

Nanonis Programming Interface

THE POWER OF CUSTOMIZATION – LEGO FOR SPMS

Nanonis provides an extensive LabVIEW programming environment for its SPM Control System. This interface allows to programmatically control all Nanonis modules. The user can quickly develop routines and automate specifically designed experiments. Most SPM parameters can be set/read, have call-back routines or trigger events from such programming framework.

Key advantages of the programming interface are:

- **No limitation in scope:** Nanonis provides a real programming interface and not a scripting language. The advantage is that the user has full access to all features LabVIEW provides (graphs, database access, convenient data handling, GPIB/RS232/USB access to other instruments,...) and is not limited to the features that the scripting engine provides.
- **Full development environment:** The user can benefit from the full LabVIEW programming environment and debugging features. Programs can be traced step-by-step, variables inspected and run-time behavior tested. All Nanonis VIs are incorporated in LabVIEW as a *User Library* and are readily accessed through the *Functions Palette* (see Fig. 2).
- **User interface:** Using LabVIEW the user can immediately design an appealing user interface coming with his routine.
- **Offline development:** The development and testing of user routines can be done “offline” without the need of a real-time system. Programs can be tested with the Nanonis STM Simulator and transferred to a “life” system only when all the bugs have been squashed.
- **Ease of use and quick prototyping:** LabVIEW's ease of use makes it possible to quickly develop your own custom routines without an extensive knowledge a programming language.
- **Transparent network access:** The user routine can run on the same computer as the Nanonis software or on a computer on the network.
- **No electronics knowledge required:** Old controllers require TTL handshakes and difficult to debug trigger electronics to communicate with external equipment - now you can directly talk to the hardware, read back full data sets from other instruments and integrate them directly with the SPM control system.

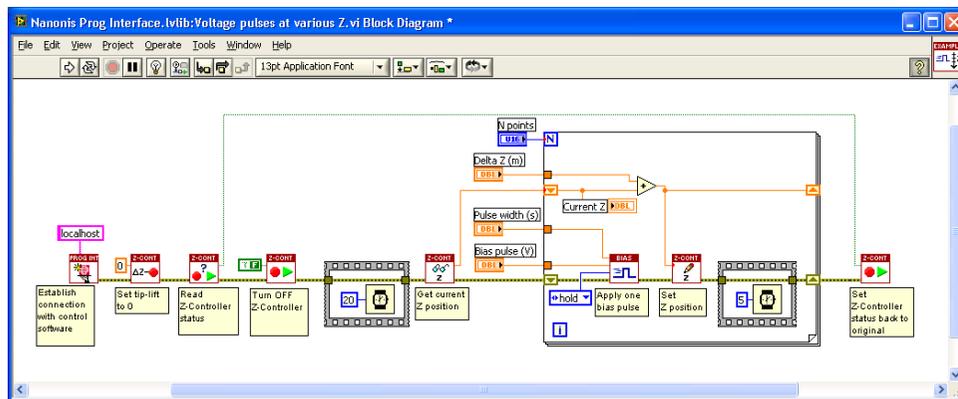


Figure 1: Example script to apply voltage pulses at 10 different heights above the sample.

SELECTION OF EXAMPLES INCLUDED WITH THE PROGRAMMING INTERFACE:

Spectroscopy:

- **Non-linear Sweep:** Define your own sweep function for bias or Z and acquire data while running the sweep. This makes it possible to perform e.g. a logarithmic backward sweep in z while acquiring the tunnelling current.
- **Multi-Segment bias spectroscopy with user-defined interpolation points:** The user can define a sequence of base points (and timings) from which one single spectroscopy curve is composed. The acquired data points do not necessarily have to be equidistant and the sequence does not have to be continuous. The acquired data is displayed immediately.
- **Store and recall multiple spectroscopy settings:** The user can store spectroscopy settings of a particular experiment and recall them later at the click of a button. Not limited in the number of settings that can be stored.
- **Run arbitrary spectroscopy routine:** The user can write his own spectroscopy routine reading data via GPIB/RS232/... from external equipment. The routine is then called at each grid point and the data directly displayed on screen.

