

27 May 2026

ResistaMet GUI: open-source electrical characterization for Keithley 2400 series sourcemeters

Brenden Ferland¹

1. Polymer Program, University of Connecticut

Abstract

ResistaMet GUI is a desktop application for electrical characterization using Keithley 2400 series source measure units (SMUs) over a GPIB interface. These instruments are commonly used by electrical engineers, materials scientists, and other physical scientists to characterize electrically active devices and materials. This software is benchwritten and tested with 6 primary measurement modes: two- and four-wire resistance, constant-voltage and constant-current biasing for long runs (with in-run event annotations), I-V sweep, collinear 4-point probe per ASTM F84, and van der Pauw per ASTM F76 Method A. Each reading carries an instrument uncertainty propagated from the Keithley datasheet's per-range accuracy specs, all the way through to derived quantities like sheet resistance and resistivity. An in-package Keithley simulator, calibrated against captured SCPI traces from our bench Keithley 2420 and replayed in CI, allows users to test and practice each mode without an SMU, and keeps documented hardware quirks from regressing.

Keywords

Keithley 2400, sourcemeter, four-point probe, van der Pauw, ASTM F84, ASTM F76, electrical characterization, sheet resistance, laboratory software, open source, PyVISA, PySide6

ResistaMet GUI: open-source electrical characterization for Keithley 2400 series sourcemeters

Brenden Ferland^{1,*}

¹Polymer Program, Institute of Materials Science, University of Connecticut, Storrs, CT 06269

Abstract ResistaMet GUI is a desktop application for electrical characterization using Keithley 2400 series source measure units (SMUs) over a GPIB interface. These instruments are commonly used by electrical engineers, materials scientists, and other physical scientists to characterize electrically active devices and materials. This software is bench-written and tested with 6 primary measurement modes: two- and four-wire resistance, constant-voltage and constant-current biasing for long runs (with in-run event annotations), I-V sweep, collinear 4-point probe per ASTM F84, and van der Pauw per ASTM F76 Method A. Each reading carries an instrument uncertainty propagated from the Keithley datasheet's per-range accuracy specs, all the way through to derived quantities like sheet resistance and resistivity. An in-package Keithley simulator, calibrated against captured SCPI traces from our bench Keithley 2420 and replayed in CI, allows users to test and practice each mode without an SMU, and keeps documented hardware quirks from regressing. The software includes live plotting, built-in safety mechanisms for both technicians (touch-safety voltage warnings) and samples (a probe-power envelope), crash-safe streaming output, and many other features and bug fixes hard-earned from using the software in our lab.

*E-mail: brenden.ferland@uconn.edu

Keywords: Keithley 2400, sourcemeter, four-point probe, van der Pauw, ASTM F84, ASTM F76, electrical characterization, sheet resistance, laboratory software, open source, PyVISA, PySide6

Statement of need

Keithley 2400 Series SMUs are standard equipment in academic and industry laboratories specializing in electrical characterization. A typical experiment can be anything from long-duration fixed-bias logging with event annotations, periodic four-point-probe surveys, I-V sweeps, chronoamperometry, and 2- or 4-wire resistance measurements. These instruments are traditionally driven by hand-written scripts in LabVIEW, MATLAB, or Python libraries like `pymeasure` and `QCoDeS`. All of these require the user to write and maintain code, and none ship four-point-probe analysis. Modern paid software is sold by Tek as `KickStart` for \$610–\$2,200 per seat, omits four-point-probe analysis, and is restricted to recent touchscreen 2400-series models. Keithley's bundled `LabTracer 2` (circa 2006, since discontinued) is a transistor-focused two-SMU I-V configurator with no four-point probe, van der Pauw, long fixed-bias logging, or in-run event annotation. No-code free alternatives are sparse and fragmented, often written in a bespoke manner for a single application. ResistaMet GUI covers that combined workflow on the same hardware academic groups already own. It is free, open, and aims to match or exceed paid software. Beyond the instrument and a computer, the only purchase needed is a GPIB to USB adapter.

State of the field

The closest direct peer to ResistaMet GUI is commercial software sold by Tektronix (the OEM for Keithley) called `KickStart`. This is a Windows-only point-and-click GUI sold both as a subscription and perpetual licensing at a per-seat cost of \$610 per year or \$2,200 to own it. Most features are limited per-instrument and the Keithley 2400 series is

limited to only I-V sweep mode. The now-discontinued LabTracer 2 (c. 2006) was a two-SMU I-V configurator with no four-point-probe or long-duration logging bundled with the instrument.

