The GXSM-4.0 manual

Percy Zahl, Andreas Klust, Stefan Schröder, Thorsten Wagner
and more `http://gxsm.sf.net`

May 14, 2023

Part I.

Preface

Preface to the GXSM-4.0 manual

The GXSM history and very early predecessors are dating back to 1995. It came a long way even lived on different operating systems with initial DSP code fragments in C, Turbo-Pascal and ideas I have to give credit to Gerhard Meyer to get me started with the very very early stages of my experiences with STM control. It evolved and diverted quickly into it's very own system from then on. Other branches related to work of Ullrich Köhler aka PM-STM had some influences and eventually a PM-STM operating on OS/2 was a live.

With that operating system phasing out a move to Linux was undertaken and some software named 'Xxsm' based on the X-Forms GUI was evolving and turned pretty mature by end of 2000 with a big move over to the early Gtk toolkit and GXSM got established. From there a long and steady journey was up on it. With more and more multi-core/threading machine little related pit-falls evolved and were hard to find and even harder to fix – if not even unfixable without major reworking.

But now in 2017 a huge move and undertaking whopped that grown up GXSM2 over into what now is the all new GXSM3 with a rewritten GUI at all levels and logics behind. Lasted big hit is the reimplementation of the OpenGL based 3D/Volume data view to inspect and visualize up to 5D data sets in real-time and dynamically unleashing the full power of a gaming grade GPU. GL-4.0 essential.

For historical reasons the first section of the following preface will remain in German.

Das Jahr 2000 ist vorraussichtlich mein letztes am Institut für Festkörperphysik der Universität Hannover. In diesem Vorwort möchte ich die Historie der ehemals 'nur' STM-Software festhalten. Ich habe die letzten fünf Jahre, die Zeit vor meiner Diplomarbeit eingeschlossen, dort verbracht und in der Arbeitsgruppe von Michael Horn-von-Hoegen begonnen, mich mit der Tunnelmikroskopie und den zugehörigen Techniken zu beschäftigen. Im Sommer 1995 zeigte mir A.Meier ein kleines, noch unberührtes, unter einer Schutzhaube verborgenes Gerät – ein Micro STM. Wir besaßen nur dieses Gerät zusammen mit einem Tunnelstromverstärker, einer Signalprozessor gesteuerte Meßkarte (PC31) – und – dem Wissen, daß damit ein gewisser G.Meyer in Berlin erfolgreich ein Gerät gleichen Prinzips bedient. Ich machte es mir zur Aufgabe, der ganzen Sache Leben einzuhauchen ...

Von G.Meyer hatte ich unterdessen einige Software zusammengesammelt und wir versuchten, die diversen DOS/Pascal Programme auf unserem hochmodernen P90 zum Laufen zu bringen. Währenddessen vertiefte ich mich in die Geheimnisse der DSP-Programmierung und der Kommunikation zwischen Host-PC und DSP.

Des Fortschritts wegen beschloß ich, ein bereits vorhandenes OS/2-Programm namens PMSTM weiter zu entwickeln. PMSTM basierte auf einer schon erheblich älteren OS/2 Version von L.Anderson und wurde von H.Bethge zum Messen eingesetzt.

Gleichzeitig produzierte R.Kumpe einige Assemblerroutinen zur Ansteuerung seiner Datatranslation-Karte unter OS/2 – welche ebenfalls einen spezial STM-Eigenbau, jedoch noch mit analogem Regler, bedienen sollte.

Parallel wurden mit H.Pietsch die notwendigen Zusätze für AFM implementiert.

Es verging einige Zeit, aber dann war es soweit: Die neue OS/2 Software konnte das Gerät steuern und Daten aufnehmen und anzeigen. An Luft wurden erste Versuche mit Goldproben und abgeknipster Wolframspitze erfolgreich abgeschlossen. Weitere Verbesserungen in nahezu jeder Hinsicht konnte ich im Verlauf meiner Diplomarbeit einbringen – das Gerät war unterdessen im UHV der Maschine 'Quantum' im Einsatz.

Die Geschichte des Micro STM´s wurde im weiteren wesentlich von R. Hild bestimmt, der das höchst empfindliche Gerät in einem an Federn aufgehängten Kupferblock mit 3D-Wirbelstromdämpfung versenkte und somit zu Höchtstleistung brachte.

Die Zeit war gegen das wunderbar stabile OS/2, es wurde zum Außenseiter und eigentlich benötigte man es nur noch zum Messen ...

Windows erschien mir als durchaus erfahrenem und gebrantmarktem DOS/Win3.X Programmierer als völlig ungeeinete Platform, so hatte ich doch die Vorzüge einer stabilen Betriebssystemplatform mit OS/2 zu schätzen gelernt. Ich konnte jedoch mit meinen Unix/Linux Grundkenntnissen schnell eine zukunftssichere Alternative finden und entschied, erste Versuchte mit einer neuen Software für mein SPA–LEED zu unternehmen. Das SPA–LEED sollte nämlich ebenfalls mit einer Signalprozessorkarte gesteuert werden, um später ggf. ohne zusätzlichen Aufwandt ein STM nachrüsten zu können.

Das Resultat war ein Programm namens xspa, welches unter Verwendung der xforms Libary unter X11 lief.

Aus diesen Erfahrungen schöpfend entwickelte ich ein völlig neues Konzept für eine grundlegend neue Struktur eines neuen STM-Programmes – der alte Code war zu steif und beinhaltete viel zu viele globale Variablen. Auch das fixe Datenformat dat wurde als historisch abgelegt und es fand ein Übergang zu dem NetCDF-Format statt. Das neue Konzept ist objektorientiert und ermöglicht erstmals eine flexible Mehrkanal-Verwaltung und -Messung (dies wurde in PMSTM nur per Trick in unflexibler Weise zur Datenaufnahme hineigebastelt, da das AFM die gleichzeitige Aufzeichnung von Kraft und Reibungssignal ermöglicht).

