

Python 'IOC'

Klemen Vodopivec, Kay Kasemir,
Feb. 2022

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Why Python? Why Python IOC?

- Python widely adopted language, easy to learn, develop, test
- Large set of readily available high-level functions
 - Crunching data (numpy, scipy, etc.)
 - Interfacing external sources, ie. databases, web services
- When existing EPICS functionality doesn't cover the new use case...
- ...but want to use robust EPICS infrastructure
 - IOC shell (interactive console, commands, logging)
 - Reliable CA & PVA communication
 - Archives, alarms, autosave
 - GUI tools

Example: Proposal Database

ipts_items x

Beamline: **bl-14b** Run State: **Run** Only change when run state is IDLE! Proposal ID: **21188**

ID	Title	Start	Members
21376	Commissioning HYSPEC with new IRP & moderator	2018-05-17	2XY
21188	Study of intrinsic resonance mode found in ab-initio simulations of NaBr	2018-10-24	B_FULT...
20806	Magnetic Field Dependence of the Spin Dynamics at the Quantum Critical P...	2018-11-14	CLANC...
20536	Evolution of spinon excitations under external magnetic fields in TbIn0.95Mn...	2018-10-31	MGKIM;...
18299	BlueSky Ophyd testing with EPICS PV's, live	2016-12-19	19G;2X...
14871	Isolating magnetic excitations from phonons in isotopic Gadolinium	2016-02-12	19G;2X...
14664	Commissioning with EPICS, NED	2015-08-14	2L1;2X...
13589	Measuring the Magnetic Form Factor in the Topological Kondo Insulator Sm...	2015-05-20	2L1;2X...
12265	Magnetic excitations of the sawtooth Fe chains in Rb2Fe2O(AsO4)2	2014-08-15	2XY;2X...

ID	Name	Description	Mass	Container	Nature
-1	No sample	N/A	N/A	N/A	N/A
60430	La2CuO4	Solid polycrystalline di...	25 g	Aluminum Mount	Polycrystal
60431	LSCO	Solid polycrystalline di...	25 g	Aluminum Mount	Polycrystal

Proposal ID: **21188** Start: **2018-10-24**

Title: Study of intrinsic resonance mode found in ab-initio simulations of NaBr

Members: B_FULTZ;MEM;RH3;YSHEN

Contacts: **[REDACTED]**

Sample Environment: Micas furnace GEN II

Devices:

SMS Update: ● OK

Sample ID: **-1** Name: No sample

Mass: 0.0000 g Container: N/A

Formula: N/A Nature: N/A

Lattice: a,b,c: 0.0000 0.0000 0.0000 α, β, γ : 0.00 deg 0.00 deg 0.00 deg

Description: N/A

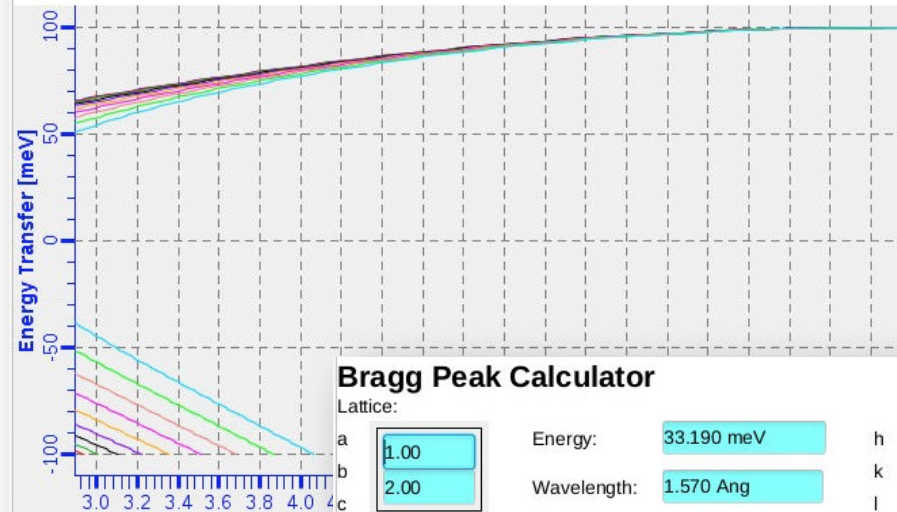
Comments: N/A

Example: Computations

Powder Planning Tool

Energy: Scattering Angle min: Calculate

Wavelength: Scattering Angle max: Calculate



Bragg Peak Calculator

Lattice: a: b: c: Maximum Values: h: k: l: Calculate

Energy: Wavelength:

