

# KAME ver 4.1.10 (Partially in English)

Sep 23, 2015 Kentaro Kitagawa

## Preface

---

“kame” is a C++ multithreaded program which focuses on automated measurement by scripting with Ruby language.

It is especially intended to be used in solid-state NMR experiment, but includes many types of measurement drivers. The source is cross-platform (Linux/Mac OS X/Windows).

The license is open source, GPL(GNU public licence) version 2 or laterです。

To install kame on Mac/Linux machines	5
To install kame on Windows machines	5
<i>The case of using compiled binaries</i>	5
<i>The case of compiling source zip</i>	5
Basic usage	6
Usage for text boxes	7
Usage for Graphs/Charts	7
メニュー項目	8
ファイル	8
開く	8
保存	8
閉じる	8
ログを取る	8
終了	8
測定	8
停止	8
スクリプト	8
開始	8
新規ラインシエル	8
<i>Graphviz .dot</i> を保存	8
表示	8
ヘルプ	9
メッセージウィンドウ	9
タブ	9
ドライバ	9
グラフ	10
キャリブレーションテーブル	11
<i>Node Browser</i>	11
インターフェース	11

スカラエントリ	12
Rawストリームリーダー	13
DCソース	13
デジタルマルチメータ(DMM)	13
デジタルストレージオシロスコープ(DSO)	14
ファンクションジェネレータ	15
液面計	15
ロックインアンプ/容量計	15
超伝導マグネット電源	16
標準信号発生器 (SG)	17
ネットワークアナライザー	17
温度計/温調器	18
カウンター	19
流量制御	19
モーター制御	20
NMR Pulser	20
NMR FID/Echo measurement	22
NMR Relaxation Rate Measurement	24
NMR Spectrum Measurement (Frequency Sweep)	25
NMR Spectrum Measurement (Field sweep)	26
NMR LC autotuner	27
電流反転抵抗測定	27
例 (各温度でスペクトラムを保存する)	28
Fedoraインストール直後にネットワークが使えないときは?	29
Fedoraの「ソフトウェアの更新」が失敗するが?	29

LinuxでDAQmxを使っていて、頻繁にフリーズするときは？	29
GPIBで通信エラーが起こるときは？	29
シリアルポートの設定は？	29
シリアルポートが開けない時は？	29
kameで対応していない機器を使いたい時は？	30
NI製DAQmxデバイスを用いたNMRシステム(パルサー及びオシロスコープ /アベレージャ)	30
Sシリーズデバイス	31
Mシリーズデバイスのパルサー用接続	31

# To install kame on Mac/Linux machines

See japanese document or memos in the source zip.

# To install kame on Windows machines

## The case of using compiled binaries

Qt of version 5.3 or later including MinGW32bit is necessary.

If you need GPIB,

Install a 488.2 driver from National Instruments (support for ANSI C and direct entries are needed)

If you need DAQmx,

Install a DAQmx driver from National Instruments (support for ANSI C is needed)

Expand kame-4\*\*-win32.zip and launch it via kame.bat. Use not too old machine so that it has enough functioning support for OpenGL2.1.

If you use USB devices of Thamway, copy fx2fw.bix, slow\_dat.bin, fullspec\_dat.bin to the directory of kame executable. If something is wrong, try deleting fullspec\_dat.bin.

## The case of compiling source zip

First, try the above instructions.

Next, several depending libraries.

fftw-3.3.4-dll32.zip (Expand it in the upper directory of the kame source, and name the directory "fftw3")

<http://www.fftw.org/download.html>

Install Ruby and msys (<http://www.mingw.org/>), and activate the path for the ruby. And expand ruby-2.2.\* source in the upper directory of the kame source, and change the name to "ruby".

Launch msys.bat, and in the ruby source directory, activate the path of the mingw (suchlike... PATH=\$PATH:/c/QT/Qt-(somewhat)/Tools/mingw4(somewhat)\_32/bin), and type

```
CFLAGS="-g0" ./configure --enable-shared --disable-rubygems
```

```
make
```

Then you would obtain a DLL of Ruby

For zlib, get "Compiled DLL" from the following, and expand it in the upper directory as well and name it "zlib"

<http://www.zlib.net/>

You need to compile the newer version of GSL by yourself

<http://www.gnu.org/software/gsl/>

Download the newest gsl, and expand it in the upper dir, naming "gsl". Do the following as you have done for ruby

```
./configure;make
```

Then, copy ./libs/libgsl-0.dll and cblas/.libs/libgsl-cblas-0.dll to the directory of the executables.

Open kame.pro from Qt creator. Build it in the mingw32 environment.

To execute kame, you need to copy the following dlls to the directory of the kame executables (for example, build-kame-\*\*\*-Debug).

zlib1.dll(in C:\Program Files\*\GnuWin32\bin or so), llibgsl.dll, libgslcblas.dll (in gsl) ,  
ibftw3-3.dll (in FFTW3) , msvc\*-ruby\*.dll(in ruby)

And you can launch kame from Qt Creator.

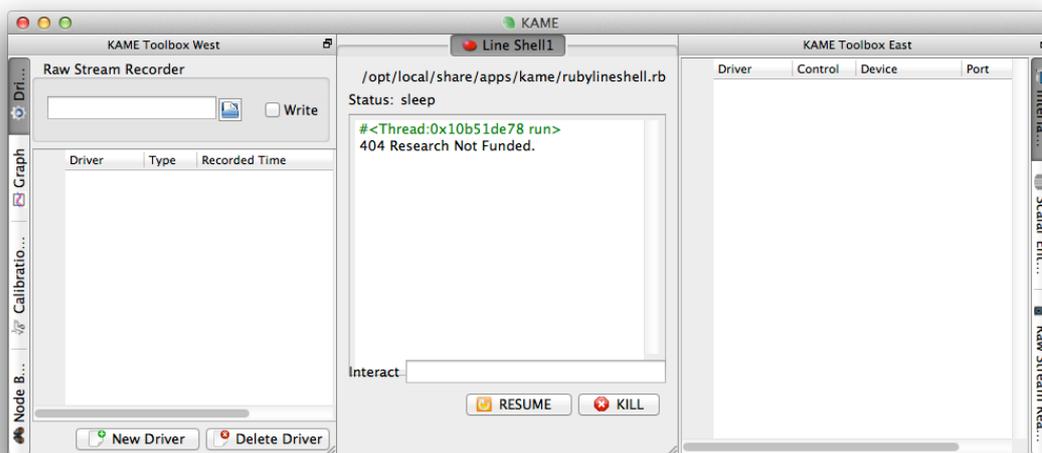
For i18n, kame\_ja.qm must be in the same directory, and “kame/script/rubylineshell.rb” is needed in the “Resources” directory to use a line shell.

## First step

---

Error logs will be written in /tmp/kame.log(Unices) or in the current directory(Windows).

## Basic usage



(First time) Add favorite driver for your instrument by invoking “New” button in the “Driver” tab.

(Next time) Open your .kam file to restore the saved state in the menu “File”-“Open”

Push “Start” button in the “Interface” tab. Before this, you must specify address and/or port for GPIB/SerialPort/TCP.