Es entstand xxsm, welches strukturell die Grundlage des heutigen GXSM darstellt. Dieses Programm entstand im Verlauf der Diplomarbeit von R. Hild, der die Entwicklung life mit seinem STM verfolgen mußte bzw. durfte :=)

Auch unser neues Spielzeug, ein Luft-AFM, wurde von xxsm gesteuert sowie eine Streulichtapparatur (SARLS).

Die Computertechnologie schreitet unaufhaltsam vorann und es ist zu befürchten, daß bald keine ISA-Slots mehr verfügbar sind. So entschieden wir für neue Geräte, eine PCI-Version der DSP-Karte (PCI32) zu kaufen. Diese Karte schien der PC31, mal von dem PCI-Bus abgesehen, doch sehr vergleichbar. Es gab jedoch einige Unwegsamkeiten, die mich einige Nerven kosteten ...

Jedenfalls entwickelte ich ein Kernelmodul für diese Karte und später eine Variation für die alte PC31, denn auf User-Space IO konnte wegen PCI Konfiguration, etc. nicht mehr

ausgewichen werden. Zusätzlich mußten alle Utilities (Loader, Terminal) an die neue Karte angepasst werden. Eine neue Library machte das anfängliche Chaos komplett – die Hoffnung, das DSP- Programm nicht umschreiben zu müssen, war vergebens. Jedoch konnte mit einigen Tricks weiterhin eine, zwar neue, aber gemeinsame Version erhalten bleiben.

Die alte xforms Libary ist zwar extrem effizient, insbesondere meine sehr schnellen MIT-SHM Bilddarstellungsroutinen, aber die Oberfläche und Menüdarstellung lassen einige Wünsche offen. Eine modernes Toolkit Namens Gtk+/Gnome weckte mein Interesse im Sinne des Fortbestandes und der Weiterentwicklung dieser unterdessen mächtigen Mikroskopie- Software. Ein Kraftakt von diversen Nächten zwischen Juni und Dezember 1999 brachte xxsm im neuen Gewand als gnomified xxsm – kurz GXSM hervor.

Das objektorientierte Konzept von Gtk+/Gnome vereinfachte es auch erheblich, die SPA–LEED-Ansteuerung in GXSM mitaufzunehmen. Darüberhinaus entstanden gleichzeitig einige Tools wie Gfit, Goszi und dsp-applet.

Dieses Werk soll im weiteren all denjenigen helfen, die einerseits mit GXSM arbeiten, aber auch etwas mehr über die Internas erfahren wollen. So gliedert sich das folgende Manual in

- einenaenwendungsbezogenen Teil;
- nützliche Tips (HOWTOs zu STM und AFM); und
- einen Versuch das Programmkonzept zu erläutern.

Ich möchte hier meinen Dank an alle aussprechen, die zu meiner Arbeit beigetragen haben, insbesondere jedoch:

M. Henzler für das immer gute Instituts-Klima und die schönen "'Almen"'.
M. Horn-von-Hoegen dafür, daß er mir die Zeit zum ständigen Arbeiten an diesem Projekt gelassen hat.
G. Meyer für seine Starthilfe bei der DSP Programmierung und diversen Gesprächen.
H. Bethge für Ihre Geduld mit mir im Keller.
R. Kumpe für seine Mitarbeit an PMSTM und Diskussionen.
H. Pietsch für sein Mitwirken an PMSTM.
L. Anderson, den ich leider nie persönlich kennengelernt habe, für seine Arbeit an PMSTM.
U. Köhler und seine Truppe für einige Diskussionen.
A. Meier für seine endlose Geduld mit mir ...
F.J. Meier zu Heringdorf für immer wieder freundlich und fröhliche BS Discussions mit diversen Guinness.
R. Hild und M. Bierkandt für deren Ausdauer mit den ewig neuen Versionen.
A. Klust für alle Beiträge zu diesem Projekt.
H. Goldbach für SPA–LEED Discussions und für den (noch nicht) herausgesuchten BF krams.
Heilo für die Gewährung einer Mehraufwandzulage.

Negenborn Januar 2000 Percy Zahl

Preface to the GXSM-4.0 manual

... GXSM has been licensed as GPL and goes to SourceForge.net ...

Es ist Sonntag, der 21.1.2001, es schneit draußen und die Zeit ist mal wieder ein Jahr fortgeschritten; Ich stecke mitten in meinen Vorbeitungen zum Ortswechsel von Negenborn bei Hannover nach Denver/Colorado und möchte ein paar Bemerkungen zum Stand des GXSM-Projekts festhalten, nachdem meine Dissertation mit der Veröffentlichung meines Werkes zum Stress auf Oberflächen abgeschlossen ist.

Das GXSM-Projekt ist seit Herbst 2000 offiziell als Open-Source auf `sourceforge.net` via Web-Interface und CVS-Access verfügbar. Die Verzeichnishirarchie wurde überarbeitet und 'GXSM' ist nun das Projekthauptverzeichnis.

Im Dezember 2000 wurde ein flexibles Plug-In Interface für GXSM entworfen und damit begonnen alle speziellen Erweiterungen in Plug-Ins auszulagern. Damit wird eine erhebliche Flexibilität erzielt. Im Janaur 2001 sind ein SPA–LEED Simulationsmodul, Peak-Finder, Fokus-Tool und Oszi-Plugin hinzugekommen. Ein Support für die alte Burr-Brown Karte für SPA-LEED ist ebenfalls verfügbar und bereits im Einsatz.