2.0 [meV] 4.0 [meV] 6.0 [meV]
16.0 [meV] 18.0 [meV] 20.0 [meV]

h	k	l	q	d	2 theta [deg]
0.0	0.0	0.0	0.000	inf	0.000
0.0	0.0	1.0	0.000	inf	0.000
0.0	0.0	2.0	0.000	inf	0.000
0.0	1.0	0.0	0.000	inf	0.000
0.0	1.0	1.0	0.000	inf	0.000
0.0	1.0	2.0	0.000	inf	0.000
0.0	2.0	0.0	6.283	1.000	103.441
0.0	2.0	1.0	6.283	1.000	103.441
0.0	2.0	2.0	6.283	1.000	103.441
1.0	0.0	0.0	6.283	1.000	103.441
1.0	0.0	1.0	6.283	1.000	103.441
1.0	0.0	2.0	6.283	1.000	103.441
1.0	1.0	0.0	6.283	1.000	103.441
1.0	1.0	1.0	6.283	1.000	103.441
1.0	1.0	2.0	6.283	1.000	103.441

Single Crystal Planner

Energy: Sample Rotation Angle min: Calculate

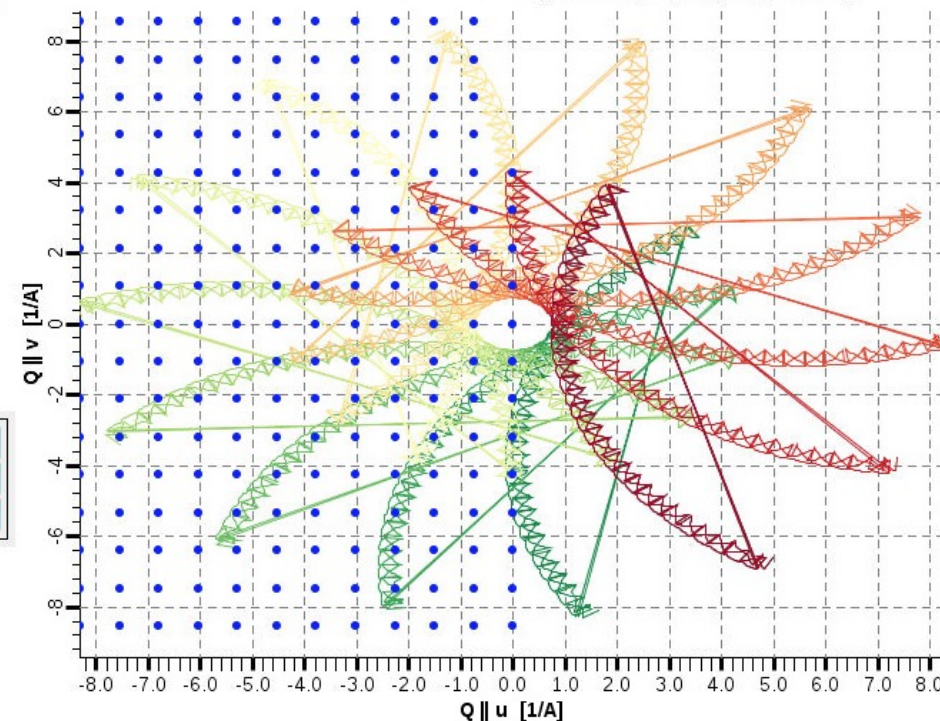
Energy transfer: Sample Rotation Angle max: Create Scan Tab.Scan

Sample Rotation Step: (updates tmp/crystal_plan_scan.csv)

Lattice Parameters [A]:

Lattice Angles [deg]:

Vectors u, v:



Min. sample angle --- Max sample angle Grid

Example: Energy Adjustment

IncidentEnergy X

Desired Incident Energy: <-- "Enter" on desired energy starts update of motors & choppers!