When you click a label of driver in “Driver” or “Interface” tab, you can open the driver specific window

The tab “Scalar Entry” is intended to prepare the conditions for storing line-by-line text data. The entries would be temperature or voltage etc.. When the difference after the last written value exceeds “Delta”, one line will be appended in the file. If “Delta” is negative, it is appended on getting new value. When you click the cell, a chart as a function of time will appear.

The tab “Graph” is for making X-Y-(Z) graphs from scalar entries.

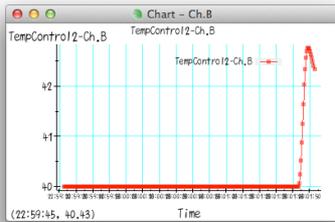
To save the state in kame, invoke “File”-“Save”. Frequent saving may save your time.

## Usage for text boxes

When you edit the text of UIs, you have to press “Enter” key at the end, because the instruments should not reflect the intermediate text. Unless so, the original state is recovered.

The values in most of the text boxes can be operated from the Ruby scripts.

## Usage for Graphs/Charts



They are drawn by OpenGL in 3D space as transparent entities.

Left double click opens detailed dialog.

Right double click opens hints for you.

Left click and dragging on the plot can be used to expand scales.

Right click on the plot enables auto scaling.

Wheeling on the plot is zooming in or out.

Right click and dragging on the axis expands the axis.

Right click on the axis enables auto scaling for the axis.

Wheeling on the axis inclines the plot in the 3D space.

Middle click and dragging inclines the plot in the 3D space more freely.

Middle click restores a view point.

If you are dealing with the wave data, which can be stored in the text, you have a filepath bar above or beneath the graph. The directory icon can choose the file name, but you must not forget to click “Dump” when you want to save the data.

## メニュー項目

### ファイル

#### 開く

.kamファイルを読み込みます。中身はRubyスクリプトになっています。従って、スクリプト実行画面が中央部に開き、エラーが出れば赤く表示されます。

#### 保存

現在開いている全ドライバや他の一部の設定を全て.kamファイルに保存します。内部的には、Rubyからコントロール可能なノードの内、保存属性が設定されているもののスナップショットが保存されます。

#### 閉じる

全ドライバ、キャリブレーションテーブルが削除されます。誤って押さないようにしましょう。

#### ログを取る

測定器との通信ログ、ノードの追加／削除情報も/tmp/kame.log（Windowsでは実行フォルダのkame.log）に保存されます。ディスクを圧迫するので、不必要な時は止めましょう。

#### 終了

測定中はキャンセルされます。保存を忘れないようにしましょう。

### 測定

#### 停止

全測定を停止させます。うっかり押さないようにしましょう。

### スクリプト

#### 開始

自動測定のためのRubyスクリプトを読み込みます。標準では.seqファイルです。書き方は別章で説明します。

#### 新規ラインシエル

Rubyスクリプトを一行ごとに打ち込める画面を開きます。起動時に既に1つ開いています。

#### Graphviz .dotを保存

生成した.dotファイルをgraphvizで処理すれば、ノードツリーをグラフィカルに把握できます。私も使わない機能です。

### 表示

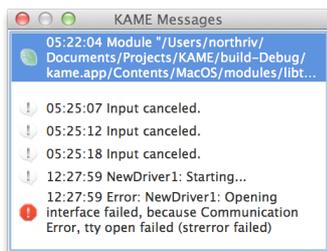
タブをクリックしても同じです。

## ヘルプ

用意してません。このマニュアルを見て下さい。

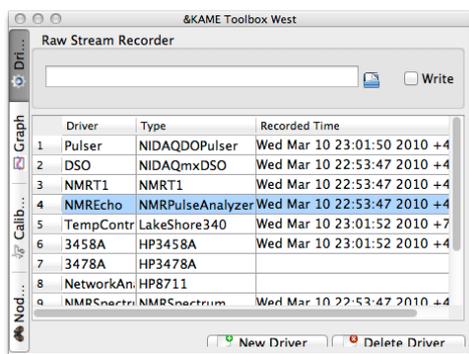
## メッセージウィンドウ

起動時から左端にあります。主要なメッセージ、警告やエラーが表示されます。もう少し細かい情報は標準出力と/tmp/kame.log（Windowsでは実行フォルダのkame.log）に保存されます。



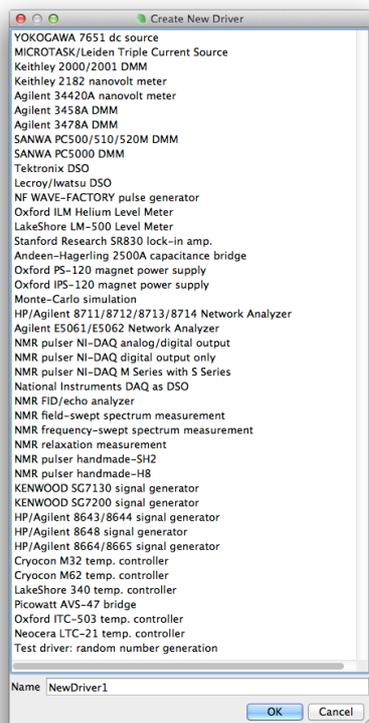
## タブ

## ドライバ



ドライバの追加・削除が出来ます。また、ドライバー一覧上でクリックするとドライバ固有の設定ウィンドウが出ます。レコード時刻は、最後に測定器から記録に値するデータが来た時間を表示します。

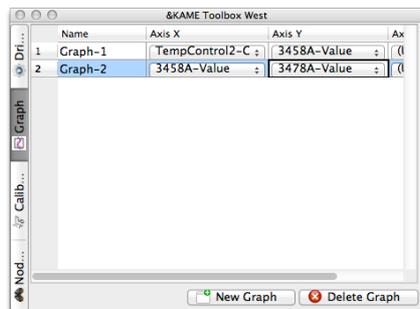
新規ドライバを押すと次のようなウィンドウが出て、必要なドライバに名前を付けて作成できます。名前は後から変更は難しいので、考えてつけて下さい。



## Rawストリームレコーダ

生データを極力保存します。「書き込み」チェックボックスが入っている間、例えば、オシロスコープの波形が全て記録されます。GZIP形式で圧縮されますが、当然それなりのディスク容量を消費します。

## グラフ

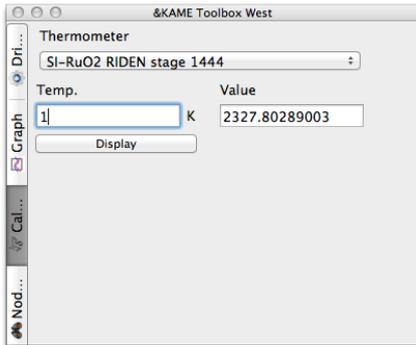


2次元/3次元グラフを生成します。X/Y/Z軸を「スカラエントリ」の中の項目から選んで下さい。

設定点数を超えた古いデータはグラフから消されます。

“Live”はメモリにレコードされた全データ、“Stored”は「テキストライタ」でディスクに記録されたデータを表示します。

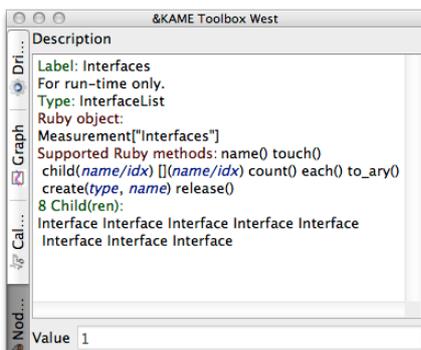
## キャリブレーションテーブル



温度計の校正データを定義するのに使います。定義ファイルは今のところ、手動でRubyスクリプトを書く必要があります。サンプルがkame/Measurementsフォルダにあります。スプライン曲線による補完、またはチェビシェフ多項式による定義が可能です。

