Lab::Measurement – measurement control with Perl

F. Olbrich[1], A. Dirnaichner[1], D. Kalok[1], D. Taubert[2], D. Schröer[2], and A. K. Hüttel[1]

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

[2] *Fakultät für Physik, LMU München, Geschwister-Scholl-Platz 1, 80539 München, 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";
```

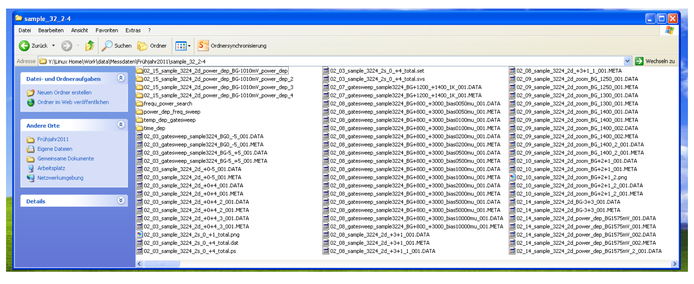
Safe measurement needed?!

- Your sample only survives small voltage steps. You make a typo. Your sample dies. Noooooooo!
- Use ‘gateprotect’ for voltage sources: pre-defined limits for sweep speed, step size, range
- Voltage sweeps, not steps, whenever needed

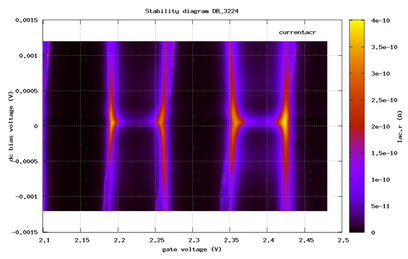
```
# declare the gate voltage source
my $YokGate=new Lab::Instrument::Yokogawa7651({
    'connection_type'=> 'LinuxGPIB',
    'gpib_board'     => 0,
    'gpib_address'   => 12,
    'gate_protect'   => 1,
    'gp_max_volt_per_second' => 0.05,  # max sweep speed
    'gp_max_step_per_second' => 10,    # max steps per second
    'gp_max_volt_per_step'   => 0.005, # max step size
    'gp_min_volt'            => -2,    # hard negative limit
    'gp_max_volt'            => 0.2,   # hard positive limit
    'fast_set' => 1,
});

# sweep to the start gate voltage -0.5V
$YokGate->set_voltage(-0.5);
```

Overview needed?!



- `make_overview.pl`
- Autogenerates a LaTeX file and images, e.g. for all data in a directory
- Includes comments, axis labels, ...
- “Generate measurement log from data files”



stability diagram for magnetic field 0 mT;

Ithaco: Verstaerkung 1e-008 , Rise Time 1 ms;

Messen der Ausgangsspannung des Ithaco über Agilent und Lock-in;

Lock-in-Einstellungen Frequency 137.69Hz, amplitude 50mV*10^-4=5µV

Voltage dividers DC: 0.01 AC: 0.0001

Temperatur = 25 mK;

...

Currently supported hardware

Hardware driver backends:

- NI-VISA (Windows and Linux) and all hardware supported by it
- LinuxGPIB and all hardware supported by it
- RS232 serial port

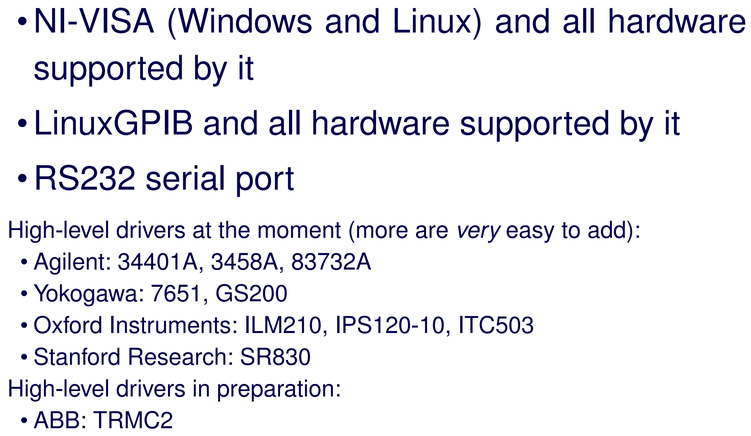
High-level drivers at the moment (more are *very* easy to add):

- Agilent: 34401A, 3458A, 83732A
- Yokogawa: 7651, GS200
- Oxford Instruments: ILM210, IPS120-10, ITC503
- Stanford Research: SR830

High-level drivers in preparation:

- ABB: TRMC2
- Agilent: 34411A, 34420A, 34970A, 8360
- Keithley: 2000, 2400, 6221
- Knick: S252
- Lakeshore: 336, 370
- Stanford Research: SR620, SR780, SIM928

Internal architecture



- Lab::Bus — hardware driver encapsulation
- Lab::Connection — thin glue layer between Instrument and Bus
- Lab::Instrument — measurement instruments, high-level drivers; these are accessed by your measurement script

Key facts

- open source / free software
- http://www.labmeasurement.de/
- license: same as Perl (GPL-2 / Artistic)
- releases on CPAN, development on Gitorious
- contributors welcome!

Results obtained using Lab::Measurement / Lab::VISA

- D. Schröer, *et al.*, Phys. Rev. B **76**, 075306 (2007).
- D. Preusche, A. K. Hüttel, *et al.*, J. Appl. Phys. **106**, 084314 (2009).
- D. Taubert, *et al.*, Phys. Rev. B **82**, 161416(R) (2010).
- D. Taubert, *et al.*, Phys. Rev. B **83**, 235404 (2011).
- D. Taubert, *et al.*, J. Appl. Phys. **109**, 102412 (2011).
- D. Taubert, *et al.*, Rev. Sci. Instrum. **82**, 123905 (2011).
- M. Gaass, A. K. Hüttel, *et al.*, Phys. Rev. Lett. **107**, 176808 (2011).

Real world example

