SPEC - A Physicist's and User's Perspective

Stuart B. Wilkins

Condensed Matter Physics and Materials Science Department

4/19/2010

BROOKHAVEN NATIONAL LABORATORY

a passion for discovery

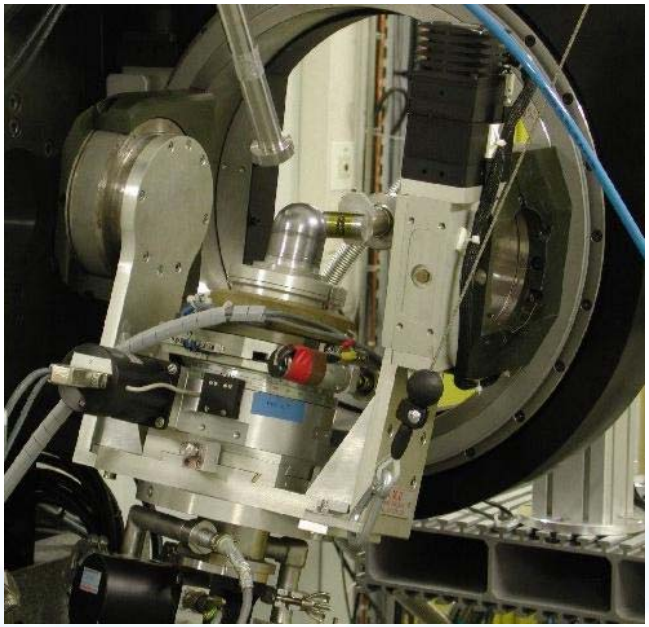
Office of Science
U.S. DEPARTMENT OF ENERGY

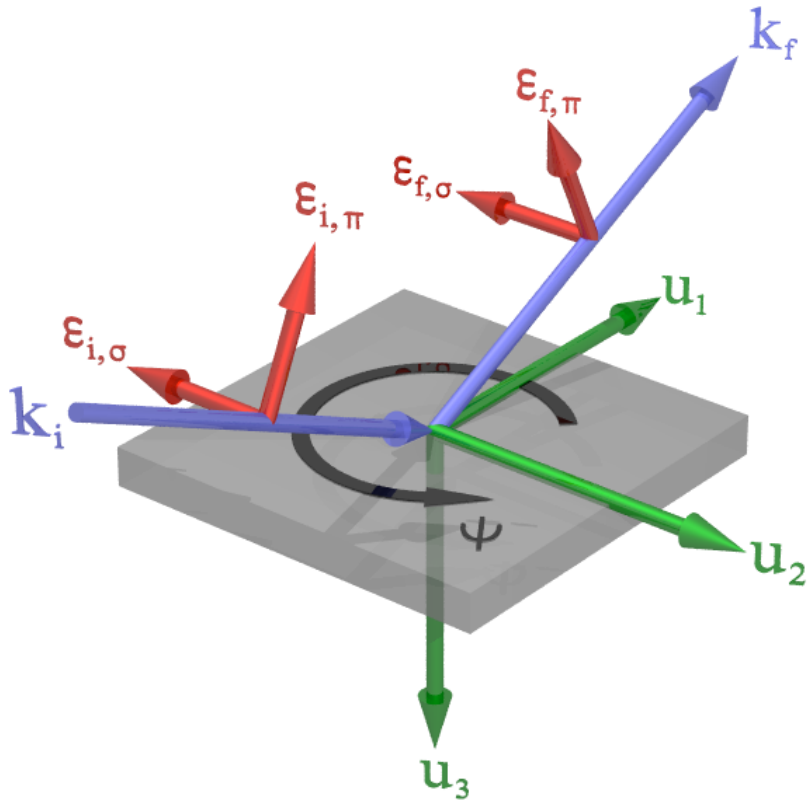
Outline of Talk

- Experiments :
  - Caveat : Condensed Matter Scatterers
  - Example of typical experiments
- Introduction to SPEC .. Interface and inner workings
- Advantages and Disadvantages