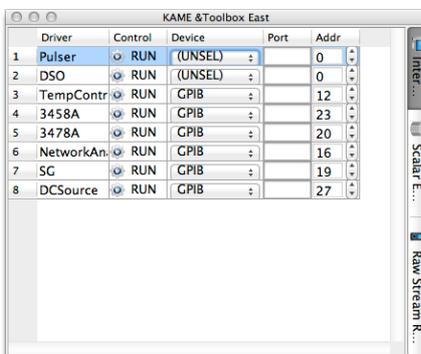
「表示」を押すと、校正曲線を表示します。

## Node Browser



kameの中のデータは全てツリー構造に管理されています。Rubyスクリプトで制御する際、`"Measurement["Drivers"]["driver1"]["foo"].value=1.0`という感じで行います。画面上の入力部がRuby上でどう扱われるか表示します。スクリプトのサンプルは、kame/Sequencesフォルダにあります。

## インターフェース



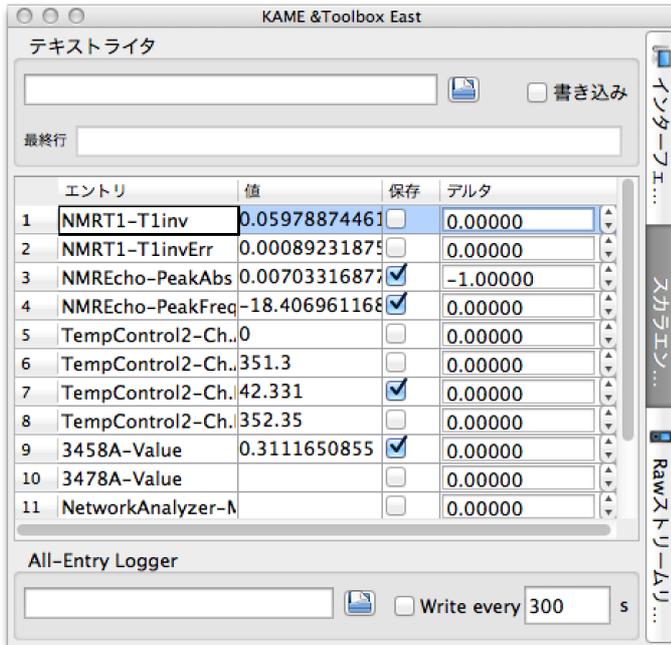
ドライバが測定器との通信経路を必要としている場合、ここに設定項目及び、起動/停止ボタンが出ます。通常はードライバに対し一つです。「デバイス」、「ポート」、「アドレス」をそれぞれ正しく設定して下さい。

GPIBの場合、ボードが一つであれば、「ポート」は空白で構いません。

シリアルポートの場合、RS-232Cではアドレスは不要ですが、ポートは/dev/ttyUSB0(Linux),/dev/tty.usbserial(Mac)等のデバイスファイルを指定します。RS-485ではスレーブのアドレスを設定して下さい。

TCP/IPの場合、「ポート」に(IPアドレス):(ポート番号)を設定して下さい。

## スカラエントリ



ドライバからレコードされた結果の数値が一覧に出ます。

該当する値をクリックすると、横軸が時間のチャートが現れる。チャートの操作法は、横軸をオートスケール出来ない以外はグラフと同じです。

## テキストライタ

テキストファイルに一行ごとにスペース区切りで、「保存」を指定された値を出力します。改行はLFです。「書き込み」をチェックしないと出力しません。

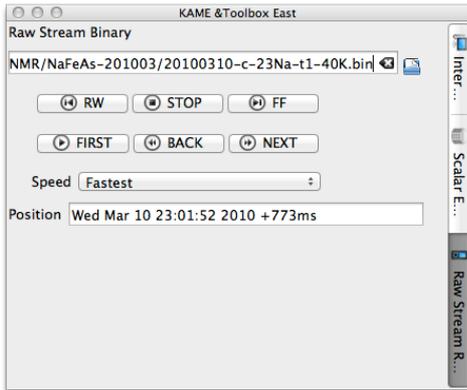
どのタイミングで一行が出力されるかは、「デルタ」で指定します。「デルタ」を正にすると、変化分がその値を超えたときに一行出力します。負にすると、その項目に新しい値が読み込まれる度に一行出力します。

項目の順番を変えるには、番号の上でCTRLキーを押しながらドラッグします。

## ロガー

テキストファイルに一行ごとにスペース区切りで、指定した間隔で全ての値を出力します。

## Rawストリームリーダ



上記のRawストリームレコーダで出力したファイルを読み込んで解析できます。巻戻し機能はバグっているので、何回も解析する場合は、「最初」に戻ってから「送り」で早送りして下さい。早送り速度は「速度」で指定します。「位置」はレコードが記録された時刻を表示しています。

## ドライバ固有の設定

---

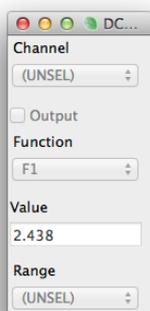
### DCソース

YOKOGAWA 7651 DC source (GPIB)

ADVANTEST TR6144/R6142/R6144 DC source (GPIB)

MICROTASK/Leiden Triple Current Source (GPIB)

7651を使う場合は、他の装置よりも先に開始したほうが無難です。



### デジタルマルチメータ (DMM)

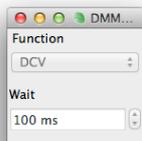
Keithley 2000/2001 DMM (GPIB)

Keithley 2851A nanovolt meter (GPIB)

HP/Agilent 34420A nanovolt meter (GPIB)

HP/Agilent 3458A/3478A DMM (GPIB)

SANWA PC500/510/520M/PC5000 DMM (IR SerialPort)



「ウェイト」は、読み込み周期を設定します。

スカラエントリに値が送られます。

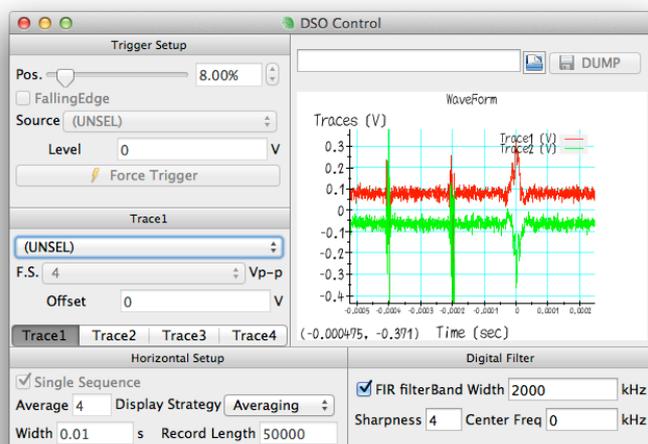
## デジタルストレージオシロスコープ(DSO)

