COM Calculator

User Manual

Version v1.7.5

August 14, 2026

Based on IEEE Std 802.3 Annex 93A/178A COM Calculation Engine

Table of Contents

Chapter 1 Software Overview 7

1.1 Key Features 7

1.2 COM Technical Background 7

1.3 Core Metrics 8

1.4 Additional Metrics 8

1.5 Applicable Scenarios 9

Chapter 2 System Requirements and Installation 10

2.1 System Requirements 10

2.2 Python Dependencies 10

2.3 One-Click Installation (Recommended) 10

2.4 Manual Installation Steps 11

2.5 setup_env.bat Detailed Description 12

Chapter 3 Software Licensing 13

3.1 Trial Mode 13

3.2 View Host ID 13

3.3 Activate Base License 14

3.4 Deactivate Base License 14

3.5 Status Bar License Information 14

3.6 Trial Mode Feature Restrictions 15

3.7 SCPI Remote Control Option License 15

Chapter 4 Interface Overview 17

4.1 Main Window Layout 17

4.2 Menu Bar 17

4.3 Right Panel Result Tabs 17

Chapter 5 Runtime Environment Settings 19

5.1 Open Settings 19

5.2 Mode A: Compiled Executable (Recommended) 19

5.3 Mode B: Full MATLAB 19

5.4 General Settings 20

5.5 Save Settings 20

Chapter 6 File Selection 21

6.1 Select Configuration File 21

6.2 Select Thru Channel S-Parameter File 21

6.3 Aggressor S-Parameter File Panel 22

6.4 Adding FEXT / NEXT Files 22

6.5 Remove / Clear Files 23

6.6 Case Selection 23

Chapter 7 Parameter Override 24

7.1 Panel Description 24

7.2 Edit Override Values 24

7.3 Button Operations 25

7.4 Parameter Sorting 25

7.5 Runtime Behavior 25

7.6 Complete List of Overridable Parameters 26

Chapter 8 Running COM Calculation 29

8.1 Pre-Run Checks 29

8.2 Start Run 29

8.3 During Execution 29

8.4 Completion Detection 30

8.5 Stop Run 30

8.6 Validation Failure Handling 30

Chapter 9 Viewing Results 31

9.1 Summary Tab 31

9.2 Detailed Data Tab 32

9.3 Channel Chart Tab 34

9.4 Report Tab 37

9.5 Run Log Tab 38

Chapter 10 Noise Tolerance Calibration 40

10.1 Calibration Principle 40

10.2 Two Calibration Paths (read this first) 40

10.3 Enable Calibration Mode 41

10.4 Set Calibration Precision 41

10.5 Run Calibration 42

10.6 View Calibration Results 42

Chapter 11 Configuration Management 43

11.1 Save Configuration 43

11.2 Load Configuration 43

11.3 Initialize 44

Chapter 12 FAQ and Troubleshooting 45

Chapter 13 Typical Workflows 48

13.1 Basic COM Calculation Workflow 48

13.2 Parameter Sensitivity Analysis 48

13.3 Noise Tolerance Calibration Workflow 49

13.4 Batch Multi-Case Comparison 49

Chapter 14 Mode Comparison and Selection 50

Chapter 15 SCPI Remote Control 51

15.1 Overview 51

15.2 Starting SCPI Service 52

15.3 IEEE 488.2 Common Commands 53

15.4 SCPI Command Reference 54

15.5 Usage Examples 62

15.6 Error Code Reference 64

15.7 Status Register Details 64

15.8 Typical Automation Workflow 65

Chapter 16 VNA Network Analyzer 67

16.1 Overview 67

16.2 Connecting to an Instrument 67

16.3 Quick Setup 68

16.4 Measurement Operations 68

16.5 SCPI Command Control 69

16.6 Remote SCPI Control of VNA 69

16.7 VNA Control Secondary Development 70

17. Online Update 72

17.1 Steps 72

17.2 Security 72

17.3 SCPI Remote Update 72

17.4 Feedback 73

Appendix A Excel Configuration File Structure 75

Case Definitions 75

Key Parameter Description 75

Appendix B Full SCPI Command Quick Reference 77

New Command Detailed Reference 87

VNA Network Analyzer Commands 89

License Remote Activation (v1.3.1+) 90

Chapter 1 Software Overview

COM (Channel Operating Margin) is a key metric defined in the IEEE Std 802.3 standard for evaluating the signal integrity of high-speed serial links. The COM value comprehensively considers channel insertion loss, return loss, crosstalk, equalizer performance, and other factors, serving as the core criterion for determining whether a physical layer link meets the standard requirements.

This tool is a PyQt6-based graphical user interface frontend for invoking the IEEE 802.3 COM calculation engine (com_ieee8023_4p14p0.m, v4.14.0). The engine is implemented based on IEEE Std 802.3 Annex 93A/178A specifications and supports COM calculation for multiple data rate standards.

1.1 Key Features

- Graphical interface for configuring COM calculation parameters, simplifying the workflow
- Two execution modes: compiled executable (Mode A) and full MATLAB (Mode B)
- Parameter override: directly modify key parameters from the Excel configuration in the GUI without editing Excel
- S-parameter channel chart real-time preview (SDD21 insertion loss / SDD11 / SDD22 return loss)
- Automatic parsing of MATLAB output CSV result files, displaying detailed calculation data
- Intuitive COM result display: PASS (green) / FAIL (red), 28pt large font at a glance
- Support for multi-case batch execution, with progress shown in the status bar
- Noise tolerance calibration (Sigma BBN / HP calibration), calibration results displayed in blue large font
- Run configuration save and load (JSON format), convenient for repeated testing
- PDF report export with run history records

1.2 COM Technical Background

COM is a statistical channel performance evaluation metric defined in IEEE Std 802.3 Annex 93A (100GBASE-KR4/CR4) and Annex 178A (200GBASE/400GBASE, etc.). Its core idea is: given the channel characteristics and transceiver parameters, evaluate whether the link bit error rate (BER) meets the standard requirements by calculating the ratio of signal amplitude to various noise components.

The COM calculation process involves the following key steps:

- **Channel characterization**: Extract differential insertion loss (SDD21), return loss (SDD11/SDD22), and crosstalk (FEXT/NEXT) characteristics from S-parameter files.
- **Equalizer optimization**: Automatically search for the optimal CTLE (Continuous Time Linear Equalizer) and DFE (Decision Feedback Equalizer) settings to maximize the COM value.
- **Noise budget**: Calculate each noise component (ISI residual, crosstalk, random noise, reflection noise, etc.) and aggregate the total noise power.
- **COM calculation**: COM = 20 · log■■(A_s / A_ni), where A_s is the signal amplitude and A_ni is the equivalent noise amplitude. COM ≥ threshold is judged as PASS.

1.3 Core Metrics

Metric	Full Name	Unit	Description & Criteria
COM	Channel Operating Margin	dB	Channel operating margin. COM ≥ 0 dB (or custom threshold) is PASS; higher values indicate better signal quality and greater margin
ERL	Effective Return Loss	dB	Effective return loss. Comprehensively evaluates the impact of channel reflections on signal quality; higher values indicate less reflection and better channel quality
VEO	Vertical Eye Opening	mV	Vertical eye opening. The vertical height of the equalized eye diagram, reflecting the signal amplitude margin
VEC	Vertical Eye Closure	dB	Vertical eye closure. Measures the degree of channel damage to the eye diagram
A_s	Signal Amplitude	mV	Equalized signal amplitude. The numerator in the COM calculation
DER	Detector Error Ratio		Detector error ratio. The equivalent BER at the COM threshold

1.4 Additional Metrics

In addition to the core metrics, the software also extracts the following information from the calculation process:

Metric	Description	Unit
Die-to-Die IL	Die-to-die insertion loss (at Nyquist frequency)	dB
Causality Correction	S-parameter causality correction amount	dB
Truncation Ratio	Time-domain truncation ratio	dB
TP0 AC RMS	AC RMS voltage at test point 0	mV
Unequalized Pulse Peak	Pulse response peak before equalization	mV
σ_hp / σ_bn	Calibration noise values (noise tolerance calibration results)	mV
Run Time	Computation time per case	min

1.5 Applicable Scenarios

This tool is suitable for the following typical application scenarios:

- **Connector/cable evaluation**: Assess whether newly designed connectors or cables meet COM requirements for the target data rate
- **PCB channel design verification**: Verify the signal integrity of PCB trace designs under the target standard
- **System-level compliance testing**: Perform IEEE standard compliance verification on complete links (TX → channel → RX)
- **Parameter sensitivity analysis**: Use the parameter override feature to quickly evaluate the impact of different parameter settings on COM
- **Noise tolerance calibration**: Determine channel noise tolerance parameters for receiver compliance testing
- **Multi-case comparison**: Batch-run different package configurations (e.g., different trace lengths) and compare results

Chapter 2 System Requirements and Installation

2.1 System Requirements

Item	Requirement
Operating System	Windows 10 / 11 (64-bit)
Python	**Must be 3.14** (other versions not supported)
Mode A Dependency	**MATLAB Runtime must be R2024a** (other versions not supported)
Mode B Dependency	Full MATLAB installation (R2024a recommended)
Disk Space	Approx. 500 MB (including MATLAB Runtime)

2.2 Python Dependencies

Package	Version	Purpose
PyQt6	≥ 6.6.0	GUI framework
openpyxl	≥ 3.1.0	Read/write Excel configuration files
scikit-rf	≥ 0.31.0	S-parameter processing and channel chart rendering
matplotlib	≥ 3.7.0	Chart rendering (scikit-rf dependency)

Note: Strict version requirements: Python must be 3.14, MATLAB Runtime (Mode A) must be R2024a. Using other versions will cause the software to malfunction or produce compatibility errors.

2.3 One-Click Installation (Recommended)

The software includes a **setup_env.bat** one-click environment configuration script that automatically completes all installation steps. Simply double-click to run.

```
setup_env.bat
```

The script will automatically perform the following operations:

- **Detect Python 3.14**: Detects via py launcher, PATH, and common installation paths. If not found, automatically downloads and installs Python 3.14.3 (64-bit) from a mirror source.

- **Detect MATLAB Runtime R2024a**: Detects via registry and common installation paths. If not found, automatically downloads (approx. 4 GB) and silently installs (requires administrator privileges).
- **Install pip dependencies**: Runs the runfirst.py script to automatically install PyQt6, openpyxl, scikit-rf, and other required Python packages.

> **Tip:** setup_env.bat supports 10 global mirror nodes (Global CDN, China, Japan, Korea, Germany, UK, Australia, India, France, Main Site) with automatic fastest source detection. MCR download supports BITS resume. If network access is restricted, you can install manually first and then run the script.

2.3.1 Package Directory Structure

After installation, the software directory structure is as follows:

```
COM_Tool_dist/
■■■ setup_env.bat            ← Double-click to install MCR + launch
■■■ COM_Tool/
    ■■■ COM_Tool.exe
    ■■■ demo/            ← Demo scenarios + S-parameters
    ■■■ resources/           ← Resource files (Excel configs/S-params/icons),
user-browsable and replaceable
    ■■■ _internal/           ← Runtime dependencies (do not modify)
        ■■■ help/               ← User manual PDF
```

2.4 Manual Installation Steps

For manual installation, follow these steps:

1. Install **Python 3.14** (must be 3.14, 64-bit) and add it to the system PATH.

   Download: https://www.python.org/downloads/

2. Open a command line terminal and navigate to the project directory:

```
cd com_tool
```

3. Install Python dependencies:

```
pip install -r requirements.txt
```

4. (Mode A) Install **MATLAB Runtime R2024a** (must be R2024a, download from the MathWorks website).

   Download: https://www.mathworks.com/products/compiler/matlab-runtime.html

5. Launch the software:

```
python main.py
```

> **Note:** Python version must be 3.14. Using 3.13 or 3.15 or other versions will cause compiled modules (.pyd) to fail to load. MATLAB Runtime version must be R2024a; other versions are incompatible with the compiled executable.

2.5 setup_env.bat Detailed Description

setup_env.bat is the one-click environment configuration script. Its execution flow is:

Step	Action	Description
Step 1	Detect Python 3.14	Three detection methods: py launcher, PATH, common paths
Step 2	Install Python 3.14	Only runs if not detected; 10 global mirror nodes with auto fastest source detection
Step 3	Detect MCR R2024a	Checks registry and default installation paths
Step 4	Install MCR R2024a	Only runs if not detected; requires admin privileges; approx. 4 GB
Step 5	Install pip dependencies	Runs runfirst.py to install all Python packages

After the script finishes, it displays an environment configuration summary confirming the status of Python and MCR. If any item shows "Not Ready", please follow the instructions to resolve it manually.

> **Tip:** If using Mode B (full MATLAB), MATLAB Runtime is not required, but you still need Python 3.14 and must install dependencies via setup_env.bat or pip.

Chapter 3 Software Licensing

This software uses a Host ID-based licensing mechanism. On first launch, it operates in trial mode with a limited number of runs. For long-term use, please contact the vendor to obtain a license key.

3.1 Trial Mode

When launched for the first time, the software automatically enters trial mode. In trial mode, a limited number of COM calculations can be performed, and the status bar will display the remaining count.

→ GUI feedback: Status bar displays orange text "Trial Mode — N runs remaining"

3.2 View Host ID

1. Click the menu bar "License" → "License Management..."

   → GUI feedback: "License Management" dialog opens

2. The dialog displays the Host ID (unique identifier) of this machine.
3. Click the "Copy" button to copy the Host ID to the clipboard.
4. Send the Host ID to the vendor to obtain the corresponding license key.

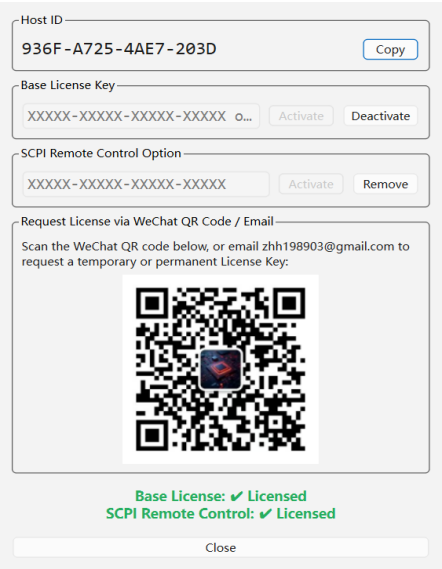

Figure 3-1 License Management Dialog (Host ID, Base License, SCPI Option License, QR Code)

3.2.1 Apply for License via QR Code

In the "License Management" dialog, you can scan the QR code below to contact the vendor and apply for a temporary or permanent license key. The QR code is also displayed in trial expiration or feature restriction dialogs for quick access.

3.3 Activate Base License

Enter the base license key (permanent or temporary) in the "Base License Key" input area:

1. In the "License Management" dialog, paste the received key into the "Base License Key" input field.

2. Click the "Activate" button.

   → GUI feedback: Success: Green message "License activated successfully!", status bar turns green "Licensed"

   → GUI feedback: Failure: Red warning "Invalid license key, please check and try again"

> **Note:** Permanent and temporary keys are mutually exclusive: you cannot enter a temporary key when a permanent license is active, and vice versa. You must deactivate the current license before switching.

