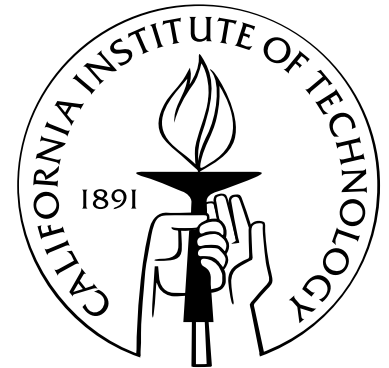


Mezurit 2: Virtual instrumentation for electronics experiments

Dr. Brian Standley

FOSDEM

2 Feb 2013



UNIVERSITY OF CALIFORNIA
UCRIVERSIDE

Origin of Mezurit 2

Problem #1: Acquire, scale, and record data

Problem #2: Sweep a region of parameter space

Problem #3: Find and trigger on rare events

Common hardware:

- Semiconductor parameter analyzer (expensive, inflexible)
- Computer with DAQ and/or GPIB cards

Common software:

- Lab-specific LabVIEW/Matlab/IDL
- Mezurit "1" by Marc Bockrath and David Cobden
- meaSureit by Vera Sazonova

Key features

Virtual channels:

- Arbitrary (Python) functions of hardware ports (and GPIB)
- *Invertible* functions can be outputs

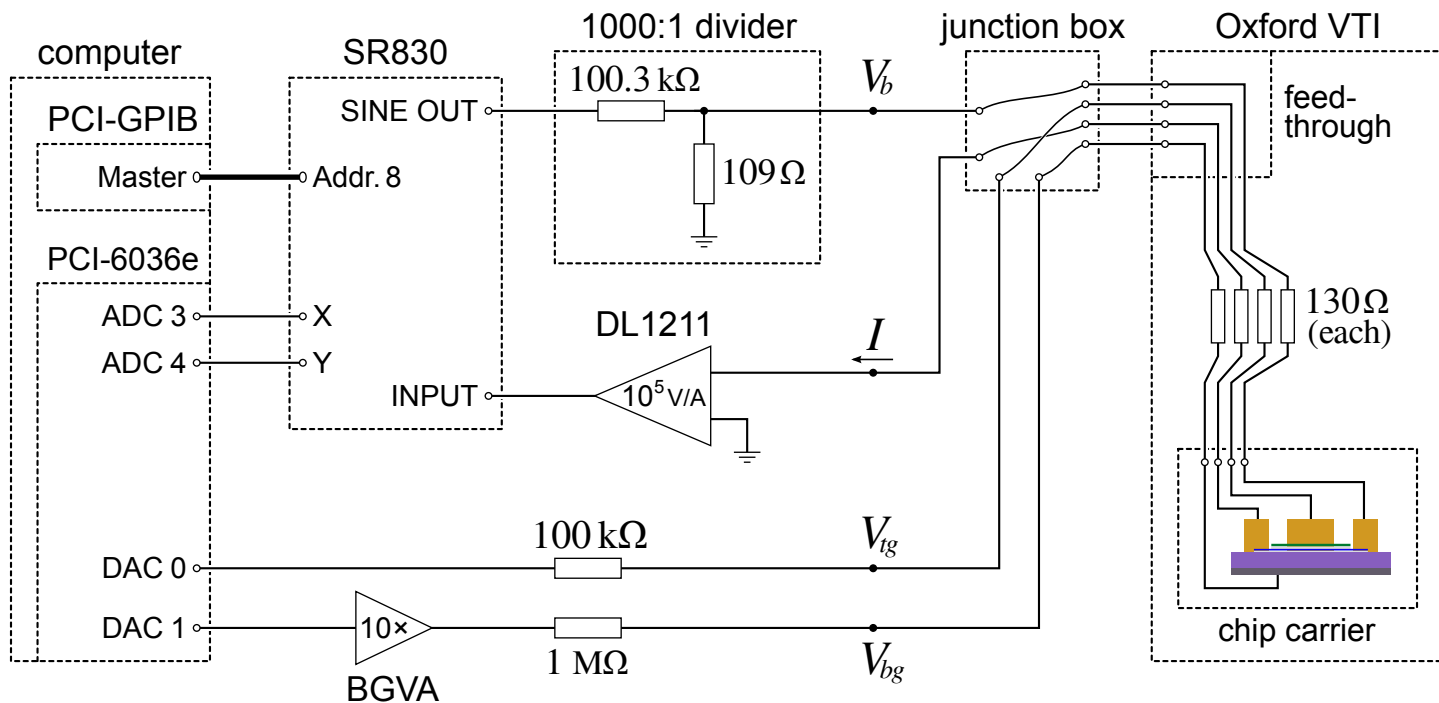
Virtual instruments:

