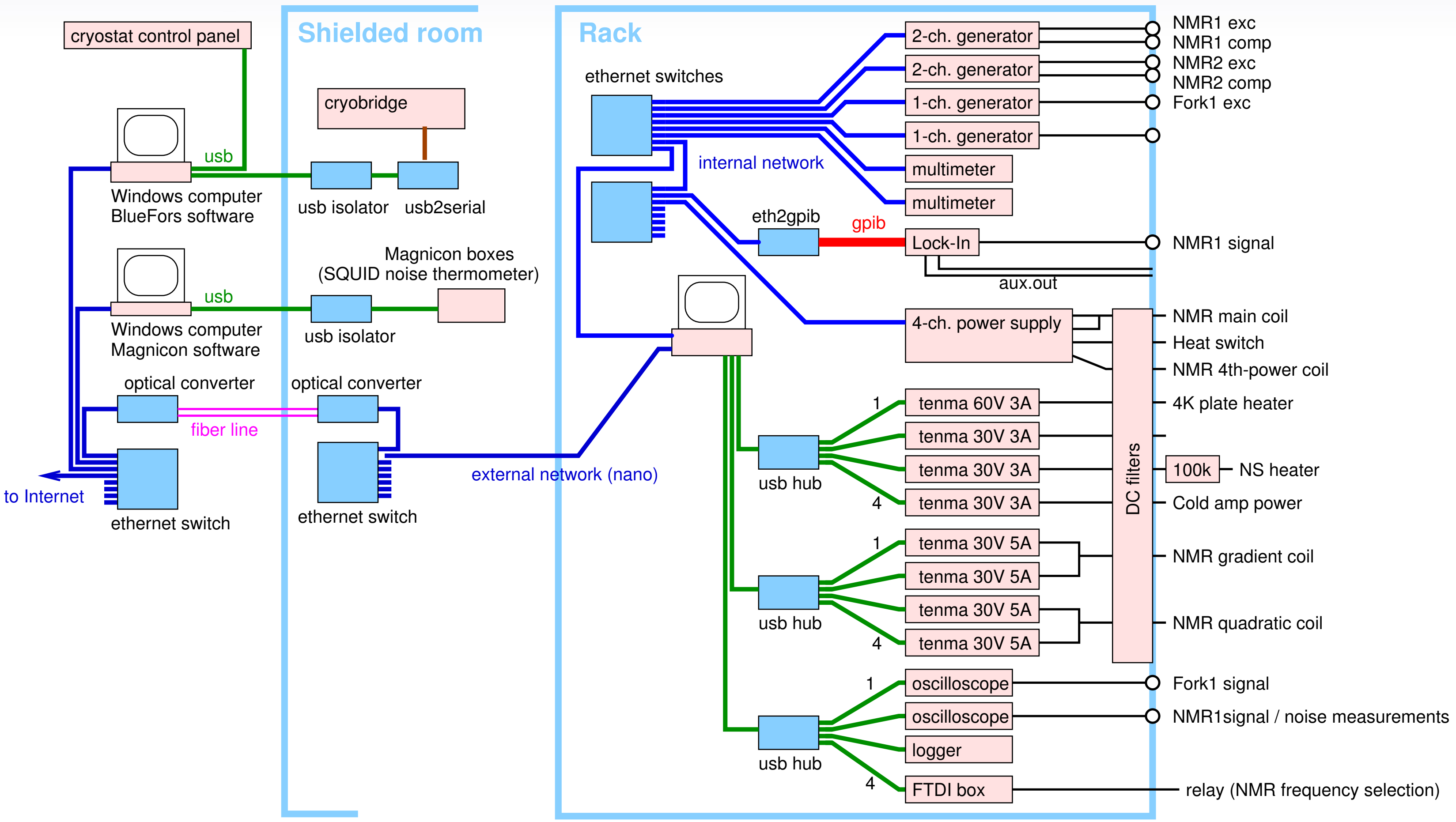
Software system for ^{3}He NMR experiments

Vladislav Zavjalov

14.3.2018

Hardware



cryostat control panel

usb

Windows computer
BlueFors software

usb

Windows computer
Magnicon software

optical converter

fiber line

to Internet

ethernet switch

Shielded room

cryobridge

usb isolator

usb2serial

Magnicon boxes
(SQUID noise thermometer)

usb isolator

optical converter

ethernet switch

external network (nano)