3.4 Deactivate Base License

1. In the "License Management" dialog, click the "Deactivate" button in the "Base License Key" area.

2. Click "Yes" in the confirmation dialog.

   → GUI feedback: License deactivated, software reverts to trial mode

> **Note:** When deactivating the base license, the SCPI option license will also be cleared.

3.5 Status Bar License Information

Status	Color	Description
Licensed	Green	License valid, unlimited runs, all features available
Trial Mode	Orange	In trial, shows remaining count, some features restricted
Unlicensed	Red	Trial runs exhausted, activation required

3.6 Trial Mode Feature Restrictions

Users who have not activated any license key (including permanent and temporary keys) are in trial mode. In addition to the run count limit, the following features are also restricted:

Restricted Feature	Trial Mode	Licensed
FEXT aggressor S-parameter files	Max 1	Unlimited
NEXT aggressor S-parameter files	Max 1	Unlimited
Noise tolerance calibration mode	Not available	Available

When a trial mode user attempts to use a restricted feature (e.g., adding a second aggressor file or enabling calibration mode), the software will display a prompt dialog directing the user to contact the vendor for a license key.

→ GUI feedback: "Feature Restricted" dialog appears, prompting to contact vendor for a key and enter it in [License - License Management]

Note: Activating any type of key (permanent or temporary) will unlock all feature restrictions. A temporary key adds run counts while also removing feature restrictions.

3.7 SCPI Remote Control Option License

The SCPI remote control feature is an independent option that requires a separate SCPI option license key. The "License Management" dialog has a dedicated "SCPI Remote Control Option License" input area.

3.7.1 License Matrix

License Type	Calibration/Multi-file	SCPI Remote Control	Run Count
No Key	Restricted	Not available	Trial N runs
Base Permanent Key	Unlimited	Not available	Unlimited
Base + SCPI Permanent Key	Unlimited	Available	Unlimited
Temporary Key	Unlimited	Available	+N runs

3.7.2 Activate SCPI Option

1. First ensure the base permanent key is activated ("Base License Key" area shows "Licensed").

2. In the "SCPI Remote Control Option License" area, paste the SCPI option key.

3. Click the "Activate" button.

→ GUI feedback: Success: Message "SCPI remote control option activated!"

→ GUI feedback: Base key not activated: Message "Please activate the base license key first"

3.7.3 Deactivate SCPI Option

The SCPI option can be deactivated independently without affecting the base license:

1. In the "SCPI Remote Control Option License" area, click the "Deactivate" button.

2. Click "Yes" in the confirmation dialog.

→ GUI feedback: Only the SCPI option is deactivated; the base license remains unchanged

3.7.4 Important Notes

• The SCPI option key can only be installed after the base permanent key is activated • The SCPI option can be deactivated independently without affecting the base license • When deactivating the base license, the SCPI option license is also cleared • Temporary keys automatically include SCPI privileges; no separate SCPI option key is needed • SCPI license is validated at runtime: every INIT/MEAS command checks authorization, not just at server startup

Chapter 4 Interface Overview

The main window uses a left-right split layout, with the input configuration area on the left and the results display area on the right.

4.1 Main Window Layout

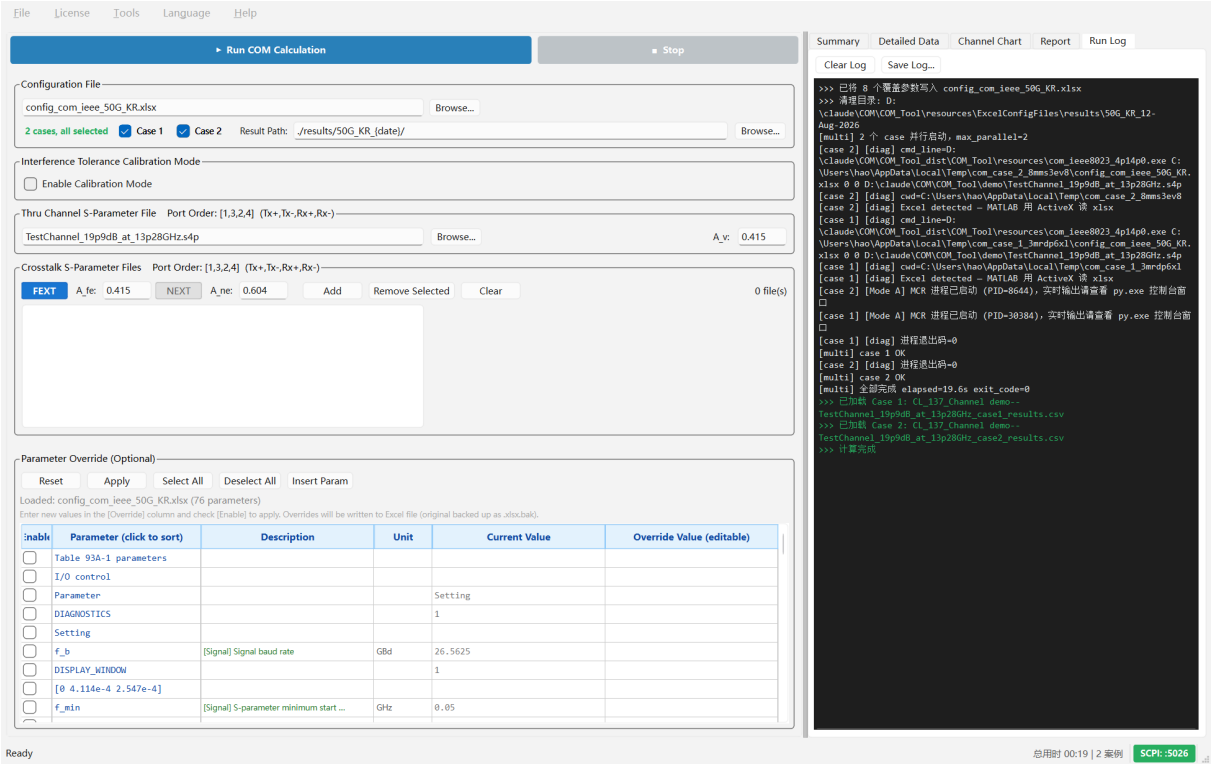

Figure 4-1 Main Interface (Left: File Selection + Parameter Override, Right: Channel Chart + Result Tabs)

4.2 Menu Bar

Menu	Functions
File	Settings, Save/Load Configuration, Initialize, Exit
License	License Management
Tools	SCPI Remote Control Service Start/Stop (Ch.15), Connect VNA (Ch.16)
Help	About, User Manual

4.3 Right Panel Result Tabs

Tab	Content
Summary	COM/ERL/VEO core metrics, PASS/FAIL judgment, case comparison table
Detailed Data	Full CSV field display per case
Channel Chart	SDD21/SDD11/SDD22 frequency-domain curves
Report	Run history tree, PDF export
Run Log	Real-time log output, color-coded by level

Chapter 5 Runtime Environment Settings

Before first use, you need to configure the MATLAB runtime environment.

5.1 Open Settings

Click the menu bar "File" → "Settings..."

→ GUI feedback: "Runtime Environment Settings" dialog opens

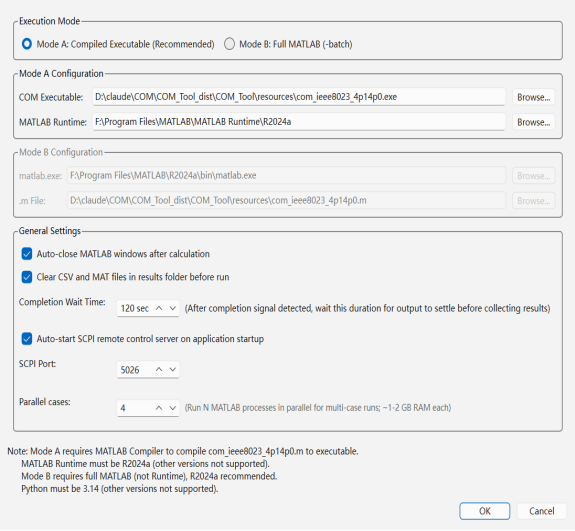

Figure 5-1 Runtime Environment Settings Dialog (Mode A/B Selection, Path Configuration, General Settings)

5.2 Mode A: Compiled Executable (Recommended)

Mode A uses a standalone executable compiled with MATLAB Compiler. No full MATLAB installation is needed; only the free MATLAB Runtime is required.

1. Select the "Mode A: Compiled Executable (Recommended)" radio button.

   → GUI feedback: Mode B configuration area becomes grayed out and non-editable

2. Click the "Browse..." button next to "COM Executable" to select com_ieee8023_4p14p0.exe.

3. Click the "Browse..." button next to "MATLAB Runtime Path" to select the MCR **R2024a** installation directory.

   Example: F:\Program Files\MATLAB\MATLAB Runtime\R2024a

> **Note:** MATLAB Runtime **must be R2024a**. Using other versions (e.g., R2023b, R2024b) will cause the executable to fail to launch or produce runtime errors. MATLAB Runtime can be downloaded for free from the MathWorks website, or automatically installed via setup_env.bat.

5.3 Mode B: Full MATLAB

Mode B directly invokes the locally installed MATLAB via the matlab -batch command to execute .m scripts. This mode supports real-time log output.

1. Select the "Mode B: Full MATLAB (-batch)" radio button.

   → GUI feedback: Mode A configuration area becomes grayed out

2. Click the "Browse..." button next to "matlab.exe Path" to select matlab.exe from your MATLAB installation.

   Example: C:\Program Files\MATLAB\R2024a\bin\matlab.exe

3. Click the "Browse..." button next to ".m File" to select com_ieee8023_4p14p0.m.

5.4 General Settings

Setting	Default	Description
Auto-close MATLAB window after calculation	Checked	Automatically terminate MATLAB popup (Mode A)
Clear result directory before run	Checked	Clears the directory specified by RESULT_DIR (or the results subfolder if not set) to avoid old results interfering
Completion wait time (seconds)	120	Silent period timeout, range 10 — 600 seconds
Auto-start SCPI service	Unchecked	Automatically start SCPI remote control on app launch
SCPI Port	5026	TCP port for auto-started SCPI service
Parallel cases	1	Run N MATLAB processes in parallel for multi-case runs (1=serial). Each instance ~1-2 GB RAM; calibration mode forces 1 (binary search cannot be split). Range 1-16.

5.5 Save Settings

After configuration, click "OK" to save settings. Click "Cancel" to discard changes.

→ GUI feedback: Settings saved to system registry (QSettings), automatically loaded on next launch

Chapter 6 File Selection

The file selection panel is located in the middle-left of the main window, used to specify input files required for COM calculation.

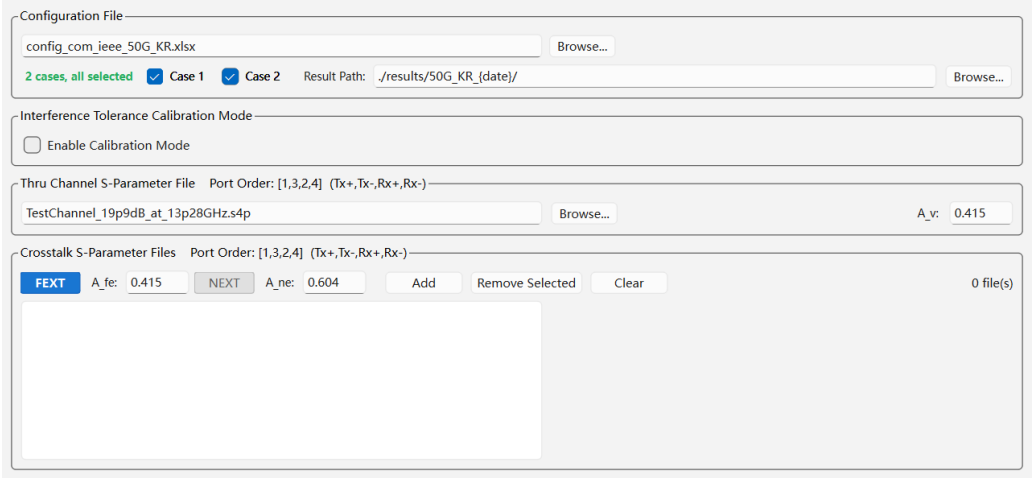

Figure 6-1 File Selection Panel (Config / Thru / Aggressors / Cases)

6.1 Select Configuration File

The configuration file is in Excel format (.xlsx) and contains all parameter settings needed for COM calculation.

1. Click the "Browse..." button next to "Configuration File (.xlsx)".
2. Select the Excel configuration file in the file dialog and click "Open".

   → GUI feedback: Path is displayed in the input field

   → GUI feedback: Parameter override panel automatically reads the Excel file and loads the parameter list

   → GUI feedback: Status label shows "Loaded: xxx.xlsx (N parameters)"

   → GUI feedback: Case selection area appears, showing Case 1, Case 2... checkboxes

6.2 Select Thru Channel S-Parameter File

The thru channel is a 4-port S-parameter file (.s4p) that describes the frequency-domain characteristics of the channel under test. This file is the core input for COM calculation.

1. Click the "Browse..." button next to "Thru S4P (.s4p)".
2. Select the .s4p file and click "Open".

   → GUI feedback: Path is displayed in the input field

   → GUI feedback: Channel chart tab automatically plots SDD21 (blue), SDD11 (red), SDD22 (orange) curves

The **A_v** input field to the right of the Browse button displays the thru channel signal amplitude (in V) from the Excel configuration. This value is automatically read when loading the configuration file; users can modify it directly, and it will be automatically written back to Excel at runtime.

S-Parameter File Requirements

S-parameter files must meet the following requirements:

- Format: Standard Touchstone .s4p file (4-port S-parameters)
- Port definition: 4 ports corresponding to the differential pair TX+/TX-/RX+/RX-
- Port order: Defined by the "Port Order" parameter in the Excel configuration (default [1 2 3 4]), i.e., Port 1=TX+, 2=TX-, 3=RX+, 4=RX-
- Frequency range: Should cover DC to at least 2x the Nyquist frequency
- Frequency step: Should not exceed the Delta_f value in the Excel configuration

> **Tip:** The port order of S-parameter files is very important. Different VNA (Vector Network Analyzer) manufacturers may use different port numbering conventions. Please verify that the Port Order in the Excel configuration matches the actual port numbering used during measurement. Common port orders are [1 2 3 4] (default) and [1 3 2 4].

6.3 Aggressor S-Parameter File Panel

The aggressor S-parameter file panel combines FEXT (Far-End Crosstalk) and NEXT (Near-End Crosstalk) files in the same area, with **[FEXT]** / **[NEXT]** toggle buttons at the top to select the active list. Both types are .s4p format and multiple files can be added for each.