```
use strict;
use Lab::Instrument::Yokogawa7651;
use Lab::Instrument::IPS12010;
use Lab::Instrument::HP34401A;
use Lab::Instrument::SR830;
use Lab::Measurement;

# measurement range and resolution
my $Vbiasstart = -0.0036;       # V, after divider
my $Vbiasstop = 0.0036;         # V, after divider
my $Vbiasstep = 0.00002;        # V, after divider
my $Bstart=0.1;                 # T
my $Bstop=0;                    # T
my $Bstep=0.01;                 # T

# general measurement settings and constants
my $Vbiasdivider = 0.01;        # <1, voltage divider value
my $currentamp = 1e-9;          # A/V
my $sample = "nanotube";
my @starttime = localtime(time);
my $startstring=sprintf("%04u-%02u-%02u_%02u-%02u-%02u",
                        $starttime[5]+1900,$starttime[4]+1,$starttime[3],
                        $starttime[2],$starttime[1],$starttime[0]);
my $title = "Bias versus magnetic field";
my $filename = $startstring."_biasfield";

# the bias voltage source
my $YokBias=new Lab::Instrument::Yokogawa7651({
    'connection_type'=> 'LinuxGPIB',
    'gpib_board'     => 0,    'gpib_address'  => 3,
    'gate_protect'   => $Vbiasprotect,
    'gp_max_volt_per_second' => 0.05/$Vbiasdivider,
    'gp_max_step_per_second' => 10,
    'gp_max_volt_per_step' => 0.005/$Vbiasdivider,
    'fast_set' => 1,
});

# the lock-in: ac measurement
my $SRS = new Lab::Instrument::SR830({
    'connection_type'=> 'LinuxGPIB',
    'gpib_board'     => 0,    'gpib_address'  => 8,
});

# the multimeter: dc measurement
my $HP = new Lab::Instrument::HP34401A({
    'connection_type'=> 'LinuxGPIB',
    'gpib_board'     => 0,    'gpib_address'  => 12,
});

# the superconducting magnet control
my $magnet=new Lab::Instrument::IPS12010({
    'connection_type' => 'LinuxGPIB',
    'gpib_board'     => 0,    'gpib_address'  => 24,
});

# general comments for the log
my $comment=<<COMMENT;
Bias sweeps versus magnetic field; gate voltage -3.74 V
B from $Bstart to $Bstop step size $Bstep
Bias voltage from $Vbiasstart to $Vbiasstop step size $Vbiasstep
Current preamp $currentamp A/V
SRS lock-in: integrate 100ms, freq 117.25Hz, sensit. 10mV
COMMENT