Rack

ethernet switches

internal network

eth2gpib

gpib

2-ch. generator

2-ch. generator

1-ch. generator

1-ch. generator

multimeter

multimeter

Lock-In

aux.out

4-ch. power supply

usb hub

1

4

tenma 60V 3A

tenma 30V 3A

tenma 30V 3A

tenma 30V 3A

usb hub

1

4

tenma 30V 5A

tenma 30V 5A

tenma 30V 5A

tenma 30V 5A

usb hub

1

4

oscilloscope

oscilloscope

logger

FTDI box

DC filters

NMR1 exc
NMR1 comp
NMR2 exc
NMR2 comp
Fork1 exc

NMR1 signal

NMR main coil
Heat switch
NMR 4th-power coil
4K plate heater

100k
NS heater

Cold amp power

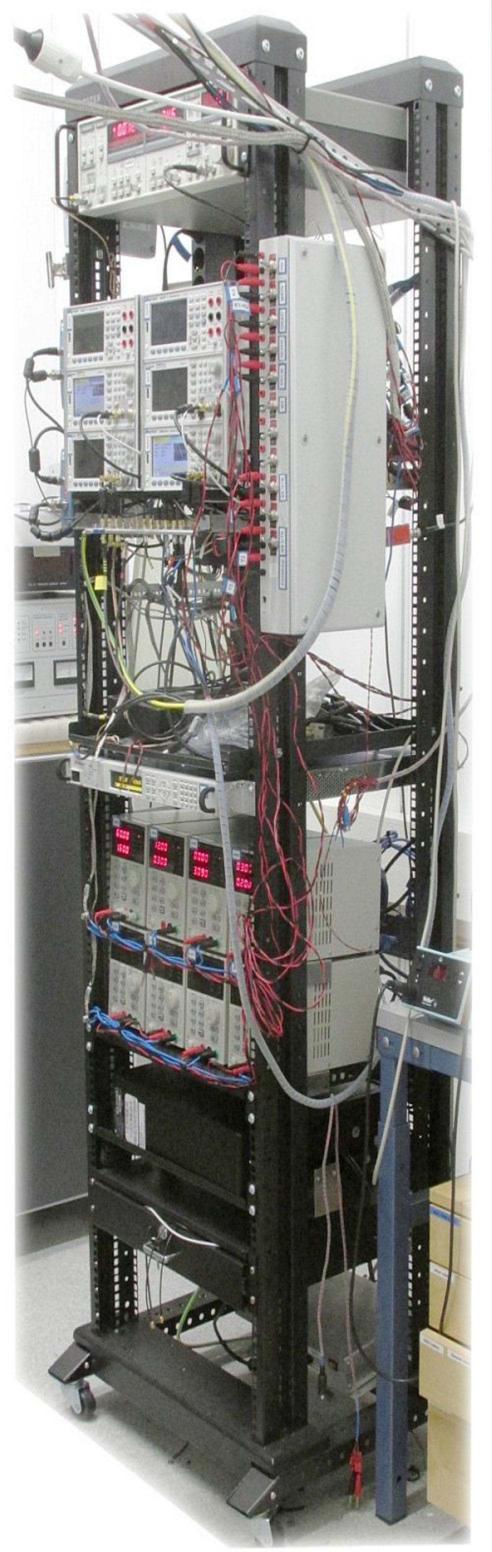
NMR gradient coil

NMR quadratic coil

Fork1 signal

NMR1signal / noise measurements

relay (NMR frequency selection)



Device library

https://github.com/slazav/tcl-device

TCL language:

- easy to make graphical interfaces
- used in ROTA (some programs can be used)

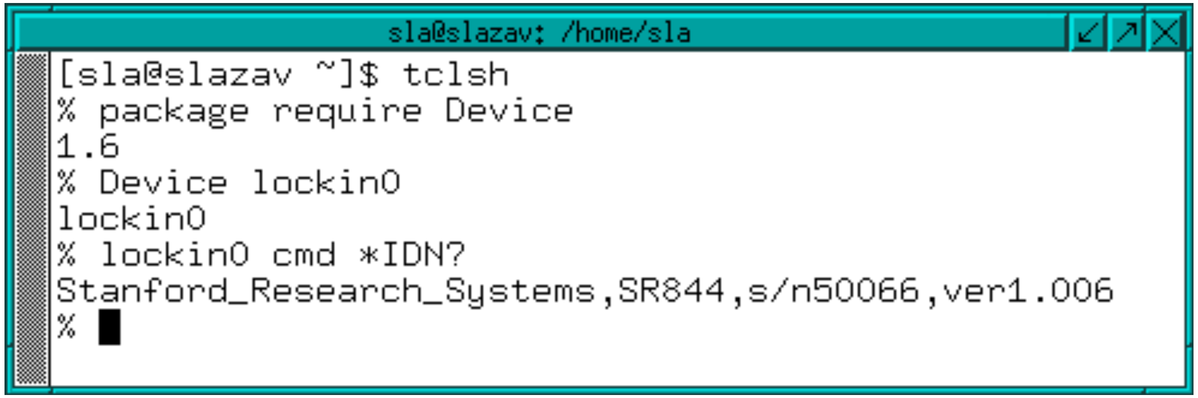
-good for interaction between programs

Main idea: programs do not care about how devices are connected.