Negenborn Januar 2001 Percy Zahl

... GXSM goes international ...

Just a few up-to-date remarks:

As far as I know GXSM is used now in more than four countries around the globe.

Kernel 2.4.x support was implemented and is proofed to work well.

More and more Plug-Ins are added ...

A Gnome Druid to guide the new GXSM user along all most important settings was added.

Golden, Colorado USA, August 2001 Percy Zahl

Preface to the GXSM-4.0 manual

... about one year later and the GXSM user community is expanding worldwide – some success?

A lot of new features were added and a so called 'multi-layer' capability was implemented, especially for use with 2D probing. The modularization via Plug-Ins is driven further, so the scanning and probing control was released from GXSM core and turned into Plug-Ins. There is still a lot to do, but it is already producing valuable 2D STS data! Together with this development the probing modes are expanded and a experimental digital Lock-In was implemented on a combined DSP and host level.

In today ongoing process the DSP software undergoes a complete redesign. While the DSP hardware future is still not clear (some great options are in sight) – future DSP platform setups are very expensive due to hardware and development tools as well, and as long there is no funding there can't be a new implementation for this non profit project.

And today we can announce the beginning of the new composed GXSM Manual. It is partly automatic generated from Plug-In source files. Lots of thanks to Andreas for initiating this development!

Golden, Colorado USA, June 16, 2002 Percy Zahl

... GXSM2.0 is coming and it also starts support for a new DSP platform ...

It's Sunday, my official part of my two year Postdoc job at Colorado School of Mines in Golden has just passed and I'm working on my GXSM project – it's snowing outside for the first time since about three month of no perception at all ...

Two important milestones in the GXSM history are just in progress:

Gnome2 is now available and features several good changes and redesigns but also some non-compatible issues to be taking into account. This has now led to the new GXSM2.0 CVS branch and results into a new 'gxsm2'. The port is completed, including over 70 PlugIns. Some minor issues are to be treated but a functional alpha version of GXSM2 is available.

The old PCI32,PC31 DSP cards are hopelessly out of date and not any longer easily available. And just in time the 'SignalRanger' DSP occurred, it's a via USB connected standalone DSP board. So the Linux support for SRanger-USB and some new DSP tools were developed and are available via there own SF project (`http://sranger.sf.net`). The 3rd DSP soft generation for the fixed point SRanger board DSP (TMS320C5402) is currently in development, in parallel the GXSM2 SRanger support is created – No results yet, but it all looks promising.

Golden, Colorado USA, February 2, 2003 Percy Zahl

. . . GXSM2 and Signal Ranger in daily data production!

It's a rainy Sunday, and the GXSM manual is going to be revised to the GXSM2 version. The Signal Ranger documentation part is to be started . . . but the Signal Ranger boards (SR-STD and SR-SP2) are now both supported – thanks to SoftDB for the friendly loan of the STD board!

Also I'm happy to say, the SR does a really good job in our lab as second/spare and testing DSP subsystem. The analog performance (noise level) is outstanding: Using a huge scanning tube (XY: 1000Å/V, Z: 200Å/V) it's possible to resolve the Au-herringbone and Buckyballs on Au(111)!

Adliswil, Switzerland, October 5th, 2003 Percy Zahl

... GXSM2 V1.8.4 and Signal Ranger going Multidimensional and Vector Probing

Before all that, the SRanger kernel module for Linux Kernel 2.6.12 and higher was stabilized. The completion of the Vector Probe implementation just happened. The design of the low level (DSP) Vector Probe was finished for over an year but the streaming of the data was only limited working for some time, also a really user friendly GUI was pending. That all changed now. The invention of the GXSM Event mechanism finalized the new capabilities, including the raster probe mode. Also the approach and coarse motion control was polished up and now features an arbitrary wave form mode for best possible results for any inertial motion driven positioning and maximized the flexibility. And for all the future, the whole GXSM core undergo an extensive code cleanup which now totally removed all remainings of hardware close parameters. These are now banned into the corresponding HwI plugins.

Rocky Point, Long Island, NY, USA, November 16th, 2005 Percy Zahl

Preface to the GXSM-4.0 manual

. . . GXSM2 V1.12.0 is released with full multidimensional data processing and visualization power

Several new add-ons were added, in brief:

- more pre defined Vector-Probe modes
- a SignalRanger/CoolRunner CPLD based and hardware or CPLD times (gated) 32bit counter channel to acquire pulse counts/rates of any source
- a event logging mechanism to attach important parameter changes and probe data to the eventually running scan
- full core 4-dim data support and visualization now enabled
- transition of several math plugins to run on multi layers and time dimension on demand
- marker object to manually count things of different flavor (color)

But the major achievement is that the GXSM core now allows handling of 2-dimensional images sets. In particular, a single scan channel can now hold stacks of images (layers) for multiple times making a true 4-dimensional data set. Data can be played like a movie in layer (at a specific time) or in time dimension (at a specific layer). Visualization of profiles (or series of profiles) can be navigated in real time with the 'Show Point' or 'Show Line' tool in any dimension, also image-slices in any dimension can be generated on the fly.

Export of movies is possible using the Quicktime library. The new OSD (On Scan Display) allows to overlay real-time informations, like time any other parameters.