The **A_fe** input field next to the [FEXT] button and the **A_ne** input field next to the [NEXT] button display the FEXT and NEXT aggressor signal amplitudes (in V) respectively. These values are automatically read from the Excel when loading the configuration file; users can modify them directly, and they will be automatically written back to Excel at runtime.

The currently active button is highlighted in blue, while the inactive button is gray. The Add, Remove Selected, and Clear buttons always operate on the currently visible list, and the count label shows the number of files in the current list.

6.4 Adding FEXT / NEXT Files

1. Click the **[FEXT]** or **[NEXT]** button at the top of the panel to switch to the target list.
2. Click the "Add" button.
3. Select one or more .s4p files in the file dialog (multi-select supported).

   → GUI feedback: File list updates, showing the added file names

   → GUI feedback: Count label updates to "N files"

→ GUI feedback: Channel chart overlays corresponding curves (FEXT green series / NEXT cyan series)

6.5 Remove / Clear Files

- "Remove Selected": Select one or more items in the current visible list, click "Remove Selected" to remove them.
- "Clear": Click "Clear" to remove all files from the current list.

> **Tip:** Switching between [FEXT] / [NEXT] buttons does not affect files in the other list; the two lists are independent.

6.6 Case Selection

The Excel configuration file may contain multiple cases. After loading the configuration file, case checkboxes appear at the bottom of the panel, allowing you to select which cases to run.

→ GUI feedback: All selected: info label shows green; partially selected: orange; none selected: red

> **Note:** At least one case must be selected to run COM calculation.

Chapter 7 Parameter Override

The parameter override panel allows you to temporarily override certain key parameter values without modifying the original Excel file. Overridden parameters will be written to the Excel at runtime (a backup .xlsx.bak is automatically created on the first write; subsequent overrides will not overwrite the backup).

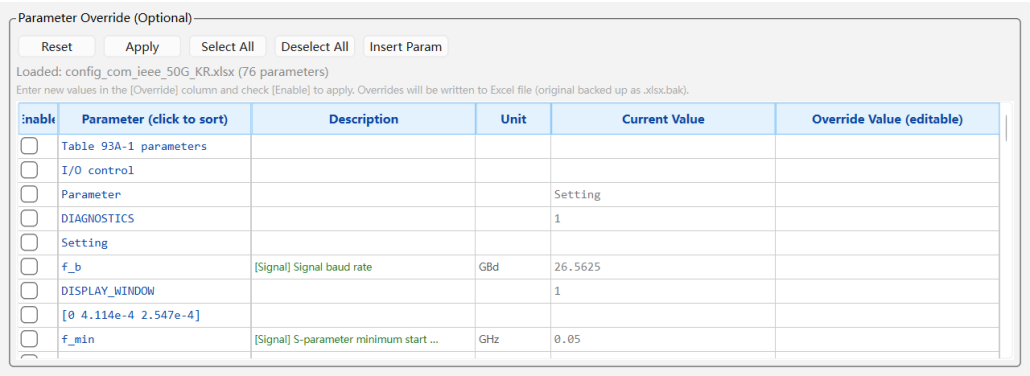

Figure 7-1 Parameter Override Panel (Enable / Param / Description / Unit / Current / Override)

7.1 Panel Description

The parameter override panel is displayed as a table with the following columns:

Column	Description
Enable	Checkbox; when checked, the parameter in this row will take effect at runtime
Parameter	Parameter identifier name from the Excel
Description	Description of the parameter
Unit	Physical unit of the parameter
Current Value	Original value read from the Excel (displayed in gray italic)
Override Value	User-entered new value (red font, yellow background)

7.2 Edit Override Values

1. Double-click or single-click the cell in the "Override Value" column and enter the new value.

   → GUI feedback: The "Enable" checkbox for that row is automatically checked

→ GUI feedback: Cell background turns yellow, font turns red, indicating the value has been modified

2. Press Enter to confirm the input, or click elsewhere to end editing.

7.3 Button Operations

Button	Function
Select All	Check the "Enable" checkbox for all rows
Deselect All	Uncheck all rows
Restore	Clear all override values, reverting to default values from Excel
Apply	Write override values of checked rows to the "Current Value" column
Insert Param	Append a new keyword row to the xlsx template (e.g., AUTO_TFX)

The "Insert Param" button opens a dialog to add a keyword row missing from the xlsx template but read by the .m engine. Common cases: AUTO_TFX for the ERL=2 workflow, or extra parameters for the Module Input flow.

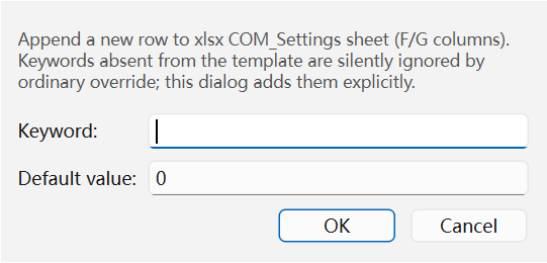

Figure 7-2 Insert Param Dialog (new keyword + default value)

7.4 Parameter Sorting

Click the "Parameter" column header to toggle between alphabetical order and original order.

7.5 Runtime Behavior

When clicking "Run", all parameters with "Enable" checked and an override value filled in will be written to the Excel configuration file. A backup (.xlsx.bak) is automatically created only on the first write; subsequent overrides will not overwrite the backup, ensuring .xlsx.bak always preserves the original file.

Note: Parameter override modifies the Excel file content. To restore, use the automatically generated .xlsx.bak backup file. The .bak file is only created on the first override and always corresponds to the original configuration.

7.6 Complete List of Overridable Parameters

The following table lists all parameters supported by the parameter override panel, grouped by category:

Signal Basic Parameters

Parameter	Description	Unit	Notes
f_b	Signal rate (baud rate)	GBd	Symbol rate of the signal, e.g., 212.5 GBd for 200G/lane
f_min	S-parameter minimum start frequency	GHz	Minimum valid frequency for S-parameters, typically set to a small value
Delta_f	Frequency step	GHz	Frequency resolution requirement for S-parameters

Equalizer Parameters

Parameter	Description	Unit	Notes
g_DC	CTLE DC gain list	dB	CTLE equalizer DC gain search range, e.g., "[-1 0 1 2 3]"
f_z	CTLE zero frequency	GHz	Zero frequency of the CTLE transfer function
f_p1	CTLE pole 1 frequency	GHz	First pole frequency of the CTLE transfer function
f_p2	CTLE pole 2 frequency	GHz	Second pole frequency of the CTLE transfer function
g_DC_HP	CTF low-frequency AC gain list	dB	CTF high-pass filter gain search range

Parameter	Description	Unit	Notes
c(0)	TX FFE main cursor minimum		Transmitter feed-forward equalizer main cursor constraint
N_b	DFE fixed tap count	tap	Number of taps in the decision feedback equalizer

Noise Parameters

Parameter	Description	Unit	Notes
sigma_RJ	Random jitter RJ	UI	RMS value of transmitter random jitter
A_DD	DD noise power density	V^2/GHz	Power spectral density of deterministic noise (Dual-Dirac)
eta_0	Noise floor	dB	Receiver thermal noise floor
SNR_TX	Transmitter SNR		Transmitter signal-to-noise ratio
R_LM	Level mismatch ratio		Level matching parameter for multi-level modulation

Threshold and Control Parameters

Parameter	Description	Unit	Notes
COM Pass threshold	COM pass threshold	dB	Default 0 dB, COM ≥ this value is judged as PASS
ERL	Enable ERL calculation	0/1	Set to 1 to enable ERL calculation, 0 to skip
ERL_ONLY	Calculate ERL only	0/1	Set to 1 to calculate ERL only, skip COM
CSV_REPORT	Output CSV results	0/1	Set to 1 to output detailed CSV result files

Parameter	Description	Unit	Notes
RESULT_DIR	Result output directory		Specifies the output path for CSV and result files. Supports {date} placeholder (expands to DD-Mon-YYYY format). Relative paths are resolved from the configuration file directory
z_p select	Case selection		List of case indices to run, e.g., "[1 2 3]"
RX_CALIBRATION	Noise tolerance calibration	0/1	Set to 1 to enable calibration mode
Sigma BBN step	Calibration step size	V	Calibration search step (smaller = more precise)

Chapter 8 Running COM Calculation

8.1 Pre-Run Checks

Before running, the software automatically checks the following conditions:

- A valid configuration file (.xlsx) is selected
- A thru channel S-parameter file (.s4p) is selected
- Executable / MATLAB path is configured in settings
- At least one case is selected
- License is valid, or trial runs remain

If any condition is not met, an error dialog listing all issues will be displayed.

→ GUI feedback: Red error dialog appears, listing each failed check item

8.2 Start Run

After confirming all conditions are met, click the "■ Run COM Calculation" button.

→ GUI feedback: "■ Run COM Calculation" button is grayed out (cannot be clicked again)

→ GUI feedback: "■ Stop" button activates in red

→ GUI feedback: Run log tab automatically switches to the foreground

→ GUI feedback: Log outputs startup information: command line, working directory, process PID

→ GUI feedback: Status bar begins showing elapsed time

→ GUI feedback: If parameters are overridden, log shows write confirmation information

8.3 During Execution

Mode A Characteristics

Mode A uses the compiled executable. Due to MATLAB Runtime limitations, intermediate calculation logs cannot be displayed in real-time in the GUI log panel.

- Log panel shows diagnostic information (command line, PID, exit code) and CSV results
- Full MATLAB output can be viewed in the command line window (py.exe console) used to launch the software

Mode B Characteristics

Mode B directly invokes MATLAB and supports real-time stdout capture.

- Log panel displays MATLAB computation output in real-time

Status Bar Information

During execution, the status bar displays:

- Elapsed time (min:sec)
- Completed cases / total cases
- Average time per case

8.4 Completion Detection

The software detects calculation completion via:

- stdout keyword detection ("run time = X min")
- CSV result file generation detection
- Expected case count auto-detected from Excel config (z_p select); completes immediately when reached

If the process exits without detecting the expected number of results, the software enters a silent period countdown, waiting for possible delayed output. A progress bar and "Click to finish now" button are displayed.

After calculation completes, the software automatically cleans up orphaned processes spawned by MATLAB (EXCEL.EXE, conhost.exe) to prevent resource leaks. This also applies in SCPI remote control mode.

8.5 Stop Run

To stop mid-run, click the "■ Stop" button.

→ GUI feedback: MATLAB process is terminated

→ GUI feedback: Log shows the process exit code

→ GUI feedback: "■ Run" button returns to clickable state

8.6 Validation Failure Handling

If pre-run checks fail, detailed error information is displayed:

→ GUI feedback: Error dialog appears, listing all unmet conditions

Please correct each issue according to the prompts and retry.

Chapter 9 Viewing Results

After calculation completes, results are displayed in the various tabs on the right panel.

9.1 Summary Tab

The summary page displays the COM value in large font (28pt) with color-coded PASS / FAIL indication:

- COM ≥ threshold (default 0 dB) → **Green "PASS ✓"**
- COM < threshold → **Red "FAIL ✗"**

If there are multiple cases, the summary page shows the worst-case COM value (i.e., the minimum COM across all cases). A comparison table also displays the core metrics for each case:

Metric	Description	Unit	Criteria
COM	Channel Operating Margin	dB	≥ threshold: PASS; < threshold: FAIL
ERL	Effective Return Loss	dB	Higher is better (less reflection)
VEO	Vertical Eye Opening	mV	Higher means more open eye diagram
VEC	Vertical Eye Closure	dB	Lower means less channel damage
DER	Detector Error Ratio	—	Equivalent BER
Die-to-Die IL	Die-to-die insertion loss	dB	Total loss at Nyquist frequency
Run Time	Computation time per case	min	—

In the comparison table, PASS metric rows are marked with green background, and FAIL rows with red background, for quick identification of non-compliant items.

> **Tip:** The COM threshold can be modified via the "COM Pass threshold" parameter in the parameter override panel (default 0 dB). Different standards may specify different COM thresholds; please refer to the specific data rate standard requirements.

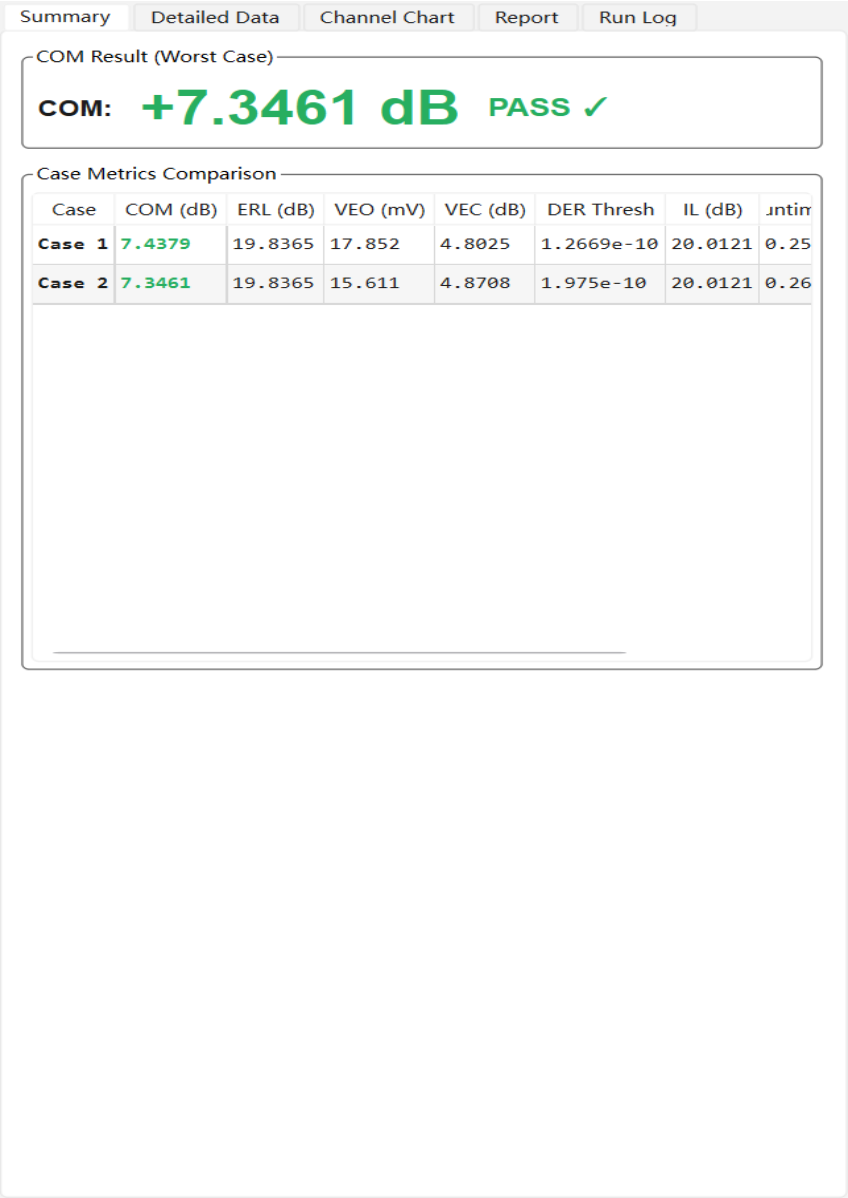

Figure 9-1 Summary Tab (Large COM Value + PASS/FAIL Indicator + Multi-Case Comparison Table)