Researchers needing to use older instruments have other options, though none are complete. Programmatic scripting frameworks (pymeasure [1], QCoDeS [2]) offer procedure scaffolding and instrument drivers for users capable of writing their own code. Community-developed open source GUIs are most commonly tuned for narrow-scope subfields or tests. MeaSSUre:I-V [3] targets transistor-memory and neuromorphic device characterization. QKeithleyControl [4] covers transistor sweeps and solar-cell tracking. keithleygui [5] is built for organic-FET work, specifically using the Keithley 2600 series. F  bba et al.'s LLM-driven controller [6] fits single-diode parameters to solar-cell I-V curves and is the published precedent for using a large language model to write Keithley control software, an approach this work also follows (see AI usage disclosure). GatherLab/four-point-sheet-resistance covers one specific four-point probe setup.

ResistaMet GUI covers a combined materials-characterization workflow that, to my knowledge, has not been previously available in a single point-and-click application. There are six measurement modes, backed up by measurement standards where available: resistance over time, fixed-bias modes for current and voltage, a dedicated mode for collinear 4-point probe measurements (ASTM F84-aligned) [7, 8], ASTM F76 van der Pauw resistivity with the §11.1 homogeneity gate [9], and an I-V sweep mode. ResistaMet is explicitly designed for a broad array of uses; while individual programs may be better suited for niche use cases, I aim to have the best general-purpose application for Keithley 2400 series electrical characterization.

Software design

ResistaMet GUI evolved from a set of bespoke CLI scripts used for sensor and power-generation monitoring over long windows. Eventually, non-computer-savvy lab members needed to use the software as well and had difficulty with the abrasive CLI interface. The GUI was thus born, and has been developed from the ground up with a philosophy of 'per-user' organization and personalization. This unlocked a better scientific user experience in the form of live feedback and easily switchable controls. The name ResistaMet comes from a weak and now embarrassing portmanteau of 'resistance + metrology', as the software originally measured only resistance over time for electrically conductive foams. Unfortunately, I've determined that it is too late for a proper rename.

As different needs were flagged by labmates and myself, more features were added. Voltage and current source covered chronoamperometry, electrochemistry, and device biasing. Four-point probe added in-app analysis grounded in standards for the lab's Signatone S-302 stand, replacing a workflow of raw I-V capture in Keithley's bundled LabTracer 2 (circa 2006, since discontinued) plus hand calculation. I-V sweep added transistor and diode work as a 'standard' feature, and van der Pauw mode handles samples where a collinear four-point probe head will not seat correctly.

Over time, the software evolved through bench necessities and frustrations. For example, an unexpectedly conductive sample resulted in sending far too much power to a Signatone SP4 tungsten-carbide collinear probe tip, resulting in it melting. In response to this, I added over-power protections. While running a van der Pauw measurement, it became abundantly clear that swapping electrodes needs to have a visual guide to prevent confusion.

The implementation originally used PyQt5 and matplotlib but was migrated to PySide6 due to Qt5 EOL and weaker copleft. pyqtgraph was used for enhanced visualization at higher framerates. NumPy [10] is used for buffer statistics and four-point-probe math, and PyVISA [11] for instrument communication on Python 3.9 or later. The output section of settings offers HDF5 (when h5py is installed). Gzip compression is off by default because lab tools like Excel and Origin cannot open .gz and the core philosophy is that this software should be usable by someone who doesn't know what a .gz file is.

Most per-mode SCPI sequences were tested and corrected against a live Keithley 2420 (firmware C30) across two parallel passes in April 2026. The first pass walked each command individually with :SYST:ERR? checks. Five defects surfaced, including silent re-ordering of :FORM:ELEM outputs and a misaligned compliance bit. The second pass was a full read of the ~300-page Keithley 2400-series User Manual [12], which surfaced on-instrument features the GUI was not using (hardware sweep engine, auto-zero ONCE, hardware averaging filter, offset-compensated ohms, cable-null reference) that allowed for faster or more accurate measurements. An in-package SCPI simulator (described in Validation) keeps the fixes from regressing in CI, and a --simulate flag exposes it to end users so labmates can learn the workflow without GPIB hardware and reviewers can review without owning a Keithley. Per-reading measurement uncertainty is reported