Rocky Point, Long Island, NY, USA, September 18th, 2007 Percy Zahl

... GXSM2 V1.22.0 releases and all new SPM dedicated SR-A810

The year 2009 is already going to its end, a nice sunny colorful fall day... This year has brought big changes or better upgrades on DSP level and in particular on the side of analog signal conversions. The new Signal Ranger Mark 2 (MK2) was already around for a while in 2008 and GXSM supported with a test HwI and ported DSP code the new platform, which brought us USB-2.0 and more DSP speed and memory. But it had a drawback, even a newer revision of the AICs, it was a slow bottle neck, as it had un acceptable long loop delays. The need for some thing dedicated was even more pressing now. All started out in early summer 2008, discussions of the GXSM team and affiliated leading GXSM users and institutions (in particular myself and Rolf Möller from Department of Physics, University of Duisburg-Essen, Germany) with SoftdB and their hardware engineer B. Paillard were paving the way to the now available Analog-810 interface for the MK2 (MK2-A810). Final designs and decisions for the DA/AD converter were made about following the STM conference in Keystone, Colorado.

Just a few days before Christmas 2008 I received the first prototype for the MK2-A810 assembly – big thanks to SoftdB! As the MK2 was already fully supported by GXSM and the driver infrastructure for the A810 was seamless to the previous, getting it going was done in a snap! This new thing was holding to it's great specs greatly and things were moving along very well. By January the first working HwI release was going into CVS and very soon was also field/laboratory STM proven to work and preform very well. In all aspects it is superior, precision, stability and speed – 75kHz feed back loop with full loop delay less than 5 samples. Sampling rates at up to 150kHz possible (and used in latest versions). A little later two optional counter channels were added the FPGA logic.

The new MK2-A810 is available since February 2009 and now also 19" rack ready on a GXSM/SPM customized enclosure.

All basics done, the new power of the A810 was getting explored in depth and a couple of new features and software based performance improvements were implemented, let me just list:

- 2 channel 32-bit counter timer, syncronized with sampling
- Evaluation of software based resolution enhancements, Z and X/Y HR-mode
- Real time magnitude dependent bandwith adjusting via FIR for current input
- Transition to Float in GXSM in conjunction with transfer of full statistical data (32-bit resolution summed values from DSP + normalization count)
- FIFO data transfer optimization: using custom byte packing, first order linear predictor...
- Real time configurable 4-channel feedback input mixer, lin/log/fuzzy modes
- Enhanced and expanded Vector-Probe modes and visualization modes (GUI)
- Many more details...

And: A update on GXSM and the new MK2-A810 is submitted/accepted to Journal of Vacuum Science and Technology B (JVST B), proceedings of the NC-AFM, Yale 2009.

Brookhaven National Laboratory - CFN, Upton, Long Island, NY, USA, October 29th, 2009 Percy Zahl

... from Version Numbers to 'Battenkill Warrior', 'Lancaster Classics', 'Lindau Historic' to GXSM2 V1.27.3 'Arosa Express' – get prepared for the high-end MK3Pro-A810/PLL and new GXSM optimized SPM HV-Amp what is now also available: the 'Smart Piezo Drive' with serious DSP power and featured under the hood not yet seen!

Not functional but just getting a little more 'social' – dedication names to major mile stone version changes – so you can chat better about versions!

'Battenkill Warrior' is introducing a separated Z-Offset signal for scan slope compensation on analog level. Also a new so far little used/tested feature tracking VP mode to follow in real time 'gradient up' or '-down'. May also name it atom-tracker.

Full 3D Offset control and Linear Drift Correction via automated Offset adjusting. Also added helping aids to easy determine drift manually from manually to be identified features in scan(s) via 'Global Position Reference Mark' and Time/Drift calculation for Point markers. (New options: `Scan/View/Coordinates/(Time + Relative)` If previously a Global Reference was set via any Point/Marker Object/Global Reference Point.

Massive selections of hundreds of VectorProbe data files for example from or live while raster probing via easy DnD read back is not supported and very useful for life data inspection of long mapping runs.

Always on going: new additions to the universal GXSM VP modes/tabs, including a dual folder with user arrangeable tabs for better work flow and overview. Example: segmented STS.

'Lancaster Classics' is not providing much visible additions but introduces now a higher precision of the DSP level integer math moving to 32- and temporary 64bit for vector scan signal generations. Also further enhancements of the HR signal output mode (native 16bit to near 20 (some limitations apply) on software level) and other optimizations are included. Additions and new indicators for the PanView to incorporate GPIO and some DPS statemachine status indications.

'Lindau Historic' – incorporating a set of GXSM community ideas discussed and collected at the 2011 NC-AFM in Lindau. Most visible, the new a red-profile history. Some patches needed for newer Gkt+/Gnome/X11 releases. Few more options for VP-Z. New VP feature allows to program limits/triggers to stop a VP section, for example when a certain max force is reached. Also the Mk3-A810/PLL activities are evolving and a PLL prototype is getting available in early 2011! The PLL is only suitable and dedicated for tuning fork systems with up to 75kHz detection – it is all software based and developed by SoftdB. This means for the future is hardware speeds are moving up we will be able follow up the bandwidth! DSP code porting in under the way and this takes more hurdles than anticipated – however with some support by SoftdB it's getting finally to a working version what still is a little beta (by end of 2011). Also new on the communities demand: A GXSM optimized Piezo / HV Amplifier 'Smart Piezo Drive' what includes a separate DSP for several control and monitoring tasks – it works all standalone and fully analog, but can be hooked up via USB to a control computer for watching signals and configuring all kind of features like gains and bandwidth – but as a novelty it also can perform linear drift corrections by itself and accepts digitally offset settings.