Tektronix DSO (GPIB)

Lecroy/Iwatsu DSO (GPIB)

DSO on NI-DAQ M,S series (NI-DAQmx)

Thamway A/D conversion DV14U25 (USB)



オシロスコープの/DIVでなく、縦軸はフルスケール「F.S.」で、水平軸は全長の「幅」で設定して下さい。

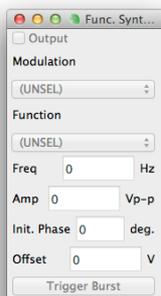
「シングルシーケンス」にすると、「アベレージ」回数分積算してはクリアするの繰り返しになります。

「表示の方法」を「Averaging」にすると適宜表示するので便利ですが、GPIBやCPUに大きな負担をかけます。「シングルシーケンス」モードで「Sequence」にすると指定積算回数に達した時にだけ表示します。

「デジタルフィルタ」、FIR(Finite Impulse Response)フィルタを掛けることができます。「中心周波数」(fc)から「バンド幅」(bw)で指定した帯域をそれぞれのチャンネルに掛けるので、 $fc-bw/2$ から $fc+bw/2$ の周波数を通すようになります。FIRは畳込みで高速に処理されます。

# ファンクションジェネレータ

NF WAVE-FACTORY (GPIB)



# 液面計

Oxford ILM Helium levelmeter (GPIB, SerialPort)

Cryomagnetics LM-500 levelmeter (GPIB)

設定画面はありません。

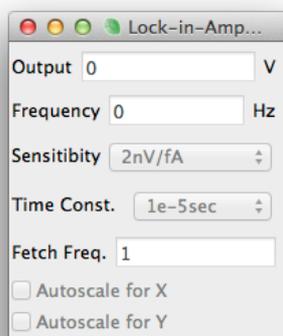
スカラエントリに値が送られます。

# ロックインアンプ / 容量計

Stanford Resresearch SR830 lock-in amplifier (GPIB)

NF LI5640 lock-in amplifier (GPIB)

Andeen-Hagerling 2500A capacitance bridge (GPIB)



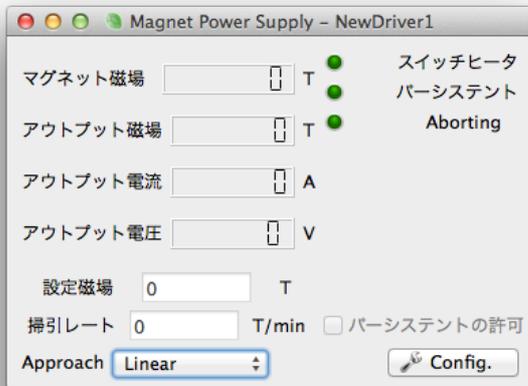
「取り込み頻度」はタイムコンスタントの逆数を基準にしています。

スカラエントリにX/Y値が送られます。

# 超伝導マグネット電源

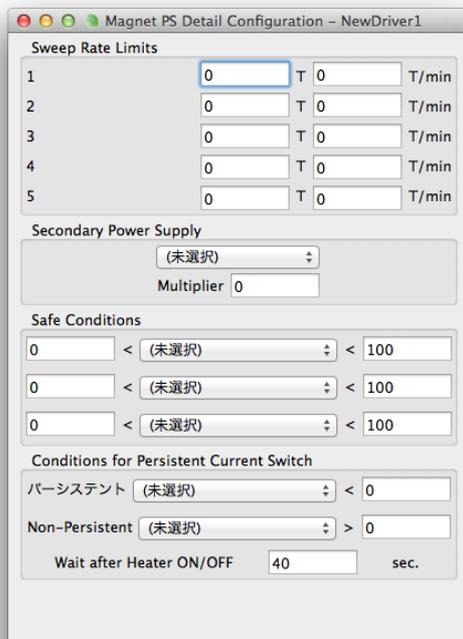
Oxford PS/IPS-120 magnet power supply (GPIB, SerialPort)

Cryogenic SMS10/30/120/150C magnet power supply (USB Serial Port)



パーシステントスイッチヒータの制御は自動で行います。「パーシステントの許可」をチェックすると、磁場が設定値に到達した時にヒータを切ります。"Approach"を"Oscillating"にすると、20%オーバーシュートしながら磁場をスイープします。

「設定」ボタンを押すと次の画面が開きます。



掃引レート及び、最高磁場に制限をかける場合は、“Sweep Rate Limits”に”最大磁場”-”レート”のペアを小さい順に必要な数だけ入れて下さい。

“Secondary Power Supply”はシムコイルの駆動に使います。

“Safe Conditions”は、スカラエントリの値を監視して、範囲から外れると磁場を0にします。

同様に、パーシステントスイッチ用の条件を付加することが出来ます。

“Wait after Heater ON/OFF”に、スイッチヒーターの待ち時間を入れて下さい。

## 標準信号発生器 (SG)

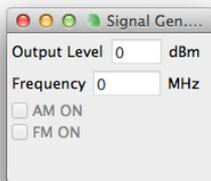
KENWOOD SG7130/SG7200 signal generator (GPIB)

DSTech. DPL-3.2XGF (SerialPort)

HP/Agilent 8643/8644/8648/8664/8665 signal generator (GPIB)

Rhode-Schwartz SML-01/02/03/SMV-03 signal generator (GPIB, SerialPort)

Thamway NMR PROT built-in signal generator (TCP/IP, GPIB, USB)



PROTの場合は、ゲイン等のNMR設定が制御可能です。

## ネットワークアナライザー

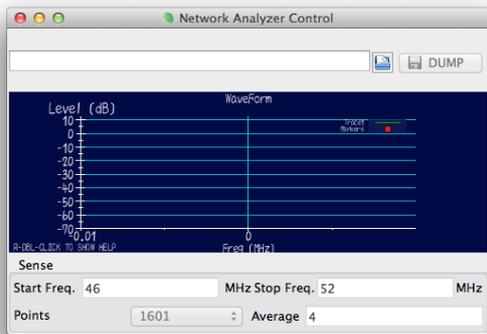
HP/Agilent 8711/8712/8713/8714 network analyzer (GPIB)

Agilent E5061/E5062 network analyzer (GPIB)

DG8SAQ VNWA3E w/ custom dll network analyzer (TCP/IP)

Thamway T300-1049A Impedance Analyzer (SerialPort)

NMR Built-in network analyzer (NMRPulser+SG+DSO)



スカラエントリにマーカーの値を出力します。

## 温度計／温調器

Cryocon M32/M62 temperature controller (GPIB)

LakeShore 340 temperature controller (GPIB)

LakeShore 370 temperature controller (GPIB)

Picowatt AVS-47 bridge (GPIB)

