Lab::Measurement – measurement control with Perl

C. Butschkow[1], A. Iankilevitch[1], S. Geißler[1], A. Dirnaichner[1], and A. K. Hüttel[1]

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

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!

```
# Read out SR830 lock-in at GPIB address 13
use Lab::Instrument::SR830;

my $sr=new Lab::Instrument::SR830(
        connection_type=>'LinuxGPIB',
        gpib_address => 13,
        gpib_board=>0,
);

my $amp=$sr->get_amplitude();
print "Reference amplitude: $amp V\n";

my $freq=$sr->get_frequency();
print "Reference frequency: $freq Hz\n";

my ($r,$phi)=$sr->get_rphi();
print "Signal             : r=$r V   phi=$phi\n";
```

Currently supported hardware

Hardware driver backends:

- LinuxGPIB and all hardware supported by it
- Linux USB-TMC kernel driver
- NI-VISA (Windows and Linux) and all hardware supported by it
- Oxford Instruments IsoBus
- Raw TCP connection, generic network socket
- RS232 serial port

High-level drivers at the moment (more are *very* easy to add):

ABB TRMC2; Agilent 34401A, 34420A, 3458A, 83732A, U2000; Anritsu MG369xB; Oxford Instruments ILM210, IPS, ITC503; Rohde & Schwarz FSV7, SMB100A, ZVB; Signal Recovery 726x; Stanford Research SR830; Trinamic PD11042; Yokogawa 7651, GS200

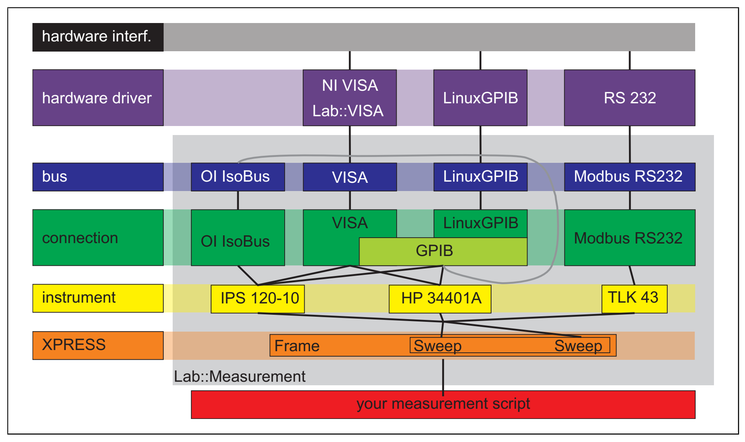
High-level drivers in preparation:

Agilent 34411A, 34970A, 8360; Keithley 2000, 2400, 6221; Knick S252; Lakeshore 336, 370; Oxford Instruments Triton, Mercury; Stanford Research SR620, SR780, SIM928

Key facts

- open source / free software
- http://www.labmeasurement.de/
- license: same as Perl (GPL-1+ or Artistic)
- releases on CPAN, development on Github
- contributors and cooperations welcome!

Internal architecture



- Lab::Bus — hardware driver encapsulation
- Lab::Connection — thin glue layer, one connection per instrument
- Lab::Instrument — instrument-level logic, sending / receiving command strings
- Lab::XPRESS — sweep and frame classes for nested loop measurements

Example instrument driver

- Rohde & Schwarz SMB100A signal generator

```
package Lab::Instrument::RSSMB100A;
our $VERSION = '3.40';

use strict;
use Lab::Instrument;
use Time::HiRes qw (usleep);

# inherit the generic code
our @ISA = ("Lab::Instrument");

our %fields = (
        supported_connections => [ 'GPIB', 'TCPraw' ],
        # default settings for the supported connections
        connection_settings => {
                gpib_board => 0,
                gpib_address => undef,
        },
        # default device settings
        device_settings => {
        },
);

# [...]

sub set_power {
        my $self=shift;
        my ($power) = $self->_check_args( \@_, ['value'] );
        $self->write("POWer:LEVel $power_DBM");
}

sub get_power {
        my $self = shift;
        return $self->query("POWer:LEVel?");
}

sub set_pulselength {
        my $self = shift;
        my ($length) = $self->_check_args( \@_, ['value'] );
        $self->write("PULM:WIDT $length_s");
}

sub get_pulselength {
        my $self = shift;
        my $length = $self->query("PULM:WIDT?");
        return $length;
}

# [...]

sub power_on {
    my $self=shift;
    $self->write('OUTP:STATe ON');
}

sub power_off {
    my $self=shift;
    $self->write('OUTP:STATe OFF');
}

1;

# the documentation, will be converted to HTML
=pod
=encoding utf-8

=head1 NAME

Lab::Instrument::RSSMB100A - Rohde & Schwarz SMB100A Signal Generator

=head1 AUTHOR/COPYRIGHT

  Copyright 2005 Daniel Schroeer (<schroeer@cpan.org>)
            2011 Andreas K. Huettel
            2014 Andreas K. Huettel

This library is free software; you can redistribute it and/or modify it
under the same terms as Perl itself.

=cut
```

- GPIB? Ethernet? doesn't matter here, the connection code takes care of that!
- fast implementation of many different instrument functions
- additional features possible, e.g. "device cache": stores device parameters if setting a value is fast but reading it out is slow / impossible