It's now 2012 and GXSM 'Arosa Express' is out – get ready for more remote and freely

via Python programmable SPM actions! Now VP probing and Mover/Autoapproach can by fully controlled via GXSM-Python-Remote (via the 'emb' interface). Here is a little sneak peak script:

```
import emb as gxsm

gxsm.set ("DSP_Bias","0.1")
gxsm.set ("DSP_SCurrent","1.5")
gxsm.set ("OffsetX","0")
gxsm.set ("OffsetY","0")

gxsm.action ("DSP_CMD_AUTOAPP")

for x in range (0,1000,100):
  gxsm.set ("OffsetX","%f"%x)
  gxsm.sleep (10)
  gxsm.set ("DSP_IV-Start-End","-1")
  gxsm.set ("DSP_IV-End","1")
  gxsm.action ("DSP_VP_IV_EXECUTE")
  gxsm.sleep (10)
```

PS: New Motto: 'GXSM is a glove to your SPM experiment' – this said, we claim you have free hands to do pretty much all you like in a instant real time feedback experience – but be warned, this also means you can dig your tip into the surface if you want to also if you do not take care as there is pretty much no way to distinguish unless introduction mostly annoying safety 'blocks'.

Brookhaven National Laboratory - CFN, Upton, Long Island, NY, USA,
March 6th, 2012 Percy Zahl

Signal Revolution: 'Snowy Janus Hack' + 'Snowy Janus Signal Warrior' with Mark3-Pro-A810-PLL

'And soon I realized the power of the new dimension I added!'

An evolving idea leading out of an dilemma unleashing not yet seen power and configurability – a digital patch-rack with a lot of transparency and insights. Double power for users and developers – live 'signal' or variable debugging and monitoring.

Why I am doing this? How? What is it and where will it go and what can I do with it? Let me try to explain: A need for even more real time flexibility and the need to access even more data channels/signals – to manage a huge expansion of the 'signal space'. Tis is no more possible and not good to handle efficiently with a fixed signal to channel assignment due to a limited number of total channel I can handle for several reasons. However, there are more than enough channels, just a growing variety of different 'data' or 'signals' beyond the raw analog inputs and you will never need all of them the same time. So that lead me to a new approach I already started initially to get out of the dilemma I found myself getting into and soon I realized the power of the new dimension I added!

The 'PAC/PLL' signals needed to be accessible... that we have now for a while (MK3) via the configuration of a signal source via a pull down menu for now each Feedback Mixer Input 0..3.

And again those can be remapped (MIX0..3) – still names (PAC1..3 in the scan channel source selector for 2d scans). I need to clean up the signal naming conventions – part of the over haul. That's the harder part on GUI level. Plan executed: Each signal got a unique name, a value range and real world unit associated.

You are still with me? I hope ...

Signals are simply variables and connections are made via just pointing the input to the source – as simple as this – 'signals' will be 'hot pluggable'.

The current DSP code in state machine design will remain unchanged in its proven design form, only so far fixed signal paths will be 'broken up' and made open to be hot reroutable with more connectivity options, more data sources exposed to GXSM via existing channels and at other locations as needed. The existing data processing blocks starting with ADC inputs, PAC-siganls, counters, control values (like Bias)..., Scan, Offset Move, Probe, LockIn, ... and finally a HR Output block with new options... I will call from now on modules. And data/variables resulting from module data processing I will name signals. The default power-up signal linkage will be pretty much the default as known from the past. A kind of netlist will be exposed to GXSM and configuration tools to be manipulated hot. So we have module input 'nodes' and module output 'signals' made available for netting up!

Let me break up the existing design in existing modules as sketched: Outputs will be more configurable and not any more hard assigned – just a default! They will get a input stage with the signal input and two additional adding inputs (modulation, etc.) with optional gain.

To handle all this a new cleaned up python basic based support class and a configurator script with monitoring goodies and a signal graph visualization tool is now available. Also

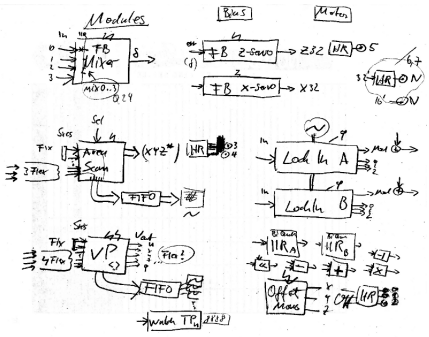

Figure 1.: Idea Sketch: Modules and signals.

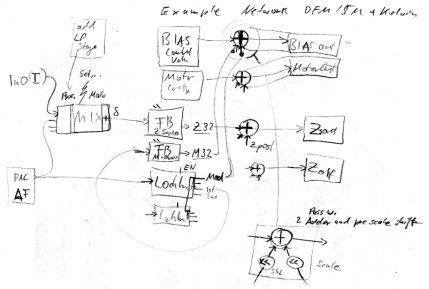

Figure 2.: Idea Sketch: Example SPM configuration.

a signal configuration management with to flash store and auto restore there is.

On GXSM level you will feel home pretty much right away – only several sources are now not fixed any more but allow to choose form multiple signals. Also signals are in classes so not always all need to be exposed to prevent a 'GUI over load'.

That said – enjoy the new next generation and a few handy and eye candy monitoring galvos, a signal scope and a new tuning tool with peak auto-fit.

Well, read about the actual outcome in the SRanger Mk2/3 HwI plugin.

GXSM Central, Rocky Point, Long Island, NY, USA,
March 29th, 2014 Percy Zahl

Got High Speed? Real Time Engine 4 GXSM3-3.50: 'Next Level RTime Engine' with Mark3-Pro-A810-PLL and RedPitaya

'It was long rocky way! It is way too long since my last update here. But finally a e- or re-volution.'

We got a all new fully high definition capable GTK3 compliant GUI! That was some act.

Beside many, many new features, GXSM is now all into nc-AFM and provided dedicated tools for molecular imaging.

Alongside a in many aspects enhanced GXSM python console with GXSM utility libraries and include functionality and a growing python support library.