9.2 Detailed Data Tab

Creates a separate sub-tab for each case, displaying all fields from the MATLAB output CSV file. Data is organized in the following groups for quick access to key information:

Summary | Detailed Data | Channel Chart | Report | Run Log

Case 1 | Case 2

字段名	值
Z22est	101.7183
Z11est	99.8407
VTF_loss_dB_at_Fnq	24.7749
VMA	
VIP_to_VMP_IL_dB_at_Fnq	24.6087
VEO_window_mUI	
VEO_mV	17.852
VEC_dB	4.8025
VCM_DC_HF_mV	29.9
VCM_CD_HF_mV	29.7
uneq_FIR_peak_time	5.3941e-09
TXLE_taps	[0.025 -0.15 0.775 …
Tr_measured_from_step_ps	114.1176
tfx_estimate	
TD_ILN	
tail_RSS	0
steady_state_voltage_weq_mV	98.5279
steady_state_voltage_mV	847.7628
SNR_ISI_XTK_normalized_1_sigma	25.2856
SNR_ISI_est	36.8772
Sigma_ts_sqrd	0.15527
sigma_N	0.00043748
sigma_bn	
sigma_ACCM_at_tp0_mV	
sigma_AC_CCM_at_rxpkg_output_mV	
sgm_xt	0
sgm_TX	0.0011619

Figure 9-2 Detailed Data Tab (CSV fields shown by group)

Group	Fields Included	Description
COM Core Results	COM, VEC_dB, VEO_mV, A_s, A_ni, threshold_DER, EW_UI	Most important calculation results
ERL Results	ERL_min, ERL, ERL1, ERL2	Return loss related metrics

Group	Fields Included	Description
Noise Components	sigma_ISI, sigma_J, sigma_XT, sigma_Rn, ICN, etc.	RMS values (mV) of each noise component
Channel Parameters	IL_db_die_to_die, SCMR_dB, VMC_HF_mV, etc.	Channel characteristic parameters
Equalizer Settings	ctle, gdc, best_G_high_pass, txffe	Optimized equalizer parameters
Run Information	code_revision, config_file, rtmin	Engine version and run time

9.3 Channel Chart Tab

The channel chart displays S-parameter characteristics as frequency-domain curves:

Curve	Meaning	Color
SDD21	Differential Insertion Loss	Blue
SDD11	TX Return Loss	Red
SDD22	RX Return Loss	Orange
FEXT	Far-end crosstalk curves	High-contrast colors
NEXT	Near-end crosstalk curves	High-contrast colors

X-axis: frequency (GHz), Y-axis: magnitude (dB).

> **Tip:** Channel charts are drawn instantly after selecting S-parameter files, without running COM calculation.

Sub-chart Switching

Tabs in the upper-right corner of the channel chart allow switching display modes: All (showing SDD21/SDD11/SDD22 subplots simultaneously), SDD21, SDD11, SDD22. Markers are automatically preserved when switching.

Chart Height Adjustment

Below the channel chart is a gray drag handle (horizontal bar). Drag it up or down to freely adjust the chart height, giving the curves more pixel space. In "All" mode, the three SDD21/SDD11/SDD22 subplots are proportionally adjusted together. Scrolling is available

when exceeding the visible area. Double-click the handle to restore auto height.

Curve Display Toggle

Below each subplot is a checkbox list for independently toggling the display of each curve. When curves are hidden, the Y-axis automatically rescales to the visible curve range.

Interactive Markers

Right-click on the channel chart to open a context menu with the following options:

Menu Item	Function
Add Marker	Add a marker on the nearest visible curve, color matches the curve
Add Markers for All Curves	Add a marker at this frequency for all visible curves in all subplots
Delete Nearest Marker	Delete the marker closest to the right-click position
Clear All Markers	Delete all markers in the current subplot
Edit Marker...	Open a dialog to edit the marker frequency or amplitude
Set Axis Range...	Manually set X/Y axis ranges
Reset Axis Range	Restore to initial axis ranges

Marker labels automatically adjust position to avoid going out of bounds and overlapping. Left-click a marker dot or label to drag the marker along the curve. Right-click a label to directly operate on the corresponding marker.

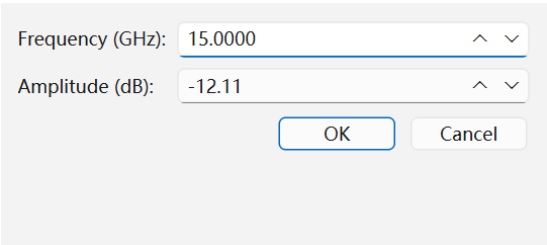

Figure 9-3 Edit Marker Dialog (modify frequency / amplitude, snaps to nearest curve point)

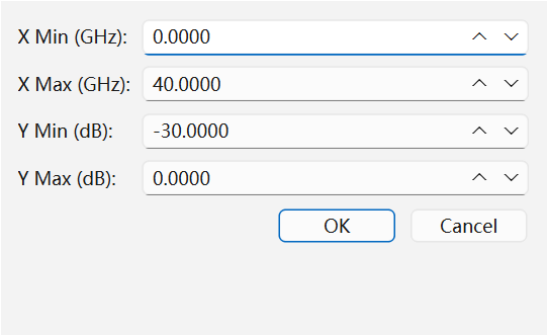

Figure 9-4 Set Axis Range Dialog (X / Y min and max)

Tip: Marker information is included in PDF reports, recording frequency and amplitude. When exporting PDF, the current state of the channel chart is captured, including manually added markers and adjusted axis ranges. Markers are not lost when switching subplot display modes.

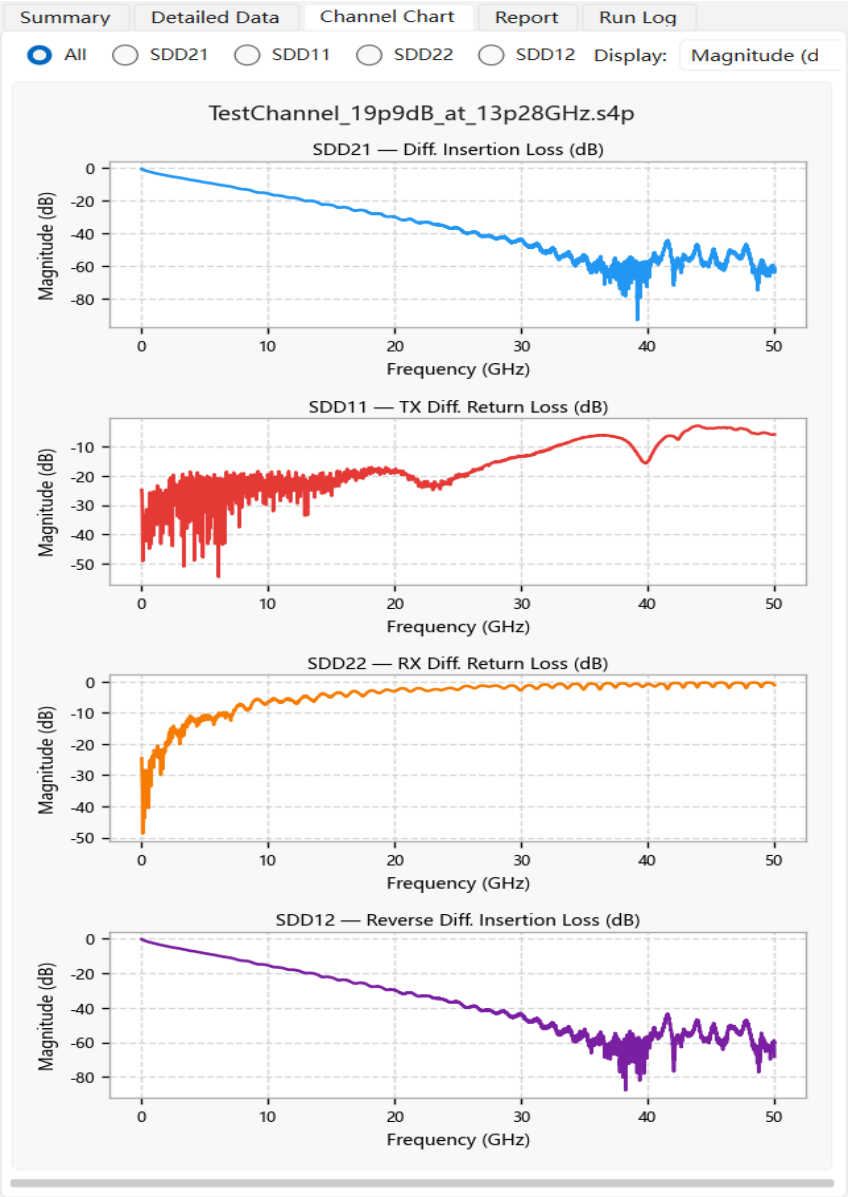

Figure 9-2 Channel Chart Tab (SDD21 Insertion Loss / SDD11 Return Loss / SDD22 Return Loss Curves)

9.4 Report Tab

Figure 9-5 Report Tab (Tree-structured run history + Export PDF button)

The report tab displays each run as a tree structure, including run time, configuration file, result values, and PASS/FAIL judgment. Supports run history across multiple runs.

- Click the "Export PDF..." button to export run results as a PDF report file. The report contains complete calculation results for all cases, suitable for archiving or submission to clients.
- Click the "Clear Report" button to clear all history records.

PDF Report Page Structure

The exported PDF report is structured with the following pages, each major section on its own page:

Page	Content
Cover Page	Title + Test result summary (COM value, PASS/FAIL, duration) + Run conditions + Parameter overrides
Configuration Settings	All parameters from the Excel configuration; short values (≤120 chars) in 4-column compact layout, long values (e.g., MATLAB arrays) in 2-column full-width display, auto-wrapped without truncation
Case Metrics	Case comparison table + all CSV detailed fields for each case
Channel Charts	SDD21/SDD11/SDD22 frequency-domain curve plots + marker info table

Tip: When exporting PDF, the current channel chart state is captured in real-time. Before exporting, manually added markers, adjusted axis ranges, or dragged chart heights will all be reflected in the PDF.

9.5 Run Log Tab

The run log is displayed on a dark background, with different levels color-coded:

Level	Color	Purpose
INFO	White	General information
WARNING	Yellow	Warning messages
ERROR	Red	Error messages

- "Clear Log": Clear the log panel content.
- "Save Log...": Save the log to a text file.

Figure 9-3 Run Log Tab (Color-coded Log: INFO/WARNING/SUCCESS Levels)

Chapter 10 Noise Tolerance Calibration

Noise tolerance calibration is used to determine the channel noise tolerance parameters σ_hp (high-pass calibration noise) or σ_bn (baseband calibration noise). This is an independent operating mode that does not affect the normal COM calculation flow.

The purpose of calibration is to determine the noise level at which COM equals exactly the threshold value under the given channel conditions. This noise value can be used for subsequent receiver compliance testing as a reference baseline for noise tolerance.

10.1 Calibration Principle

The calibration process gradually adjusts the noise parameter (σ_bn), repeatedly calculating COM, until finding the noise level that makes COM exactly equal to the pass threshold. Smaller step sizes yield more precise results but require longer computation time.

- σ_hp: Transmitter high-pass calibration noise (TX calibration noise), reflecting the impact of transmitter noise on the channel
- σ_bn: Baseband calibration noise (BBN calibration noise), reflecting the channel's inherent noise tolerance

10.2 Two Calibration Paths (read this first)

The official IEEE script contains **two mutually exclusive calibration paths**. Which one runs is **decided automatically by the configuration file**, and the script does not announce the switch. The two paths require different input files, and providing the wrong ones fails the run outright.

Path	Trigger condition	Files you must provide
Legacy RX_CALIBRATION	The configuration does not enable RxFFE (receiver feed-forward equaliser)	Exactly 1 FEXT file (used as the noise path); NEXT count must be 0
New PSDRXCAL	The configuration enables RxFFE and FFE_OPT_METHOD = MMSE (true for essentially all 200G/212G configs)	A PSDRXCAL Noise file (its own field in the UI); FEXT / NEXT counts are unrestricted

The switch inside com_ieee8023_4p14p0.m: ticking calibration mode writes RX_CALIBRATION=1; the script then checks whether the FFE optimisation method is MMSE and RxFFE is enabled, and if so **sets PSDRXCAL=1 and clears RX_CALIBRATION back to 0**, taking the new path. FFE_OPT_METHOD defaults to MMSE, and RxFFE counts as enabled whenever ffe_pre_tap_len or ffe_post_tap_len is non-zero. Consequently **a configuration can take the new path without containing a PSDRXCAL row at all**.

> **Tip:** You do not have to work this out yourself: after you tick "Enable Calibration Mode", the UI switches its blue hint text according to the current configuration and shows the **PSDRXCAL Noise** field when it is required. If that field appears, the new path is in use and the field is mandatory.

> **Note: Common failure**: when the new path is used but the PSDRXCAL Noise file is empty, the script indexes past the end of the file list at position (2+num_fext+num_next); the run ends within seconds with **no result at all**, and in Mode A (MATLAB Runtime) the log does not show why. The symptom is "it stopped after a dozen seconds and the Summary tab is blank" - check that field first. To see the real script error, switch temporarily to Mode B (full MATLAB), whose stdout is readable.

The bundled demo scenarios cover both paths for comparison: scenario_2 (50G KR) and scenario_4 (100G C2C) take the legacy path; scenario_6 (C2M module input) takes the new PSDRXCAL path.

10.3 Enable Calibration Mode

> **Note: License requirement**: Calibration mode is only available to users with an activated license key (permanent or temporary). It cannot be enabled in trial mode; a prompt dialog will appear.

1. In the file selection panel, check the "Enable Calibration Mode" checkbox.

   → GUI feedback: Run button text changes to "■ Run COM Calibration", button turns orange

   → GUI feedback: Blue info text appears: Calibration mode requires 1 FEXT file, 0 NEXT files

10.4 Set Calibration Precision

In the calibration parameters area, enter the Sigma BBN step value (default 0.01). This value controls the calibration search step size, in units of V.

Step Value	Precision	Typical Duration	Use Case
0.001	High	Longer	Precise calibration values for final reports
0.01	Medium (Recommended)	Moderate	Routine engineering evaluation

Step Value	Precision	Typical Duration	Use Case
0.1	Low	Shorter	Quick rough estimation

10.5 Run Calibration

1. Confirm that a configuration file, thru S-parameter file, and **exactly 1** FEXT file are selected.
2. Switch to the [NEXT] list and confirm it is **empty** (calibration mode does not use NEXT).
3. Click the "■ Run COM Calibration" button.

   → GUI feedback: The MATLAB engine pops up a reminder dialog (asking you to confirm the measured signal_RJ / A_DD / SNR_TX values). The tool closes it automatically — no user action required; the run log records "[Calibration] engine dialog auto-closed".

