GXSM - Smart & Customizable SPM Control

Ideal for STM+AFM Control Allowing Complex Data Acquisitions

The open source and community driven software project GXSM takes the next level to provide a highly and in-operando adaptable scanning probe microscopy (SPM) control system. A highly efficient digital signal processing (DSP) interfaces any SPM to a Linux based PC. Via a Gnome based GUI default scanning but also highly advanced mapping and probing/manipulations can be performed. Standalone it can perform image analysis tasks and multidimensional visualization.

Introduction

The invention of the scanning tunnelling microscope in 1982 [1,2] (SPM) triggered today’s tremendous potential of scanning probe microscopy (SPM) to image, spectroscopically analyze, and manipulate atoms and molecules on surfaces at length scales from ~100 μm down to the atomic scale. Many different SPM techniques and instruments such as the STM and the atomic force microscope (AFM) were developed in the last 32 years [3,4]. Especially the potential of local, spin or even time resolved spectroscopic measurements and manipulations on atomic scales is still explored by many research groups. This broad range of local probing and manipulation requires highly customizable instrument controls, which only a few commercial SPM control and software solutions are supporting. Furthermore for most systems the source code is not available as open source. Therefore, the extension of these software packages by their users to special dedicated tasks is either impossible or limited to the capability of a built-in scripting language.

The Gnome X Scanning Microscopy project (GXSM) [5-7] is a modular software suite including everything needed from low level DSP code, system driver to the GXSM core and graphical user interface. Its core is designed for in size unlimited, fast multidimensional/movie data handling and visualization - including 3D (Open GL based). An object oriented (C++), modular and plug-in based key design has proven now for over 15 years to accommodate any future developments. Plug-ins are available for background correction, data analysis, math, feature-detection (via OpenCV lib), statistics, scan data im-/export, Python [14] scripting
remote control, etc.

In particular, an abstracted hardware class allows a hardware plug-in to connect to any specific hardware to be used with the GXSM core for basic SPM functions. While more hardware specific and control tasks are purely remaining on plug-in level.

There is full hardware support for a range of DSP hardware commercially available from an independent company SoftdB [8,9] developing and manufacturing the open source SPM controller based on the MK2-A810 and MK3-A810/PLL boards. Lately, also a high voltage amplifier "Smart Piezo Drive MK3-HV1" was released which is optional but fully integrated with the control system.*

Along with community driven developments and implementation of several unique features, a highly flexible and even in-operando adaptable signal configuration on DSP level, easy custom addition and implementation of special needs via Python scripting makes it a very versatile solution for most researchers needs [23-26].

The GXSM software project [5] is released under the GNU general public license and can be obtained via the internet. GWYDDION [15], a related open source SPM data analysis project, can now handle seamlessly the GXSM NetCDF data files thanks to some collaborations. Also there are a few other SPM related software projects [16-22] to be mentioned, but none include a completely open, free and all GPL platform, building tools and operating system.

**Complete and Open SPM Control System**

A single digital signal processor subsystem [5-9] executes independently from the GUI running on a PC signal processing tasks via a set of SPM specific modules. A robust state machine therefore supervises the different acquisition, processing and output processes in the main kernel located on the DSP. A standard USB-2.0 link to the host is used for control and data streaming. The DSP subsystem is host
independent and continues to operate the SPM safely even if unplugged from the host or while host restart - GXSM will reconnect at any time to take over control.

While scanning basic parameter changes are tracked (event logging) and stored along with image data and additional information using the Unidata NetCDF data format [10]. Image data im-/export in formats listed in the GXSM manual [12] (see Chapter 23, "Plug-Ins: Scan") is possible.

Beyond the basic XY scan generator and Z-servo loop for constant current (STM) or force (AFM) imaging the monolithic control system includes a fully digital Lock-In and a PLL suitable for tuning fork style (commercialized as QPlus or Akiyama probe) sensors [29-31]. The new digital signal scheme allows to assign any of the digital DSP generated signals as data source for imaging. The generic Vector Probe (VP) engine allows in addition manipulation of all output channels so that any kind of spectroscopy or manipulation task can be carried out.

