BioSciences

Berkeley Center for
Structural Biology
(BCSB)

Developing and Integrating EPICS Driver in Python

Gabriel Gazolla
(Computer Systems Engineer)

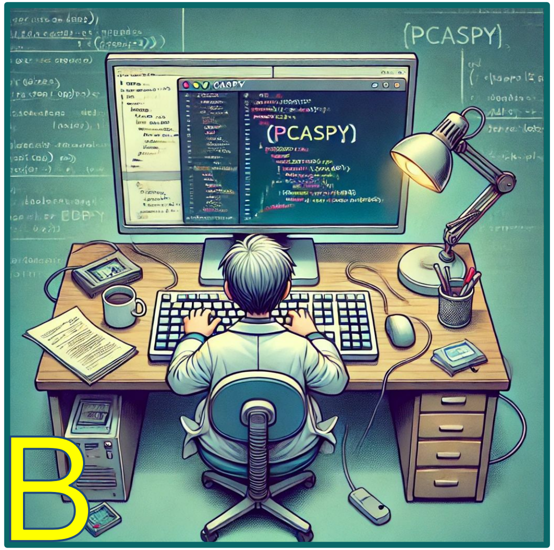
John Taylor
(Head of Software and Instrumentation)

U.S. DEPARTMENT OF ENERGY

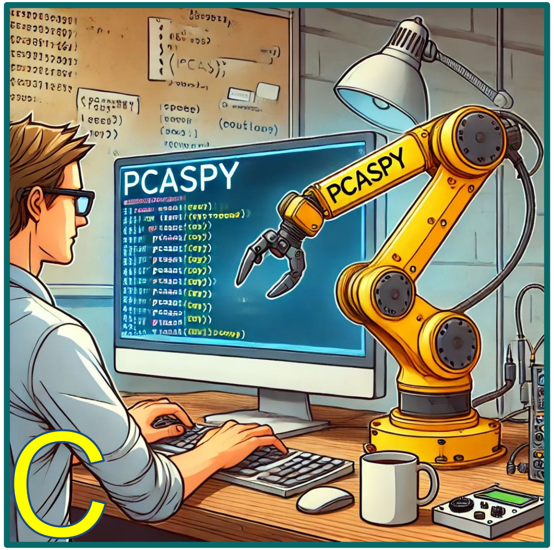
BERKELEY LAB

Agenda & Objectives

Learn what PCASpy is capable of.

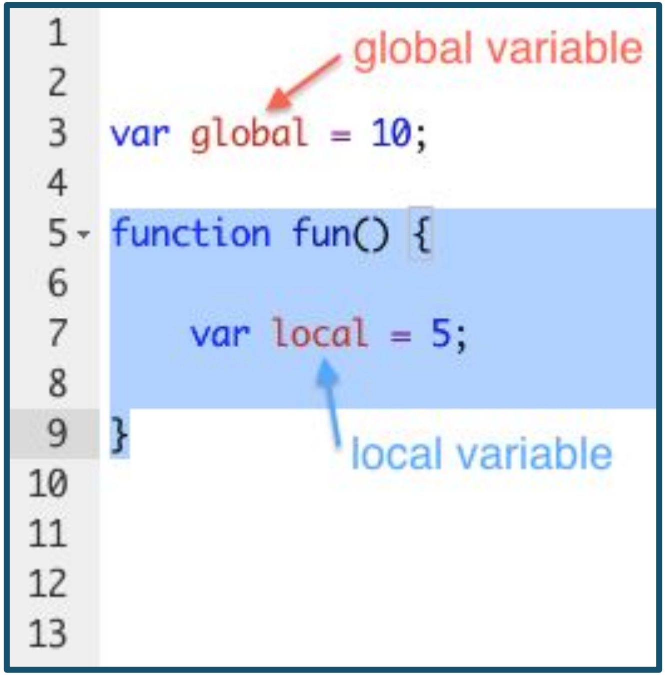


Be able to understand and read a PCASpy implementation.

