

Lab::Measurement – measurement control with Perl

S. Reinhardt¹, C. E. Lane², C. Butschkow¹, A. Iankilevitch¹, A. Dirnachner¹, and A. K. Hüttel¹

¹Institute for Experimental and Applied Physics, University of Regensburg, 93040 Regensburg, Germany

²Department of Physics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA

Flexible measurement needed?!

- Tired of following your wires in square meters of LabVIEW diagrams?
- Tired of clumsy string handling and low-level driver functions in your looong C program?
- Use a text processing language to manage your measurement! Use Perl!

```
#!/usr/bin/env perl
# Read out SR830 Lock In Amplifier at GPIB address 13
use 5.010;
use Lab::Moose;

my $lia = instrument(
    type          => 'SR830',
    connection_type => 'LinuxGPIB',
    connection_options => { pad => 13 }
);

my $amp = $lia->get_amplitude();
say "Reference output amplitude: $amp V";

my $freq = $lia->get_freq();
say "Reference frequency: $freq Hz";

my $r_phi = $lia->get_rphi();
my ($r, $phi) = ($r_phi->(r), $r_phi->(phi));
say "Signal: r=$r V, phi=$phi degree";
```

Currently supported hardware



Hardware driver backends:

- NI-VISA (MS Windows) and all hardware supported by it
- LinuxGPIB and all hardware supported by it
- TCP connection, generic network socket (both Linux and MS Windows)
- USB-TMC lightweight driver (Linux, libusb)
- VXI-11 lightweight driver (Linux, libtirpc)
- Zurich Instruments LabOne API (both Linux and MS Windows)

Growing number of high-level drivers (more are very easy to add):

- Multimeters: HP / Agilent / Keysight
- DC sources: Yokogawa / Keithley
- Lock-in amplifiers: Stanford Research / Signal Recovery / Zurich Instruments
- Temperature controllers: Lakeshore / Oxford Instruments
- RF / microwave sources, spectrum analyzers, VNAs: Rohde & Schwarz
- and many more...

Key facts

- Open source / free software
- <https://www.labmeasurement.de/>
- License: same as Perl (GPL-1+ or Artistic)
- Releases on CPAN, development on Github
- Contributors and cooperations welcome!



Real world measurement

- Ferromagnetic resonance measurement with vector network analyzer (VNA)
- Outer loop: continuous sweep of magnetic field with OI Mercury iPS magnet controller
- Inner loop: VNA transmission measurement at multiple discrete microwave frequencies

```
#!/usr/bin/env perl
use 5.010;
use Lab::Moose;

# Define the two instruments
my $ips = instrument(
    type          => 'OI_Mercury::Magnet',
    connection_type => 'Socket',
    connection_options => { host => '192.168.3.15' },
);

my $vna = instrument(
    type          => 'RS_ZVA',
    connection_type => 'VXI11',
    connection_options => { host => '192.168.3.27' },
);

# Set VNA's IF filter bandwidth (Hz)
$vna->sense_bandwidth_resolution( value => 1 );

# The outer sweep: continuous magnetic field sweep
my $field_sweep = sweep(
    type        => 'Continuous::Magnet',
    instrument  => $ips,
    from        => 2,      # Tesla
    to         => 10,     # Tesla
    rate       => 0.01,   # Tesla/min
    start_rate => 1,     # Tesla/min (rate to approach start point)
    interval   => 0,     # run slave sweep as often as possible
);

# The inner sweep: set frequency to 1GHz, 2GHz, ..., 10GHz
my $frq_sweep = sweep(
    type        => 'Step::Frequency',
    instrument  => $vna,
    from        => 1e9,
    to         => 10e9,
    step       => 1e9
);

# The data file: gnuplot-style, VNA data prefixed with B
my $datafile = sweep_datafile(
    columns => [ 'B', 'f', 'Re', 'Im', 'r', 'phi' ]
);
# Add a live plot of the transmission amplitude
$vna->add_plot( x => 'B', y => 'r' );
# Define the measurement instructions per (B,f) point
my $meas = sub {
    my $sweep = shift;
    say "frequency f: ", $sweep->get_value();
    my $field = $sweep->get_field();
    # This is not really a VNA "sweep", but only a
    # point measurement at one frequency
    my $pd1 = $vna->sparam_sweep( timeout => 10 );
    # Record the result, prefixed with B
    $sweep->log_block(
        prefix => { field => $field },
        block  => $pd1
    );
};

# And go!
$field_sweep->start(
    slave      => $frq_sweep,
    datafile   => $datafile,
    measurement => $meas,
    folder     => 'Magnetic_Resonance',
);
```