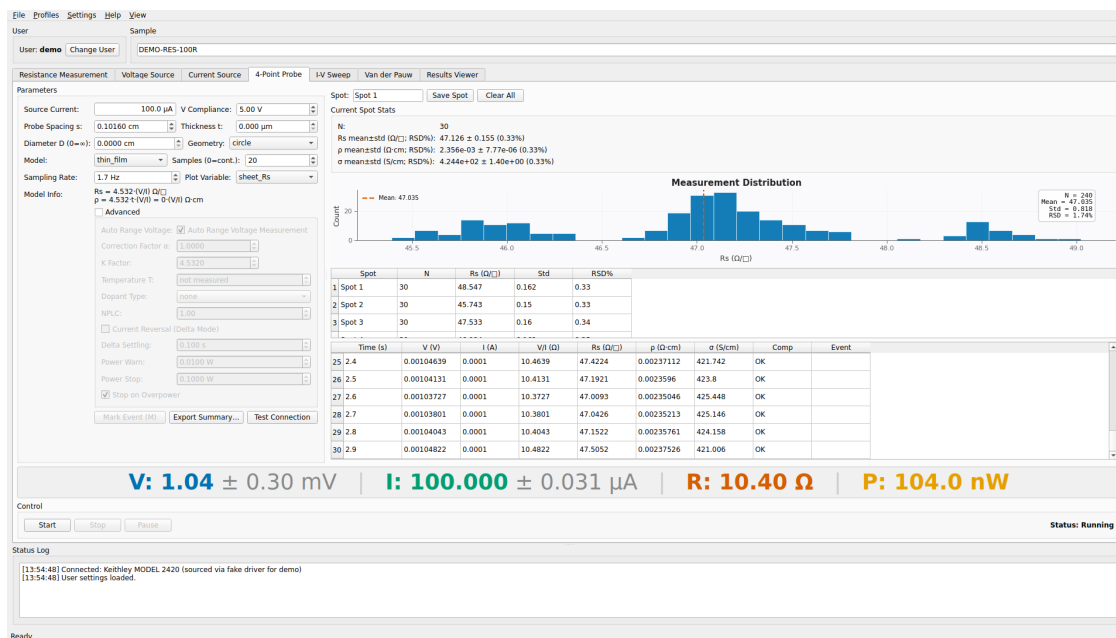


Figure 1. ResistaMet GUI four-point-probe tab: live readout, per-spot statistics panel, and sheet-resistance histogram populated from a multi-spot survey. The Advanced panel exposes ASTM F84 correction factors, current-reversal (delta) mode, and the probe-power safety envelope.

based on values from the manual and Keithley datasheet.

The 4PP mode implements ASTM F84-02's correction-factor decomposition $F = F_2 \cdot F(w/S) \cdot F_T$ [7, 8] with unit tests pinned to the standard's tabulated reference values for finite sample diameter, thickness (out to $w/S = 2.0$), and temperature (n- and p-type silicon). The van der Pauw worker implements F76-08 Method A [9] as a state machine that emits `geometry_ready`, blocks on a `threading.Event` while the user reconnects four corner clips, sources $+I$ and $-I$, and enforces `:OUTP OFF` between geometries. F76 §11.1's homogeneity gate runs against the eight resulting voltages. Standards are enforced in software with sensible defaults while still exposing raw per-reading values, so users can run their own calculations when needed.

Validation

A functional Keithley simulator was built during the validation process for both reviewers without access to an instrument and for training/demonstration purposes. A Keithley 2420 (firmware C30) was wired to reference resistors at three different operating decades (100 Ω, 10 kΩ, 1 MΩ) in an asymmetric 4-wire configuration (symmetry was not required for this as instrument response was desired, not perfect accuracy). The FakeKeithley was written backwards from these traces; everything is reverse-engineered from real instrument behavior, not written from specs. The simulator is cross-validated against a Keithley 2400, and there is an open invitation for the community to validate it against their own instrument using `scripts/community_capture.py`.

End-to-end tests drive every measurement feature through the UI-to-CSV pipeline and check validity against a simulated DUT. A hardware-in-the-loop suite executes against the bench instrument as a secondary validation if one is available. Accepted submissions are replayed automatically in CI on Python 3.9–3.12 across Linux and Windows.