The MK3 convergence detector aka PAC-PLL is great, but suffers on bandwidth and statistics. So after a major act of learning and FPGA hacking the first ever 125MHz dual PAC-PLL and all bells and whistles including super fast amplitude controller, Q-control option and ultra wide range PLL operating with 48 bit phase/freq. control was born.

After proof of principle some unexpected efforts needed to put in to create a reliable, fast and real time digital data link to the DSP – featuring now a McBSP serial link operating over a potentially longer set of two CAT5 cables (about 2m in use right now). This triggered some interrupt collision issues serving the McBSP on the DSP side fast and on time. With great help and insights from Alex at SoftdB the problem was tracked and a solution sketched up. May be for the good, a major revision of the historically more monolytic data processing scheme was on the table. A single hint... and I ended up creating my very own micro kernel and real time machine on a 1/150kHz time scale. Evaluating and optimizing tasks, moving every bit of code not fully real time critical into new idle tasks. Also new a automatic RT task scheduling and enable/disable control – all DSP based – still provides a seamless and 100% backwards compatible DSP code from the 'outside' point of view.

Oh no.... I need to rewrite the DSP under the hood section now.

But for the good, things just got better and faster!

GXSM Central, Upton, Long Island, NY, USA,
April 19th, 2019 Percy Zahl

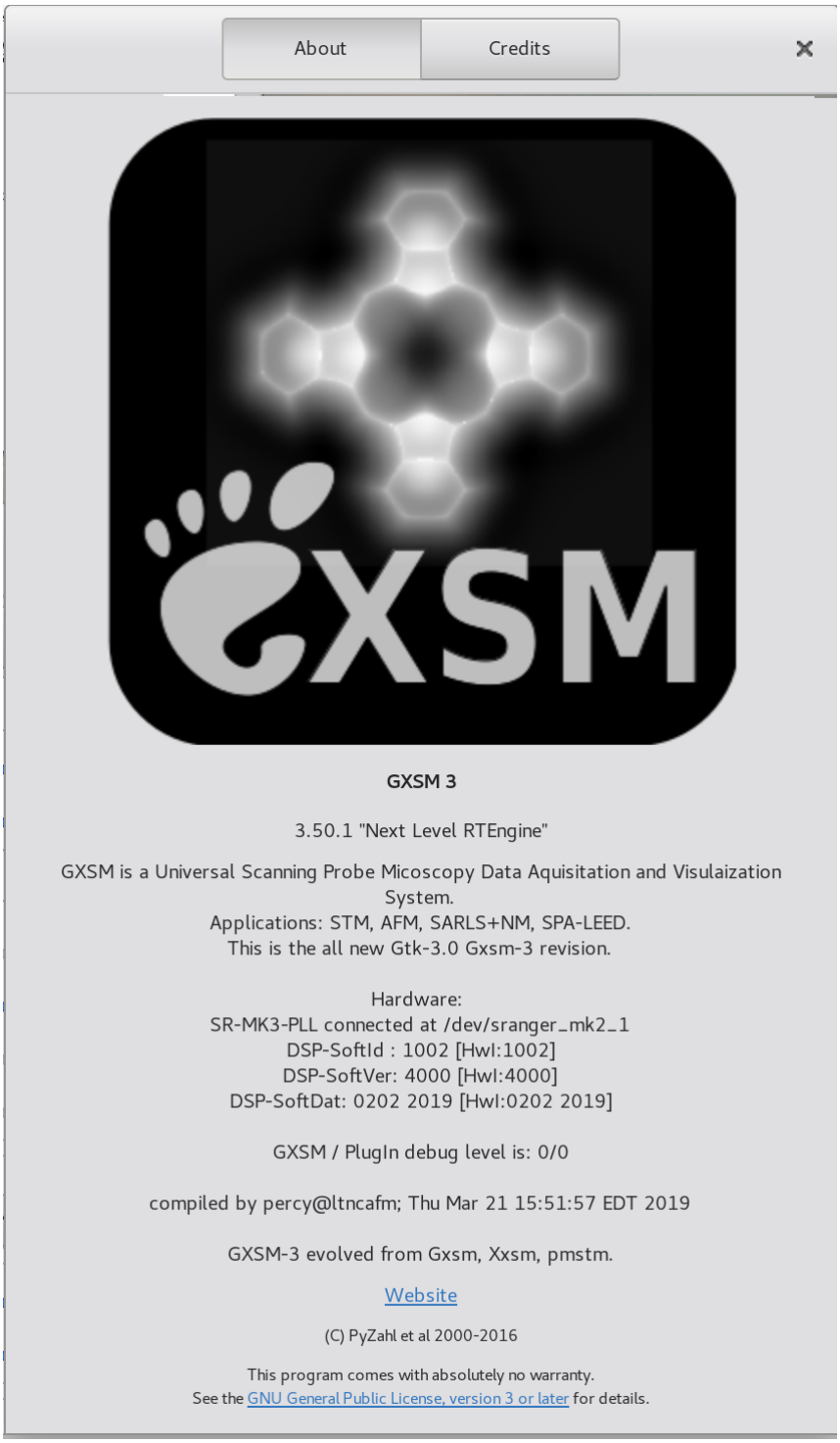

Figure 3.: GXSM3 3.50.1 about screen.