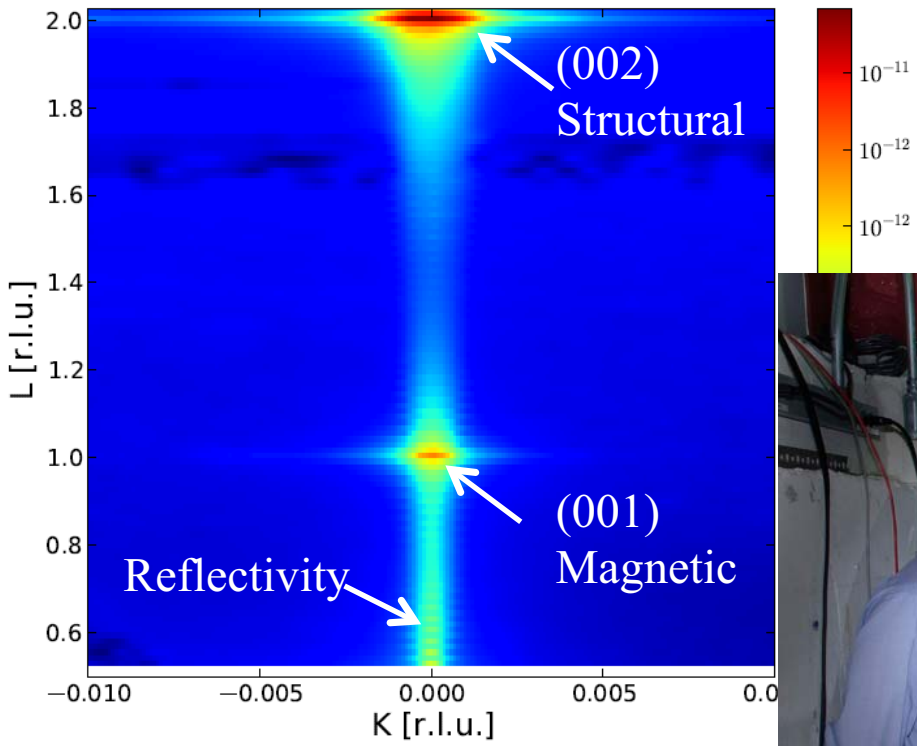
Condensed Matter Scattering Experiments

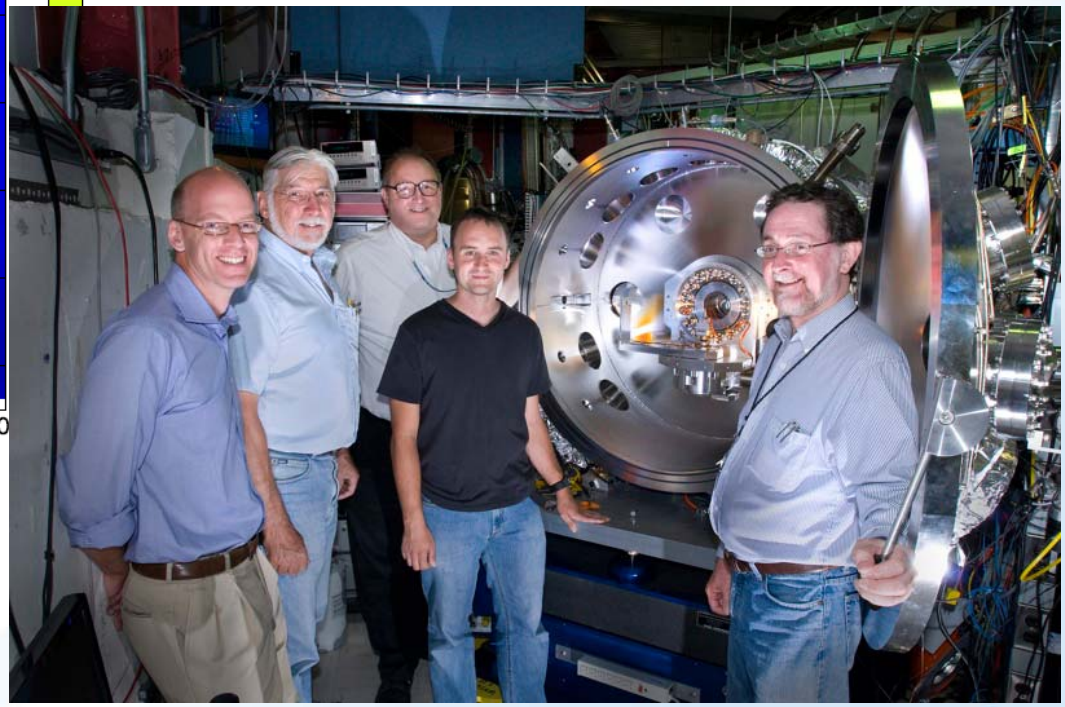
- Complicated scattering experiments, using difractometers with multiple degrees of freedom.





Next Generation Instruments

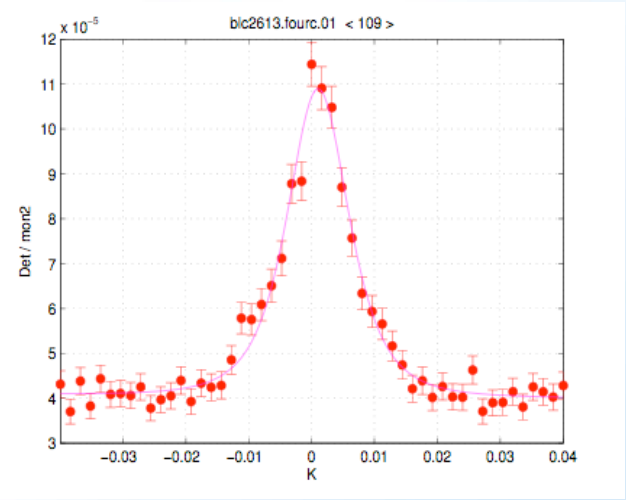




Every experiment is unique....

- No experiment is identical. Sample is not the only thing that changes in a Physics experiment.
- Sample environment.
- In-situ measurements