4. Wait for calibration to complete.

> **Note:** In calibration mode, the program will not automatically timeout. You must wait for the MATLAB process to exit naturally, or manually click the "Finish" button after confirming that calculation is complete. Calibration time depends on step settings and may take from several minutes to tens of minutes.

10.6 View Calibration Results

After calibration completes, results are displayed in the "Noise Tolerance Calibration Results" area of the summary tab:

- Calibration noise value is displayed in blue 28pt large font, in units of mV
- Title shows the parameter name (σ_hp or σ_bn)
- Status indicator shows blue "Calibration Complete ✓"

Calibration results are also recorded in the CSV data on the detailed data tab, under the field name sigma_bn. The report tab also includes the calibration noise value.

> **Tip:** The calibration noise value (mV) can be directly used for subsequent noise tolerance testing. It is recommended to record the configuration file, S-parameter file, and step value used during calibration to ensure reproducibility.

Chapter 11 Configuration Management

The software supports saving the current file selection and parameter override configuration to a JSON file for quick restoration later.

11.1 Save Configuration

Click the menu bar "File" → "Save Configuration..."

→ GUI feedback: File save dialog opens

Choose a save path and filename (.json), and click "Save".

Saved content includes:

- Configuration file path
- Thru S-parameter file path
- FEXT / NEXT file lists
- Parameter override settings
- Case selection status

11.2 Load Configuration

Click the menu bar "File" → "Load Configuration..."

→ GUI feedback: File selection dialog opens

Select a previously saved .json file and click "Open".

→ GUI feedback: All panels automatically update to the loaded configuration values

→ GUI feedback: Parameter override panel reloads

11.2.1 Demo Scenarios (demo\ folder)

From v1.7.0 the demo configuration comes from the **demo** folder in the installation directory (demo.json has been removed). Each **scenario_*.json** in that folder is one demo scenario, and the software **auto-loads the lowest-numbered one on first launch** (the slowest rate, easiest to start with).

Three scenarios ship with the tool, covering three IEEE 802.3 generations:

Scenario file	Baud rate (Nyquist)	Standard
scenario_1_26p5625GBd_50G_KR.json	26.5625 GBd (13.28 GHz)	802.3cd 50G/lane
scenario_2_53p125GBd_100G_C2C.json	53.125 GBd (26.56 GHz)	802.3ck 100G/lane

Scenario file	Baud rate (Nyquist)	Standard
scenario_3_106p25GBd_200G_KR.json	106.25 GBd (53.13 GHz)	802.3dj 200G/lane

Scenario JSON uses the same workspace format as "Save Configuration":

- **description**: shown in the status bar and the SCPI listing
- **config_file**: Excel configuration path
- **thru_file**: thru-channel S-parameter path
- **fext_files** / **next_files**: crosstalk file lists

> **Tip:** Paths resolve relative to **the folder holding the scenario JSON**, so the whole demo\ directory can be moved with the installation to any drive. The S-parameters referenced by the scenarios live inside demo\, while the Excel configuration points at ../resources/ExcelConfigFiles/. Copy one of the files and edit the paths to add your own scenario - the software lists whatever it finds.

Use [File - Load Configuration] to switch scenarios at any time; over SCPI use CONF:DEMO:LIST? to enumerate and CONF:DEMO:LOAD to load.

11.3 Initialize

Click the menu bar "File" → "Initialize"

→ GUI feedback: Confirmation dialog appears: "Are you sure you want to clear all settings?"

After clicking "Yes" to confirm:

→ GUI feedback: All file selections are cleared

→ GUI feedback: Parameter override panel is reset

→ GUI feedback: Case selections are cleared

Chapter 12 FAQ and Troubleshooting

Q: No output in the log panel after running (Mode A)

A: This is normal behavior. Mode A uses the compiled executable, and due to MATLAB Runtime limitations, intermediate output cannot be captured by the GUI. Full MATLAB output can be viewed in the command line window (py.exe console) used to launch the software. The log panel will show diagnostic information and the final CSV results.

Q: "Executable not found" error

A: Please check the path configuration in "File → Settings". Verify that the selected .exe or matlab.exe path is correct and the file exists. If the path contains spaces or special characters, ensure the path format is correct.

Q: "Insufficient privileges" error

A: Please try running the software as administrator. Right-click python.exe or the shortcut and select "Run as administrator".

Q: Parameter override does not seem to take effect

A: Please verify: (1) The "Enable" checkbox for the row is checked; (2) A valid value has been entered in the "Override Value" column; (3) Check the run log for parameter write confirmation. Also, if some parameters in Excel are formula references, parameter override will replace formulas with fixed values.

Q: COM value shows "—"

A: This means MATLAB computation did not complete normally or the output format is abnormal. Check: (1) Whether there are error messages in the run log; (2) Whether the MATLAB process exited normally (exit code 0); (3) Whether CSV result files were generated in the directory specified by RESULT_DIR (default: results subfolder).

Q: Channel chart is not displayed

A: Please confirm a valid .s4p file is selected. The file must be a standard Touchstone format 4-port S-parameter file. If the file format is incorrect, the channel chart area will display an error message. Common causes: file encoding issues, port count is not 4, file corruption, etc.

Q: Trial runs exhausted

A: Please contact the vendor to obtain a license key. View your Host ID in "License → License Management" and provide it to the vendor.

Q: "Feature Restricted" prompt when adding more aggressor S-parameter files or enabling calibration

A: In trial mode (no key activated), FEXT and NEXT aggressor S-parameter files are limited to 1 each, and calibration mode is not available. Activating any type of license key (permanent or temporary) will unlock all feature restrictions. Contact the vendor for a key and enter it in "License → License Management".

Q: Excel configuration file has been modified

A: The parameter override feature modifies the Excel file at runtime. The software only creates a backup (.xlsx.bak) on the first override; subsequent overrides do not overwrite the backup, so .bak always preserves the original file. To restore, rename .xlsx.bak to .xlsx.

Q: Calculation takes a long time, is this normal?

A: COM calculation time depends on parameter configuration and hardware performance, typically taking several minutes to tens of minutes per case. Check elapsed time and completed case count in the status bar. Click "Stop" to abort if needed. First runs may take longer due to MATLAB Runtime initialization.

Q: How to run multiple cases simultaneously?

A: After loading the configuration file, the case selection area shows all available cases. Check the cases you want to run; the software will execute them sequentially. The status bar shows current progress during execution. You can also specify a case list via the "z_p select" parameter override.

Q: Which is better, Mode A or Mode B?

A: Mode A (Recommended): No full MATLAB installation needed, only the free MATLAB Runtime. Easy deployment. Drawback: cannot view real-time output in the GUI log panel. Mode B: Requires full MATLAB installation, supports real-time log output, suitable for development and debugging environments. Both modes produce identical calculation results.

Q: Unsure about S-parameter port order?

A: The "Port Order" parameter in the Excel configuration defines the port mapping. Common port orders are [1 2 3 4] (TX+ TX- RX+ RX-) and [1 3 2 4]. Refer to the port connection method used during VNA measurement, or check the S-parameter file comment lines. Incorrect port order will lead to incorrect COM calculation results.

Q: Calibration mode does not show calibration noise value

A: Possible causes: (1) Calibration has not completed yet; wait for the MATLAB process to finish; (2) In Mode A, results are obtained via CSV fallback; ensure the CSV_REPORT parameter is set to 1; (3) Check the detailed data tab to confirm the CSV file contains the sigma_bn field.

Q: How to compare results from different channel designs?

A: You can compare using the following methods: (1) Save different configurations as JSON files and load them sequentially; (2) Use the multi-case feature to define different trace lengths in Excel; (3) After runs complete, the report tab retains all history records for vertical comparison.

Q: Where are CSV result files saved?

A: They are saved in the directory specified by RESULT_DIR (default: results subfolder of the configuration file directory), with filenames in the format "*_caseN_results.csv". RESULT_DIR supports the {date} placeholder (expands to DD-Mon-YYYY format). Relative

paths are resolved from the configuration file directory.

> **Tip:** If you encounter issues not covered in this manual, please check the detailed output in the run log tab, or contact technical support.

Chapter 13 Typical Workflows

This chapter describes complete operational workflows from configuration to results through typical application scenarios.

13.1 Basic COM Calculation Workflow

Scenario: Evaluate whether a high-speed serial link meets the COM requirements of the target data rate standard.

Step 1 — Prepare Input Files

- Excel configuration file (.xlsx): Contains parameter settings for the target data rate standard
- Thru channel S-parameter (.s4p): 4-port S-parameter measurement data of the channel under test
- FEXT files (.s4p, optional): S-parameters of far-end crosstalk aggressors
- NEXT files (.s4p, optional): S-parameters of near-end crosstalk aggressors

Step 2 — Configure Software

- First use: Go to "File → Settings" to configure the MATLAB runtime environment
- Load configuration file → Select thru S4P → Add FEXT/NEXT files
- Preview frequency-domain curves in the channel chart tab to confirm files loaded correctly
- If parameters need modification, edit them in the parameter override panel

Step 3 — Run Calculation

- Select cases to run, click "■ Run COM Calculation"
- Monitor the run log and status bar progress

Step 4 — View Results

- Summary page: Confirm whether COM is PASS
- Detailed data page: View complete noise decomposition and equalizer parameters
- Report page: Export PDF report for archiving

13.2 Parameter Sensitivity Analysis

Scenario: Evaluate the impact of a parameter change on COM (e.g., different CTLE settings).

1. Complete a baseline run and record the COM value.
2. Modify the target parameter override value in the parameter override panel.

3. Re-run COM calculation.
4. Compare the history records in the report tab to analyze the impact of parameter changes.

> **Tip:** You can save different parameter settings as configurations (JSON files) for quick switching and reproduction later.

13.3 Noise Tolerance Calibration Workflow

Scenario: Determine channel noise tolerance parameters for receiver compliance testing.

1. Select the configuration file and thru S-parameter file.
2. In the aggressor panel, click [FEXT] to add **exactly 1** FEXT file, then switch to [NEXT] and confirm the list is empty.
3. Check "Enable Calibration Mode" and set the Sigma BBN step value.
4. Click "■ Run COM Calibration".
5. Wait for calibration to complete; the summary page displays σ_bn or σ_hp value (mV).
6. Record the calibration noise value for subsequent testing.

13.4 Batch Multi-Case Comparison

Scenario: The Excel configuration file defines multiple package cases (e.g., different trace lengths), and you need to run and compare them all at once.

1. Load the Excel configuration file containing multiple case definitions.
2. The case selection area will show Case 1, Case 2, ... checkboxes.
3. Check all cases to run (or use Select All).
4. Click "■ Run COM Calculation"; the software will compute each case sequentially.
5. After completion, the summary page comparison table shows core metrics for each case.
6. The detailed data page creates a separate data tab for each case.

Chapter 14 Mode Comparison and Selection

This software supports two execution modes for different use cases. Both modes produce identical calculation results; they differ only in deployment method and feature characteristics.

Comparison Item	Mode A: Compiled Executable	Mode B: Full MATLAB
Deployment Requirement	Only MATLAB Runtime **R2024a** needed (free)	Full MATLAB installation required (paid license)
Version Requirement	**Must be R2024a** (other versions not supported)	R2024a recommended
Installation Size	Runtime approx. 2-3 GB	Full MATLAB 10+ GB
Real-time Log	Not supported (GUI cannot capture)	Supported (displayed in log panel)
Console Output	Viewable in py.exe console	Redirected to GUI log panel
Startup Speed	Slower on first run (Runtime initialization)	Relatively fast
Use Case	Production/customer deployment (Recommended)	Development/debugging
CSV Results	Normal output	Normal output
Calibration	Fully supported	Fully supported
Recommendation	✓ Recommended for daily use	Use when real-time log is needed

Tip: If both MATLAB Runtime and full MATLAB are installed, you can switch modes in settings at any time without restarting the software.

Chapter 15 SCPI Remote Control

This software supports a SCPI (Standard Commands for Programmable Instruments) remote control interface, enabling control of COM calculations from automated test systems, CI/CD pipelines, or scripts.

15.1 Overview

SCPI Protocol Introduction

SCPI (Standard Commands for Programmable Instruments) is an industry-standard protocol for test and measurement, defining the syntax and semantics of instrument control commands. IEEE 488.2 defines common commands (*IDN?/*RST/*OPC, etc.) and the status register model.

The COM Calculator implements SCPI-raw over TCP/IP (default port 5026), allowing clients to control COM calculations from any SCPI-capable automation environment.

Use Cases

- ATE (Automated Test Equipment) integration
- CI/CD pipeline automated verification
- Batch parameter sweeps (parameter scanning and optimization)
- LabVIEW / MATLAB / Python automation scripts

Technical Architecture

- Protocol: SCPI-raw over TCP/IP
- Port: 5026 (COM Tool default SCPI port)
- Terminator: LF (0x0A)
- Encoding: ASCII
- Connection model: Single client (serves one connection at a time)
- Idle timeout: 300 seconds
- VISA address: TCPIP0::<IP>::5026::SOCKET

Response Format

All responses are single-line ASCII strings terminated by LF (0x0A). List-type responses use comma or semicolon separators:

- SYST:HELP:HEAD? — comma-separated: *CLS,*ESE,...
- CONF:PAR:LIST? — semicolon-separated: f_b=212.5;sigma_RJ=0.01
- CONF:FILE:FEXT:LIST? — semicolon-separated: "f1.s4p";"f2.s4p"
- LOG:DATA? / LOG:LAST? — semicolon-separated: line1;line2;...

- Error response: -100,"Invalid mode: BOGUS"
- Empty list: "" (double-quoted empty string)

15.2 Starting SCPI Service

Auto-Start (Recommended)

In "File" → "Settings", check "Auto-start SCPI remote control server on application startup", set the port, and click OK. The SCPI service will start automatically on next launch.

From GUI (Manual)

In the main window menu bar, click "Tools" → "Start SCPI Service". Enter the port number (default 5026) in the popup dialog, then click OK to start.

- Status bar will show a green label "SCPI: :5026"
- Click "Tools" → "Stop SCPI Service" to shut down
- SCPI service stops automatically when the main window is closed

> **Tip:** On first launch, Windows Firewall may display a prompt. Please allow Python through the private network.

Headless Mode

Headless mode requires no GUI, suitable for server or embedded environments:

```
python scpi_main.py --port 5026 --host 0.0.0.0
```

- --port: Listening port (default 5026)
- --host: Listening address (default 0.0.0.0, i.e., all interfaces)
- --ready-signal: Print SCPI_READY when server is listening (for automation)
- --quiet / -q: Minimal output, suppress command list
- Ctrl+C for graceful shutdown

Windows Firewall Settings

On first launch, Windows Firewall may display a prompt. Please allow Python through the private network. Or add a rule via the administrator command line:

```
netsh advfirewall firewall add rule name="COM_SCPI" dir=in action=allow
protocol=TCP localport=5026
```

15.3 IEEE 488.2 Common Commands

*IDN? / *RST / *CLS

Command	Type	Description
*IDN?	Query	Returns "IEEE_COM,COM_Tool,0,<VERSION>"
*RST	Set	Reset SCPI state: clears files/params/cases/calibration/chart markers; run options return to power-on defaults (quiet=120, parallel=1, etc.), then GUI-persisted paths and run options are reloaded from QSettings; SPAR tool state cleared; License is unaffected
*CLS	Set	Clear status byte, event registers, error queue

*OPC / *OPC? (Operation Complete Synchronization)

*OPC sets the OPC bit in ESR when operation completes. *OPC? blocks until operation completes, returns "1". Typical usage: INIT; *OPC? → wait for calculation to complete.