**Signal Processing Modules**

A SPM dedicated set of generic data processing modules is implemented on the DSP to create a versatile user customizable SPM control logic. A default configuration implements a standard STM control. Here a module is a digital data processing block (fig. 3). It usually has signal input(s) and provides one or more signals as output.

**8 Channel Analog Signal Input Stage:**
Allows the configuration of digital differential inputs derived from two individual inputs.

**4 Channel mixer/IIR:**
Filter with one special real time adaptive filter [7].

**Control Values:**
User manipulated values dictated by GXSM for bias and a like.

**Z-Servo:**
The main feedback loop controller.

**M-Servo:**
A auxiliary loop controller. The M- and Z-Servos are providing both polarity output signals. Example: M-Servo configured as a Kelvin controller in conjunction with the PLL and Lock-In.

**Lock-In:**
A two channel digital Lock-In with common internal digital reference but separate
phase settings. It generates over a given number of correlation periods the basic signal average, 1st and 2nd order signals.

**PAC/PLL**: Phase amplitude control and digital PLL frequency detection up to 75 kHz frequency locking. Note: If a higher bandwidth is needed any external PLL can be used with the available 4-channel signal mixer to generate an error signal for the Z-servo.

**Scan, Fast-Scan, Slope and Offset:**
The "Fast-Scan" provides a sine wave X-scan actuation signal.

**Coarse/Wave/Auto-approach:**
Controller for coarse sample/tip positioning with 3-axis step counter.

**VP Engine:**
Real time Vector Probe program interpreter.

**8+2 Channel Signal mixing and HR output stage**
Provides the capability to configure a linear combination of up to three signals with scaling for each of the 8 actual analog outputs. The resulting value is passed through the GXSM HR mode filter (fig 1). Two additional simple special mixing channels are reserved for custom wave signal (Wave-X/Y) mapping, only used for coarse motion signal generation.

The fundamental sampling rate and thus the signal processing frequency of this MK2&3/A810 system is 150 kHz with a total loop delay of 4.5 samples (2 in + 2.5 out, see specifications SR-A810 [8,9]).
Our DSP level real-time VP [7] (fig. 2) interpreter can perform virtually any thinkable spectroscopy and manipulation tasks, such as scanning tunneling spectroscopy (STS), tip formation or feature tracking. A VP vector represents the change of all coordinates (Bias, X, Y, Z, ..., state of feedback (on/off) and a list of data channels to be acquired plus program control parameters for timing, repetitions, etc.) from one to the next point in time. It can be invoked manually starting at the current tip position, on a grid or at every pixel automatically for mapping purposes. It can also do complex signal adjustments on pixel level to acquire any custom mode image data: Data at a given segment of the VP can be mapped to an image channel. The VP program interpreter allows the execution of nested repeat loops. A VP section execution can be delayed until a trigger signal condition occurs. Also a limiter option can be set to watch for a signal min/max conditions to prevent further manipulations. A "atom" tracker option can be programmed to follow a max or min condition with the tip while recording the trace [27,28].

**Hot Pluggable Digital Signals**

Now our new DSP level signal management (fig. 3) gives the user full control over the actual signal to module interconnection using any of the signals available on DSP level. The assignment from module outputs (signal sources) to module inputs is "in-operando" reconfigurable. Just like pulling a plug on a rack and connecting it elsewhere - just it's an ideal swap over with no glitch! No need to recompile or restart any program. Live and visual signal monitoring is available as digital reading, analog meter or scope via python tools. These also support low level configuration of the DSP unit.