- Predominantly datasets are either
  - Multiple line scans
  - 2D CCD images

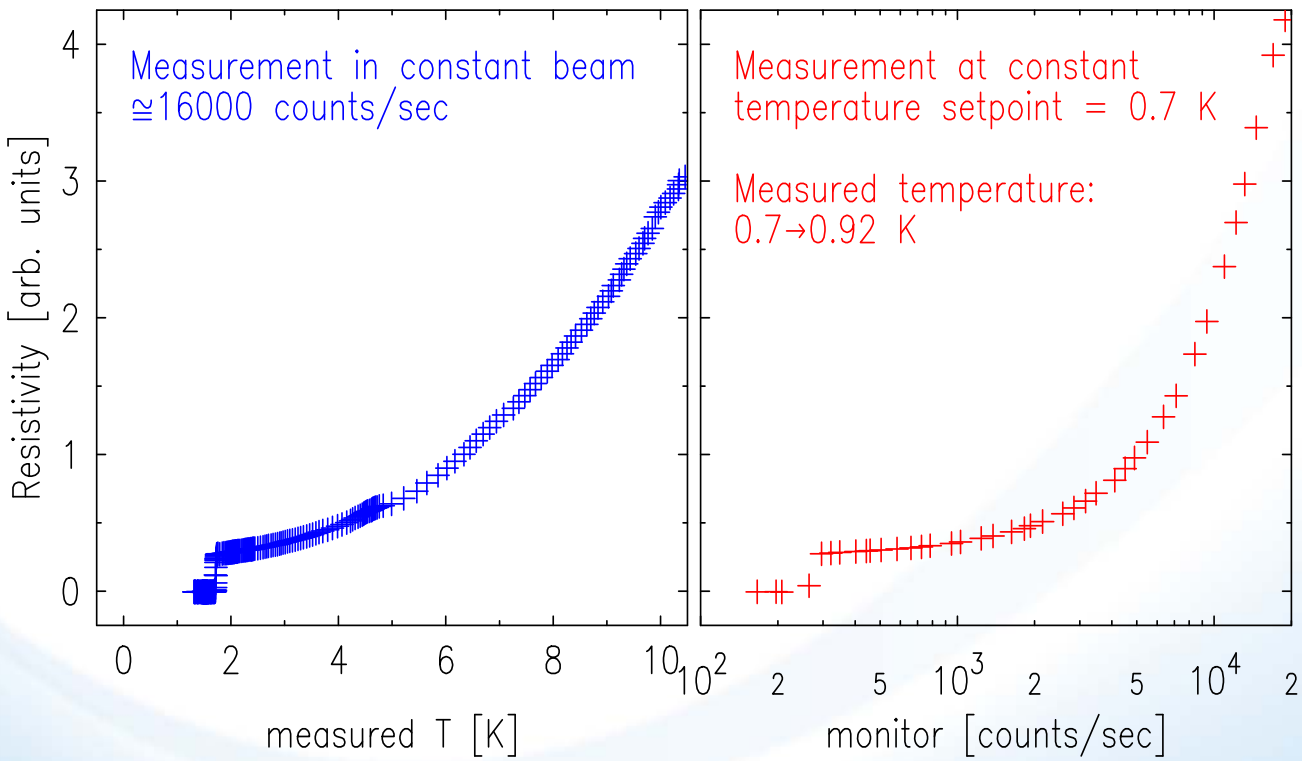


Sample environment is key ...

- Complicated sample environments.
  - Magnetic Field
  - Low temperature
  - In-situ resistivity
  - Electric Field

Why is this important?

- Sample diagnostics. Magnetic scattering UPd_2Al_3



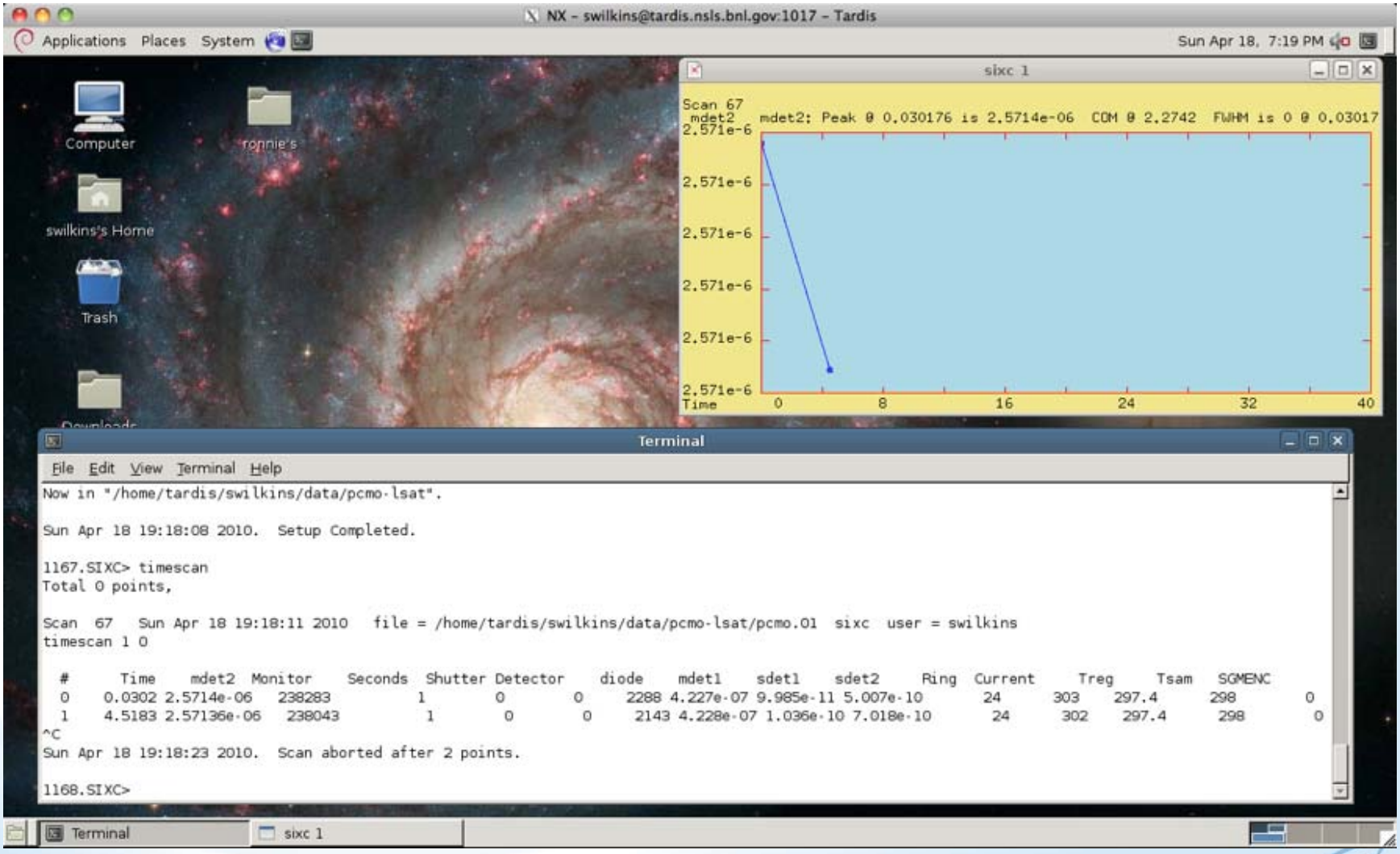
This all leads to ....

- What are the needs of a control software / system?
  - Stable
  - Command line (CLI) based ....
  - Access to beamline motors / counters (EPICS / TACO)
  - Hardware level access to interfaces (SERIAL / GPIB)
  - Scriptable
  - User level virtual motors / counters
  - Hackable!

What is SPEC?

- Spec is a UNIX-based software package for instrument control and data acquisition.
- First produced in 1986..
- Built-in code to regulate motor controllers and detection electronics using VME, CAMAC, GPIB, RS-232, PC-board and ethernet-socket interfaces.
- Talks to EPICS / (TACO / TANGO) for abstracted hardware.
- Command Line Based.
- C-like syntax for “macroable” functions.
- Scans etc. are run in user space.

How does it look?



Geometry code ...