Last successfully set: 17.000 meV ● Kill Air And Stop

Fermi Chopper Speed: Flat Focusing Test Mode

*Fermi speed, flat focusing become active the next time a desired energy is entered.
Consider using Test Mode before actual energy update.*

```

2018-10-24 16:00:53 INFO Would set BL14B:Chop:Skf4:EnergyReq = 17
2018-10-24 16:00:53 INFO Adjust positions for PG focus element
2018-10-24 16:00:53 INFO Current values m1pg 20.3678 deg, vm2 40.736 deg, mfgp 0.897044 1/m
2018-10-24 16:00:53 INFO Moving to m1pg 19.0826 deg, mfgp 0.983556 1/m
2018-10-24 16:00:53 INFO Would set BL14B:Mot:mfgp = 0.983556
2018-10-24 16:00:53 INFO Would set BL14B:Mot:m1pg = 19.0826
2018-10-24 16:00:53 INFO Evaluating safety of moving drum shield to 38.1653 deg
2018-10-24 16:00:53 INFO Moving to vm2 38.1653 deg
2018-10-24 16:00:53 INFO Would set BL14B:Mot:vm2 = 38.1653
2018-10-24 16:00:53 INFO PG Focus element angle at 20.368 deg
2018-10-24 16:00:53 INFO PG Focus element focus at 0.897 1/m
2018-10-24 16:00:53 INFO Drum shield angle at 40.736 deg
2018-10-24 16:00:53 INFO Successfully tested Ei=17 meV
2018-10-24 16:01:45 INFO ----- Setting Ei to 17 meV -----
2018-10-24 16:01:45 INFO Setting T0=30 Hz and Fermi=300 Hz
2018-10-24 16:01:45 INFO T0 is already at requested frequency.
2018-10-24 16:01:45 INFO Fermi frequency is already at requested frequency
2018-10-24 16:01:45 INFO Changing incident energy from 15 meV to 17 meV
2018-10-24 16:01:45 INFO Changing incident energy from 15 meV to 17 meV
2018-10-24 16:01:45 INFO Changing incident energy from 15 meV to 17 meV
2018-10-24 16:01:45 INFO Changing incident energy from 15 meV to 17 meV
2018-10-24 16:02:23 INFO Adjust positions for PG focus element
2018-10-24 16:02:23 INFO Current values m1pg 20.3678 deg, vm2 40.736 deg, mfgp 0.897044 1/m
2018-10-24 16:02:23 INFO Moving to m1pg 19.0826 deg, mfgp 0.983556 1/m
2018-10-24 16:02:40 INFO Evaluating safety of moving drum shield to 38.1653 deg
2018-10-24 16:02:40 INFO Moving to vm2 38.1653 deg
2018-10-24 16:04:10 INFO PG Focus element angle at 19.082 deg
2018-10-24 16:04:10 INFO PG Focus element focus at 0.98 1/m
2018-10-24 16:04:10 INFO Drum shield angle at 38.1653 deg
2018-10-24 16:04:10 INFO Successfully set Ei=17 meV
                
```

	Speed Req.	Energy Req.	Lock	OK	
T0:	30 Hz	17.0 meV	●	●	Details
T1A:		17.0 meV	●	●	
T1B:		17.0 meV	●	●	
Fermi:	300 Hz	17.0 meV	●	●	
Monochromator:	PG				Details
Heusler Rotation:	42.42300 deg		●		Details
Heusler Focus:	0.413333 mm		●		
P.G. Rotation:	19.08200 deg		●		Details
P.G. Focus:	0.98000 mm		●		
Detector Vessel:	-71.9913 deg				Details
Drum Shield:	1803.87111 mm				
Drum Shield:	38.1657 deg		●		

- Could use sequencer, but had existing python code

Available Python IOC solutions

pcaspy

<https://github.com/paulscherrerinstitute/pcaspy>

- Most mature, started in 2011
- Purely in Python code
- But missing EPICS goodies
 - autosave, IOC commands
- Custom records implementation
 - No RTYP, EGU, SCAN etc. fields
 - Affects GUI tools

pyDevSup

<http://mdavidsaver.github.io/pyDevSup/>

- Plugs into standard EPICS IOC
- Invoked from EPICS records
- Invoked Python code must extend provided class
- Most flexibility to update records and fields

PyDevice

<https://github.com/klemenv/PyDevice>

- Plugs into existing EPICS IOCs, simplest to use
- Invoked from EPICS records
- Arbitrary Python code invoked from EPICS records
- Can't change record fields directly
- EPICS-independent Python code
 - Can be reused or tested outside EPICS environment

pcaspy in more details

- Can be used as an EPICS IOC or a client
- All code is in Python
 - Defining database records
 - Serving the connections
 - Handling requests
 - Interfacing with other IOCs
- Needs to reimplement common EPICS functionality
 - autosave
 - Access Security
 - Data types