Real world measurement

```
#-------- 0. Import Lab::Measurement -------
use Lab::Measurement;
#-------- 1. Initialize Instruments --------
my $bias = Instrument('Yokogawa7651',
        {
        connection_type => 'VISA_GPIB',
        gpib_address => 3,
        gate_protect => 0
        });
my $multimeter = Instrument('Agilent34410A',
        {
        connection_type => 'VISA_GPIB',
        gpib_address => 17,
        nplc => 10                          # integration time in number of
                                            # powerline cylces [10*(1/50)]
        });
my $gate = Instrument('Yokogawa7651',
        {
        connection_type => 'VISA_GPIB',
        gpib_address => 6,

        gate_protect => 1,
        gp_min_units => -10,
        gp_max_units => 15,
        gp_max_units_per_second => 10e-3
        });
#-------- 2. Define the Sweeps -------------
my $gate_sweep = Sweep('Voltage',
        {
        mode => 'step',
        instrument => $gate,
        points => [-5, 5],                  # [starting point, target] in Volt
        stepwidth => [0.1],
        rate => [5e-3],                     # [rate to approach start, sweeping
                                            # rate for measurement] in Volt/s
        });
my $bias_sweep = Sweep('Voltage',
        {
        instrument => $bias,
        points => [-5e-3, 5e-3],            # [starting point, target] in Volt
        rate => [0.1, 0.5e-3],              # [rate to approach start, sweeping
                                            # rate for measurement] in Volt/s
        interval => 1,                      # measurement interval in s

        delay_before_loop => 10             # delay before sweep begins in s
        });
#-------- 3. Create a DataFile -------------
my $DataFile = DataFile('Gate_IV_sample1.dat');

$DataFile->add_column('GateVoltage');
$DataFile->add_column('BiasVoltage');
$DataFile->add_column('Current');
$DataFile->add_column('Resistance');

$DataFile->add_plot({
        'type'    => 'pm3d'
        'x-axis'  => 'GateVoltage',
        'y-axis'  => 'BiasVoltage',
        'cb-axis' => 'Current'
        });
#-------- 4. Measurement Instructions -------
my $my_measurement = sub {

        my $sweep = shift;

        my $gate_voltage = $gate->get_value();          # gate voltage source
        my $bias_voltage = $bias->get_value();          # bias voltage source
        my $current = $multimeter->get_value()*1e-7;    # I/V converter output
        my $resistance = ($current != 0) ? $voltage/$current : '?';

        $sweep->LOG({
                GateVoltage => $gate_voltage,
                Voltage => $bias_voltage,
                Current => $current,
                Resistance => $resistance
                });
};
#-------- 5. Put everything together -------

$DataFile->add_measurement($my_measurement);
$voltage_sweep->add_DataFile($DataFile);

my $frame = Frame();
$frame->add_master($gate_sweep);                    # the outer, slow loop
$frame->add_slave($bias_sweep);                     # the inner, fast loop

$frame->start();
```

- two nested loops:
  gate voltage V_g and bias voltage V_{sd}
- read out multimeter for each point

Data and metadata

each measurement generates subdirectory

```
huettel@kailua ~/tmp/MEAS_2894 $ ls
Config.txt  Ifcurve.dat  Ifcurve_01.png  freqsweep_SMB100A.pl
```

- archival copy of the measurement script
- configuration of all controlled devices
- data, tab-separated GnuPlot format
- automated plot (identical to live view)

Results obtained using Lab::Measurement

- A. Dirnaichner, A. K. Hüttel, *et al.*, submitted, arXiv:1502.02005 (2015).
- D. R. Schmid, A. K. Hüttel, *et al.*, submitted, arXiv:1407.2114 (2014).
- D. R. Schmid, A. K. Hüttel, *et al.*, accepted by Phys. Rev. B, arXiv:1312.6586 (2014).
- S. Ratz, A. K. Hüttel, *et al.*, New J. Phys. **16**, 123040 (2014).
- M. Gaass, A. K. Hüttel, *et al.*, Phys. Rev. B **89**, 241405(R) (2014).
- A. Kumar, A. K. Hüttel, *et al.*, Phys. Rev. B **89**, 075428 (2014).
- P. L. Stiller, A. K. Hüttel, *et al.*, physica status solidi(b) **250**, 2518 (2013).
- G. Granger, *et al.*, Nature Physics **8**, 522 (2012).
- D. R. Schmid, A. K. Hüttel, *et al.*, New J. Phys. **14**, 083024 (2012).
- C. Butschkow, *et al.*, submitted, arXiv:1110.5507 (2011).
- M. Gaass, A. K. Hüttel, *et al.*, Phys. Rev. Lett. **107**, 176808 (2011).
- D. Taubert, *et al.*, Rev. Sci. Instrum. **82**, 123905 (2011).
- D. Taubert, *et al.*, J. Appl. Phys. **109**, 102412 (2011).
- D. Taubert, *et al.*, Phys. Rev. B **83**, 235404 (2011).
- D. Taubert, *et al.*, Phys. Rev. B **82**, 161416(R) (2010).
- D. Preusche, A. K. Hüttel, *et al.*, J. Appl. Phys. **106**, 084314 (2009).
- D. Schröer, *et al.*, Phys. Rev. B **76**, 075306 (2007).

We gratefully acknowlegde funding by the DFG via the Emmy Noether grant Hu1808/1-1 "Carbon nanotubes as electronical and nano-electromechanical hybrid systems in the quantum limit", the collaborative research centre SFB 689 "Spin phenomena in reduced dimensions", and the graduate research school GRK 1570 "Electronic properties of carbon based nanostructures".