- SPEC Has “built in” geometry calculation code.
- Essential for diffractometer control!
- Developed for many different geometries
  - SIXC
  - PSI-C
  - FOURC
  - SURF

Acta Cryst. (1967). **22**, 457

Angle Calculations for 3- and 4- Circle X-ray and Neutron Diffractometers*

By William R. Busing and Henri A. Levy
Chemistry Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, *U.S.A.*

(*Received* 13 *June* 1966)

Methods are derived for calculations useful in the operation of 3- and 4-circle X-ray or neutron single-crystal diffractometers. These include: (1) establishing the sample orientation from the cell parameters and the observed angles for two reflections, or from the observed angles for three reflections only, (2) calculating the angles for observing a given reflection either in a special setting or at a specified azimuthal angle, (3) obtaining the vectors needed for calculating absorption corrections, and (4) using observations of several reflections to refine cell and orientation parameters by the method of least squares.

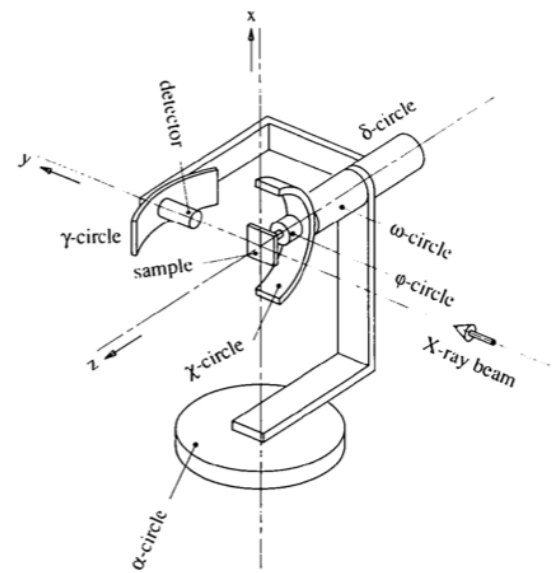


Fig. 1. A schematic drawing of the six-circle diffractometer with all circles set to the zero position.

Macros in user space.

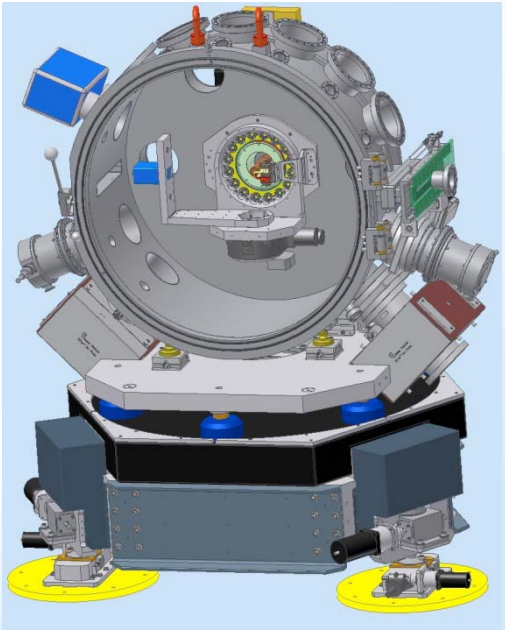
- Spec runs all “scans” and macros in user space.
- Allows for “on-the-fly” different scans
- Allows for complicated “HKL Scans”
- Virtual motors / counters

```
1168.SIXC> prdef ascan

# SPECD/standard.mac
def ascan '
        if ($# != 5) {
                eprint "Usage:  ascan  motor start finish  intervals time"
                exit
        }
        _check0 "$1"
        _m[0] = $1; _s[0] = $2; _f[0] = $3
        _n1 = int($4); _ctime = $5
        _nm = 1
        _ascan
'

swilkins's state is stored

1169.SIXC>
```

“Virtual” devices

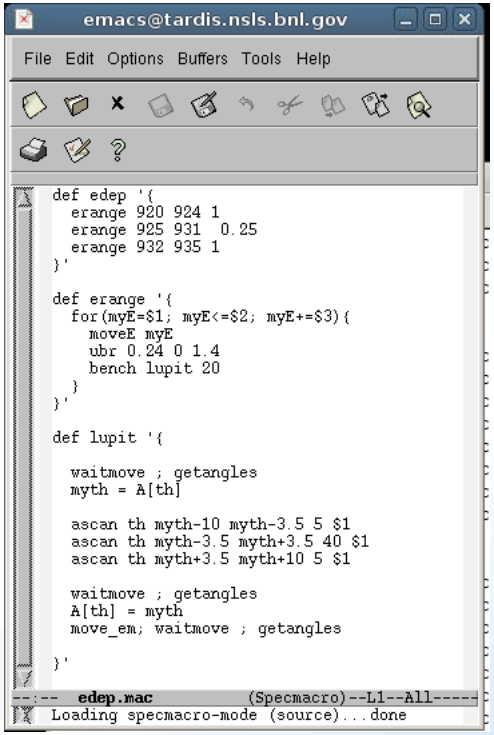
- User space macros allow for virtual counters and motors ..
- Useful for 3-jack tables.
- Diffractometer angles can be formed from multiple motions (rotation / translation)
- Providing these in user-space allows for easy debugging.
- User-space macros can be developed “on the fly” for new measurements

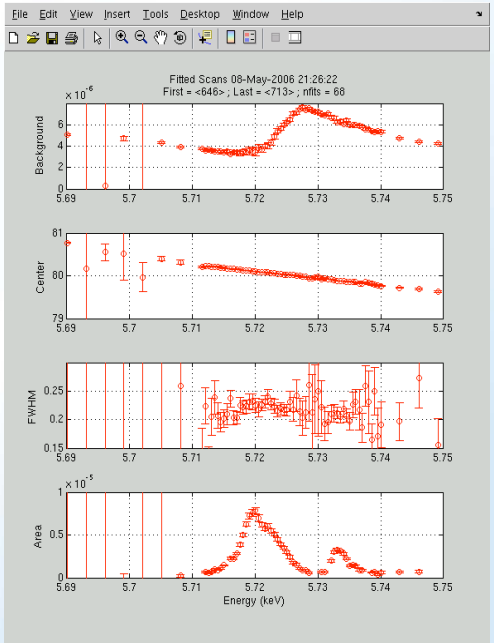
```
1177.SIXC> wdiff
                ZDiff      XDiff      RDiff      PDiff      YDiff
                zdiff      xdiff      rdiff      pdiff      ydiff
   User       -3.8200    -5.8036    -0.0000     0.0000     0.0000
   Dial       -3.8200    -5.8035     0.0000     0.0000     0.0000

swilkins's state is stored

1178.SIXC>
```



User macros.

