PICA: Advanced High-Precision Transport Measurement Automation with Python

Prathamesh Deshmukh[1,2], Sudip Mukherjee[1,2,*]

[1] UGC-DAE Consortium for Scientific Research, Mumbai Centre, Bhabha Atomic Research Centre, Mumbai, 400 085, Maharashtra, India (ROR: 04cmez398)

[2] Savitribai Phule Pune University, Ganeshkhind Road, Pune, 411 007, Maharashtra, India (ROR: 044g6d731)

[*] Corresponding author: sudipm@csr.res.in

DOI: 10.5281/zenodo.18377216

22 December 2025

Summary

High-precision, low-noise transport measurements are essential for advancing research in spintronics and materials characterisation. To enable such progress, highly precise and accurate automation software is required. PICA (Python-based Instrument Control and Automation) is a modular open-source software suite designed to automate advanced transport measurements for electronic devices and material samples. PICA is designed as a versatile framework capable of operating on any standard laboratory workstation. It provides an extensible, unified graphical user interface (GUI)-based standalone program to coordinate high-precision instruments, specifically ultra-low current source (DC/AC) units, nanovoltmeters, high-resistance electrometers, impedance analyser, and temperature controllers. Built on the robust Python scientific ecosystem, PICA leverages community standard libraries as an alternative to licensed commercial software for instrument control. By utilising `threading` and `multiprocessing` capabilities, PICA ensures that the entire hardware ecosystem functions seamlessly and as a single cohesive unit. This allows the system to perform automated protocols, including temperature-dependent wide ranges of resistance measurement (10^{-8} - 10^{16} Ω), current-voltage (I-V) characterisation, capacitance characterisation and pyroelectric current measurement, and orchestrate measurements under varying magnetic fields and temperatures without requiring physical reconfiguration of the measurement setups.

Statement of need

Advancements in experimental physics and device manufacturing depend on the precise characterisation of material properties under extreme physical conditions (e.g. low temperature and high magnetic/electric fields). For automating experiments, researchers have to choose between expensive proprietary graphical programming environments or developing a custom measurement script from scratch. While powerful ecosystem libraries such as PyVISA (Grecco et al. 2023) and PyMeasure (PyMeasure Developers 2025) provide the foundational drivers for instrument communication, they are fundamentally software libraries that require the user to write and maintain code, creating a barrier to entry for researchers requiring direct data acquisition without the overhead of developing and maintaining a custom codebase.

Research impact

PICA addresses this gap by functioning as a turnkey application rather than as a library. It offers a "ready-to-run" graphical interface that abstracts the underlying control logic, allowing experimentalists to focus on data acquisition without needing to develop custom software scripts for the supported hardware configurations. PICA's architecture is designed to be highly configurable, enabling users to readily adapt it to their specific requirements and to implement user-defined protocols in addition to the standard measurement protocols already provided. It eliminates the need for reconfiguring the measurement setup to achieve comprehensive characterisation, enabling continuous operation across the full range from ultra-low-resistance measurements (with the current reversal technique effectively removing constant offsets and improving the signal-to-noise ratio) for superconductors to high-impedance electrometric measurements for high-band gap insulators (covering 24 orders of magnitude in resistance), using a single unified framework. Pyroelectric measurement performed using an electrometer enables a highly sensitive

characterisation of ferroelectric phase transitions by detecting extremely small pyroelectric currents, with a resolution on the order of 10^{-15} A. The impedance analyser enables the characterisation of capacitance anomalies over the frequency range from 20 Hz to 2 MHz and is utilised for magnetocapacitance and photoinduced characterisation across a wide variety of multiferroic systems. Thus, the primary objective of PICA is to serve as a robust software platform that enables advanced high-precision characterisation of materials.

The system is currently validated with scientific benchmark SMU (Source Measure Unit) hardware, including the AC-DC current source (Model: 6221, Keithley), the Nanovoltmeter (Model: 2182, Keithley), the Electrometer (Model: 6517B, Keithley), the DC Source Measure Unit (Model: 2400, Keithley), the impedance analyser (Model: E4980A, Keysight), and the temperature controller (Model: 350/340, Lakeshore). While the current implementation drives specific instruments, the underlying framework is highly customisable. Researchers using different hardware models need only replace the specific SCPI (Standard Commands for Programmable Instruments) commands with their instrument equivalent commands to utilise the suite.