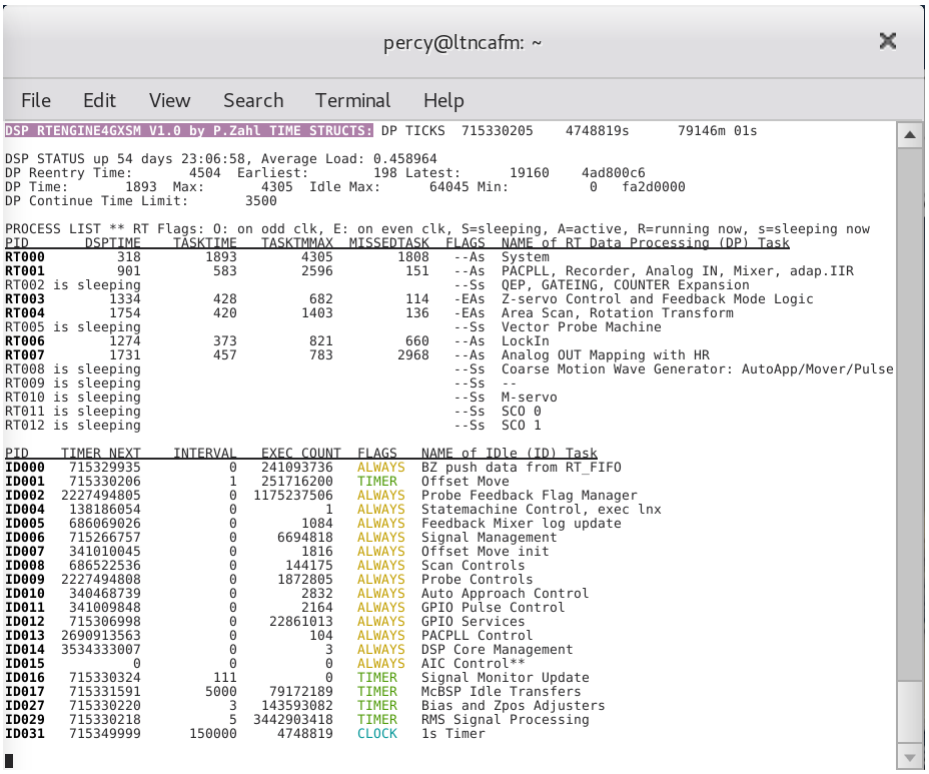

Figure 4.: Real Time Engine 4 GXSM3 – a RT process view.

Part II.

How to

1. HowTo: Install GXSM-4.0 from source code

1.1. System requirements

GXSM-4.0 needs a reasonably up-to-date machine running a recent version of almost any Linux variant. GXSM-4.0 is developed on a Debian Testing. Some description to install GXSM-4.0 on a clean Debian bullseye are found in this forum posting: `https://sourceforge.net/p/gxsm/discussion/297458/thread/3f0faafe/` It also runs on Ubuntu 22.04 LTS for which dedicated binary packages are available.

System memory requirements are ranging from little to several gigabytes if you want to deal with big scans, movies or multidimensional data sets.

To compile GXSM-4.0 on a Debian Testing install these devel packages:

```
$ apt-get install libgnomeui-dev intltool \
yelp-tools gtk-doc-tools gnome-common \
libgail-3-dev libnetcdf-dev \
libnetcdf-cxx-legacy-dev libfftw3-dev \
libgtk-4-0 libgtksourceview-5.0-dev \
gsettings-desktop-schemas-dev \
python-gobject-2-dev libgtksourceviewmm-5.0-dev \
libquicktime-dev libglew-dev freeglut3-dev \
libgl1-mesa-dev libopencv-core-dev \
libopencv-features2d-dev libopencv-highgui-dev \
libopencv-objdetect-dev libnlopt-dev libglm-dev \
fonts-freefont-ttf meson ninja
```

The '\' will connect the command lines. You can also skip it an write everything in one long command line.

1.1.1. Wayland, NVIDIA, openGL4...

The 3D visualization requires an openGL4 support based on a NVIDIA GPU (and the nouveau driver). GXSM-4.0 will run without 3D visualization. If 3D is selected without in the channelselector, possible a lot of warnings will pop up. This will prevent GXSM-4.0 from crashing but the warning dialog windows make GXSM-4.0 effectively unresponsive.

To run GXSM4 with Wayland as window manager, you have two alternative to tweak your linux: i) In Ubuntu 22.04 Wayland is the default window manager if you are not

using an nvidia gpu. To deactivate Wayland support, please add/enalbe as root in /etc/gdm3/custom.conf the line.

```
WaylandEnable=false
```

ii) Alternatively, deactivate the splash screen during GXSM4 startup. Open the dconf-editor and navigate to org/gnome/gxsm4. Here change the entry "splash" to off.

1.2. Source code

Recently, the source code from `https://sf.net` to `https://github.com`. To obtain a copy of the source code, please run in a terminal:

```
$ cd ~
$ git clone https://github.com/pyzahl/Gxsm4 gxsm4-git
```

The repository will be cloned into /gxsm4-git.

1.2.1. Meson

Since GXSM-4.0 the meson build system (instead of make) is used. After download the source code (into /gxsm4-git) change into the respective directory from a terminal and run the following commands:

```
$ meson builddir
$ cd builddir
$ meson compile
$ meson install
```

First command creates the 'builddir' in the project's root folder. Here, '$meson compile, meson install' calls ninja. You can simply call 'ninja install' to do it all. You may have to run the last line with a preceding 'sudo' to gain the rights to copy files in system directories.

To uninstall call in the 'buildir'

```
$ ninja uninstall
```

2. HowTo: Install GXSM-4.0 via flatpak

2.1. System requirements

Flatpak is a distribution independent way to install linux based software. To use it, install flatpak and flathub

```
$ sudo apt install flatpak -y
$ sudo flatpak remote-add --if-not-exists flathub \
https://flathub.org/repo/flathub.flatpakrepo
```

on your system. Here, we assume an Ubuntu derivate.