Research impact

ResistaMet GUI has been used to collect electrical data for two publications within the Adamson lab at UConn. Voltage and current were both measured over time for a peer-reviewed study in *Journal of Materials Chemistry A* for moist-electric generators based on graphene hydrogel composites [13]. The software was not cited directly as it was not known to be possible. The software was also used for piezoresistive characterization and voltage-controlled de-icing [14]. The Sun

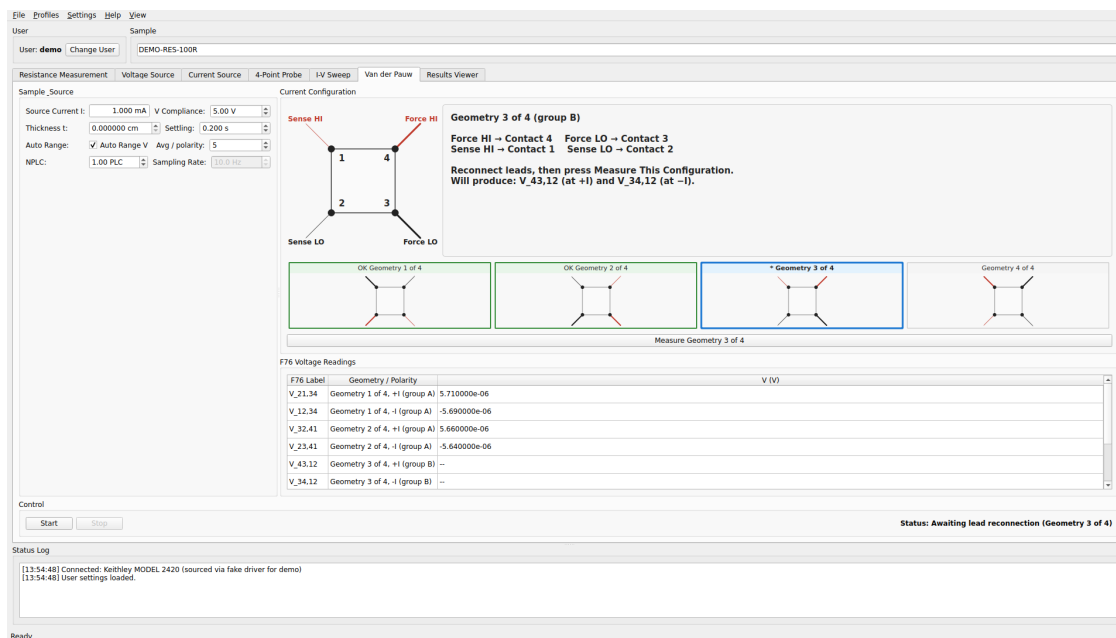


Figure 2. ResistaMet GUI Van der Pauw tab, mid-protocol. The large sample diagram shows which corner each instrument lead should attach to for the active F76 geometry; the filmstrip below tracks progress through the four geometries (green = complete, blue = active, gray = pending). Output is disabled between geometries so the operator can safely reconnect leads.

group at UConn Institute of Materials Science has adopted the software for four-point probe measurements.

AI usage disclosure

Claude (Anthropic) was used extensively as a coding assistant; the majority of the source code, test scaffolding, and manuscript drafts were AI-generated under the author's direction, using Claude models from 3.7 Sonnet (Feb 2025) through Opus 4.7 (Apr 2026). All AI output was grounded against the Keithley Series 2400 SMU Datasheet and User Manual, ASTM F84 and F76, and SCPI traces captured from a live Keithley 2420 (firmware C30). Every measurement mode was bench-tested on that instrument by the author. Design decisions, scientific interpretation, and final responsibility for the code and claims in this paper rest with the author.

Code and Data Availability

ResistaMet GUI source code is hosted at <https://github.com/PEEKPerformer/ResistaMet-GUI> under the MIT license. Tagged releases are archived on Zenodo; the concept DOI 10.5281/zenodo.19919751 always resolves to the latest archived version. Captured SCPI traces used for simulator validation are committed in `tests/fixtures/scpi_traces/` and replayed by `tests/test_fake_matches_hardware.py` on every CI run.