Oxford ITC-503 temperature controller (GPIB, SerialPort)

Neocera LTC-21 temperature controller (GPIB)

Keithley 2700 DMM w/ 7700 scanner (GPIB)

OMRON E5°C controller (Serial Port Modbus RTU 57600bps)



「温度計」を選択した場合、kameが抵抗値等から温度に変換します。温度計の校正曲線は「キャリブレーションテーブル」タブで確認できます。

「外部機器」を選ぶと、そのDCソースまたは流量制御弁を用いてPID制御を行います。

スカラエントリに値（抵抗値等と温度）が送られます。

表示はされていませんが、"Stabilized"というノードに「設定温度」からの「現在の温度」のエラーを一定時間平均した値が保存されています。

## カウンター

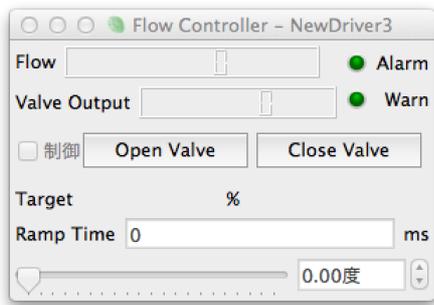
Mutoh NPS digital counter (SerialPort)

設定画面はありません。

スカラエントリに値が送られます。

## 流量制御

Fujikin FCST1000 Series Mass Flow Controllers (Serial Port 38400bps)



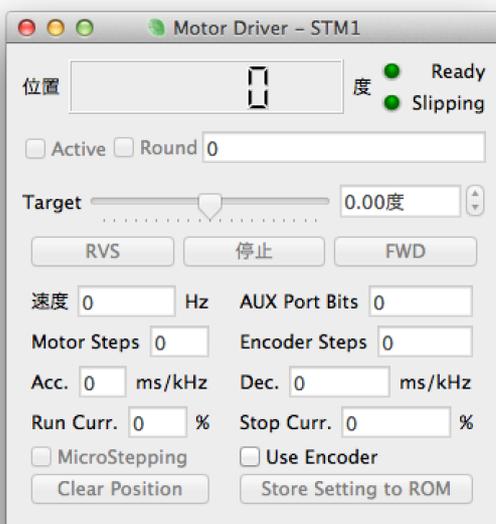
マスフローコントローラドライバです。

スカラエントリに値が送られます。

## モーター制御

OrientalMotor FLEX CRK motor controller (Serial Port Modbus RTU 57600bps)

OrientalMotor FLEX AR/DG2 motor controller (Serial Port Modbus RTU 57600bps)



ステッピングモータードライバです。

「Active」で励磁のON/OFFを設定します。

スカラエントリに値が送られます。

## NMR Pulser

NMR pulser on NI-DAQ M series (NI-DAQmx)

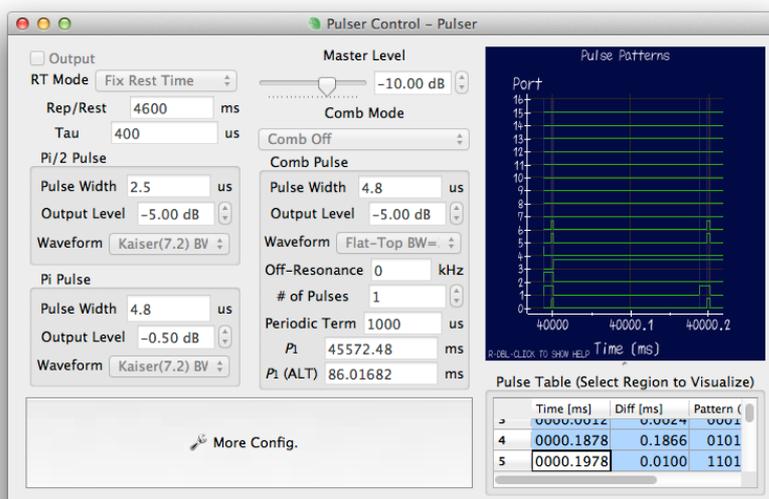
NMR handmade pulser on H8 (SerialPort)

NMR handmade pulser on SH2 (SerialPort)

NMR pulser Thamway N210-1026 PG32U40 (USB)

NMR pulser Thamway N210-1026S/T (GPIB, TCP/IP: experimental)

See appendix for DAQmx



The pulser drivers without featuring QAM cannot change the items of intensities and waveforms.

Digital waves will appear right side.

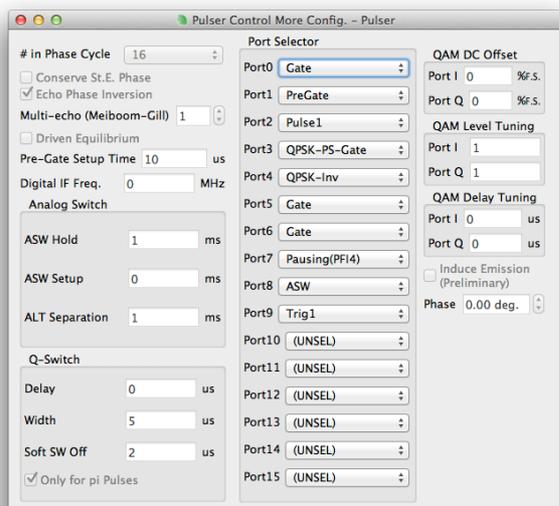
When "RT mode" is "Fix Rep. Time", the total period between the 1st (Pi/2) pulse is fixed to "Rep/Rest" time. Otherwise, ("Fix Rest Time"), the clearance time between the last pulse or  $2\tau$  to Saturation (Comb) Pulse is fixed to the described value.

The times (suchlike "Tau", " $P_1$ ") are defined to the center of pulses, in principle. For "Comb pulse", the center of the last saturation pulse is considered.

When "Comb mode" is "P1 ALT", it uses two  $P_1$  values alternatively. In that case, two FID/echo drivers are necessary to acquire  $T_1$ . Hardware gating using "ASW" signal is necessary.

Seemingly, "Comb ALT" mode alternatively issues the pulse sequence with and without saturation pulses.

The following window appears when "More Config." button is invoked.



“# in Phase Cycle” determines the cycles for  $\pi/2, \pi$  pulses up to the value of 4, and in addition, the phases for comb pulses can be altered when the # > 4.

“Conserve St.E. Phase” allows to measure Stimulated Echo with phase cycling. For this, use two pulses for the comb, and the period of the comb should be the same of  $\tau$ , and width of  $\pi/2$  pulse is zero, and the width of  $\pi$  pulse should be same of that of comb.

“Multi-echo” more than 2 uses Carr-Purcell-Meiboom-Gill (CPMG) cycles. It features phase cycling even for CPMG. For this, you have to do proper setting in the FID/echo driver.

“Port selector” is used for associating the hardware ports and the functions.

“Gate” is gating.

“PreGate” is gating with a setup time for a power amp.

“Trig1” is a trigger rising at  $2\tau$ .

“Trig2” is a trigger rising at  $\pi/2$  pulse and falling at  $\pi$  pulse.

“ASW” is gating for the receiver.

“QSW” becomes H level only after the pulse. This is for Q switch.