- Acquisition – Data logging up to ~5 kHz
- Scope – Asynchronous acquisition up to hardware limits (~1 MHz)
- Sweeps – Linear or non-linear output ramps
- Triggers – Event detection with predefined responses (also ~5 kHz)

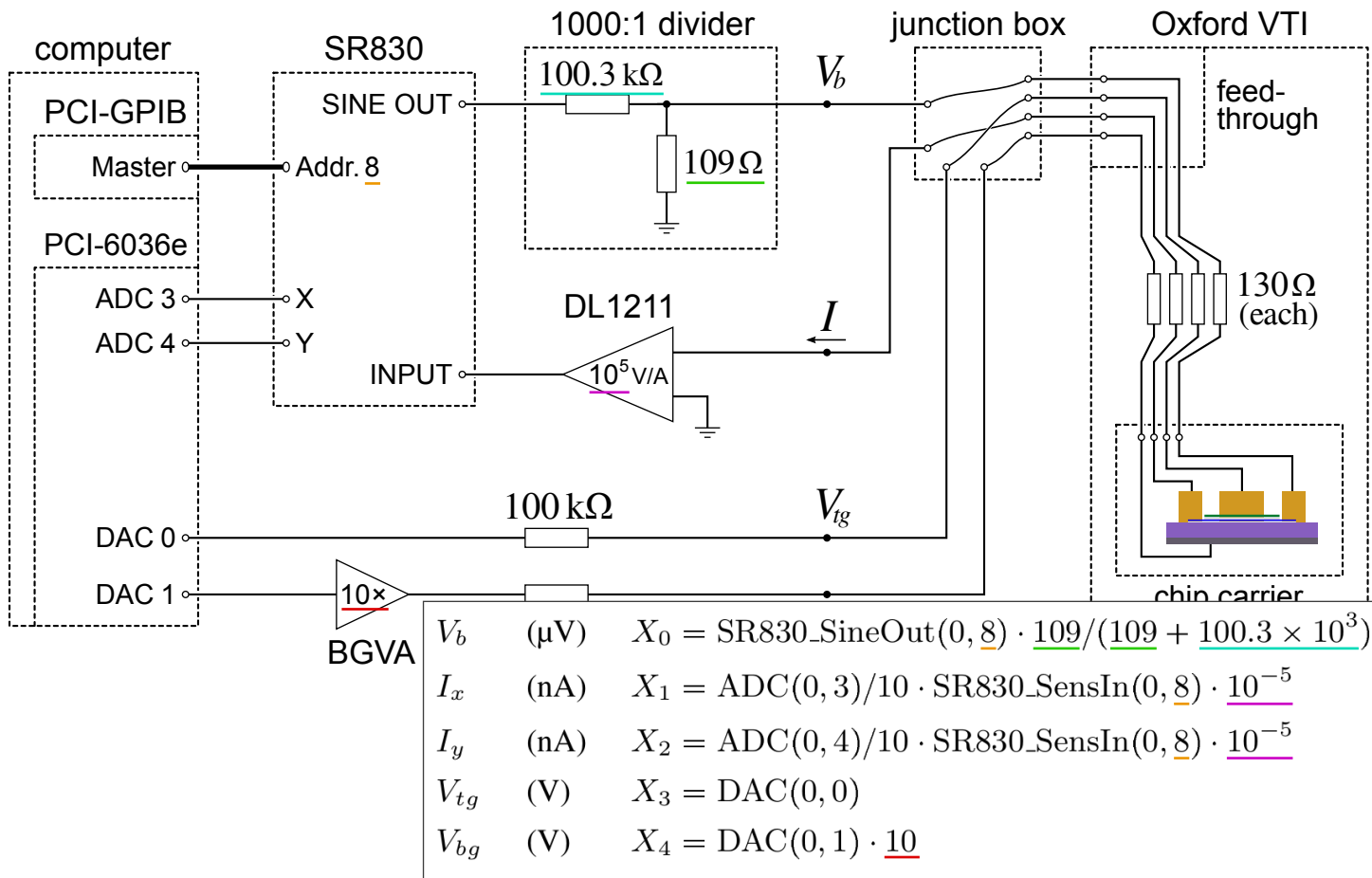
Command-line terminal

Continuously-updated plot and data readout