*STB? / *ESR? / *ESE

*STB? reads the status byte (non-destructive). *ESR? reads and clears the standard event status register. *ESE <mask> sets the event status enable register.

Command	Type	Description
*ESE <mask> / *ESE?	Set/Query	Event status enable register
*SRE <mask> / *SRE?	Set/Query	Service request enable
*WAI	Set	Wait for all pending operations to complete
*TST?	Query	Self-test (checks path validity), 0=pass

15.4 SCPI Command Reference

GUI/SCPI bidirectional sync: every actionable GUI feature (>99%) has a corresponding SCPI command. SCPI changes are reflected in the GUI in real time, and GUI edits are pushed to the SCPI cache. Pure UI actions (opening/closing dialogs, image-viewer zoom, etc.) have no SCPI counterpart because they do not change underlying state.

Design-by-intent gap: the SCPI listening port is configurable only via the GUI Settings dialog (the SCPI command cannot change its own port — that would orphan the active connection).

15.4.1 File Configuration (CONFigure:FILE)

Command	Description	Example
CONF:FILE:CONF <path>	Set Excel configuration file	CONF:FILE:CONF "D:/config.xlsx"
CONF:FILE:CONF?	Query configuration file path	
CONF:FILE:THRU <path>	Set Thru S4P file	CONF:FILE:THRU "D:/thru.s4p"
CONF:FILE:FEXT:ADD <path>	Add FEXT file	CONF:FILE:FEXT:ADD "D:/fext1.s4p"
CONF:FILE:FEXT:DEL	Clear all FEXT files	
CONF:FILE:FEXT:COUN?	FEXT file count	→ 2
CONF:FILE:FEXT:LIST?	FEXT file list	→ "f1.s4p";"f2.s4p"
CONF:FILE:NEXT:ADD/DEL/COUN?/LIST?	NEXT files (same as FEXT)	
CONF:FILE:VAL?	Validate all files	→ 0 (success)

15.4.2 Parameter Override (CONFigure:PARameter)

Command	Description	Example
CONF:PAR:SET <key>,<val>	Set parameter override	CONF:PAR:SET f_b,212.5
CONF:PAR:SET? <key>	Query override value	CONF:PAR:SET? f_b
CONF:PAR:CLE [<key>]	Clear override	CONF:PAR:CLE f_b

Command	Description	Example
CONF:PAR:INS <key>,<val>	Insert new keyword row into xlsx (for templates missing the keyword)	CONF:PAR:INS AUTO_TFX,1
CONF:PAR:APPL	Apply current overrides to xlsx without running MATLAB (v1.5.12)	
CONF:PAR:LIST?	List all overrides	→ f_b=212.5;sigma_RJ=0.01
CONF:PAR:CAT?	Available parameter list	

15.4.3 Signal Amplitude (CONFigure:AMPlitude)

Command	Description	Example
CONF:AMPL:AV <val>	Set thru channel amplitude A_v (V)	CONF:AMPL:AV 0.385
CONF:AMPL:AV?	Query A_v value	
CONF:AMPL:AFE <val>	Set FEXT aggressor amplitude A_fe (V)	CONF:AMPL:AFE 0.385
CONF:AMPL:AFE?	Query A_fe value	
CONF:AMPL:ANE <val>	Set NEXT aggressor amplitude A_ne (V)	CONF:AMPL:ANE 0.481
CONF:AMPL:ANE?	Query A_ne value	

15.4.4 Execution Mode and Paths

Command	Description
CONF:MODE <EXE|MATLAB>	Mode A (EXE) or Mode B (MATLAB)
CONF:PATH:EXE <path>	Compiled exe path
CONF:PATH:MCR <path>	MATLAB Runtime path
CONF:PATH:MATLAB <path>	matlab.exe path
CONF:PATH:SCR <path>	.m file path (Mode B; parent is addpath-ed, stem called as function)

15.4.5 Run Settings (CONFigure:GENeral)

Command	Description
CONF:GEN:QUI <seconds> / ?	Silent wait timeout (default 120)
CONF:GEN:CLOS <ON|OFF> / ?	Auto-close MATLAB figure window
CONF:GEN:EXP <N> / ?	Expected case count (0 = auto-detect from Excel)
CONF:GEN:CLEA <ON|OFF> / ?	Clear result dir before run (default ON)
CONF:GEN:PAR <N> / ?	Parallel case count (1=serial, 2-16=parallel); multi-case runs spawn N MATLAB processes; calibration mode forces 1

15.4.6 Demo Configuration Load (CONFigure:DEMO)

Command	Description
CONF:DEMO:LOAD [path]	Load a demo scenario from demo/. Accepts a scenario name or path; defaults to the first scenario. Automatically sets file paths and parameter overrides.
CONF:DEMO:LOAD?	Returns path of last loaded configuration file

Tip: CONF:DEMO:LOAD supports both relative and absolute paths. Without arguments it loads the default demo scenario, suitable for one-click instrument configuration.

15.4.7 Trigger Control (INITiate / ABORt)

Command	Description
INIT[:IMM]	Start COM calculation (asynchronous)
ABOR	Abort running calculation (kills process)
ABOR:FIN	Skip the silent-wait period and accept current results (v1.5.12)

Tip: After starting, use *OPC? to block until completion, or poll STAT:OPER:COND? to check bit4 (MEASuring). ABOR vs ABOR:FIN: ABOR marks ABORTED; ABOR:FIN takes the normal completion path with current data.

15.4.8 Result Query (FETCh)

Command	Returns	Description
FETC:COM?	3.42 (dB)	COM value
FETC:ERL?	18.5 (dB)	ERL value
FETC:VEO?	25.3 (mV)	VEO value
FETC:VEC?	-0.5 (dB)	VEC value
FETC:AS?	100.0 (mV)	Signal amplitude A_s
FETC:DER?	1.2E-7	Detector error ratio
FETC:PASS?	1/0/-1	PASS/FAIL/Unknown
FETC:ALL?	COM=3.42,...	All metrics
FETC:RUNT?	0.69 (min)	Run time
FETC:CASE:COUN?	4	CSV case count
FETC:CASE:COM? <n>	3.42	COM for case n
FETC:CASE:DATA? <n>	COM_dB=3.42,...	All data for case n
FETC:CASE:LIST?	"Case 1",...	Case label list

15.4.9 Combined Measurement (MEASure)

MEASure commands are equivalent to INITiate + wait + FETCh, blocking until calculation completes.

Command	Description
MEAS:COM?	Start → wait → return COM
MEAS:ALL?	Start → wait → return all metrics
MEAS:PASS?	Start → wait → return PASS/FAIL

Note: MEASure commands may block for several minutes. Ensure client socket timeout is set large enough (600+ seconds recommended).

15.4.10 Status Registers (STATus)

Command	Description
STAT:OPER[:EVEN]?	Read and clear operation event register

Command	Description
STAT:OPER:COND?	Read operation condition register (bit4=calculating)
STAT:OPER:ENAB <mask>	Operation enable register
STAT:QUES[:EVEN]? / :COND? / :ENAB	Questionable status register
STAT:PRES	Reset enable registers to default values

15.4.11 System Commands (SYSTem)

Command	Description
SYST:ERR[:NEXT]?	Pop error queue → 0,"No error"
SYST:ERR:COUN?	Error queue depth
SYST:ERR:ALL?	Read and clear all errors
SYST:VERS?	SCPI version → 1999.0
SYST:HELP:HEAD?	Supported command list (comma-separated)
SYST:LANG <EN|ZH> / ?	Switch GUI language (real-time)
SYST:SCPI:PORT?	Query GUI-configured SCPI server port (v1.5.12)
SYST:SCPI:AUTO <ON|OFF> / ?	Auto-start SCPI server on GUI launch (v1.5.12)

15.4.12 Run Log (LOG)

Command	Description
LOG:DATA?	Full log (semicolon-separated)
LOG:COUN?	Log line count
LOG:LAST? <n>	Last n lines (semicolon-separated)
LOG:CLE	Clear log
LOG:SAVE "<path>"	Save log to UTF-8 text file (v1.5.12)

15.4.13 Channel Chart (CHARt)

Command	Description
CHAR:SUBP <mode>	Set subplot mode: ALL / SDD21 / SDD11 / SDD22
CHAR:SUBP?	Query current subplot mode
CHAR:MARK:ADD <subplot>,<freq_ghz>	Add a marker on the specified subplot (v1.5.12)
CHAR:MARK:ADD:ALL <freq_ghz>	Add markers for all curves at specified frequency
CHAR:MARK:DEL <subplot>,<freq_ghz>	Delete the marker on the specified subplot nearest to given freq (v1.5.12)
CHAR:MARK:LIST?	Query all markers list (freq,subplot,curves)
CHAR:MARK:CLE	Clear all markers
CHAR:AXIS:RANG <subplot>,<xmin>,<xmax>,<ymin>,<ymax>	Set axis range for specified subplot
CHAR:AXIS:RANG? [<subplot>]	Query axis range settings
CHAR:AXIS:RANG:RES	Reset all axis ranges to defaults
CHAR:HEIG <pixels|AUTO>	Set channel chart height (pixels, ≥200); AUTO or 0 restores auto height
CHAR:HEIG?	Query current chart height setting (0 = auto)

15.4.14 Result Output Directory (CONFigure:FILE:RESult:DIR)

Command	Description
CONF:FILE:RES:DIR <path>	Set result output directory (overrides RESULT_DIR in Excel)
CONF:FILE:RES:DIR?	Query current result output directory

15.4.15 Calibration Settings (CONFigure:CALibration)

Command	Description
CONF:CAL:MODE <ON|OFF|1|0>	Enable/disable interference tolerance calibration mode

Command	Description
CONF:CAL:MODE?	Query calibration mode state (1=enabled, 0=disabled)
CONF:CAL:STEP <value>	Set calibration step value (default 0.001)
CONF:CAL:STEP?	Query current calibration step value

15.4.16 Case Selection (CONFigure:CASE:SELect)

Command	Description
CONF:CASE:SEL <list|ALL>	Select cases to run (e.g. 1,3,4 or ALL)
CONF:CASE:SEL?	Query current case selection

15.4.17 Calibration Results (FETCh:CALibration)

Command	Description
FETC:CAL:NOIS?	Query calibration noise values (sigma_hp,sigma_bn in mV)
FETC:CAL:PASS?	Calibration success (1=success, 0=no data)

15.4.18 Port Order Query (FETCh:PORT:ORDer)

Command	Description
FETC:PORT:ORD?	Query port order (e.g. 1,3,2,4)

15.4.19 PDF Report Export (FILe:REPort)

Command	Description
FILE:REP:EXP <filepath>	Export PDF report (GUI mode only)
FILE:REP:EXP?	Query last exported report path
FILE:REP:CLE	Clear report tab entries (mirror GUI ■■■■; v1.5.12)

15.4.20 Interface Language (SYSTem:LANGuage)

Command	Description
SYST:LANG <EN|ZH>	Set interface language (EN/ZH/en_US/zh_CN)

Command	Description
SYST:LANG?	Query current interface language

15.5 Usage Examples

Telnet Manual Operation

```
telnet localhost 5026
```

```
*IDN?
CONF:FILE:CONF "D:/config_212Gbps.xlsx"
CONF:FILE:THRU "D:/thru.s4p"
CONF:MODE EXE
CONF:PATH:EXE "D:/com_ieee8023_4p14p0.exe"
CONF:PATH:MCR "F:/MATLAB/R2024a"
INIT
*OPC?
FETC:COM?
FETC:PASS?
```

Python Socket Client

```
import socket

def scpi(sock, cmd):
    sock.sendall((cmd + "\n").encode())
    if "?" in cmd:
        return sock.recv(4096).decode().strip()

s = socket.socket()
s.connect(("localhost", 5026))
s.settimeout(600)

print(scpi(s, "*IDN?"))
scpi(s, 'CONF:FILE:CONF "D:/config.xlsx"')
scpi(s, 'CONF:FILE:THRU "D:/thru.s4p"')
scpi(s, "CONF:MODE EXE")
scpi(s, 'CONF:PATH:EXE "D:/com.exe"')
scpi(s, 'CONF:PATH:MCR "F:/MATLAB/R2024a"')
scpi(s, "INIT")
scpi(s, "*OPC?") # Block until complete
com = scpi(s, "FETC:COM?")
print(f"COM = {com} dB")
s.close()
```

Batch Parameter Sweep Script

```
params = [("f_b", v) for v in ["106.25", "212.5"]]

for key, val in params:
    scpi(s, f"CONF:PAR:SET {key},{val}")
    scpi(s, "INIT")
    scpi(s, "*OPC?")
    com = scpi(s, "FETC:COM?")
    passed = scpi(s, "FETC:PASS?")
    print(f"{key}={val}: COM={com} dB, PASS={passed}")
    scpi(s, "CONF:PAR:CLE")
```

LabVIEW / MATLAB Integration

LabVIEW: Use TCP/IP function nodes to connect to port 5026, send SCPI command strings and read responses.

MATLAB: Use tcpclient object or instrfind/visa interface.

```
t = tcpclient("localhost", 5026);
writeline(t, "*IDN?");
idn = readline(t);
disp(idn);
```

Keysight IO / PyVISA Connection

After installing Keysight IO Libraries Suite, you can connect via VISA TCPIP SOCKET. VISA address format:

```
TCPIP0::127.0.0.1::5026::SOCKET
```

In Keysight Connection Expert, manually add a TCPIP Socket instrument with the IP and port, then use Interactive IO (Send & Read) to send commands.

Python PyVISA example:

```
import pyvisa

rm = pyvisa.ResourceManager()
inst = rm.open_resource("TCPIP0::127.0.0.1::5026::SOCKET")
inst.read_termination = "\n"
inst.write_termination = "\n"
inst.timeout = 600000 # 10 min

print(inst.query("*IDN?"))
inst.write('CONF:FILE:CONF "D:/config.xlsx"')
inst.write("INIT")
print(inst.query("*OPC?")) # Block until complete
print("COM =", inst.query("FETC:COM?"), "dB")

inst.close()
rm.close()
```

> **Tip:** Keysight IO Libraries VISA backend (visa32.dll) has been fully verified. PyVISA installation: pip install pyvisa

15.6 Error Code Reference

SCPI Standard Error Codes

Range	Type	Description
-100 ~ -199	Command Error (CME)	Syntax error, unknown command
-200 ~ -299	Execution Error (EXE)	File not found, parameter out of range, SCPI license not authorized
-300 ~ -399	Device Error (DDE)	MATLAB crash, run failure
-400 ~ -499	Query Error (QYE)	Query when no results available

Device-Specific Errors

Device-specific error codes are positive numbers; the current version does not define any. All errors can be read via SYST:ERR?.