A few details on the internal fixed point data representation: All internal signals are 32-bit precision. Native 16-bit AD inputs are represented in a signed "16.16" bit scheme where the lower 16 bits are representing additional precision from oversampling or other math. HR Outputs are 16-bit (native) + 3-bit (HR) "16.3" what gives an effective resolution of 19-bit. The only exception are special mixer output signals which may lead to numbers greater than 16-bit for the upper 16 and thus those are represented in a "24.8" bit scheme. For scaled signal mixing, fractional (Q15) fixed point math is used, i.e., saturated multiply shift and accumulate (SMAC) operations*.e.

An example is given below for a scaled (by AC-Amplitude, Q15) adding of one signal (LockIn-Ref) to the output signal (Bias-Voltage). This operation is generically executed by the output mixing module for every channel (fig. 1):
Bias-Out$_{16.16} = \text{Bias-Voltage}_{16.16} + (\text{LockIn-Ref-Signal}_{16.16} \times \text{AC-Amplitude}_{Q15})/2$

Some critical module internal calculations are executed with 64-bit temporary resolution, examples are the digital Lock-Ins correlation math or coordinate rotation transformations.

**Python Interface**

Using the GXSM build in python [14] console scripting interface any complex task can be programmed. A python script can access any GXSM entry field (giving read/write access to all parameters) via a unique alias name (conveniently shown as tooltip via mouse-over) and any button widget to start a scan or execute a VP for example can be initiated. Several additional functions allow flow control and real time DSP signal monitoring by the script as well (fig. 4). The build in scripting allows the implementation of customized auto approach and coarse positioning methods (beyond the various build in options), monitoring and warning routines for potential tip unsafe conditions, etc. In combination or standalone a python MK3-SPM support library/class is available to do lower level DSP communications including full control over signal configurations.

**Acknowledgements**

The authors thank all volunteers contributing to the GXSM project [5,13] our user community for many ideas and discussions, Stefan Schröder for initial implementation of the python interface and Niv Levy for adding an interactive console. Particular thank goes to Bruno Paillard and Alex Bourdeau from SoftdB [8,9,13] for their outstanding support and the design of the new MK3-A810/PLL analog module and the MK3-HV1. Research carried out in part at the Center for Functional Nanomaterials, Brookhaven National Laboratory, which is supported by the U.S. Department of Energy, Office of Basic Energy Sciences, under Contract No. DE-AC02-98CH10886.

**Technical Notes**

[*a] any kind of SPM or scanning "something", home build or customized commercial instrument, etc.

[*b] Smart Piezo Drive: 3-channel HV Amplifier for SPM typical Piezos, each channel drives +/-200V, 1x/5x/10x/20x gain settings, 1/5/10/50 kHz bandwidth. It integrates with the GXSM control and a digital link provides automatic offset and gain information exchange (http://www.softdb.com/dsp-products-MK3-HV1.php).

[*c] The PAC/PLL function depends on the pacpll-library supplied by SoftdB's DSP
support libraries and drivers and is not open source.

[*d*] HR: High Resolution Output via software based sigma-delta style 1 bit modulation.

[*e*] $Q_{15}$, SMAC: $Q_{15}$ is a number representation where $1.0000 = 2^{15}$. It is used for fixed point DSP multiplication in fractional mode. It implies a post multiplication right shift by 15 bit of a 32-bit result from a two 16-bit number multiplication. See also: www.ti.com/lit/ug/spru281f/spru281f.pdf, page: 6-31.

References

[10] UNIDATA, NetCDF Homepage: http://www.unidata.ucar.edu/
[13] A complete list of all GXSM developers can be found at http://sourceforge.net/projects/gxsm/, click on "view members."
[23] Zaki N. *et al.*: Physical Review B (Condensed Matter and Materials Physics) 87,

Authors

Dr. Percy Zahl (corresponding author)
Brookhaven National Laboratory
Center for Functional Nanomaterials
Upton, NY, USA

Dr. Thorsten Wagner
Johannes Kepler University
Institute of Experimental Physics
Linz, Austria

Contact

Brookhaven National Laboratory
Bldg. 134
Upton, NY 11973
USA
Phone: +1 631 344 2968
Telefax: +1 631 344 3368