VIAQS

Overview and setup

What is VIAQS?

- A colaborative open-source project.

- A hardware and software solution for controlling instrumentation.

- A very easy to use way to get data into EPICS, Labview and Web clients.

What instrumentation can it support?

- Beam position monitors
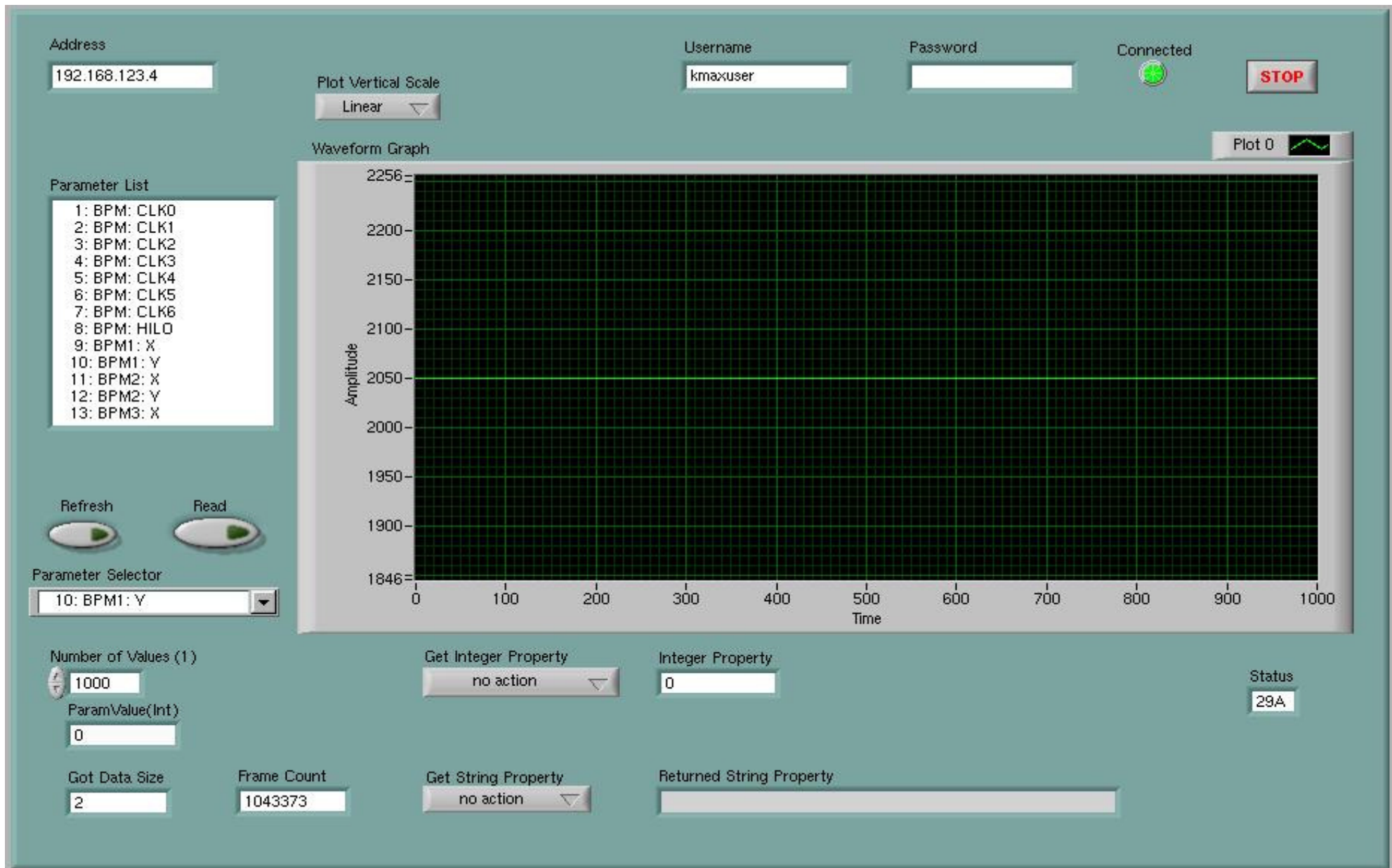- Current transformers
- Loss monitors

- ......

What platform does it run on?

- It runs on Linux (No licenses!)

- Uses widely available I/O cards for easy maintenance

Is it easy to use?

- You can get a low cost pre-loaded system

- Typically this then does not require any setup!

- To try it out - plug it in, attach screen and keyboard, and you can see your data.

- Menu driven configuration if you wish to modify network settings, device names or calibration values

- Examples are given of medm and labview screens

Example LabVIEW screen

Example medm screen

VIAQS medm Demonstration Page

NAME	X	Y
BPM1	-9.376	-9.372
BPM2	-9.369	-9.378
BPM3	-9.378	-9.378
BPM4	-9.954	-9.381
BPM5	-9.378	-9.378
BPM6	-9.369	-9.379
BPM7	-9.380	-9.380
BPM8	-9.380	-9.381

bergoz

Precision Beam Instrumentation

Example web screen

VIAQS http Demonstration Page

Beam Position Monitor (BPM)

Saturday 20th of November 2004 01:11:54 PM

ID	Owner Name	Param Name	Value	Frame
1	1 BPM	CLK0	0	0
2	2 BPM	CLK1	0	0
3	3 BPM	CLK2	0	0
4	4 BPM	CLK3	0	0
5	5 BPM	CLK4	0	0
6	6 BPM	CLK5	0	0
7	7 BPM	CLK6	0	0
8	8 BPM	HILO	0	0
9	9 BPM1	X	2043	2587824
10	10 BPM1	Y	2059	2587824

Can I get the source code?