Install the gnome SDK within your flatpak environment:

```
$ flatpak install flathub org.gnome.Sdk//41 org.gnome.Platform//41
```

GXSM-4.0was also tested with newer SDKs up to 43.

2.2. Source code

Recently, the source code from `https://sf.net` to `https://github.com`. To obtain a copy of the source code, please run in a terminal:

```
$ cd ~
$ git clone https://github.com/pyzahl/Gxsm4 gxsm4-git
```

The repository will be cloned into /gxsm4-git.

2.3. Compilation and installation

Now install and run GXSM-4.0(assuming that the json-file/source is in the folder /gxsm4-git

```
$ mkdir flatpak_builddir
$ flatpak-builder --user --install --force-clean flatpak_builddir \
gxsm4-git/org.gnome.gxsm4.json
```

2.4. Starting GXSM-4.0

To start GXSM-4.0 open a terminal and type

```
$ flatpak run org.gnome.Gxsm4
```

3. HowTo: Make GXSM-4.0 Ubuntu packages

3.1. System requirements

Make sure that you have all required packages installed. After downloading the source you might have a look at the respective control-file stores in the subfolder debian:

```
$ apt-get install libgnomeui-dev intltool \
yelp-tools gtk-doc-tools gnome-common \
libgail-3-dev libnetcdf-dev \
libnetcdf-cxx-legacy-dev libfftw3-dev \
libgtk-4-0 libgtksourceview-5.0-dev \
gsettings-desktop-schemas-dev \
python-gobject-2-dev libgtksourceviewmm-5.0-dev \
libquicktime-dev libglew-dev freeglut3-dev \
libgl1-mesa-dev libopencv-core-dev \
libopencv-features2d-dev libopencv-highgui-dev \
libopencv-objdetect-dev libnlopt-dev libglm-dev \
fonts-freefont-ttf meson ninja \
debhelper dh-autoreconf dh-make devscripts
```

The '\' will connect the command lines. You can also skip it an write everything in one long command line.

3.2. Source code

Recently, the source code from `https://sf.net` to `https://github.com`. To obtain a copy of the source code, please run in a terminal:

```
$ cd ~
$ git clone https://github.com/pyzahl/Gxsm4 gxsm4-git
```

The repository will be cloned into /gxsm4-git.

3.2.1. Packing

The source code also contains all files to make your own deb-package. To do so, checkout the source from `https://github.com` as described above. Then enter the folder /gxsm4-

git and the subfolder debian (assuming that you checked out the source code in gxsm4-git) and rename control_<your ubuntu> to control.

Then you can use

```
$ dch -i
$ dch -r
```

to update the changelog file of the package. In particular, you might want to increase the version number. Then use

```
$ debuild -b -I -uc -us
```

to make your (unsigned) package. To make a signed package use

```
$ debuild -b -I -k<your email>
```

The last command will require the variables DEBEMAIL and DEBFULLNAME to be set, i.e., you may want to add to your .bashrc

```
DEBEMAIL="<your email>"
DEBFULLNAME="<your full name>"
export DEBEMAIL DEBFULLNAME
```

3.3. Installation

Package built this way can be installed via

```
$ sudo dpkg -i <package-name>
```

3.4. Starting GXSM-4.0

You can start GXSM-4.0via the terminal by using 'gxsm4' or via the activity panel.

3.5. Ubuntu binaries

An easy alternative to compiling the source code is to obtain binaries from `https://launchpad.net`. Right now, the repository just hosts binaries of GXSM3 for Ubuntu (up to 22.04).

3.5.1. Adding the package repository

First, you have to add the PPA (Personal Package Archive) hosted on `https://launchpad.net` and refresh the local database:

```
$ apt-add-repositry ppa:totto/gxsm
$ apt update
```

Please, follow the instructions in the terminal.

3.5.2. Package installation

You can install GXSM-4.0 via

```
$ apt install gxsm4
```

This command will not only install GXSM-4.0 but also makes sure that the required dependencies are installed.

For SRanger MK2 or MK3 based hardware support, do not forget to install the respective kernel modules via

```
$ apt install sranger-modules-std-dkms sranger-modules-mk23-dkms
```

Once you have installed GXSM-4.0 you will be noticed by your package manager (in the lastest Ubuntu version this is called Ubuntu Software Center) about updates.

NOTE: In any case, please do not mix up an installation based on the source code and one via APT. The installation via meson usually will use the folder '/usr/local/', whereas apt will put GXSM-4.0 into the folder '/usr/'. The flatpak installation is done in a kind of sandbox so that it does not interfere with other ways of installation.

Part III.

Core

Part IV.

Appendix

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

Terms and Conditions

0. Definitions.

   "This License" refers to version 3 of the GNU General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

   "The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

   To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

   A "covered work" means either the unmodified Program or a work based on the Program.

   To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

   To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

   An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for

the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

   The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

   The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

   The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

   All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source

may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a

further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

   You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this

License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the

patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the

covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

    Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

    THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE

OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see `https://www.gnu.org/licenses/`.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read `https://www.gnu.org/licenses/why-not-lgpl.html`.

References

[1] Percy Zahl et al. "The flexible and modern open source scanning probe microscopy software package GXSM". In: *Review of Scientific Instruments* 74 (2003), pp. 1222–1227. DOI: 10.1063/1.1540718.

[2] Percy Zahl et al. "Open source scanning probe microscopy control software package GXSM". In: *J. Vac. Sci. Technol. B.* Vol. 28. 2010, C4E39–C4E47. DOI: 10.1116/1.3374719.

[3] Percy Zahl and Thorsten Wagner. "GXSM - Smart & Customizable SPM Control". In: *Imaging & Microscopy* 17 (2015), pp. 38–41.