Acknowledgements

Special thanks to Prof. Douglas H. Adamson (UConn) for providing the instrumentation required for this project and for allowing me to pursue interests like this despite not being direct materials research. Thanks to Adamson group members Jakiya Sultana Joya, Rumesha Anthoni M. Pererage, and Mahdad Mahmoudi for both intentionally and unintentionally breaking the software during debugging.

This work was supported in part by the U.S. Department of Education Graduate Assistance in Areas of National Need (GAANN) program under award P200A240111 to the University of Connecticut Polymer Program (PI: Mu-Ping Nieh). The contents do not necessarily represent the policy of the Department of Education, and you should not assume endorsement by the Federal Government.

Conflict of Interest

The author declares no competing financial interests.

References

- [1] PyMeasure Developers. *PyMeasure: scientific measurement library for Python*. Version 0.16.0. 2024. DOI: 10.5281/zenodo.595633. URL: <https://github.com/pymeaure/pymeaure>.
- [2] Jens H. Nielsen and QCoDeS contributors. *QCoDeS: a modular data acquisition framework*. Version 0.58.0. 2026. URL: <https://github.com/microsoft/Qcodes>.
- [3] Hongseok Oh, Hyunsoo Kim, and Hyerin Jo. “MeaSSUre:I-V: Open software for transistor characterization using source-meter units”. In: *SoftwareX* 21 (2023), p. 101318. DOI: 10.1016/j.softx.2023.101318. URL: <https://doi.org/10.1016/j.softx.2023.101318>.
- [4] solarpx. *QKeithleyControl: Qt5 UI for controlling Keithley 2400 sourcemeters*. Repository attributed only to GitHub handle solarpx; no formal release tag, last commit February 2020. 2020. URL: <https://github.com/solarpx/QKeithleyControl>.
- [5] Sam Schott. *keithleygui: a Python-based user interface for Keithley 2600-series instruments*. Version 1.2.1. 2021. URL: <https://github.com/OE-FET/keithleygui>.
- [6] Davi Fébba et al. “From text to test: AI-generated control software for materials science instruments”. In: *Digital Discovery* 4.1 (2025), pp. 35–45. DOI: 10.1039/D4DD00143E. URL: <https://doi.org/10.1039/D4DD00143E>.
- [7] ASTM International. *Standard Test Method for Measuring Resistivity of Silicon Wafers With an In-Line Four-Point Probe*. Standard ASTM F84-02. West Conshohocken, PA: ASTM International, 2002. DOI: 10.1520/F0084-02. URL: <https://www.astm.org/f0084-02.html>.
- [8] F. M. Smits. “Measurement of Sheet Resistivities with the Four-Point Probe”. In: *Bell System Technical Journal* 37.3 (1958), pp. 711–718. DOI: 10.1002/j.1538-7305.1958.tb03883.x.
- [9] ASTM International. *Standard Test Methods for Measuring Resistivity and Hall Coefficient and Determining Hall Mobility in Single-Crystal Semiconductors*. Standard ASTM F76-08(2016)e1. West Conshohocken, PA: ASTM International, 2016. DOI: 10.1520/F0076-08R16E01. URL: <https://www.astm.org/f0076-08r16e01.html>.
- [10] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [11] Hernán E. Grecco et al. “PyVISA: the Python instrumentation package”. In: *Journal of Open Source Software* 8.84 (2023), pp. 5304–5308. DOI: 10.21105/joss.05304. URL: <https://doi.org/10.21105/joss.05304>.
- [12] *Series 2400 SourceMeter User’s Manual*. Rev. K. Document 2400S-900-01, September 2011. Keithley Instruments, Inc. Cleveland, OH, 2011.
- [13] Nishadi M. Bandara et al. “Moisture-driven electric power generation using graphene-based hydrogels for sustained power output and self-powered sensing”. In: *Journal of Materials Chemistry A* 14.26 (2026), pp. 16724–16734. DOI: 10.1039/D5TA09936F. URL: <https://doi.org/10.1039/D5TA09936F>.
- [14] Brenden Ferland and Douglas H. Adamson. *Interfacial Self-Assembly of Graphene Networks at 1 vol% Enables Piezoresistive and Electrothermal Silicone Foams*. Preprint. 2026. DOI: 10.26434/chemrxiv-2026-vsz4h. URL: <https://doi.org/10.26434/chemrxiv-2026-vsz4h>.