“Pulse1” is identical to  $\pi/2$  pulse.

“Pulse2” is identical to  $\pi$  pulse.

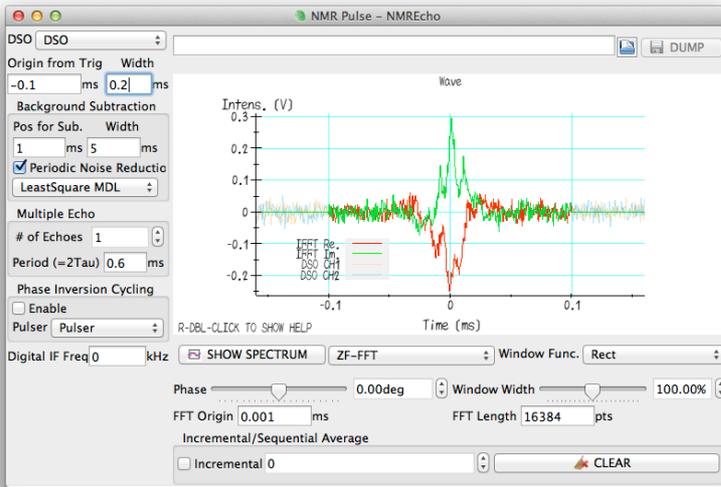
See QPSK logics below:

(Phase)	“QPSK-A”	“QPSK-B”	“QPSK-NonInv”	“QPSK-Inv”	“QPSK-PS-Gate”
0°	L	L	H	L	L
90°	H	L	H	L	H
180°	H	H	L	H	L
270°	L	H	L	H	H

Thamway receiver would use a pair of “QPSK-A”, “QPSK-B” or “QPSK-PS-Gate”, “QPSK-Inv”

## NMR FID/Echo measurement

This driver is an analyzer, which subtracts a background level from the DSO wave, cuts it using a window, sums up, and makes a frequency-domain spectrum.



For ordinary echo measurement, set the trigger of DSO to  $2\tau$  (= "Trig1"), and use "Single Sequence" mode for DSO.

It cuts the wave of DSO from the "Origin from Trig." by "Width". But for "Multiple Echo", displayed waves are averaged ones for "# of Echoes" times.

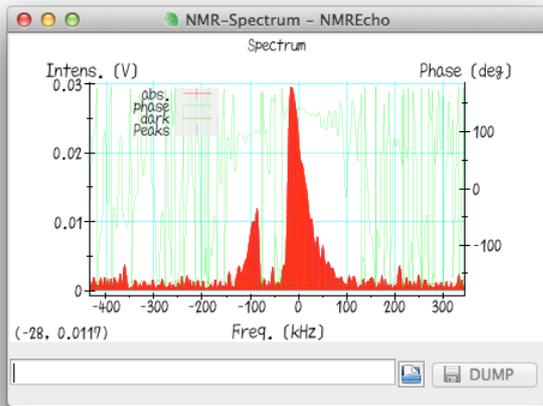
When "Width" of "Background Subtraction" is not zero, it displays a subtracted data using the average level from "Pos for Sub." with "Width". For better S/N, you have to use much larger "Width" than that of "Width" for echoes.

If "Period Noise Reduction" is activated, a least square fitting for periodic noise in the background part is attempted to extrapolate the noises. The number of noise frequencies are estimated by the information criterion. This function requires a lot of CPU powers.

When the "Phase Inversion Cycling" is on, it averages a positive and negative echoes alternatively after an inversion for the latter. This procedure allows to cancel out the time-dependent background.

When the "Incremental" is activated, once the stored wave is cleared and it averages all waves forever unless you invoke "CLEAR" button. The number box displays the average count.

Clicking "SHOW SPECTRUM" puts the following window.



Usually we use "ZF-FFT" (Zero-Filling Fast Fourier Transform) mode. The origin for the FFT is given by "FFT Origin", which is counted from the trigger. Of course, you can use "Window func."

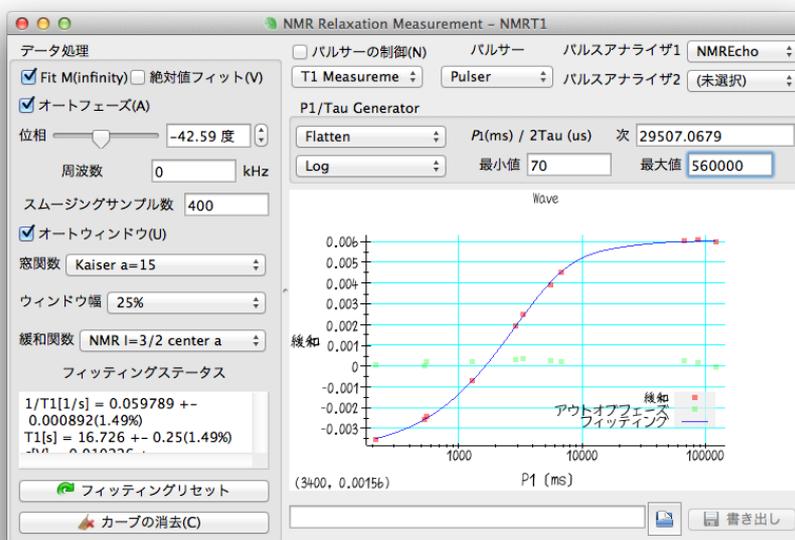
The red bars are the absolute in the complex FFT. The phase (green) should be flat if the "FFT Origin" is proper. "dark" is a noise level which is estimated by the background. "Peaks" are maxima in "abs."

The displayed "IFFT Re. / IFFT Im." in the main window are IFFT result for the spectrum analysis. In ZF-FFT, it is nothing but a wave after windowing. "DSO Re. / DSO Im." is the waves of DSO after the background subtraction.

In the scalar entries, the maximum value and its frequency is emitted.

## NMR Relaxation Rate Measurement

The driver is analyzer, which measures relaxation rates using waves in FID/Echo drivers. Of course the pulser is controlled by this. Configurations for window function are not shared with FID/Echo drivers, and can be freely changed during your measurement.



Invoking “Control Pulser” first clears the results and new  $P_1$  value (written in “Next”) will be transferred to the pulser driver when a new data come in. When the generator is “Random”,  $P_1$  will be randomly chosen between “Min” and “Max”. The “Flatten” generator tries to find the most vacant region. “Log” distribution is standard for the generators. You can change “Next” value for  $P_1$  at any time.

“Fitting Status” summaries a result of a least square fitting for a relaxation curve.

“Auto Phase” choses automatically the phase that can maximize the difference in the curve.

“Smooth Samples” specifies a number of regions for the truncation for  $P_1$  before fitting.

“Auto Windows” choses automatically the window function that can maximize S/N. It is not suited for an experiment where a frequency dependence of  $T_1$  is physically important.

“Fit M(infinity)” adds a fitting parameter for a value at  $t \rightarrow \infty$ . This should be turned off for  $T_2$  measurement.

“Frequency” is a center frequency for the relaxation analysis. Change of this entity clears the stored result.