- YES – it is an open system
- All code is available
- It is released under the GPL license

What data does it typically provide?

- Last value of a reading
- Array of the last n values
- FFT of latest values
- Arrays of latest values of all sensors

Can I extend the system

- It is designed to be a general purpose extend able system.

- It is easy to add user defined functions for data processing or data reduction on the server.

How do I get it

- Purchase a pre-configured turnkey solution.
- Build it yourself !

Can I get support?

- Email support is available
- Specialists can be contracted to provide custom extensions.
- Training is available for advanced users.

The end

Questions?

Viaqs in more detail

Getting the code

- From the viaqs home page www.viaqs.com
- From sourceforge www.sf.net/projects/viaqs
- From cvs (anonymous@cvs.sf.net:/cvsroot/viaqs)

    – viaqs – contains core viaqs code

    – doc – contains documentation and examples

    – ui_viaqs – contains code for setup menus

    – epics_viaqs – contains viaqs drivers for epics

    – php_viaqs – contains code to access viaqs from web

'viaqs' – the core -directories

- bin (linux binaries)
- lib (linux shared object libraries)
- include (include files)
- developer
  - startvaiqs (the runtime 'main loop')
  - viaqslib (libraries to access the data)
  - console (low level debug program)

Epics Viaqs

- lairApp (device suport for reading viaqs)
- joinApp (record + device support for joinArray)
- maxApp (record + device support for maxArray)
- CoApp (ioc application for closed orbit)

ui_viaqs – menus and utilities

- start_menu – main menu program (perl)
- Scripts – directory containing scripts
  - view_net – shows network config
  - set_dhcp – sets system to use dhcp
  - gw_sub – sets gateway address
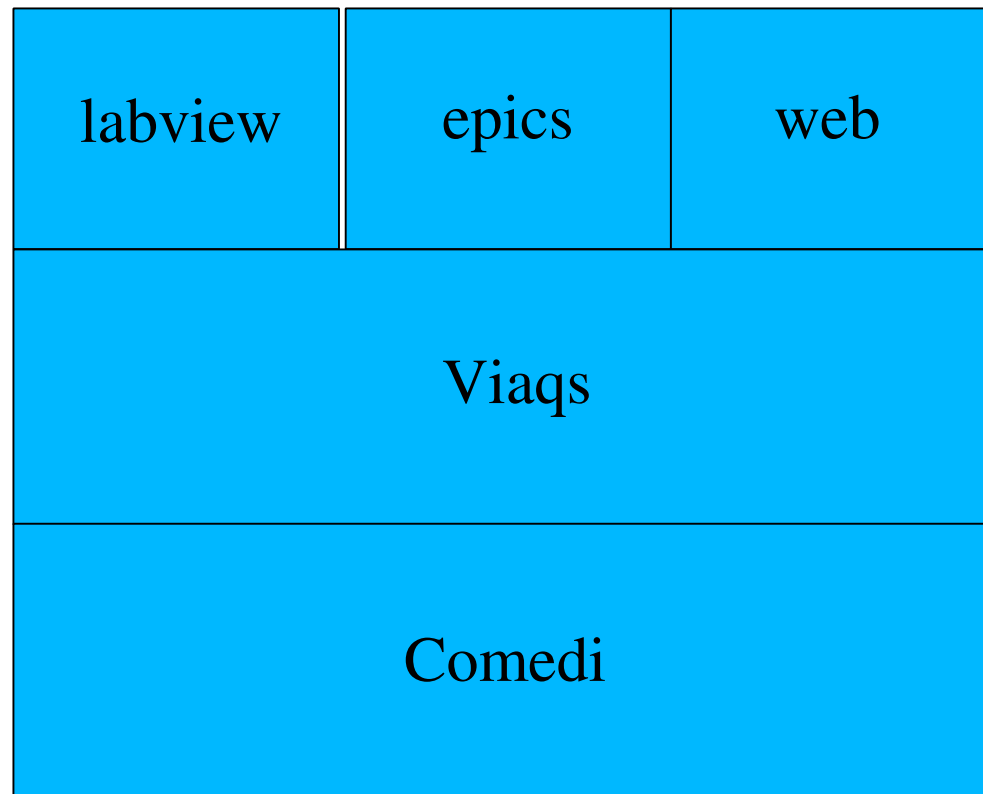 - epics_sub – set epics device names etc
  - ....

php_viaqs

- VIAQS.c – code for php to access viaqs
- config.m4 – configuration
- php_viaqs.h – heaer file

This code needs to be built in you apache source tree – under extensions – look into php documentation for details.

An example php screen is in the doc directory

Software Layout

labview | epics | web

Viaqs

Comedi

Starting viaqs (init.d ?)

modprobe ni_pcimio

comedi_config /dev/comedi0 ni_pcimio

export LD_LIBRARY_PATH=/usr/local/lib:/root/viaqs/viaqs/lib:/usr/local/epics/base3.14.4/lib/linux-x86

/usr/local/bin/comedi_calibrate

cd /root/viaqs/viaqs/bin

screen -d -m ./startviaqs config/coConfig.conf

/usr/local/apache2/bin/apachectl -k start

cd /root/viaqs/epics_viaqs/iocBoot/iocco

screen -d -m ./st.cmdcd /root/viaqs/viaqs/bin

cd /root/viaqs/viaqs/bin

screen -d -m /usr/lib/j2se/1.4/bin/java -jar Server.jar

Checking system

- Screen -r should list 3 running screens

  - Connect to each (startviaqs, epics, labview server) and check for errors

  - If not three stop them and start pocesses by hand

- Connect to web server – check web server running

Getting help

- Submit bug reports via www.sf.net/projects/viaqs

- Training courses are available

- Consider buying one turnkey system to get you started

The end

Thank you