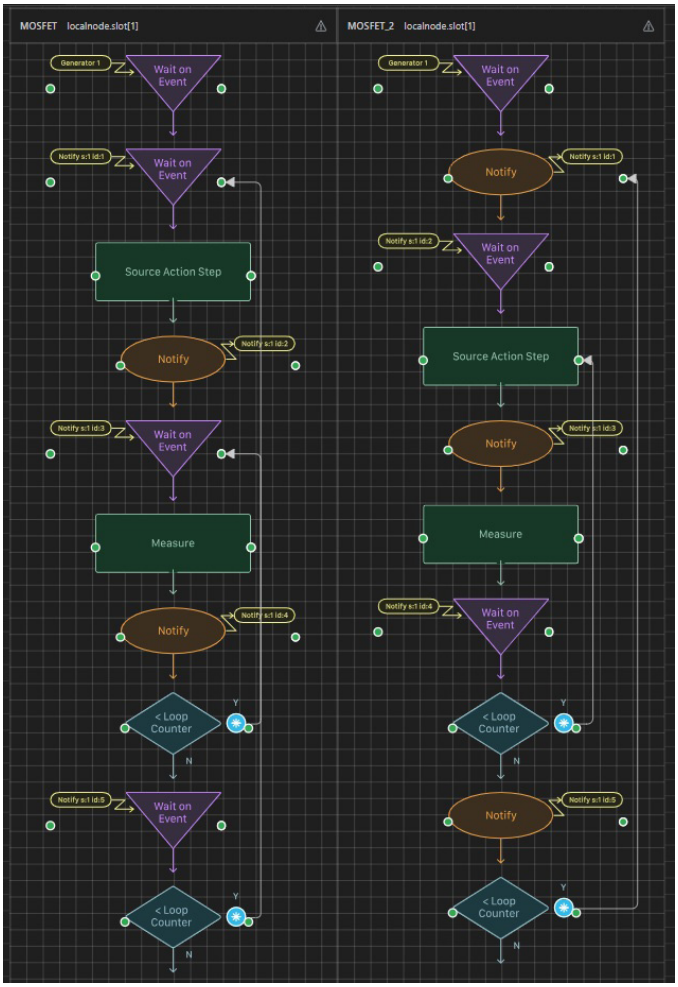

Figure 14: MOSFET Family of Curves trigger models configured for synchronization

Configuring Settings and Buffers Before Generated Code

For the generated trigger model code to be functional, the source settings, measurement settings, sweeps, and buffers need to be configured. This code is inserted before the generated trigger model code.

```
--SMU channel assignment
local gateSMU                           = slot[1].smu[1]
local drainSMU                          = slot[1].smu[2]
local smu_id                            = {gateSMU,drainSMU}
--Configure both channels for a voltage sweep
for i = 1, 2 do
    smu_id[i].reset()
    smu_id[i].source.func               = smu_id[i].FUNC_DC_VOLTAGE
    smu_id[i].sense                     = smu_id[i].SENSE_2WIRE

    smu_id[i].source.rangev             = 2
    smu_id[i].source.limiti             = 1e-3
    smu_id[i].measure.rangei            = 1e-3

    smu_id[i].source.levelv             = 0
    smu_id[i].measure.nplc              = 1

    smu_id[i].defbuffer1.clear()
    smu_id[i].defbuffer1.appendmode     = 1
    smu_id[i].defbuffer2.clear()
    smu_id[i].defbuffer2.appendmode     = 1
end

-- Initialize any sweeps that your trigger model(s) use.
slot[1].smu[1].trigger.source.linearv(1.4, 1.8, 3) --gate linear voltage sweep from 1.4 to 1.8 V in 3 points
slot[1].smu[2].trigger.source.linearv(0, 2, 41) --drain linear voltage sweep from 0 to 2 V in 41 points
-- Configure reading buffers to put your measurements into when using measure blocks
slot[1].smu[1].trigger.measure.iv(slot[1].smu[1].defbuffer1, slot[1].smu[1].defbuffer2) --gate buffers
slot[1].smu[2].trigger.measure.iv(slot[1].smu[2].defbuffer1, slot[1].smu[2].defbuffer2) --drain buffers
```