Output files

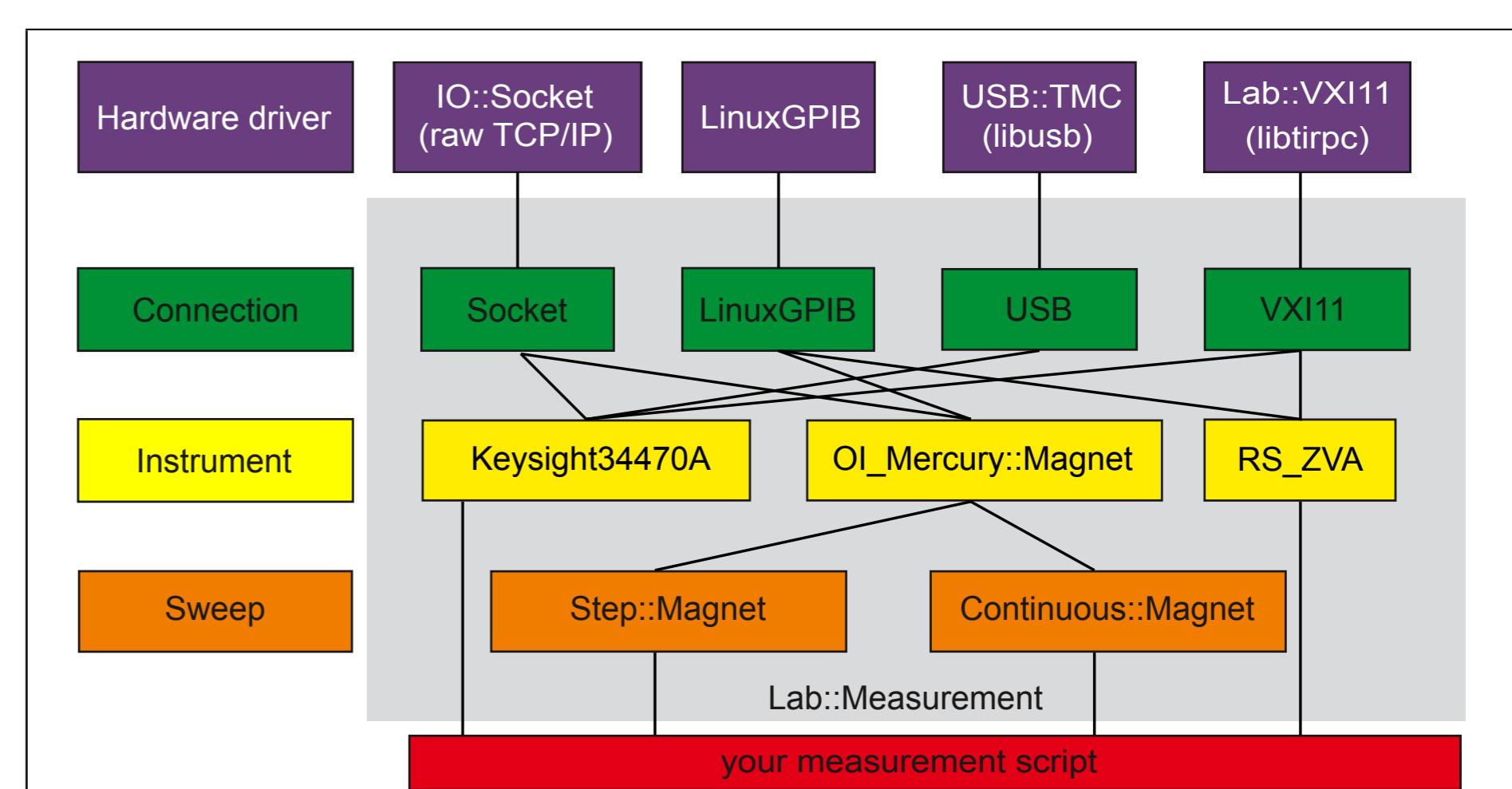
```
simon@dilfridge ~/FMR$ ls Magnetic_Resonance_001/
META.yml data.dat data.png magnet-resonance.pl
```

- **META.yml**: various metadata (host, user, date, L::M version, command line arguments)
- **data.dat**: measured data, in tab-separated Gnuplot format
- **data.png**: live plot at the end of the measurement, as a png image
- **magnetic-resonance.pl**: archival copy of the measurement script

