Mezurit 2: Virtual instrumentation for electronics experiments

Dr. Brian Standley

FOSDEM
2 Feb 2013
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

\[ V_b \text{ (\mu V)} \quad X_0 = \text{SR830}_\text{SineOut}(0, 8) \cdot \frac{109}{(109 + 100.3 \times 10^3)} \]

\[ I_x \text{ (nA)} \quad X_1 = \frac{\text{ADC}(0, 3)}{10} \cdot \text{SR830}_\text{SensIn}(0, 8) \cdot 10^{-5} \]

\[ I_y \text{ (nA)} \quad X_2 = \frac{\text{ADC}(0, 4)}{10} \cdot \text{SR830}_\text{SensIn}(0, 8) \cdot 10^{-5} \]

\[ V_{tg} \text{ (V)} \quad X_3 = \text{DAC}(0, 0) \]

\[ V_{bg} \text{ (V)} \quad X_4 = \text{DAC}(0, 1) \cdot 10 \]
A BRIEF DEMO
# Implementation

## Code (v0.91):

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>67</td>
<td>2456</td>
<td>1314</td>
<td>9591</td>
</tr>
<tr>
<td>C/C++ Header</td>
<td>37</td>
<td>513</td>
<td>747</td>
<td>1263</td>
</tr>
<tr>
<td>Python</td>
<td>3</td>
<td>134</td>
<td>40</td>
<td>333</td>
</tr>
</tbody>
</table>

(1inux) 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/](http://www.ugcs.caltech.edu/~mezurit2/)
Architecture

Interface loop:
- Receive input
- Display status
- Plot data

Tools:
- Widgets
- Settings
  - MCF nodes
- RPC nodes

Refresh logic
Operation logic

Terminal:
- Python interpreter

Acquisition loop:
- Binning system
- GPIB loop
- Logging
- Sweep
- Scanning
- Trigger

Wrapper libraries:
- DAQ
  - COMEDI
  - NI-DAQ
- GPIB
  - linux-gpib
  - NI-488.2
- Timing
  - GTimer
  - Perf. count.
- Varset
  - glibc
  - msvr

Systems:
- Virtual chan. computation
- MCF system
- RPC server
- Status messaging
Architecture

Interface loop:
- Receive input
- Display status

Tools:
- Widgets
- Settings
- MCF nodes
- RPC nodes

Acquisition loop:
- Binning system
- GPIB loop

Wrapper libraries:
- DAQ
  - COMEDI
  - NI-DAQ
- GPIB
  - linux-gpib
  - NI-488.2
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.*
Configuration system:

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

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

Config is scriptable through the "set_var" terminal function.

```
lock(mutex)
var = new_value
unlock(mutex)
set_value(widget, new_value)
```
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?

<table>
<thead>
<tr>
<th>Model definition (Python):</th>
<th>Control and display GUI (Python):</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-supplied script</td>
<td>User-supplied script</td>
</tr>
<tr>
<td>Instrument libraries</td>
<td>Widget libraries</td>
</tr>
<tr>
<td>Interface library</td>
<td>Interface library</td>
</tr>
</tbody>
</table>

Realtime layer (C):

- General DAQ / control engine
- Drivers
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/