Assert Trigger and Wait for Model to Complete After Generated Code

Because the trigger models are being synchronized, it is important to avoid accidentally overlapping operations. For this reason, the code needs to be sequenced in a particular way. First, the source outputs are turned on, then the trigger models are initiated, and then the trigger generator is asserted to begin trigger model execution. The waitcomplete() command will wait for the trigger model to finish executing before the data is printed to the terminal and source outputs are turned off.

```
--Turn on the outputs of the SMUs before starting the trigger model
slot[1].smu[1].source.output = 1
slot[1].smu[2].source.output = 1

--Initiate the trigger model(s) AFTER the SMU outputs are turned on
slot[1].trigger.model.initiate("MOSFET")
slot[1].trigger.model.initiate("MOSFET_2")

--Assert trigger to start the trigger model execution.
trigger.generator[1].assert()

--wait for all pending operations (including trigger models) to complete.
waitcomplete()

-- Process and/or print your data as needed after this point.
  print("Vg\t\t Ig\t\t Vd\t\t Id")
    for j = 1, drainSMU.defbuffer1.n , 1 do

print(gateSMU.defbuffer2[j],gateSMU.defbuffer1[j],drainSMU.defbuffer2[j],drainSMU.defbuffer1[j])
    end

--turn off the outputs of the SMUs after the trigger model has completed
slot[1].smu[1].source.output = 0
slot[1].smu[2].source.output = 0
print("done")
```

Results

You can copy/paste this data into Notepad or use TSP Toolkit to export the data and name the file with the *.csv extension so that it can be opened in Microsoft Excel for graphing.

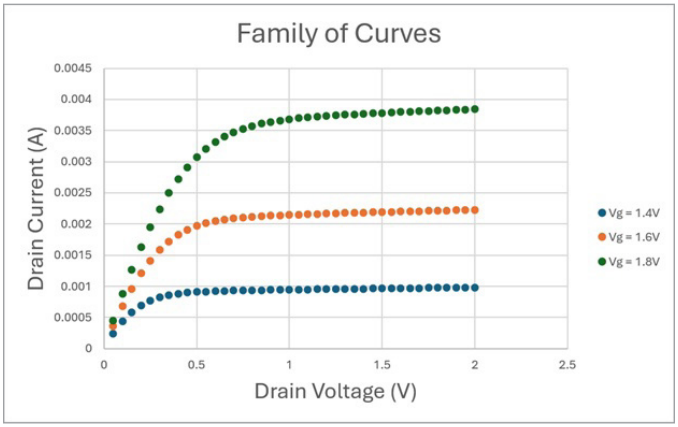

Figure 15: The Resulting Data from the MOSFET Family of Curves Test Plotted in Microsoft Excel

Conclusion

The TSP Toolkit TriggerFlow feature simplifies one of the most powerful features of TSP by transforming code-intensive development into an intuitive visual design experience. Through drag-and-drop model creation, automatic TSP code generation, built-in templates, and extensive script guidance, users can rapidly develop, modify, and deploy synchronized test sequences with reduced development time and a lower learning curve.

Read More

Synchronizing Parallel Testing with the MP5000 Series

Getting Started with the MP5000 Series Modular Precision Test System and Test Automation

Products

MP5000 Series Modular Precision Test System

TSP Toolkit VSCode Extension

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
Sweden* 00800 2255 4835
Switzerland* 00800 2255 4835
Taiwan 886 (2) 2656 6688
Thailand 1 800 011 931
United Kingdom / Ireland* 00800 2255 4835
USA 1 800 833 9200
Vietnam 12060128

* European toll-free number. If not accessible, call: +41 52 675 3777

Rev. 02.2022

Find more valuable resources at TEK.COM

062626 SBG 1KW-74275-0