Example experiment



Example experiment



A BRIEF DEMO

Implementation

Code (v0.91):

Language	files	blank	comment	code
C	67	2456	1314	9591
C/C++ Header	37	513	747	1263
Python	3	134	40	333

(Linux) libraries:

- COMEDI
- Linux-GPIB
- C Python API
- Python interpreter
- GTK+ 2
- VTE

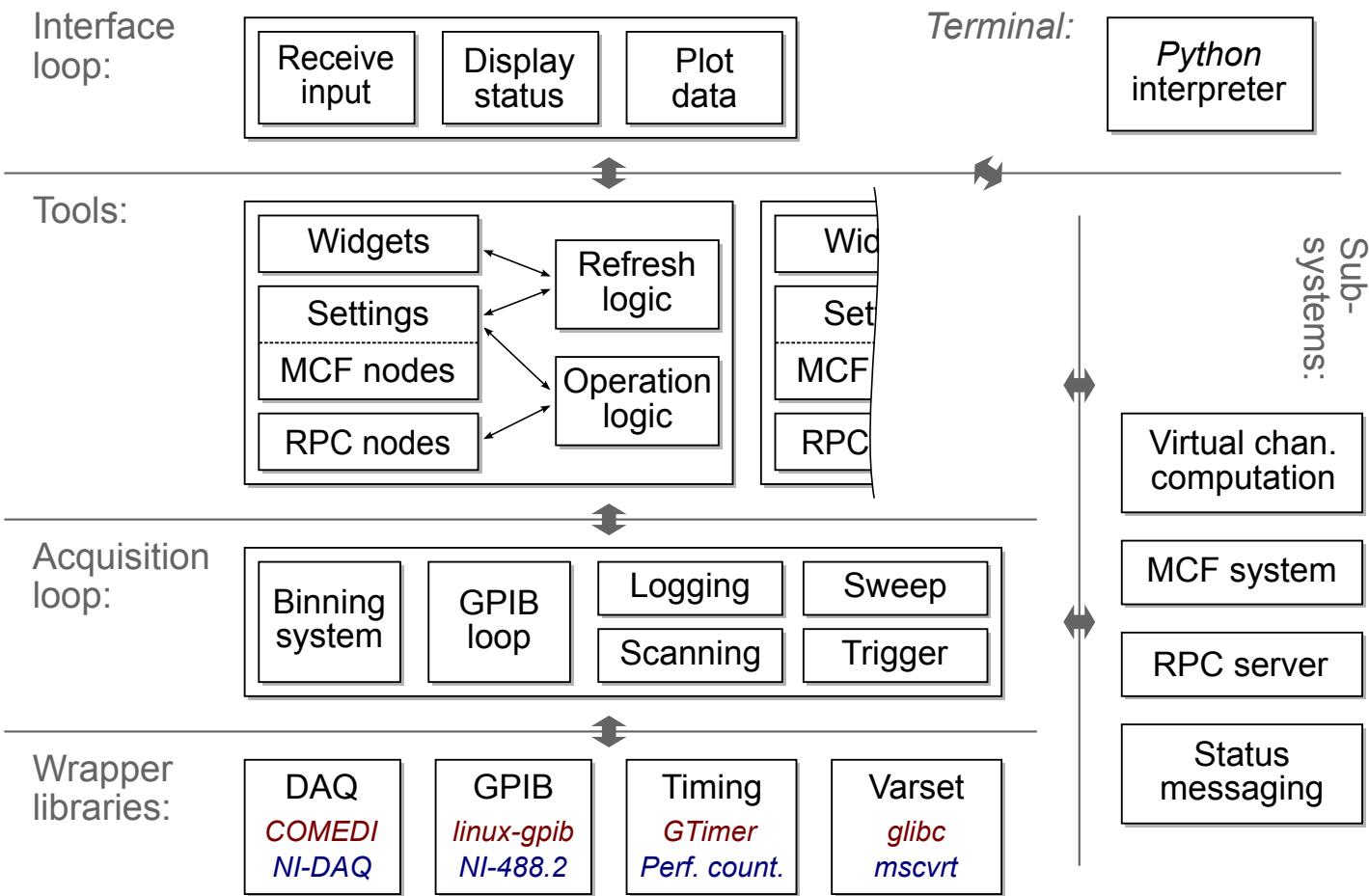
Platforms:

- GNU/Linux
- Windows XP/7 (via MinGW)

License: GPL3

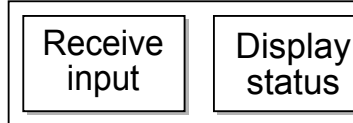
www.ugcs.caltech.edu/~mezurit2/

Architecture

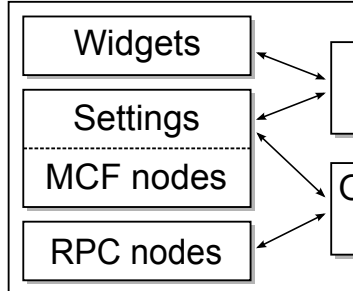


Architecture

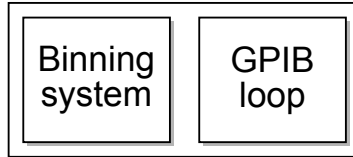
Interface loop:



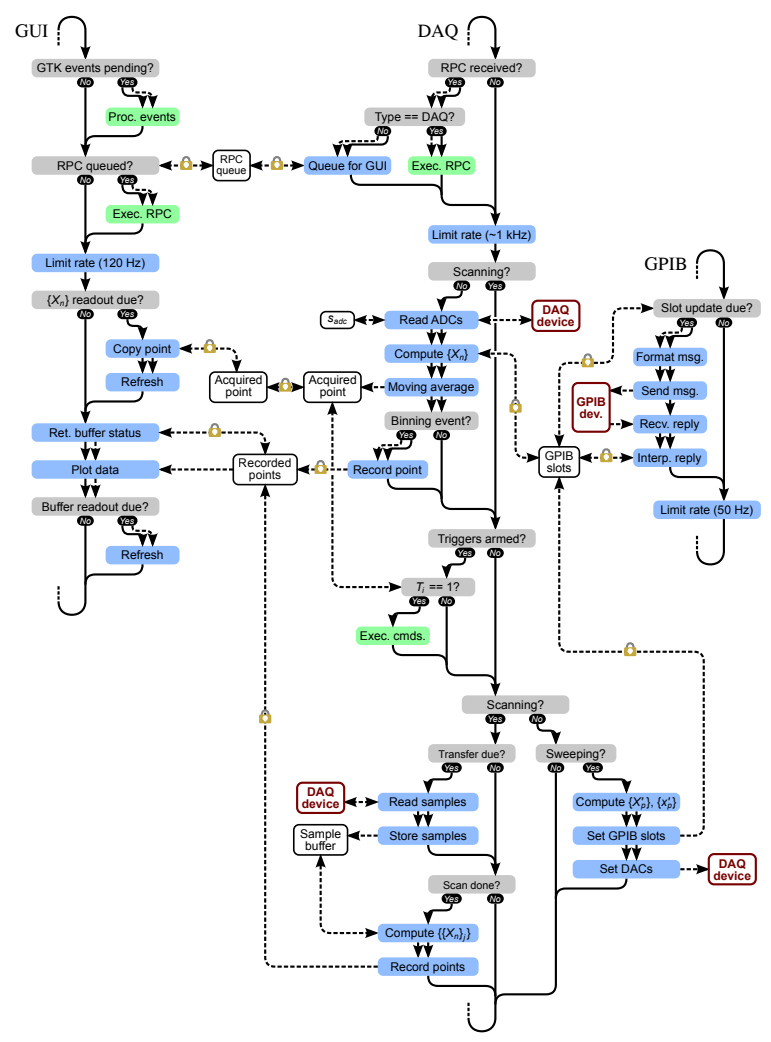
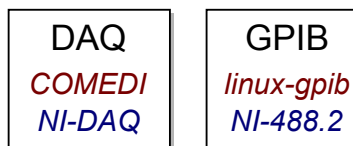
Tools:



Acquisition loop:



Wrapper libraries:



Configuration system

Text-based config files —
each line maps to an "MCF node":

- String identifier
- Data type
- Pointer to the setting variable
- Callback

*Config is scriptable through the
"set_var" terminal function.*

-
-
-
-

GUI thread:

```

rpc_closure(setvar_rpc,
            'set_var')
mcf_closure(setvar_mcf,
            'sweep_rate',
            rate_var,
            rate_widget)

while (running) {

    if (rpc_queued()) {
        cmd, arg = get_msg()
        func = rpc_lookup(cmd)
        rpc_call(func, arg)
    }

}

def setvar_rpc(arg) {
    key, value = parse(arg)
    func = mcf_lookup(key)
    mcf_call(func, value)
}

def setvar_mcf(var, widget,
              new_value) {
    lock(mutex)
    var = new_value
    unlock(mutex)
    set_value(widget,
              new_value)
}

```

Terminal:

```
M2> set_var('sweep_rate=1.2')
```

RPC Closures	
'get_var'	getvar_rpc()
'set_var'	setvar_rpc()
'save_data'	savedata_rpc()

MCF Closures	
'sweep_delta'	setvar_mcf() + var = delta_var + widget = delta_widget
'sweep_rate'	setvar_mcf() + var = rate_var + widget = rate_widget
'hold_time'	setvar_mcf() + var = hold_var + widget = hold_widget

Widgets	
ΔX_1	0.0 V
r_{sweep}	1.2 V/s
t_{hold}	0.05 s

Variables	
delta_var	0.0E+0
rate_var	1.2E+0
hold_var	5.0E-2

DAQ thread:

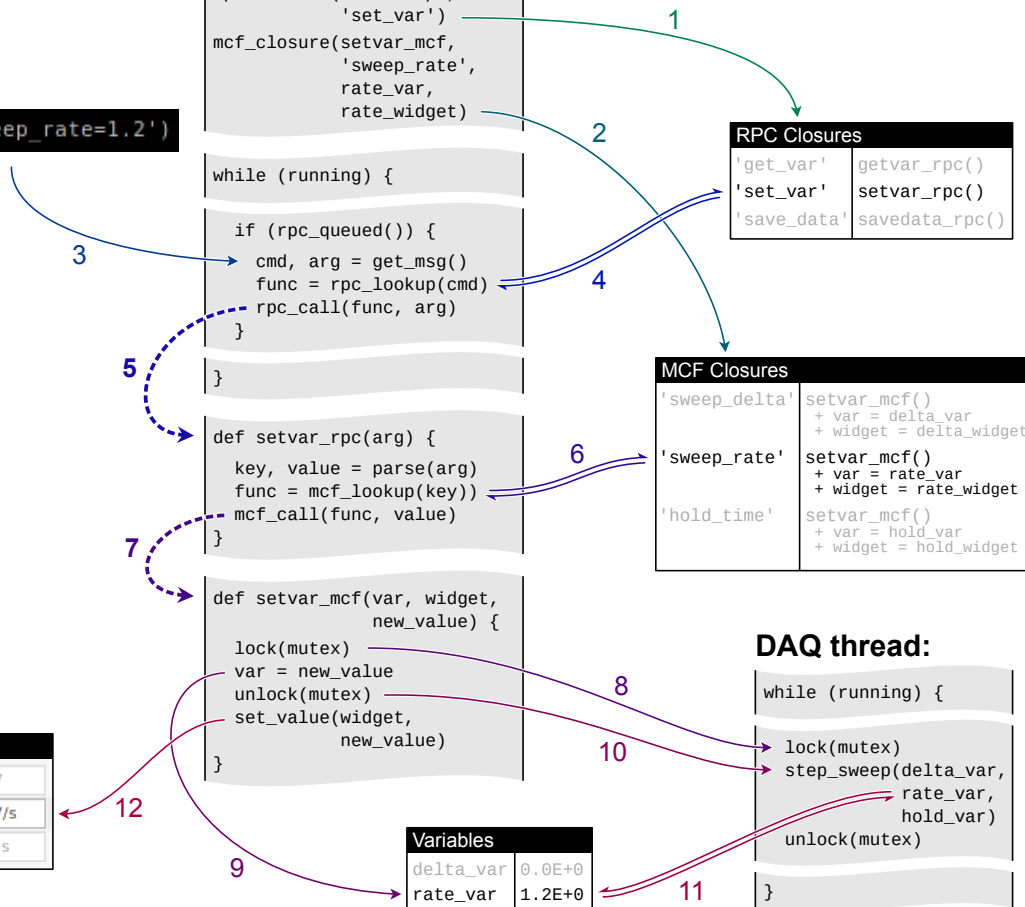
```

while (running) {

    lock(mutex)
    step_sweep(delta_var,
              rate_var,
              hold_var)
    unlock(mutex)

}