- User macros can be used for scripting and automatic data collection
- Macros can be simple command files (Thesis advisor mode)
- Complicated macros possible due to access to hardware / motor scaler values





Data Format

NX - swilkins@tardis.nsls.bnl.gov:1017 - Tardis

Applications Places System Sun Apr 18, 10:05 PM

emacs@tardis.nsls.bnl.gov

File Edit Options Buffers Tools Help

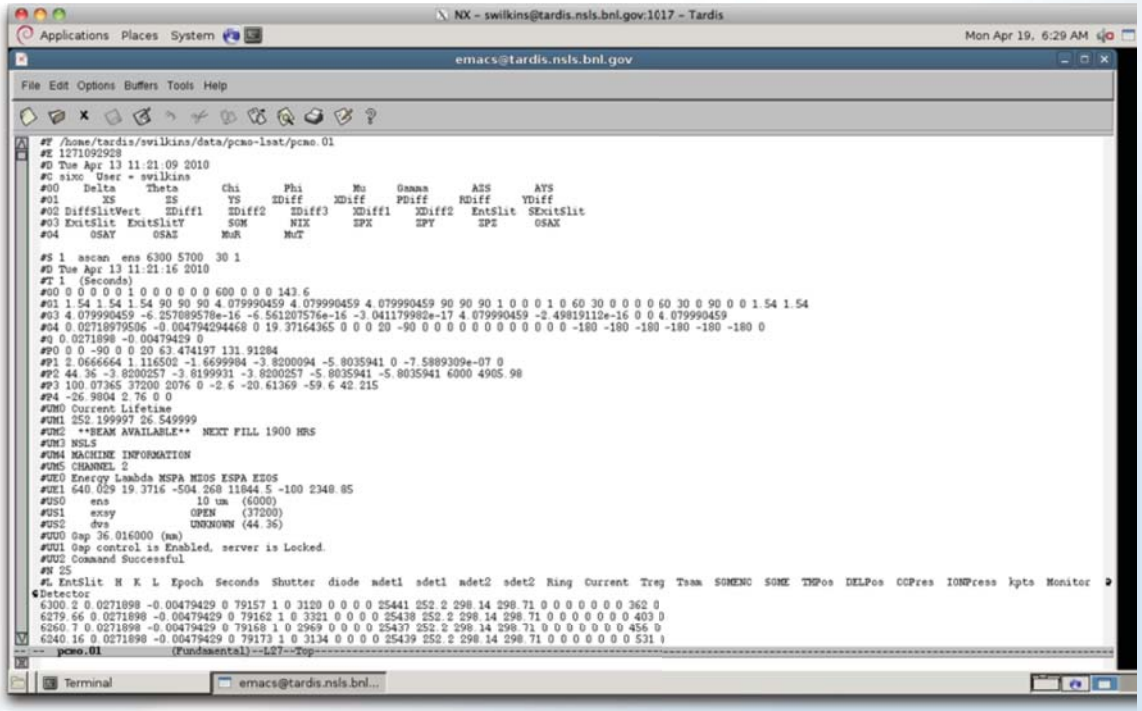
```
#F /home/tardis/swilkins/data/pcmo-lsat/pcmo.01
#E 1271092928
#D Tue Apr 13 11:21:09 2010
#C sixc  User = swilkins
#00    Delta     Theta       Chi       Phi        Mu     Gamma       AZS       AYS
#01       XS        ZS        YS     ZDiff     XDiff     PDiff     RDiff     YDiff
#02 DiffSlitVert    ZDiff1    ZDiff2    ZDiff3    XDiff1    XDiff2   EntSlit  SExitSlit
#03 ExitSlit  ExitSlitY       SGM       NIX       ZPX       ZPY       ZPZ      OSAX
#04     OSAY      OSAZ       MuR       MuT

#S 1  ascan  ens 6300 5700  30 1
#D Tue Apr 13 11:21:16 2010
#T 1  (Seconds)
#00 0 0 0 0 0 1 0 0 0 0 0 0 600 0 0 0 143.6
#01 1.54 1.54 1.54 90 90 90 4.079990459 4.079990459 4.079990459 90 90 90 1 0 0 0 1 0 60 30 0 0 0 0 60 30 0 90 0 0 1.54 1.54
#03 4.079990459 -6.257089578e-16 -6.561207576e-16 -3.041179982e-17 4.079990459 -2.49819112e-16 0 0 4.079990459
#04 0.02718979506 -0.004794294468 0 19.37164365 0 0 0 20 -90 0 0 0 0 0 0 0 0 0 0 0 0 -180 -180 -180 -180 -180 -180 0
#Q 0.0271898 -0.00479429 0
#P0 0 0 -90 0 0 20 63.474197 131.91284
#P1 2.0666664 1.116502 -1.6699984 -3.8200094 -5.8035941 0 -7.5889309e-07 0
#P2 44.36 -3.8200257 -3.8199931 -3.8200257 -5.8035941 -5.8035941 6000 4905.98
#P3 100.07365 37200 2076 0 -2.6 -20.61369 -59.6 42.215
#P4 -26.9804 2.76 0 0
#UM0 Current Lifetime
#UM1 252.199997 26.549999
#UM2  **BEAM AVAILABLE**  NEXT FILL 1900 HRS
#UM3 NSLS
#UM4 MACHINE INFORMATION
#UM5 CHANNEL 2
#UE0 Energy Lambda MSPA MZOS ESPA EZOS
#UE1 640.029 19.3716 -504.268 11844.5 -100 2348.85
#US0    ens              10 um  (6000)
#US1    exsy            OPEN    (37200)
#US2    dvs             UNKNOWN (44.36)
#UU0 Gap 36.016000 (mm)
#UU1 Gap control is Enabled, server is Locked.
#UU2 Command Successful
#N 25
#L EntSlit  H  K  L  Epoch  Seconds  Shutter  diode  mdet1  sdet1  mdet2  sdet2  Ring  Current  Treg  Tsam  SGMENC  SGME  THPos  DELPos  CCPres  IONPress  kpts  Monitor  D
Detector
6300.2 0.0271898 -0.00479429 0 79157 1 0 3120 0 0 0 0 25441 252.2 298.14 298.71 0 0 0 0 0 0 0 362 0
6279.66 0.0271898 -0.00479429 0 79162 1 0 3321 0 0 0 0 25438 252.2 298.14 298.71 0 0 0 0 0 0 0 403 0
6260.7 0.0271898 -0.00479429 0 79168 1 0 2969 0 0 0 0 25437 252.2 298.14 298.71 0 0 0 0 0 0 0 456 0
6240.16 0.0271898 -0.00479429 0 79173 1 0 3134 0 0 0 0 25439 252.2 298.14 298.71 0 0 0 0 0 0 0 531 0
--:--  pcmo.01          (Fundamental)--L27--Top----------------------------------------------------------------
```