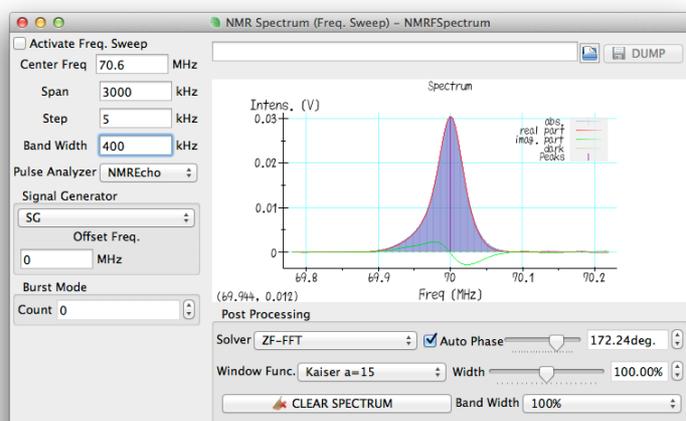
“Reset Fitting” could be used when the fitting is not well functioning due to the initial values.

$1/T_{1,2, \text{st.e.}}$  values and errors will be emitted to the scalar entries.

## NMR Spectrum Measurement (Frequency Sweep)

This driver is an analyser which measures frequency-swept spectrum using waves in FID/Echo driver. Configurations for window function are not shared with FID/Echo drivers, and can be freely changed during your measurement.

Fourier Step Sum(FSS) technique, which sums ZF-FFT results by changing the center frequency, is employed. The internal memory for the summation is in time domain, which enables you to select window function any time.



“Activate Freq. Sweep” first clears the stored result and start changing a frequency of the SG from “Center Freq.” - “Span”/2 by every “Step”.

“Auto Phase” choses the phase that can maximize the overall real part. However, usually the absolute part should be relevant.

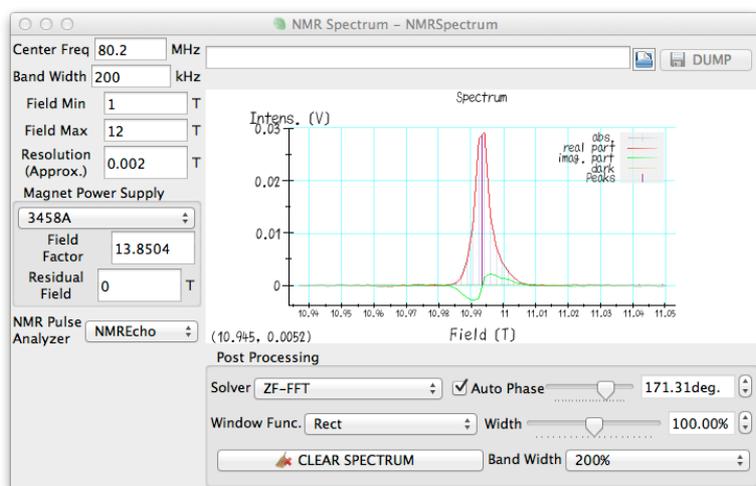
You have to consider “Band Width” for FSS before starting measurement. But you can choose it by multiplying “50%”, “100%”, “200%” anytime.

“LC tuning” determines a method for tuning during the sweep. “As is” does nothing. “Auto Tune” employs an auto-tuner driver and tunes the circuit every “Step”. “Await” turns off the pulser every “Step”, asks you, and waits until you turn on the pulse again after your manual tuning.

## NMR Spectrum Measurement (Field sweep)

This driver is an analyser which measures frequency-swept spectrum using waves in FID/Echo driver. Values of the magnetic field is taken from a magnet power supply driver or DMM driver. Configurations for window function are not shared with FID/Echo drivers, and can be freely changed during your measurement.

Fourier Step Sum(FSS) technique, which sums ZF-FFT results by changing the center frequency, is employed. The internal memory for the summation is in time domain, which enables you to select window function any time.



See also the frequency sweep driver.

Internally, the center of the frequency is calculated as  $-\log(H) \times \text{“Center Freq”}$  during FSS. It is justified as long as the resonance frequency is proportional to the field, or the case “Center Freq”  $\gg 0.001 \times \text{“Band Width”}$ .

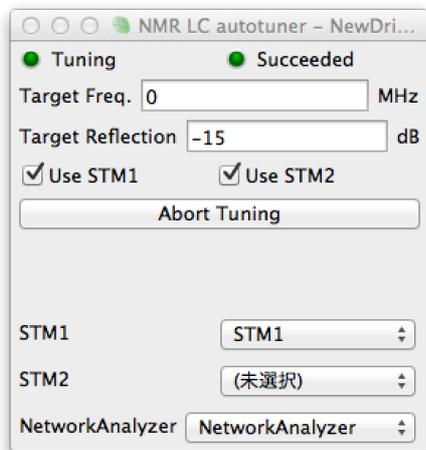
The actual field is calculated as (raw value)  $\times \text{“Field Factor”} + \text{“Residual Field”}$ .

You must sweep the magnetic field externally somehow.

“Resolution (Approx.)” specifies a typical resolution of the horizontal axis. Change of this entity may clear the result.

# NMR LC autotuner

This driver is an analyzer, which tunes the LC circuit by monitoring a vector network analyzer driver. Before and after the tuning, it inverts the bits of "AUX" in the motor drivers, which can be used to control an RF relay.



## 電流反転抵抗測定

このドライバは、解析専用で、DCソースを反転させながら、DMMからの電圧データの差分を取ります。



スカラーエントリに値が送られます。

## スクリプトによる制御

Ruby言語による制御が可能です。kame内部では、Ruby専用のスレッドが常時走っており、モニタープログラム(kame/script/rubysupport.rb)がさらにRubyのスレッドを新しく作ってスクリプト(.seq)を読み込み実行します。

標準入力と標準出力はリダイレクトされてkameの中央部に表示されます。文字コードはUTF-8です。

エラーが起これば、バックトレースが赤色で表示されます。

Mathモジュールは標準で読み込まれています。

kameとの情報のやり取り／制御は、XNodeクラスのオブジェクトを介して行います。それらオブジェクトはツリー構造になっており、Measurementオブジェクトがルートです。

マウスポインタを制御対象に置くと“Node Browser”タブにRubyでのオブジェクト名とメソッド一覧及び現在の値が表示されます。

XNodeクラスは、Arrayと似たようなメソッドをサポートしています。

例えば、Measurement["Drivers"]["NMRT1"]["P1Min"]で制御対象のオブジェクト（この場合はXDoubleNodeクラス）にアクセスできます。