Program can just open a device, send a command and get an answer.

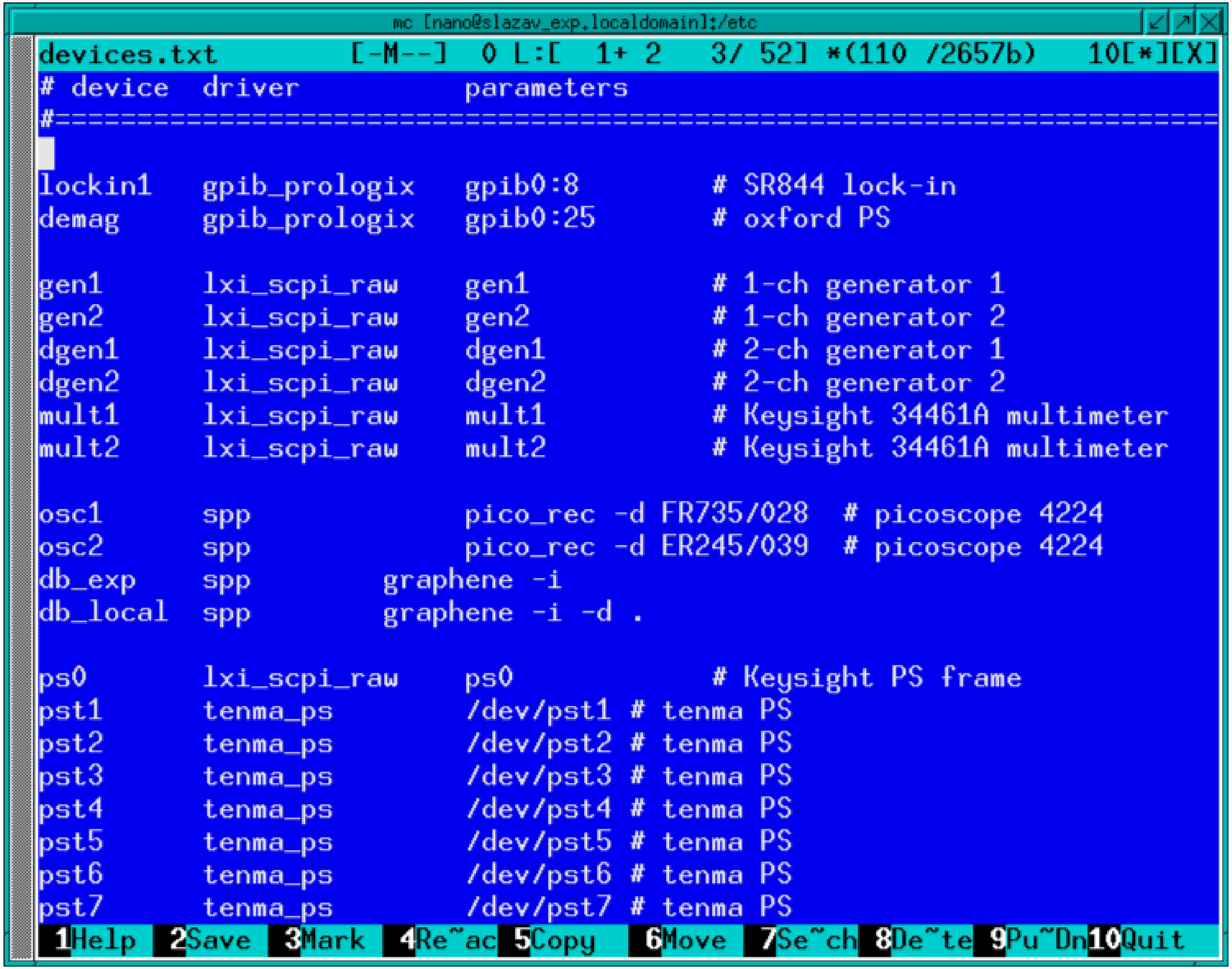
Other features:

- error handling
- IO locks
- user locks
- timeouts
- logging



```
[sla@slazav ~]$ tclsh
% package require Device
1.6
% Device lockin0
lockin0
% lockin0 cmd *IDN?
Stanford_Research_Systems,SR844,s/n50066,ver1.006
% █
```

Device library – configuration



```
devices.txt        [-M--]  0 L:[  1+ 2   3/ 52] *(110 /2657b)   10[*][X]
# device  driver          parameters
#=======================================================================

lockin1   gpib_prologix   gpib0:8        # SR844 lock-in
demag     gpib_prologix   gpib0:25       # oxford PS

gen1      lxi_scpi_raw    gen1           # 1-ch generator 1
gen2      lxi_scpi_raw    gen2           # 1-ch generator 2
dgen1     lxi_scpi_raw    dgen1          # 2-ch generator 1
dgen2     lxi_scpi_raw    dgen2          # 2-ch generator 2
mult1     lxi_scpi_raw    mult1          # Keysight 34461A multimeter
mult2     lxi_scpi_raw    mult2          # Keysight 34461A multimeter

osc1      spp             pico_rec -d FR735/028  # picoscope 4224
osc2      spp             pico_rec -d ER245/039  # picoscope 4224
db_exp    spp        graphene -i
db_local  spp        graphene -i -d .

ps0       lxi_scpi_raw    ps0            # Keysight PS frame
pst1      tenma_ps        /dev/pst1 # tenma PS
pst2      tenma_ps        /dev/pst2 # tenma PS
pst3      tenma_ps        /dev/pst3 # tenma PS
pst4      tenma_ps        /dev/pst4 # tenma PS
pst5      tenma_ps        /dev/pst5 # tenma PS
pst6      tenma_ps        /dev/pst6 # tenma PS
pst7      tenma_ps        /dev/pst7 # tenma PS
```

Graphene database

https://github.com/slazav/graphene

Main idea: you can put a few numbers or text with a timestamp into a database. Then you can extract data for any time range

Features:

- based on BerkleyDB
- integer, floating point or text values
- nanosecond-precision timestamps
- multi-column numerical values
- fast access to data, interpolation, downsampling
- command line interface
- http interface for web-applications (Grafana viewer)

DeviceRole library

https://github.com/slazav/tcl-device_role

Main idea: program can use a device in some simple role, without a knowladge about its model and command set.

Program can just open a device "as a voltage source", and run "set voltage" method.

Existing roles and supported devices:

power_supply – a power supply with constant current and constant voltage modes

* Keysight N6700B frame with N6762A or N6762A modules
* Korad/Velleman/Tenma 72-2535, 72-2540, 72-2550 power supplies

dc_source – a simple DC voltage source

* Korad/Velleman/Tenma power supplies
* SR844 lock-in (auxilary outputs)
* Keysight generators (1 and 2 channels)

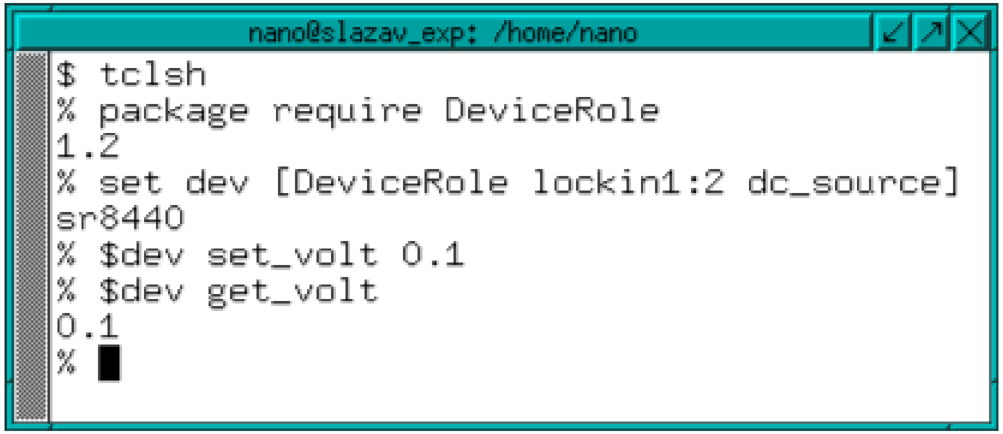
ac_source, noise_source

* Keysight generators (1 and 2 channels)

gauge – a gauge device

* SR844 lock-in
* Keysight multimeters

q



```
$ tclsh
% package require DeviceRole
1.2
% set dev [DeviceRole lockin1:2 dc_source]
sr8440
% $dev set_volt 0.1
% $dev get_volt
0.1
% 
```