Terminal emacs@tardis.nsls.bnl...

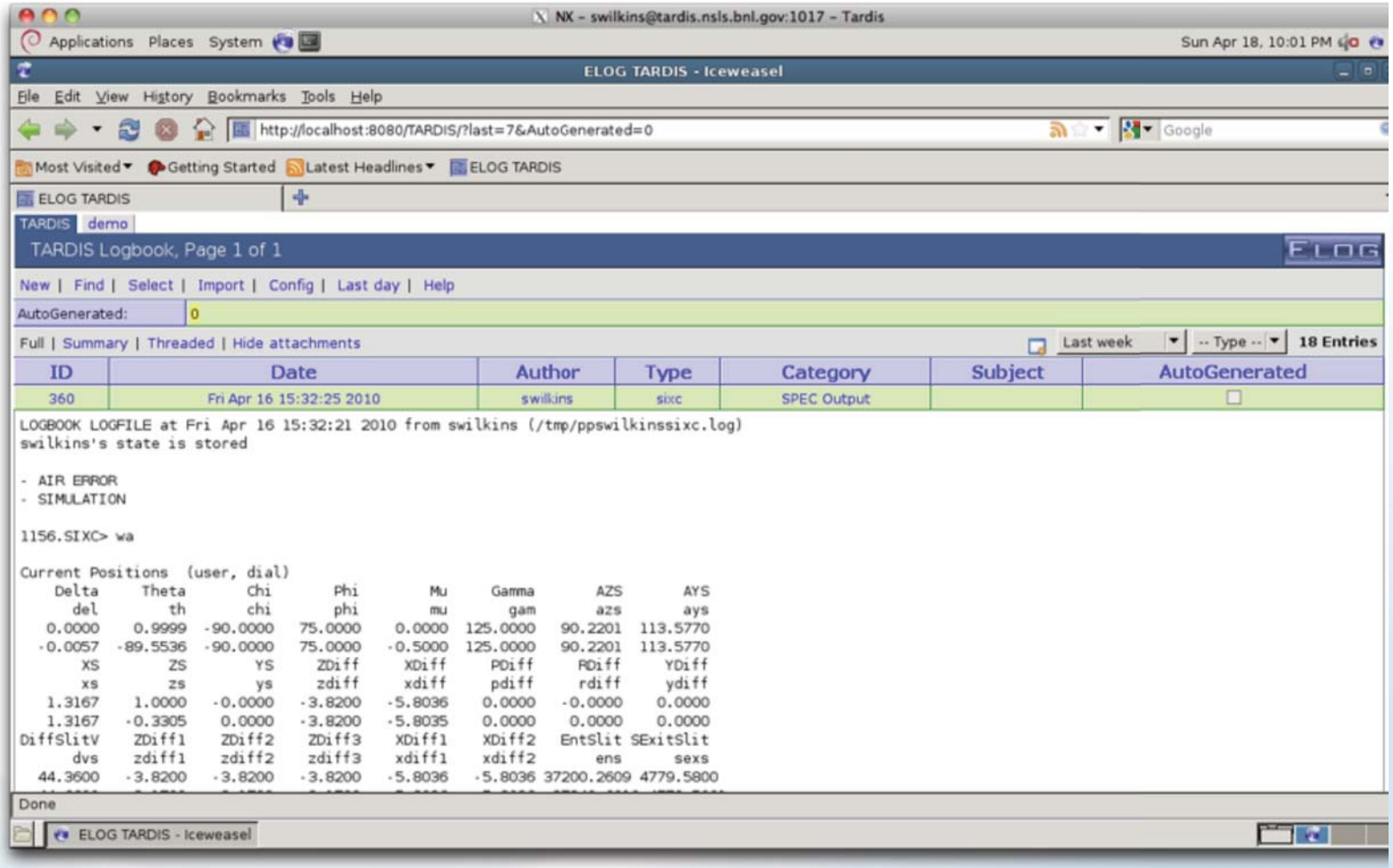
Data Format Advantages

- Scans are formed in a single file...
- Scan “headers” form part of an electronic logbook.
- Easy to imbed comments into data (stream) file for changes in beamline.

```
#C Tue Apr 13 17:18:14 2010.  g_h1 reset from 0 to 1.
#C Tue Apr 13 17:18:14 2010.  g_u13 reset from 90 to 0.
#C Tue Apr 13 17:18:33 2010.  Mode reset from 12 to 16.
#C Tue Apr 13 17:21:33 2010.  Shutter was OPEN is now CLOSED.
#C Tue Apr 13 17:21:42 2010.  Shutter was CLOSED is now OPEN.
```