15.7 Status Register Details

Standard Event Status Register (ESR)

Bit	Name	Description
7	PON	Power-on (set when server starts)
5	CME	Command error (syntax/unknown command)
4	EXE	Execution error (file not found, parameter out of range)
3	DDE	Device error (MATLAB crash)
2	QYE	Query error (query when no results available)
0	OPC	Operation complete

Status Byte (STB)

Bit	Name	Description
7	OPER	Operation status summary
5	ESB	ESR summary (ESR & ESE non-zero)
4	MAV	Response data available
3	QUES	Questionable status summary
2	ERR	Error queue not empty

Operation Status Register

Bit	Name	Description
4	MEAS	Calculation in progress
5	WTRIG	Configured, waiting for trigger

15.8 Typical Automation Workflow

```
# 1. Identify
*IDN?
→ IEEE_COM,COM_Tool,0,1.2.0

# 2. Configure files
CONF:FILE:CONF "D:/config_212Gbps.xlsx"
CONF:FILE:THRU "D:/thru.s4p"
CONF:FILE:FEXT:ADD "D:/fext1.s4p"

# 3. Set mode and paths
CONF:MODE EXE
CONF:PATH:EXE "D:/com_ieee8023_4p14p0.exe"
CONF:PATH:MCR "F:/MATLAB/R2024a"

# 4. Parameter overrides
CONF:PAR:SET f_b,212.5
CONF:PAR:SET CSV_REPORT,1

# 5. Start calculation and wait
INIT
*OPC?
→ 1 (blocks until complete)

# 6. Read results
```

```
FETC:COM?
→ 3.42
FETC:PASS?
→ 1
FETC:ALL?
→ COM=3.42,ERL=18.5,VEO=25.3,VEC=-0.5,A_s=100.0
```

Chapter 16 VNA Network Analyzer

The software integrates VNA (Vector Network Analyzer) control, allowing direct S-parameter extraction from VNA instruments without leaving the COM calculation environment. VNA functionality requires an SCPI license.

16.1 Overview

The VNA window is opened via "Tools" → "Connect VNA" and provides:

- Connect to network analyzers via VISA (TCP/IP, GPIB, USB)
- Configure measurement parameters (frequency, points, S-param, format)
- Execute frequency sweeps and single measurements
- Acquire SnP parameter data (text/table/chart views)
- Capture VNA screenshots
- Send custom SCPI commands to the VNA
- Import/export Touchstone files

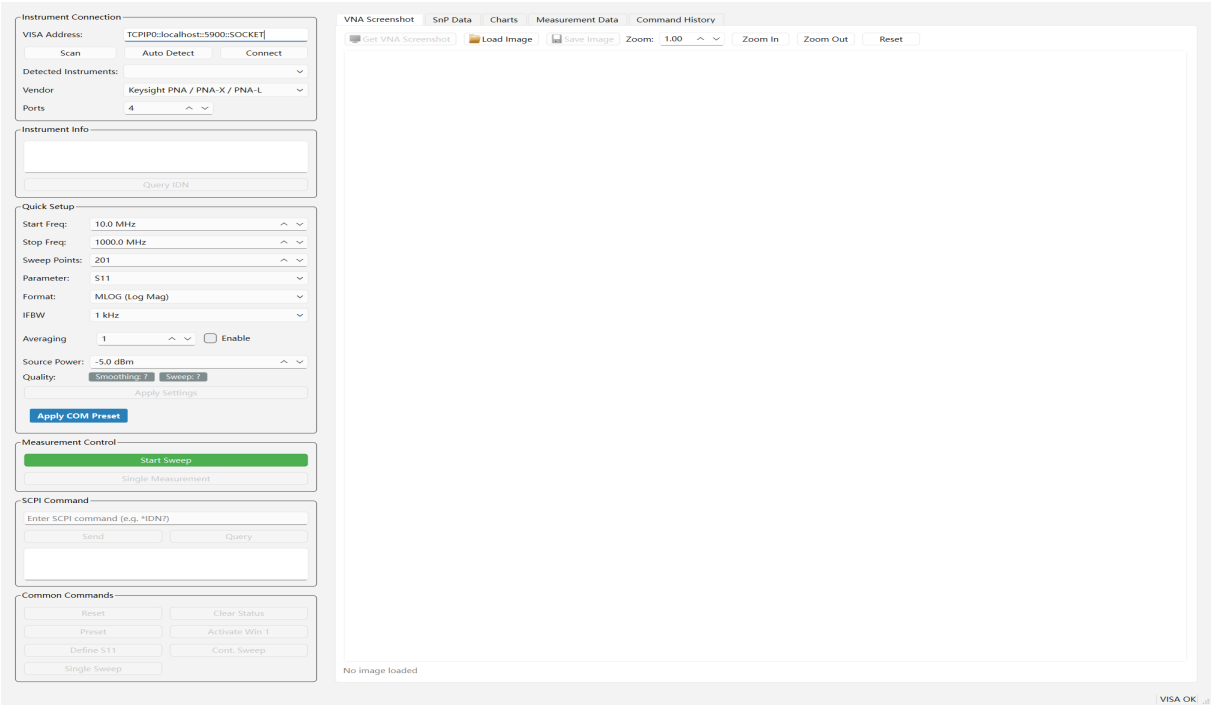

Figure 16-1 VNA Network Analyzer Control Window

16.2 Connecting to an Instrument

Steps (updated in v1.3):

- **Select vendor first** from the Vendor dropdown:

- • Keysight PNA / PNA-X / PNA-L (N5247B, N522xB, M983x)
- • Keysight ENA (E5080A/B, E5061B, E5063A, E5071C)
- • Rohde & Schwarz ZNB / ZVA / ZNA / ZNL
- • Anritsu ShockLine MS46xxx / VectorStar MS4640B
- Choose port count (S2P=2, S4P=4)
- Enter the VISA address (e.g. TCPIP0::192.168.1.100::inst0::INSTR)
- Or click "Scan" to auto-detect VISA devices on the network
- Select an instrument and click "Connect"
- After success the status bar shows Vendor | Model | Ports; vendor is locked while connected

Note: A VISA backend driver (e.g. Keysight IO Libraries or NI-VISA) and pyvisa must be installed. Each vendor's SCPI dialect is encapsulated in a driver module, so users never deal with raw commands.

S-Parameter Extraction: Each vendor driver uses the instrument's native Touchstone save command to produce a standard .s2p / .s4p file, then fetches it via MMEM:TRAN? (Keysight / Anritsu) or MMEM:DATA? (R&S). The fetched file is auto-parsed by scikit-rf and displayed in the SnP data tab.

16.3 Quick Setup

Parameter	Description	Default
Center Freq (GHz)	Measurement center frequency	1.0
Span (MHz)	Frequency sweep range	100
Sweep Points	Number of frequency samples	201
Parameter	S11 / S21 / S12 / S22	S11
Format	MLOG / PHAS / REAL / IMAG	MLOG

Click "Apply Settings" to write parameters to the instrument.

16.4 Measurement Operations

Operation	Description
Start Sweep	Execute a full frequency sweep; results shown in Charts tab

Operation	Description
Single Measurement	Trigger one immediate measurement
Get SnP Params	Extract 4-port S-parameters; shown in SnP Data tab
Get VNA Screenshot	Capture VNA display; shown in Screenshot tab

16.5 SCPI Command Control

The "SCPI Command" section on the left panel lets you send raw SCPI commands directly to the VNA. Enter a command and click "Send" (write) or "Query" (read response). Common command buttons provide one-click operations (Reset, Clear, Preset, etc.). All commands and responses are logged in the "Command History" tab.

16.6 Remote SCPI Control of VNA

VNA functionality is fully controllable via the COM Tool SCPI interface (port 5026). Remote automation example:

SCPI Command	Description
VNA:CONN "<addr>"	Connect to VNA
VNA:CONN?	Query connection status (1/0)
VNA:DISC	Disconnect
VNA:SET:CENT <GHz>	Set center frequency
VNA:SET:SPAN <MHz>	Set span
VNA:SET:POIN <N>	Set sweep points
VNA:SET:PAR <S11|S21|...>	Set measurement parameter
VNA:SET:FORM <MLOG|PHAS|...>	Set display format
VNA:SET:APPL	Apply settings to instrument
VNA:MEAS:SWE	Start frequency sweep
VNA:MEAS:SING	Single measurement

SCPI Command	Description
VNA:MEAS:SNP	Get SnP parameters
VNA:SCR:CAPT	Capture screenshot
VNA:SCPI:SEND "<cmd>"	Pass-through SCPI write to VNA
VNA:SCPI:QUER? "<cmd>"	Pass-through SCPI query to VNA

SCPI operations and GUI stay synchronized in real time: SCPI commands update the GUI display, and GUI operations are reflected in SCPI query results.

16.7 VNA Control Secondary Development

The COM calculation core is delivered as compiled binaries to protect intellectual property, but the **VNA control layer ships as fully open Python source**. It is a **sample framework**: four mainstream vendor drivers are bundled, and customer engineers are meant to extend it -- modify an existing driver to match their instrument firmware quirks, or add a brand-new vendor driver.

Open-source files (kept as .py source in the customer package, under the _internal subdirectory):

Source File	Role
drivers/vna/base.py	Abstract driver base class VnaDriver; defines the uniform control surface. Its header holds the full developer guide
drivers/vna/factory.py	Vendor registry and driver factory
drivers/vna/_binary.py	IEEE 488.2 binary-block parsing helpers
drivers/vna/keysight_pna.py	Keysight PNA driver (recommended template for a new driver)
drivers/vna/keysight_ena.py	Keysight ENA driver
drivers/vna/rohde_schwarz.py	Rohde & Schwarz driver
drivers/vna/anritsu.py	Anritsu driver
gui/vna_window.py	VNA control window

Steps to add a new VNA vendor driver:

- **(1) Copy**: copy keysight_pna.py to a new file, e.g. drivers/vna/my_vendor.py

- **(2) Edit**: rename the driver class, set a unique VENDOR_ID and DISPLAY_NAME, adjust capabilities (port count, frequency range), and translate each method body to the target instrument SCPI dialect (refer to that vendor SCPI programming guide)
- **(3) Register**: keep the @register_vendor decorator on the class, and add the new module name to the _autoload() list in factory.py

The new vendor then appears automatically in the VNA window "Vendor" dropdown and is selectable via the SCPI command VNA:VENDor -- no changes to the GUI or SCPI layer are required.

Testing without hardware: tests/virtual_vna.py is a virtual VNA (a TCP SCPI simulator emulating all four vendor dialects). Run it, then point the VNA window at TCPIP0::localhost::5900::SOCKET to verify the full connect-sweep-fetch path end to end. tests/test_vna_drivers.py demonstrates hardware-free unit testing with a FakeResource.

Tip: secondary development is best done in a Python source environment. The per-method contract (which methods are mandatory vs optional to override) is documented in the developer guide at the top of drivers/vna/base.py.

17. Online Update

The software can check for and install new versions directly from the official download site. Menu: **Help -> Check for Updates**.

17.1 Steps

1. Click [Help -> Check for Updates]; the app queries the latest version.

   → GUI feedback: Shows "You are on the latest version", or the new version number and download size.

2. Click [Download Update]. A progress bar is shown; the download can be cancelled at any time.

   → GUI feedback: Partial downloads are kept and resumed next time rather than restarting from zero.

3. Integrity is verified automatically (SHA-256); the button then becomes [Install Now].
4. Click [Install Now]. The app closes and restarts on the new version.

> **Note:** On install the old installation directory is **renamed and kept** as COM_Tool_backup_<timestamp> alongside the install folder - it is never deleted. If you edited configuration files under resources or need earlier results output, recover them from that backup; delete it once satisfied to reclaim disk space.

17.2 Security

The update manifest is signed with Ed25519. The app embeds only the public key; the signing private key is held offline by the vendor. Consequently:

- Tampering with the version, download URL or checksum makes the app refuse the update
- The package is checked against the SHA-256 in the signed manifest; a swapped or corrupt file is detected and deleted
- Download URLs are restricted to the official host and cannot be redirected to a third-party server

If the check fails, it is usually a network or proxy issue - retry, or download manually from http://106.14.76.130.

17.3 SCPI Remote Update

The same flow is available over SCPI for automation:

Command	Description
SYST:UPD:CHEC?	Check -> UPDATE|LATEST|ERROR,,
SYST:UPD:INFO?	New version details (JSON: version/size/sha256/notes)
SYST:UPD:DOWN	Start background download and verification
SYST:UPD:STAT?	Progress -> ,,,
SYST:UPD:INST	Install and restart

Note: ERROR means the **check failed** (unreachable server or invalid signature) and is a distinct state from LATEST (already up to date); automation should treat them differently.

17.4 Feedback

Menu: **Help -> Feedback**. Describe the problem or suggestion and send it straight to the author.

1. Say what you did, what you saw, and what you expected.

2. Paste screenshots with **Ctrl+V**, drag them into the window, or click [Choose images]. Up to 3; they are compressed automatically before upload.

3. [Email / WeChat] is optional - fill it in if you want a reply.

4. [Attach version and environment info] is on by default and sends the software version, OS, host ID and license status - exactly what is needed to diagnose an issue. The exact text is shown in the window; clear the box if you prefer not to.

The feedback service runs on the author machine, so **being offline is normal**. If you see "service is offline", try again later.

Equivalent SCPI commands for automation:

Command	Description
SYST:FEED:ONL?	Is the service online -> 1|0 (check before sending)
SYST:FEED ""	Send text feedback (version/host ID/license attached)
SYST:FEED:CONT ""	Leave an email or WeChat ID

Screenshots are GUI-only - a headless session has nothing to paste.

Reading replies: the software checks for the author reply in the background. When one arrives the status bar says so and the [Feedback] entry in the [Help] menu shows an unread count.

Open the window and the full conversation is at the top; unread items carry a dot and are marked read on open.

A reply is delivered only once, so the software saves it locally as soon as it arrives (%APPDATA%\COM_Tool\feedback_log.json). You can always reopen the window later and read it again.

Automation: SYST:FEED:REPL? returns the replies as a JSON array.

Appendix A Excel Configuration File Structure

The Excel configuration file (.xlsx) used for COM calculation must contain a worksheet named "COM_Settings". Parameters in this worksheet are distributed across different columns:

Area	Key Column	Value Column	Parameters Included
Main Parameters	Column A	Column B	Signal rate, equalizer, noise, and other core parameters
I/O Control	Column F	Column G	Port order, package parameters (z_p), etc.
Operational	Column J	Column K	ERL/CSV output control and other run options
Extended Control	Column O	Column P	Calibration mode, ICN parameters, etc.

Case Definitions

The Excel configuration file can define multiple package test cases. Cases are defined through the "z_p (TX)" matrix, where each column corresponds to one case. For example, a z_p (TX) matrix with 4 columns defines 4 cases.