Advanced sweep features

- Multidimensional sweeps, e.g. 3D sweep: creating one 2D datafile for each step of the outermost sweep
- Log arrays and matrices of data (PDLs). Useful for spectrum analyzers, VNAs, oscilloscopes
- Extensive support for live plots via gnuplot: line plots (2D data) and color maps (3D data)
- Customizing live plots: access to all gnuplot plot and curve options via PDL::Graphics::Gnuplot

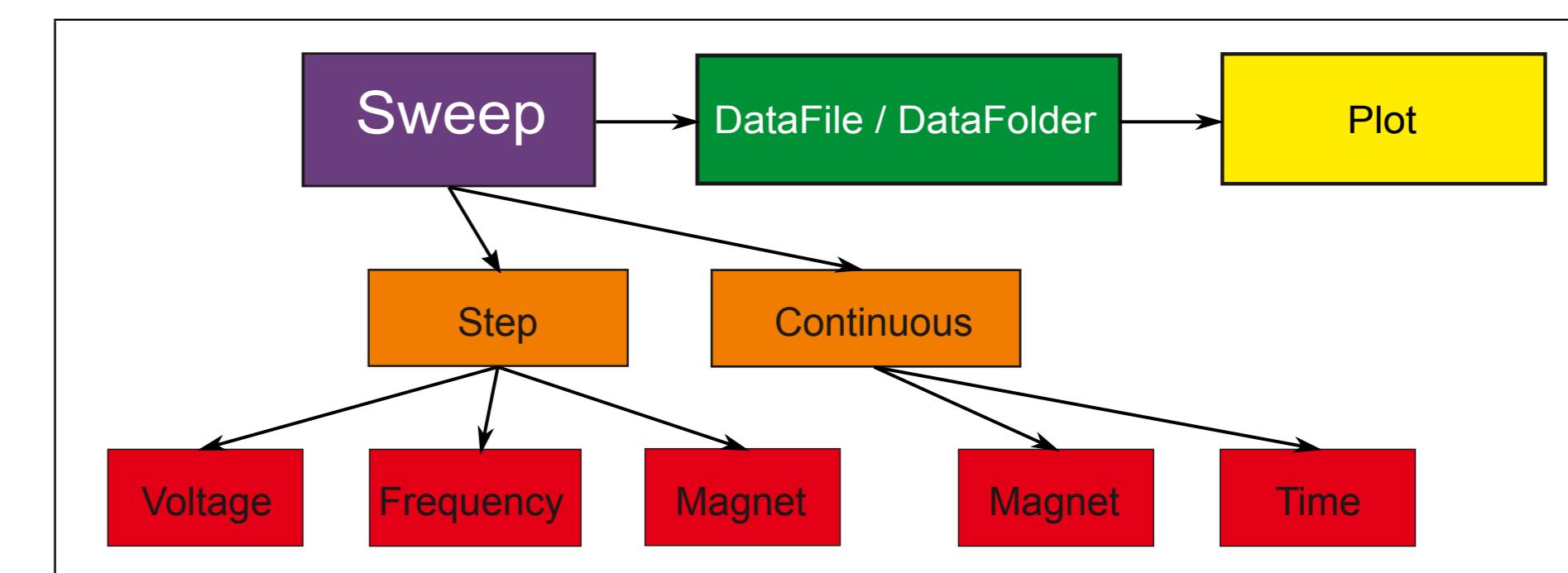
Layer structure



- Modular structure. Easy to extend with new instrument drivers and connection types
- Abstract IO layer, makes instrument drivers independent of hardware backends

Recent improvements

- New drivers
 - OI Mercury magnet controller and level meter
 - Zurich Instruments MFLI lock-in amplifier and MFIA impedance analyzer
- New connection backends
 - USB-TMC, VXI-11: lightweight user space drivers for Linux. Provide fast and cheap alternative to GPIB on Linux; high reliability since based on standard system libraries (libusb, sunrpc, libtirpc)
 - Initial support for the LabOne API of Zurich Instruments: provide Perl bindings to proprietary vendor library; instruments accessed via Ethernet or USB
- New high-level Sweep framework: Modern Perl implementation; use state of the art object-oriented programming



- Separate classes for sweeps, datafiles, datafolders, and plots
- Most operational details of the sweeps are implemented in subclasses of the Lab::Moose::Sweep base class
- High modularity: very easy to extend with new functionality

Outlook

- Implement more sweep types with the new framework, e.g. temperature sweep
- Improve on tutorials:
 - Entry-level docs for new users: basics about connections, instruments, and sweeps
 - Developer docs: how to write and test new drivers; use Moose features efficiently