PyDevice in more details

- Call any Python code directly from record
- Supports most popular EPICS records
 - ai, ao, longin, longout bi, bo, mbbo, mbbi, stringin, stringout, waveform
- Pass record fields to Python code
 - Values evaluated when record processes
 - Return value pushed to record VAL field
 - Types automatically converted
- Python exceptions translate into SEVR&STAT

```
record(ai, "Example:AbsValue") {  
    field(DTYP, "pydev")  
    field(INP, "@abs(VAL)")  
}
```

```
caput Example:AbsValue -2.2
```

```
record(ai, "Example:Delay") {  
    field(DTYP, "pydev")  
    field(INP, "@time.sleep(VAL)")  
}
```

```
caput -c Example:Delay 5
```


pcaspy

/ics/examples/python/demo_pcaspy_ioc.py

```
from pcaspy import SimpleServer, Driver
import demo
pvdb = { 'random' : {
    # 'desc': 'return value between 0 and hilim'
    'type' : 'float',
    'hilim' : 10 }
}
class DemoDriver(Driver):
    def __init__(self):
        super(demoDriver, self).__init__()
    def read(self, reason):
        if reason == 'random':
            limit = self.getParamInfo(reason)['hilim']
            value = demo.getRandom(limit)
        else:
            value = self.getParam(reason)
        return value
if __name__ == '__main__':
    server = SimpleServer()
    server.createPV('training:pcaspy:', pvdb)
    driver = DemoDriver()
    while True:
        server.process(0.1)
```

vs

PyDevice

/ics/examples/db/demo_pyioc.db

```
record(ai, "$(user):random")
{
    field(DESC, "Random value in range [0,HOPR]")
    field(HOPR, "10")
    field(DTYP, "pydev")
    field(INP, "@demo.getRandom(HOPR)")
}
```

/ics/examples/iocBoot/ioc_pydevice/st.cmd

```
#!/.../bin/linux-x86_64/pydevice
< envPaths
epicsEnvSet("PYTHONPATH", "$(TOP)/python")
cd "${TOP}"
dbLoadDatabase "dbd/demo_pyioc.dbd"
pydevice_registerRecordDeviceDriver pdbname
pydev "import demo"
dbLoadRecords "db/pydevice.db", "user=training:pydevice"
iocInit
```

/ics/examples/python/demo.py

```
import random
def getRandom(limit=1):
    # random() returns value in range [0,1]
    return random.random() * limit
```

PyDevice latest: pycalcRecord

- New record type to support passing multiple parameters
 - Similar to aSub, genSub, calc records
 - But evaluate Python code


```
record(pycalc, "Example:gcd") {  
    field(INPA, "training:pyioc:X1 CP")  
    field(INPB, "training:pyioc :Y1 CP")  
    field(INPC, "training:pyioc :X2 CP")  
    field(INPD, "training:pyioc :Y2 CP")  
    field(CALC, "gcd(A,B,C,D)")  
}
```


```
record(pycalc, "Example:GetHtml") {  
    field(INPA, "Example:Proto CP")  
    field(INPB, "Example:Hostname CP")  
    field(INPC, "Example:Port CP")  
    field(CALC, "mywebClientFetch('A', 'B', C)")  
}
```

Weather demo

[Edit] Display Display × 200 %

Weather demo

City: 


30.8 Fahrenh

Humidity:

Wind:

Clouds:

Pressure:

Show Settings

Settings

OpenWeather Key:

Units:

Handling Table, Structure

- Yes, pickled byte waveform is a hack
 - Requires scripts
 - Only works with python as server & client
- pvAccess can handle custom structures
 - Better for server side
 - Client likely needs more than dump of structure; Will still require script for user-friendly display.
- pvAccess IOC Python libraries
 - pvaPy
 - p2p

h	k	l	q	d	2 theta [deg]
0.0	0.0	0.0	0.000	inf	0.000
0.0	0.0	1.0	0.000	inf	0.000
0.0	0.0	2.0	0.000	inf	0.000
0.0	1.0	0.0	0.000	inf	0.000
0.0	1.0	1.0	0.000	inf	0.000
0.0	1.0	2.0	0.000	inf	0.000
0.0	2.0	0.0	6.283	1.000	103.441
0.0	2.0	1.0	6.283	1.000	103.441
0.0	2.0	2.0	6.283	1.000	103.441

Summary

Python with CA server & client libs can act as IOC

- Great tool to have
- Doesn't replace all IOCs