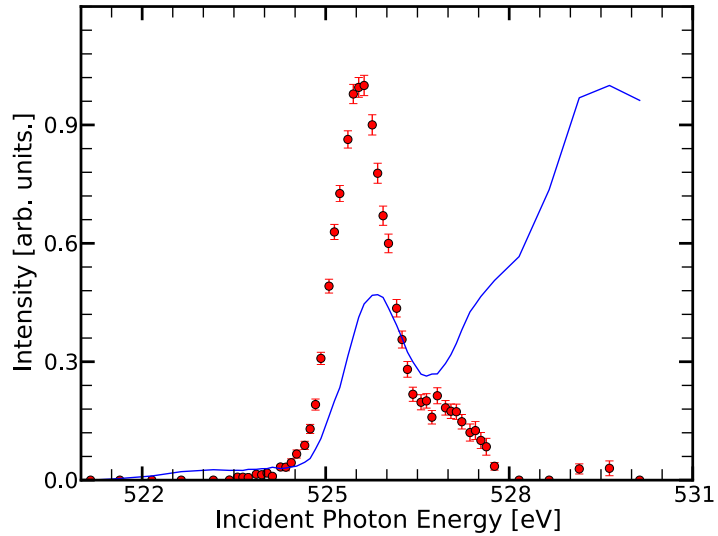
Ease of integration into Electronic Logbook....

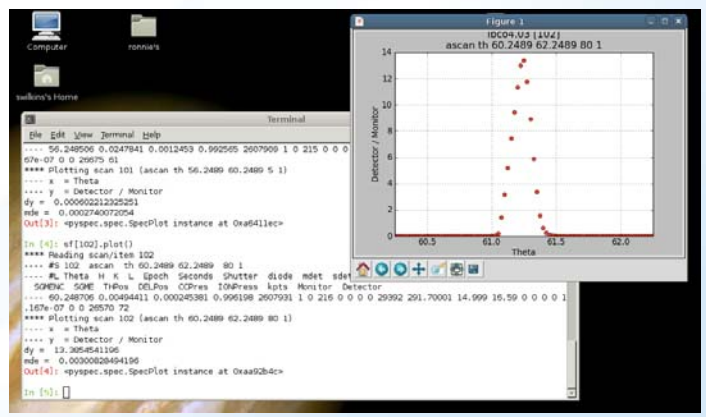


Data Analysis on the fly....

- Use a python script to produce publication quality plot....

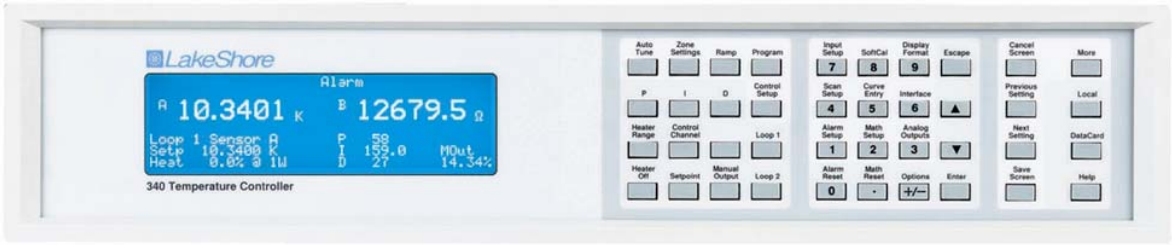
```
def Edep(sf, scans, guess, fname):

    nf = 1
    nnf = 1
    all = array([])

    for scan in scans:
        s = sf[[scan, scan+1, scan+2]]
        if nf == 1:
            figure(figsize = (8.5,11))
        subplot(3,2,nf)
        if nf == 6:
            nf = 1
            savefig('%s_%d.pdf' % (fname, nnf))
            nnf = nnf + 1
        else:
            nf = nf + 1

        errorbar(s.H, s.Detector / s.Monitor, yerr = sqrt(s.Detector) / s.Monitor, fmt = 'ro')
        x = s.H
        y = s.Detector / s.Monitor
        e = sqrt(s.Detector) / s.Monitor
        f = fit.fit(x = x, y = y, e = e, funcs = [ff.linear, lorr2],
                    guess = guess,
                    ilimited = [[0, 0], [0,0], [1, 1], [1, 1], [1, 1]],
                    ilimits = [[0,0], [0,0], [0.239, 0.241], [7e-3, 1.2e-2], [0, 1]])
        f.run()
        x, y = f.evalfitfunc(nxpts = 300)
        plot(x, y, 'b-')

        all = concatenate((all, array([12390 / s.wavelength]), f.result, f.stdev))

    all = all.reshape(-1, f.result.size * 2 + 1)

    print all
    return all
```





Control of interfaces

- SPEC gives direct access to instrument control interfaces, SERIAL (RS232), GPIB, CAMAC
- Most often it is not necessary to abstract control (eg. EPICS)
- Varies from experiment to experiment.
- User owned sample environments / measurements .....

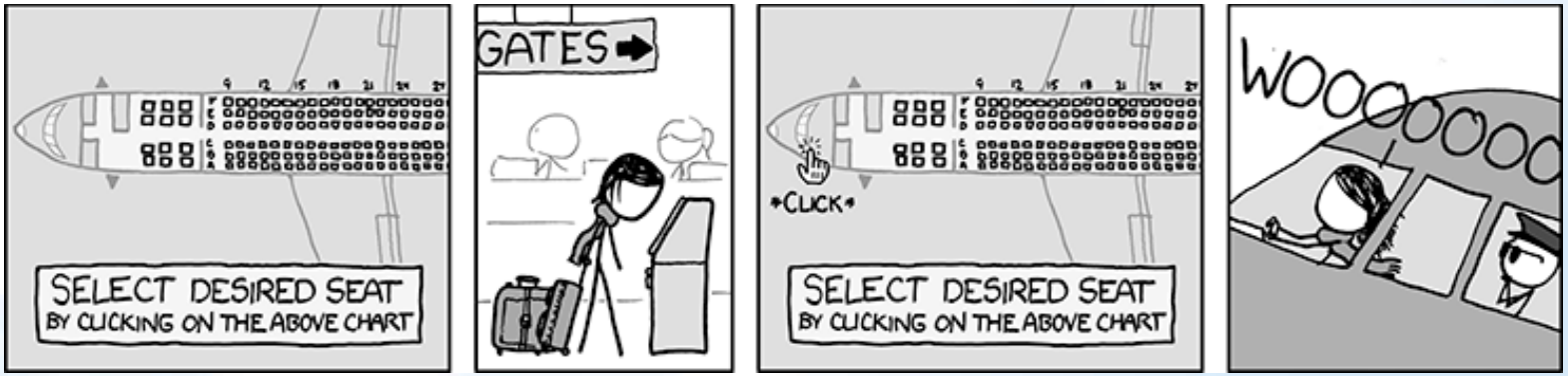


Why is it not possible to abstract?