```



TODO

Ongoing work:

- GTK+ 3 support
- Python 3 support
- Realtime operation (PREEMPT_RT, if possible)
- Incomplete features here and there
- Bug fixes

Future work:

- More users
- >1 developer?

Alternative frameworks

Current alternatives:

- LabVIEW
- Matlab/Simulink
- Scilab/Scicos/RTAI-Lab

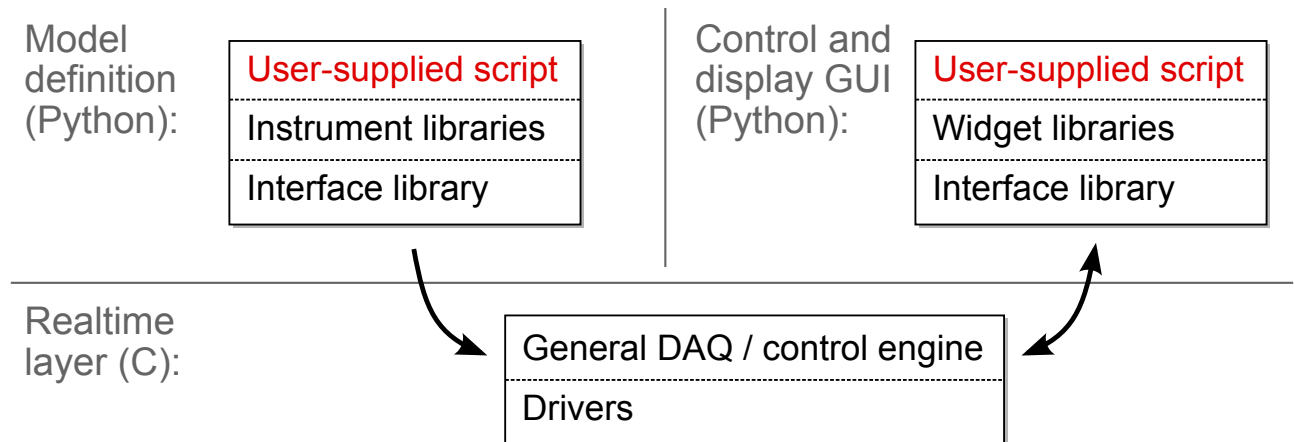
A simpler approach?

Alternative frameworks

Current alternatives:

- LabVIEW
- Matlab/Simulink
- Scilab/Scicos/RTAI-Lab

A simpler approach?



Credits

Helpful discussions:

Prof. Marc Bockrath (UC Riverside)

Prof. Henk Postma (CSU Northridge)

Testing:

Dr. Hang Zhang, Dr. Wenzhong Bao, Dr. Jairo Velasco Jr.,
Peng Wang, Tengfei Miao, Oleg Martynov

THANKS!

www.ugcs.caltech.edu/~mezurit2/