The "z_p select" parameter specifies the list of case indices to run, in MATLAB vector format. For example, "[1 2 3 4]" runs all 4 cases, and "[1]" runs only the first case.

Key Parameter Description

The following lists some key parameters in the Excel configuration file and their meanings, to help understand the configuration file structure:

Parameter	Description	Typical Value
f_b	Signal baud rate	106.25 / 212.5 GBd
Port Order	S-parameter port order	[1 2 3 4] or [1 3 2 4]
z_p (TX)	TX package trace length matrix	Matrix, one case per column

Parameter	Description	Typical Value
z_p (RX)	RX package trace length matrix	Same as above
z_p select	Case indices to run	[1 2 3 4]
ERL	Whether to calculate ERL	1 (enable) / 0 (disable)
CSV_REPORT	Whether to output CSV	1 (recommended to enable)
RESULT_DIR	Result output directory	Default .\results\C2M_host_{date}\, {date} expands to DD-Mon-YYYY
COM Pass threshold	COM pass threshold	0 dB

Appendix B Full SCPI Command Quick Reference

Command
*CLS
*ESE
*ESE?
*ESR?
*IDN?
*OPC
*OPC?
*RST
*SRE
*SRE?
*STB?
*TST?
*WAI
ABOR
ABOR:FIN
CHAR:AXIS:RANG
CHAR:AXIS:RANG:RES
CHAR:AXIS:RANG?
CHAR:CURV:LIST?
CHAR:CURV:VIS
CHAR:CURV:VIS?
CHAR:HEIG
CHAR:HEIG?

Command
CHAR:MARK:ADD
CHAR:MARK:ADD:ALL
CHAR:MARK:CLE
CHAR:MARK:CLE:SUBP
CHAR:MARK:DEL
CHAR:MARK:EDIT
CHAR:MARK:LIST?
CHAR:SUBP
CHAR:SUBP?
CONF:AMPL:AFE
CONF:AMPL:AFE?
CONF:AMPL:ANE
CONF:AMPL:ANE?
CONF:AMPL:AV
CONF:AMPL:AV?
CONF:CAL:MODE
CONF:CAL:MODE?
CONF:CAL:STEP
CONF:CAL:STEP?
CONF:CAL:THR
CONF:CAL:THR?
CONF:CASE:SEL
CONF:CASE:SEL?
CONF:DEMO:LIST?
CONF:DEMO:LOAD

Command
CONF:DEMO:LOAD?
CONF:FILE:CONF
CONF:FILE:CONF?
CONF:FILE:FEXT:ADD
CONF:FILE:FEXT:COUN?
CONF:FILE:FEXT:DEL
CONF:FILE:FEXT:LIST?
CONF:FILE:NEXT:ADD
CONF:FILE:NEXT:COUN?
CONF:FILE:NEXT:DEL
CONF:FILE:NEXT:LIST?
CONF:FILE:NOIS
CONF:FILE:NOIS:CLE
CONF:FILE:NOIS?
CONF:FILE:RES:DIR
CONF:FILE:RES:DIR?
CONF:FILE:THRU
CONF:FILE:THRU?
CONF:FILE:VAL?
CONF:GEN:CLEA
CONF:GEN:CLEA?
CONF:GEN:CLOS
CONF:GEN:CLOS?
CONF:GEN:EXP
CONF:GEN:EXP?

Command
CONF:GEN:PAR
CONF:GEN:PAR?
CONF:GEN:QUI
CONF:GEN:QUI?
CONF:LOAD
CONF:LOAD?
CONF:MODE
CONF:MODE?
CONF:PAR:APPL
CONF:PAR:CAT?
CONF:PAR:CLE
CONF:PAR:ENAB:ALL
CONF:PAR:INS
CONF:PAR:LIST?
CONF:PAR:SET
CONF:PAR:SET?
CONF:PATH:EXE
CONF:PATH:EXE?
CONF:PATH:MATLAB
CONF:PATH:MATLAB?
CONF:PATH:MCR
CONF:PATH:MCR?
CONF:PATH:SCR
CONF:PATH:SCR?
CONF:SAVE

Command
DISP:TAB
DISP:TAB?
FETC:ALL?
FETC:AS?
FETC:CAL:NOIS?
FETC:CAL:PASS?
FETC:CASE:COM?
FETC:CASE:COUN?
FETC:CASE:DATA?
FETC:CASE:LIST?
FETC:CHAR:PNG?
FETC:COM?
FETC:CSV:PATH?
FETC:DER?
FETC:ERL?
FETC:PASS?
FETC:PORT:ORD?
FETC:RUNT?
FETC:VEC?
FETC:VEO?
FIL:REP:CLE
FIL:REP:EXP
FIL:REP:EXP?
INIT
INIT:IMM

Command
LIC:ACT:BASE
LIC:ACT:LAST?
LIC:ACT:SCPI
LIC:BASE?
LIC:DEAC:BASE
LIC:DEAC:SCPI
LIC:HOST?
LIC:SCPI?
LIC:STAT?
LIC:TRIAL?
LOG:CLE
LOG:COUN?
LOG:DATA?
LOG:LAST?
LOG:SAVE
MEAS:ALL?
MEAS:COM?
MEAS:PASS?
SPAR:CASC:ADD
SPAR:CASC:CLE
SPAR:CASC:COUN?
SPAR:CASC:DEL
SPAR:CASC:DIFF:ENAB
SPAR:CASC:DIFF:ENAB?
SPAR:CASC:DIFF:MODE

Command
SPAR:CASC:DIFF:MODE?
SPAR:CASC:DIFF:PAIR
SPAR:CASC:DIFF:PAIR?
SPAR:CASC:LIST?
SPAR:CASC:PERM
SPAR:CASC:PERM?
SPAR:CASC:RUN
SPAR:DEEM:DIFF:ENAB
SPAR:DEEM:DIFF:ENAB?
SPAR:DEEM:DIFF:MODE
SPAR:DEEM:DIFF:MODE?
SPAR:DEEM:DIFF:PAIR
SPAR:DEEM:DIFF:PAIR?
SPAR:DEEM:DUT
SPAR:DEEM:DUT?
SPAR:DEEM:LEFT
SPAR:DEEM:LEFT?
SPAR:DEEM:PERM
SPAR:DEEM:PERM?
SPAR:DEEM:RIGHT
SPAR:DEEM:RIGHT?
SPAR:DEEM:RUN
STAT:OPER:COND?
STAT:OPER:ENAB
STAT:OPER:ENAB?

Command
STAT:OPER:EVEN?
STAT:OPER?
STAT:PRES
STAT:QUES:COND?
STAT:QUES:ENAB
STAT:QUES:ENAB?
STAT:QUES:EVEN?
STAT:QUES?
SYST:ERR:ALL?
SYST:ERR:COUN?
SYST:ERR:NEXT?
SYST:ERR?
SYST:FEED
SYST:FEED:CONT
SYST:FEED:ONL?
SYST:FEED:REPL?
SYST:HELP:HEAD?
SYST:INIT
SYST:LANG
SYST:LANG?
SYST:SCPI:AUTO
SYST:SCPI:AUTO?
SYST:SCPI:PORT?
SYST:UPD:CHEC?
SYST:UPD:DOWN

Command
SYST:UPD:INFO?
SYST:UPD:INST
SYST:UPD:STAT?
SYST:VERS?
VNA:ADDR?
VNA:CAL:STAT?
VNA:CAP?
VNA:CONN
VNA:CONN?
VNA:DATA:CSV:SAVE
VNA:DATA:SNP:DATA?
VNA:DATA:SNP:LOAD
VNA:DATA:SNP:SAVE
VNA:DATA:SNP:SEND
VNA:DATA:SNP:SEND?
VNA:DISC
VNA:IDN?
VNA:MEAS:SING
VNA:MEAS:SNP
VNA:MEAS:SWE
VNA:MOD?
VNA:PORT?
VNA:SCAN?
VNA:SCPI:QUER?
VNA:SCPI:SEND

Command
VNA:SCR:CAPT
VNA:SCR:SAVE
VNA:SET:APPL
VNA:SET:AVER
VNA:SET:AVER?
VNA:SET:CENT
VNA:SET:CENT?
VNA:SET:COM:APPL
VNA:SET:COM:PRES?
VNA:SET:FORM
VNA:SET:FORM?
VNA:SET:IFBW
VNA:SET:IFBW?
VNA:SET:PAR
VNA:SET:PAR?
VNA:SET:POIN
VNA:SET:POIN?
VNA:SET:POW
VNA:SET:POW?
VNA:SET:SMO
VNA:SET:SMO?
VNA:SET:SPAN
VNA:SET:SPAN?
VNA:SET:STAR
VNA:SET:STAR?

Command
VNA:SET:STOP
VNA:SET:STOP?
VNA:SET:SWEEP
VNA:SET:SWEEP?
VNA:SNP:PORT
VNA:SNP:PORT?
VNA:VEND
VNA:VEND:LIST?
VNA:VEND?

New Command Detailed Reference

Command	Parameters	Description
CONFigure:FILE:RESult:DIR <path>	Path string	Set result output directory (overrides RESULT_DIR in Excel)
CONFigure:FILE:RESult:DIR?	—	Query current result output directory
CONFigure:CALibration:MODE <ON|OFF|1|0>	ON/OFF/1/0	Enable/disable interference tolerance calibration mode
CONFigure:CALibration:MODE?	—	Query calibration mode state (1=enabled, 0=disabled)
CONFigure:CALibration:STEP <value>	Float	Set calibration step value (default 0.001)
CONFigure:CALibration:STEP?	—	Query current calibration step value
CONFigure:CASE:SELect <list|ALL>	Comma-separated ints or ALL	Select cases to run (e.g. 1,3,4 or ALL)
CONFigure:CASE:SELect?	—	Query current case selection

Command	Parameters	Description
FETCh:CALibration:NOISe?	—	Query calibration noise values (sigma_hp,sigma_bn in mV)
FETCh:CALibration:PASS?	—	Calibration success (1=success, 0=no data)
FETCh:PORT:ORDer?	—	Query port order (e.g. 1,3,2,4)
FETCh:CSV:PATHs?	—	Query comma-separated list of generated CSV result file paths
CONFigure:GENeral:CLEArresults <ON|OFF>	ON/OFF/1/0	Clear result directory before run (default ON)
CONFigure:GENeral:CLEArresults?	—	Query clear results setting
CONFigure:SAVE <filepath>	File path	Save current configuration to JSON file
CONFigure:LOAD <filepath>	File path	Load configuration from JSON file
CONFigure:LOAD?	—	Query current configuration summary
FILe:REPort:EXPort <filepath>	File path	Export PDF report (GUI mode only)
FILe:REPort:EXPort?	—	Query last exported report path
FILe:REPort:CLEar	—	Clear report tab entries (v1.5.12)
SYSTem:LANGuage <EN|ZH>	EN/ZH/en_US/zh_CN	Set interface language
SYSTem:LANGuage?	—	Query current interface language
SYSTem:SCPI:PORT?	—	Query SCPI server port (v1.5.12)
SYSTem:SCPI:AUTOstart <ON|OFF>	ON/OFF/1/0	Auto-start SCPI server on GUI launch (v1.5.12)
SYSTem:SCPI:AUTOstart?	—	Query auto-start setting (v1.5.12)
ABORt:FINish	—	Skip silent-wait period and accept current results (v1.5.12)

Command	Parameters	Description
CONFigure:PARameter:INSert <kw>,<val>	keyword,value	Insert new keyword row when xlsx template lacks it
CONFigure:PARameter:APPLy	—	Write current overrides to xlsx without running MATLAB (v1.5.12)
LOG:SAVE "<path>"	Path	Save log to file (v1.5.12)
CHARt:MARKer:ADD <subplot>,<freq>	subplot,GHz	Add marker on specified subplot (v1.5.12)
CHARt:MARKer:DELete <subplot>,<freq>	subplot,GHz	Delete nearest marker on specified subplot (v1.5.12)

VNA Network Analyzer Commands

Command	Parameter	Description
VNA:CONNect <addr>	VISA address	Connect to VNA instrument (requires SCPI license)
VNA:CONNect?	—	Query connection status (1=connected, 0=not)
VNA:DISConnect	—	Disconnect from VNA
VNA:SCAN?	—	Scan for instruments (GUI mode only)
VNA:IDN?	—	Query connected VNA IDN
VNA:ADDRess?	—	Query current connection address
VNA:SETup:CENTer <GHz>	Float	Set center frequency
VNA:SETup:SPAN <MHz>	Float	Set frequency span
VNA:SETup:POINts <N>	Integer	Set sweep points
VNA:SETup:PARameter <S11|...>	S11/S21/S12/S22	Set measurement parameter

Command	Parameter	Description
VNA:SETup:FORMat <fmt>	MLOG/PHAS/REAL/IMAG	Set display format
VNA:SETup:APPLy	—	Apply settings to instrument
VNA:MEASure:SWEep	—	Start frequency sweep
VNA:MEASure:SINGle	—	Single measurement
VNA:MEASure:SNP	—	Get SnP parameters
VNA:SCReen:CAPTure	—	Capture VNA screenshot
VNA:SCReen:SAVE <path>	File path	Save screenshot to file
VNA:SCPI:SEND <cmd>	SCPI command	Pass-through write to VNA
VNA:SCPI:QUERy? <cmd>	SCPI command	Pass-through query to VNA
VNA:DATA:SNP:SAVE <path>	File path	Save SnP data to file
VNA:DATA:CSV:SAVE <path>	File path	Export measurement CSV
VNA:VENDor <id>	keysight_pna/ena/rs_znb/anritsu_sl	Select VNA vendor (before connect)
VNA:VENDor?	—	Query current vendor id
VNA:VENDor:LIST?	—	List registered vendor ids
VNA:MODel?	—	Query instrument model (from IDN)
VNA:PORTs?	—	Query configured port count
VNA:SNP:PORTs <list>	1,2,3,4	Set Touchstone port list
VNA:CAPabilities?	—	JSON: driver capabilities

License Remote Activation (v1.3.1+)

Allows SCPI clients to activate license keys remotely. When not activated, scpi_main.py still starts (prints warning banner) but INIT/MEAS/VNA commands are rejected with -200 until activation succeeds. **Deactivate is NOT exposed** over SCPI — any deactivation must go through the GUI license dialog.

Command	Argument	Description
LICense:HOSTid?	—	Query Host ID
LICense:BASE?	—	Base license active? (1/0)
LICense:SCPI?	—	SCPI option license active? (1/0)
LICense:TRIAL?	—	Trial runs remaining
LICense:STATus?	—	Composite JSON status
LICense:ACTivate:BASE <key>	Key string	Activate base permanent or temporary key
LICense:ACTivate:SCPI <key>	Key string	Activate SCPI option (base must be active first)
LICense:ACTivate:LAST?	—	Last activation result: "1,msg" or "0,msg"

Security note: LIC:ACT transmits keys in plain text over TCP. Recommend running scpi_main.py with `--host 127.0.0.1` (localhost only) or inside a trusted VPN.