External Equipment

- **Read attributes from external equipment:** Read attributes from external equipment and include these as key-value-pairs in the file headers of your scan data, e.g. the cryostat temperature, magnetic field,...
- **Save images as PNG:** Example to demonstrate data handling
- **Auto-Approach with customer motor module:** Example to do an auto-approach with any customer provided motor module.
- **Simplified Scan Interface:** Example showing the possibilities of building a custom user-interface that include only the controls the user want to see.

Scanning

- **Scan multiple images with different bias:** Define a list of bias voltages and scan the same image multiple times with different bias voltages.

Manipulation

- **Apply series of bias pulses at different tip-heights:** see example in Figure 1, a series of bias pulse is applied at different tip heights.

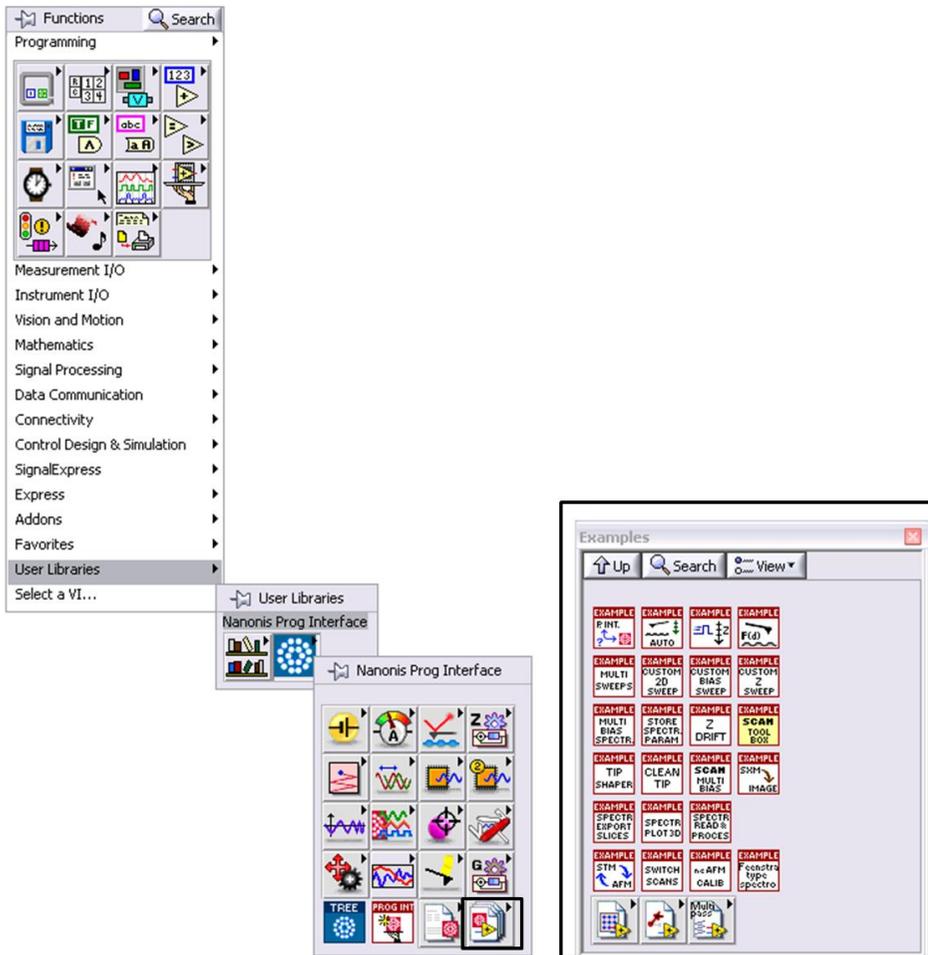


Figure 2: Palette of the Nanonis programming interface. Such sub-palette is directly accessible as a User Library from the Functions Palette within LabVIEW. How to display the vis included in one sub-palette of the programming interface, here the Examples are shown. It can also be displayed permanently if pinned as shown in the upper right corner of this image.