- No two sample environments are similar!
  - Functionality of devices varies too much.
- Not used as regular equipment.
  - User supplied hardware / one-off experiments
- No need to spend time to abstract.....
  - Why bother?
- Usually has to be “hacked”

Disadvantages of SPEC

- SPEC's language is very basic.
- Difficult to create macros which are completely type checked
- Macros work by substitution
- "umv th hdjk"



Macro Libraries are Avaliable

- SPEC has a wide range of relatively debugged macros.
- Virtual Motors / Counters
- Sample environments.
- Other hardwares....

ESRF

Beamline Instrument Software Support

SPEC Macro documentation: [Macro Index | BLISS Home]

Categories: Accelerator | Acquisition | All | BPM | Beamlines | Ccd | Counting | Detection | Generic I/O | Isg | MCA | MX | Mca | Multimeter | Obsolete | Other hardware | Positioning | Scans | Temperature | Tools | Vacuum | X-ray beam | other, |

Macros in category: **All**

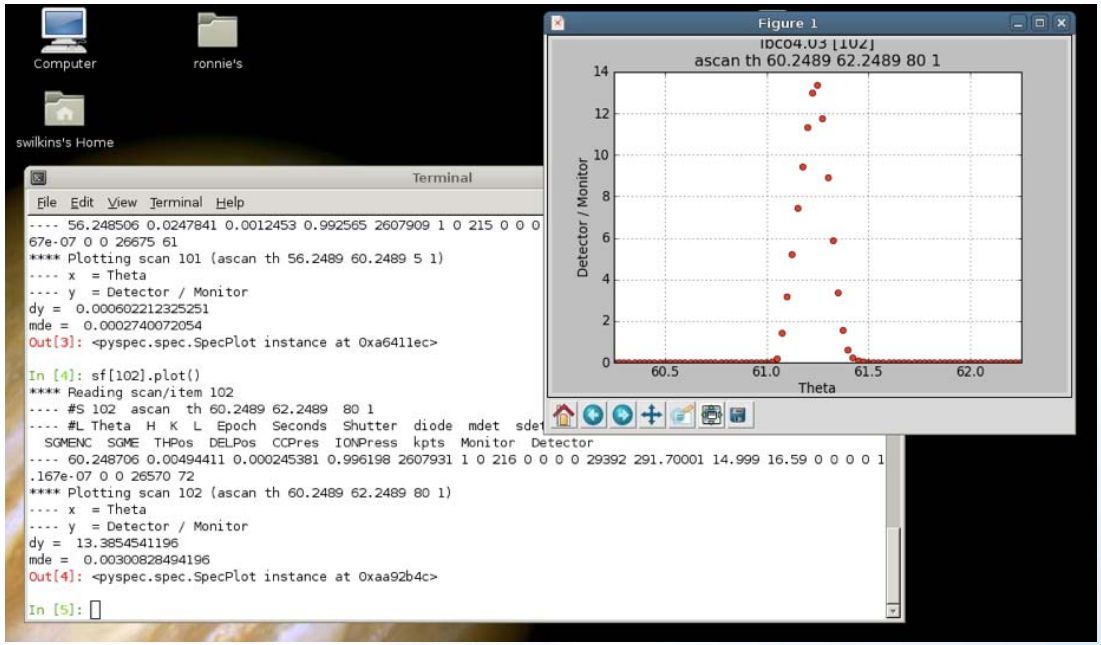
File	Description
ace	*Macros for operating and testing the APD Controller Electronics*
adc	*Macros to work with an ADAS ICV 150 ADC from SPEC*
asl	*Temperature scans with the Alpha Sigma Lambda temperature controller*
ASLf600DC	*Macro counter for Automatic System Laboratories thermometry bridge F600 DC via USB.*
ASLf700	*Macro motor for AlphaSigmaLambdaAutomatic System Laboratories temperature controller type F700.*
att_relay	*handle the relay driven attenuator box*
atte	*Attenuator Macros*
attfactor	*monochromatic beam attenuators*
attocube	*Macros to work with attocube*
attributes	*ATTRIBUTES.MAC - TANGO attribute macro counters.*
autofocus	*Macros for autofocus of minidiff camera from spec*
bac24	*Macros to control the BAC24.*
backstop	*This macro set handles the beamstop at mx beamlines.*
balzer	*Macros to read and display balzers*
beamcheck	*Checks for beam presence*
blcal	***blcal.mac** - do a calibration of the beamline flux.%BR%*
blmenu	*Macros to easily prepare a menu to access SPEC subsystems functionalities on a beamline.*
c111_gfd1d	*Access to C111 Card Linux Taco device server*
c111_mhip	*Macros to implement TDC on C111 card through device server*
c111_mhit	*Macros to implement one TDC on C111 card*
C353	*C353 bunchmarker control*
CaenV462	*Caen V462 counter timer via macro counters and vmeds*
CaenV560	*Macro to control a CAEN V560 counter card, with macro counters and vmeds*
camera	*General features for images acquisition. OBSOLETE: use ccd.mac instead*

SPEC is used in universities

- Because SPEC is commercial (?) it is used in university laboratories for data acq.
- Students arrive trained at synchrotrons to use SPEC.
- This is invaluable for beamline scientists.
- Huge user-base beyond the facility.

Summary

- Advantages
  - UNIX Based
  - Command Line Based
  - Language is C like
  - Virtual Motors / Scalars in userspace
  - Talks to Abstracted Hardware
  - Hardware level control of instrument interfaces
  - Hackable!!!
  - Wide macro base already exists
  - User base already exists / training in universities
- Disadvantages
  - Language is a bit “basic”
  - Little bit too retro ...

Software Available

- For SPEC data analysis ...
  - PYSPEC (http://sourceforge.net/projects/pyspec/)