XValueNodeクラス及びその子クラスは、value=メソッドで代入、valueメソッドで値の取得が可能です。

Rubyによる制御はわずかなタイムラグがありますので、連続して値を代入する場合は適当なスリープを入れて下さい。

## 例（各温度でスペクトラムを保存する）

```
def wait_within(node, val, min_wait, timeout, incr = 1)
  tstart = Time.now
  print "Wait for stabilize '#{node.name}' within #{val}"
  sleep min_wait
  while tstart + timeout > Time.now
    if node.get().abs <= val then
      print "OK. #{Time.now - tstart} sec. lost"
      return true
    end
    sleep incr
  end
  print "Time out #{Time.now - tstart} sec."
  false
end

tempctl = Measurement["Drivers"]["TempControl2"]
pulser = Measurement["Drivers"]["Pulser"]
pulse = Measurement["Drivers"]["NMREcho"]

for temp,rept in [[10,100],[20,50],[40,25]]
  print "T=#{temp}K\n"
  pulser["RT"].value= rept
  pulser["Output"].value= true
  tempctl["TargetTemp"].value=temp

  pulse["ExAvgIncr"].value= false
  sleep(3)
  pulse["ExAvgIncr"].value= true
  pulse["Spectrum"]["FileName"].set("hogehoge-#{temp}K-ft.dat")
  slp = [300.0, rept*100/1000.0].max()
  sleep(slp)
  pulse["Spectrum"]["Dump"].touch()
end
```

```
wait_within(tempctl["Stabilized"], temp/100.0,30,36000)
end
```

## FAQ

---

### Fedoraインストール直後にネットワークが使えないときは？

「管理」の中の「ネットワーク」を起動して、デバイスをダブルクリックしてダイアログを出して、その中の「Network Managerで管理する」、「コンピューターの起動時にデバイスを起動」をチェックして、再起動して下さい。「全てのユーザーに...許可」もチェックしてもいいかもしれません。

### Fedoraの「ソフトウェアの更新」が失敗するが？

GUIを諦めてsuでyum upgradeして下さい。そっちの方が速いです。

### LinuxでDAQmxを使っていて、頻繁にフリーズするときは？

現在のNational Instruments社DAQmx8.0.2ドライバでは、以下の制限事項というかバグがあります。  
32bit x86のみかつ、4GB以下のメモリ  
カーネルスタックサイズは8kB必要  
IOMMU (IO仮想化, VT-d等) は使えない  
SE Linuxの設定を行なってくれない。

これらを守らないと、突然フリーズします。他にも、マルチスレッド対応が完全でないようなので、問題があれば、CPUコア数を1個にしてみてください。

### GPIBで通信エラーが起こるときは？

よくあるのは、アドレスが間違っているとか、ケーブルが長すぎるとかですが、まれに機器間の相性問題があります。Oxford社全般とYOKOGAWAの7651、cryoconは、IEEE488規格に準拠していないので、非常に沢山の機器をつないだ場合に問題が起こることがあります。装置を減らすか、一部をシリアル通信に切り替える等しましょう。

GDBからkameを動かしているとエラーが出やすいです。linux-gpibのスレッド関係のバグです。

### シリアルポートの設定は？

特に指定しない限り、9600bpsノンパリティです。

### シリアルポートが開けない時は？

パーミッションをまず確認して下さい。

“Line in use”のエラーが出てmodem-managerが邪魔している場合は、NetworkManagerを停止・OFFにして、su -c 'killall modem-manager'としてください。

## kameで対応していない機器を使いたい時は？

ドライバを作成する必要があります。ソースコード中のmodules以下のフォルダを参考にして作ることができますが、難しければ北川にコンタクトしてみてください。デジボルの機種を追加する程度であれば僅かな作業で出来ます。

## 付録

---

### NI製DAQmxデバイスを用いたNMRシステム(パルサー及びオシロスコープ/アベレージャ)

私は、SシリーズのPCI-6111 (2ch12bit5MSps) をA-D変換器として利用しています。変調にQAMを使用する場合は、D-A変換部分も使います。上位機種PCI-6115も利用できます。

内部動作としては、リアルタイムに全てリングバッファに取り込んで、ソフトウェアで積算しています。従って、いかなる繰り返し時間でも、取り込み範囲が重ならない限りは取りこぼしは有りません。また、積算回数に制限はありませんが、DSOドライバでは32bitで積算しているので65536を超える積算は、NMRエコードライバとの組合せで行なって下さい。

パルサーとしては、MシリーズのPCIe-6251(DIO 8ch10MHz)を使います。PCI-6220も遅いですが、使えます。PCIe-6251はディスコンになりますが、後継機種PCIe-6351についてはまだ動作確認していません。

その他必要な外部端子等としては、

PCI-6111用に  
SH68-68-EP 1個  
BNC-2110 1個

PCIe-6251用に  
CB-68LPR 1個  
SHC68-68EPM 1個  
CA-1000 Box 1個  
CA-1000 ラックマウント 1セット  
CA-1000 BNCパネル 4個  
CA-1000 カバー 5個

を使っています。

また、2つのボード間のクロック・トリガー同期用にRTSIケーブルの接続が必要です。

CA-1000 BNCパネルは絶縁BNCを使っていますが、これを非絶縁BNCに交換しないと、グリッチがNMRエコーに混じって観測されてしまいます。

## Sシリーズデバイス

AI0,1(2ch)またはAI0-3(4ch)がA-D変換用の入力端子です。NMRではレシーバーのPSD出力に繋がります。0度と90度のポートが逆になってレシーバーもあるので注意。

ドライバで”National Instruments DAQ as DSO”を選択してください。

モジュレーターにQAMを使用する場合、パルサードライバとして、”NMR pulser NI-DAQ M Series with S Series”を選択して、2つ目のインターフェースにSシリーズを選んで下さい。AO0,1をQAMに繋がります。

## Mシリーズデバイスのパルサー用接続

QAMを使用しない場合、パルサードライバとして、”NMR pulser NI-DAQ digital output only”を選んで下さい。

SシリーズをDSOとして使用する場合、DSOの「トリガー設定」で「ソース」にパルサーの適切なラインを選んで下さい。これはソフトウェアでトリガーが掛かるので、物理的に存在しないポートでも構いません。お勧めは、ポート8を”Trig1”に設定して、「ソース」を”Pulser名/line8”にすれば、 $2\tau$ の位置にトリガーが掛かります。ソフトウェアでトリガーをかけている都合上、DSOの縦横軸の設定を変更した後は、パルサーをOFF/ONしなさいとトリガーがかかりません。

どんなパルサーにも言えますが、出力中に電源OFFした機器に繋いだり50 $\Omega$ 終端するのは止めましょう。ショートしているのとほぼ同じ事になります。

OFF時にはオープンステートにしますので、GATEとPRE-GATEのポートは数k $\Omega$ 程度でGNDに落として下さい。

以下の写真を参考に接続して下さい。

ポート7は、内部動作に予約しています（設定で変更できますが）。パルサーに信号変化がないときにカウントを停止させてボードを休ませる用途に使っています。その際、P0.7(J48)とPFI4(J41)を接続して下さい（写真中青い線）。また、NMRパルサーの設定画面の「もっと設定」内でポート7を”Pausing(PFI4)”に設定して下さい。

接続表は以下の通りです。

（ポート）（DAQポート名）（端子番号）

ポート0 P0.0 J52

ポート1 P0.1 J17

ポート2 P0.2 J49

ポート3 P0.3 J47

ポート4 P0.4 J19

ポート5 P0.5 J51

ポート6 P0.6 J16

ポート7 P0.7 J48 / PFI4 J41

GND D-GND 4,7,9,12,13,15,18,35,36,44,50,53のいずれか