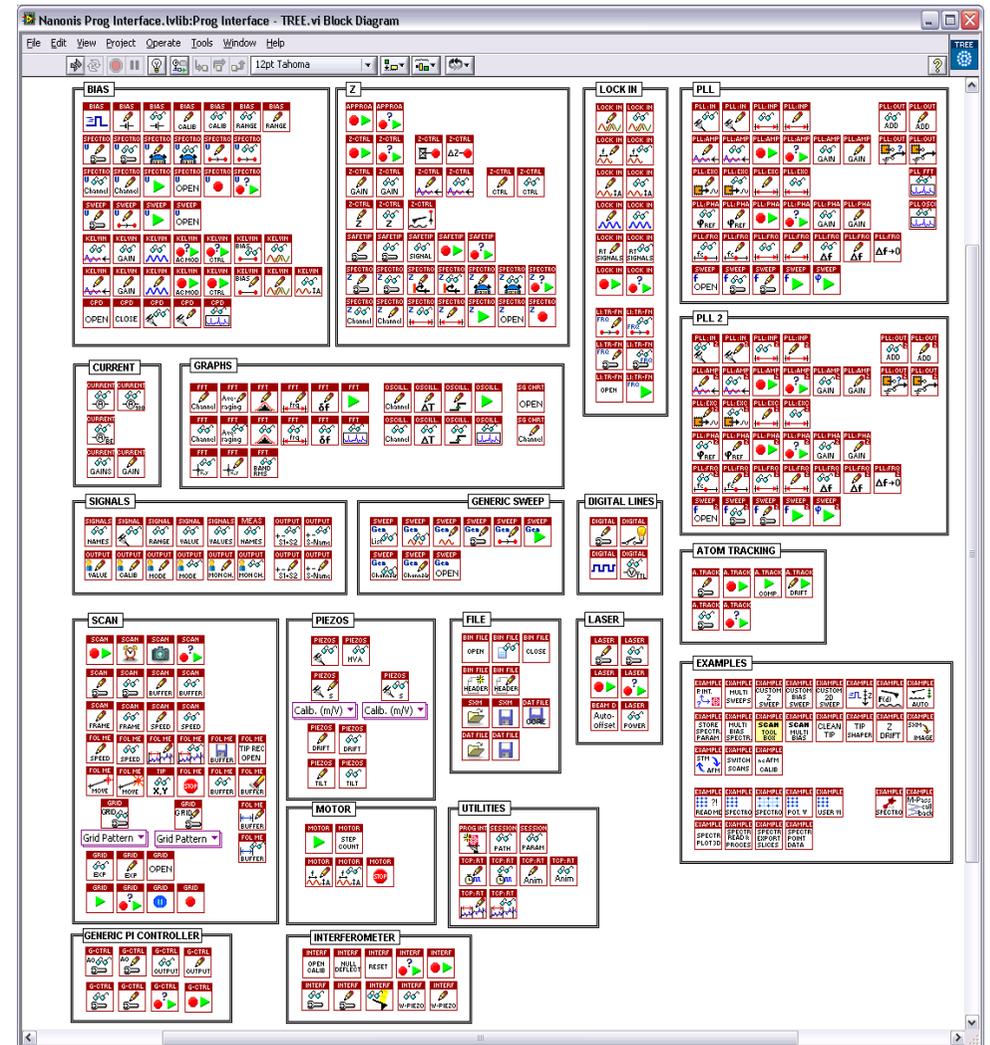


Figure 3: 'Tree' of the programming interface. Every icon represents a sub-VI that can be incorporated in the user's own VIs. Every Nanonis VI communicates with the Nanonis software to perform the requested command.