It differentiates itself through the following unique features:

- **Accessibility:** A user-friendly GUI dashboard that allows researchers without coding experience to configure and run a complex measurement protocol immediately using the suite's pre-packaged measurement modules.
- **Operational Validation:** PICA's protocols are actively used for transport measurements at cryogenic temperature using a laboratory-built, custom-designed, multifunctional cryostatic probe in-conjunction with the Physical Property Measurement System (PPMS, DynaCool, Quantum Design) (temperature range: 5-380 K, magnetic field: up to 14 T) at the UGC DAE Consortium for Scientific Research, Mumbai Centre, validating the software's core architecture in a real-world research environment and providing a stable, tested foundation for the university and researchers to build upon.
- **Fault Tolerance:** PICA prevents hardware timeouts or driver crashes from freezing the main dashboard by isolating control logic from the user interface, which is a critical advantage over single-threaded scripts.
- **Modular CLI Architecture:** As demonstrated in the repository, measurement modules also contain CLI (Command-line interface) measurement module counterparts that allow researchers to utilise PICA's measurement protocol and logic for headless automation or integration into other workflows without GUI overhead.
- **Operational Transparency:** Unlike a black box solution, PICA exposes real-time, time-stamped command logs for measurement modules, such as `[10:05:25] Keithley 6221: Ramping current to 10 mA`, Rejecting hidden automation and replacing the "black box" paradigm with transparent console logs that show every command sent to the instrument, thereby aiding debugging, ensuring the scientific reproducibility of experimental results, and allowing researchers to verify measurement protocols and troubleshoot hardware instantly.
- **Open Source Extensibility:** PICA's modular design allows researchers to easily integrate new instrument drivers or experimental protocols by subclassing existing templates, fostering a community-driven ecosystem for instrument control. This ensures that the software remains adaptable, allowing researchers to extend support for their unique instrument configurations.

Software design

PICA is built on a modular architecture characterised by self-contained modules, ensuring future extensibility. This design allows individual measurement protocols to be modified independently or added without impacting the core system stability.

Process Isolation and Concurrency

Unlike simple script-based automation, PICA decouples the User Interface (UI) from the instrumentation control logic. It utilises Python's standard `multiprocessing` library to spawn isolated processes for measurement tasks.

- **Stability:** If an instrument hangs or a communication bus times out, the isolated process can be terminated safely without freezing the main GUI or losing previous data.
- **Responsiveness:** The `tkinter`-based frontend remains responsive for live data plotting (using `matplotlib` (Hunter 2007) with blitting) even while the backend waits for hardware triggers. NumPy (Harris et al. 2020) is utilised throughout this pipeline for efficient array manipulation and data validation during real-time updates.

- **Data Integrity:** Experimental data integrity is prioritised through a "write on acquisition" strategy. Data is structured using `pandas` (Pandas Developers 2025) and is saved to a `CSV` file immediately after every acquisition point, preventing data loss in the event of a power failure or program/system crash.

Hardware Abstraction Layer

PICA utilises **PyVISA** (Grecco et al. 2023) to abstract the low-level communication protocols (GPIB, USB, Ethernet). The software implements a strict initialisation routine:

1. **Connection Verification:** A built-in "VISA (Virtual Instrument Software Architecture) Instrument Scanner" queries the bus (`*IDN?`) to map the connected instrument addresses.
2. **Instrument Reset Protocol:** To eliminate the influence of all previous experiments, all stored data, instrument buffers, and existing settings or configurations are explicitly reset, thereby providing a clean initial state before each measurement.
3. **Graceful Shutdown:** A "Safety Shutdown Routine" logic ensures that sources are ramped down to zero and heaters are disabled safely, even if the software is interrupted unexpectedly.

Testing and Simulation

To ensure measurement reliability, all of these modules were thoroughly tested with the corresponding hardware. Additionally, to facilitate development without constant access to physical instruments, PICA includes a testing suite that uses `pytest`. The suite employs `unittest.mock` to simulate VISA resources, allowing verification of backend logic streams, class structure, and command sequences in a continuous integration environment.

Acknowledgements

We acknowledge the financial support provided under the SERB-CRG project grant No. CRG/2022/005676 from the Anusandhan National Research Foundation (ANRF), a statutory body of the Department of Science and Technology (DST), Government of India.

References

Grecco, Hernán E., Matthieu C. Dartiailh, Gregor Thalhammer-Thurner, Torsten Bronger, and Florian Bauer. 2023. "PyVISA: The Python Instrumentation Package." *Journal of Open Source Software* 8 (84): 5304. https://doi.org/10.21105/joss.05304.

Harris, Charles R., K. Jarrod Millman, Stéfan J. Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, et al. 2020. "Array Programming with NumPy." *Nature* 585 (7825): 357–62. https://doi.org/10.1038/s41586-020-2649-2.

Hunter, John D. 2007. "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering* 9 (3): 90–95. https://doi.org/10.1109/MCSE.2007.55.

Pandas Developers. 2025. "pandas: Python Data Analysis Library." Zenodo. https://doi.org/10.5281/ZENODO.3509134.

PyMeasure Developers. 2025. "PyMeasure." Zenodo. https://doi.org/10.5281/ZENODO.595633.