# the "measurement": things like filename, live plot, etc,
# plus all the metadata (data file columns, axes, plots, ...)
my $measurement=new Lab::Measurement(
    sample          => $sample,         title           => $title,
    filename_base   => $filename,       description     => $comment,
    live_plot       => 'currentacx',    live_refresh    => '200',
    constants       => [
        {   'name'          => 'currentamp',
            'value'         => $currentamp,
        },
    ],
    columns         => [ # documentation of the data file columns
        {   'unit'          => 'T',   'label'         => 'B',
            'description'   => 'magnetic field perpendicular to nanotube',
        },
        {   'unit'          => 'V',   'label'         => 'Vbias',
            'description'   => "dc bias voltage",
        },
        {   'unit'          => 'A',   'label'         => 'Idc',
            'description'   => "measured dc current",
        },
        {   'unit'          => 'A',   'label'         => 'Iac,x',
            'description'   => "measured ac current, x component",
        },
        {   'unit'          => 'A',   'label'         => 'Iac,y',
            'description'   => "measured ac current, y component",
        },
    ],
    axes            => [ # possible axes for plotting, and their data columns
        {   'unit'          => 'T',   'label'         => 'B',
            'expression'    => '$C0',
            'description'   => 'magnetic field perpendicular to nanotube',
        },
        {   'unit'          => 'V',   'label'         => 'Vbias',
            'expression'    => '$C1',
            'description'   => 'dc bias voltage',
        },
        {   'unit'          => 'A',   'label'         => 'Idc',
            'expression'    => '$C2',
            'description'   => 'measured dc current',
        },
        {   'unit'          => 'I',   'label'         => 'Iac,x',
            'expression'    => '$C3',
            'description'   => 'measured ac current, x component',
        },
        {   'unit'          => 'I',   'label'         => 'Iac,y',
            'expression'    => '$C4',
            'description'   => 'measured ac current, y component',
        },
    ],
    plots           => { # plots that can be made using the axes above
        'currentdc'     => {
            'type'          => 'pm3d',
            'xaxis'         => 0,   'yaxis'         => 1,
            'cbaxis'        => 2,   'grid'          => 'xtics ytics',
        },
        'currentacx'    => {
            'type'          => 'pm3d',
            'xaxis'         => 0,   'yaxis'         => 1,
            'cbaxis'        => 3,   'grid'          => 'xtics ytics',
        },
    },
);

# correct the sign of the step sizes if required
unless (($Bstop-$Bstart)/$Bstep > 0) { $Bstep = -$Bstep; }
unless (($Vbiasstop-$Vbiasstart)/$Vbiasstep > 0) { $Vbiasstep = -$Vbiasstep; }
my $Bstepsign=$Bstep/abs($Bstep);
my $Vbiasstepsign=$Vbiasstep/abs($Vbiasstep);

## ENOUGH PREPARATION, NOW THE MEASUREMENT STARTS :)

# go to start field
print "Ramping magnet to starting field... ";
$magnet->set_field($Bstart);
print " done!\n";

# here you could eg. check the temperature

# the outer measurement loop: magnetic field
for (my $B=$Bstart;$Bstepsign*$B<=$Bstepsign*$Bstop;$B+=$Bstep) {

        $measurement->start_block();

        # set the field
        $magnet->set_field($B);

        # the inner measurement loop: bias voltage
        for (my $Vbias=$Vbiasstart;
                $Vbiasstepsign*$Vbias<=$Vbiasstepsign*$Vbiasstop;
                $Vbias+=$Vbiasstep) {

            # set the bias voltage
            $YokBias->set_voltage($Vbias/$Vbiasdivider);

            # read dc signal from multimeter
            my $Vdc = $HP->get_value();

            # read the ac signal from the lock-in
            my ($Vacx,$Vacy)=$SRS->get_xy();

            # we multiply with (-1)*$currentamp (inverting amplifier)
            my $Idc = -$Vdc*$currentamp;
            my $Iacx=-$Vacx*$currentamp;
            my $Iacy=-$Vacy*$currentamp;

            # write the values into the data file
            $measurement->log_line($B, $Vbias, $Idc, $Iacx, $Iacy);
        }
};

# all done
$measurement->finish_measurement();
print "End of Measurement!\n";
```

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”.