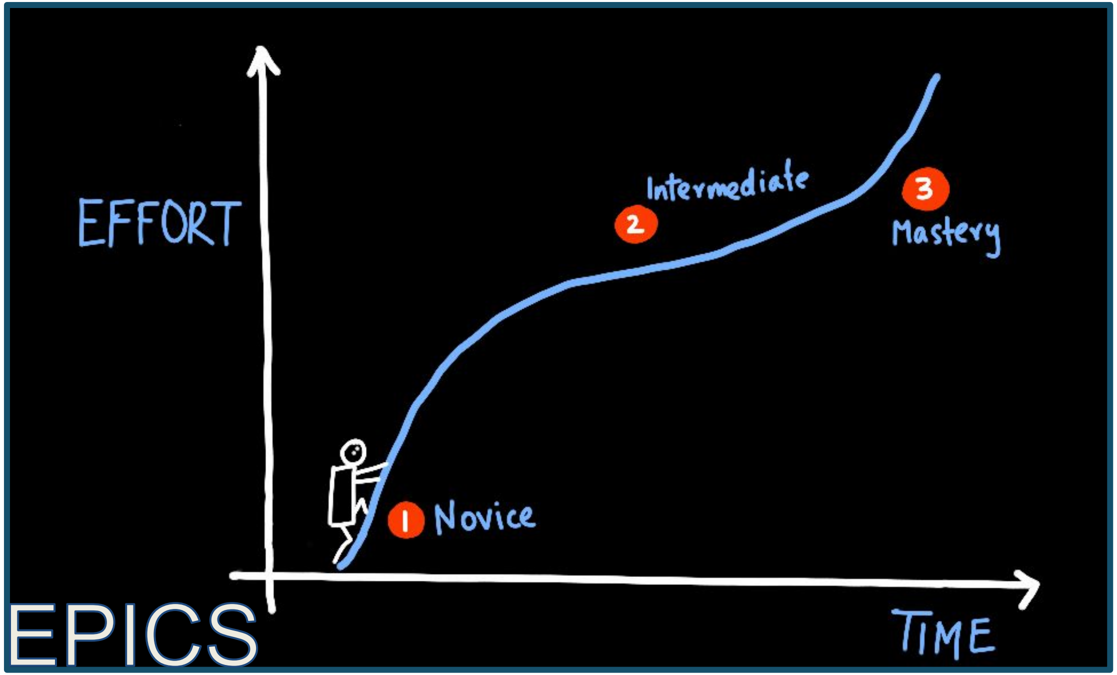


Implement a basic PCASpy driver.

What is the motivation for this talk ?



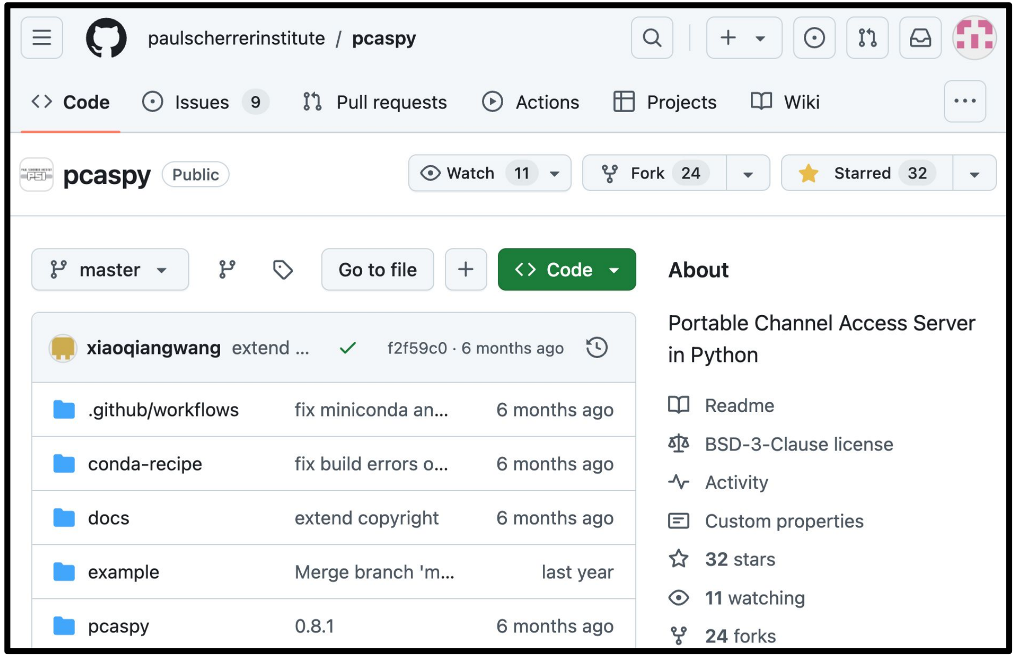
```
var global = 10;

function fun() {

    var local = 5;

}
```



EPICS

What is PCASpy ?

PCASpy /’pi:kas,pai/ provides not only the low level python binding to EPICS Portable Channel Access Server but also the necessary high level abstraction to ease the server tool programming.



paulscherrerinstitute / pcaspy

Code | Issues 9 | Pull requests | Actions | Projects | Wiki

pcaspy Public

Watch 11 | Fork 24 | Starred 32

master | Go to file | + | Code

xiaoqiangwang extend ... ✓	f2f59c0 · 6 months ago	
.github/workflows	fix miniconda an...	6 months ago
conda-recipe	fix build errors o...	6 months ago
docs	extend copyright	6 months ago
example	Merge branch 'm...	last year
pcaspy	0.8.1	6 months ago

About

Portable Channel Access Server in Python

- Readme
- BSD-3-Clause license
- Activity
- Custom properties
- 32 stars
- 11 watching
- 24 forks

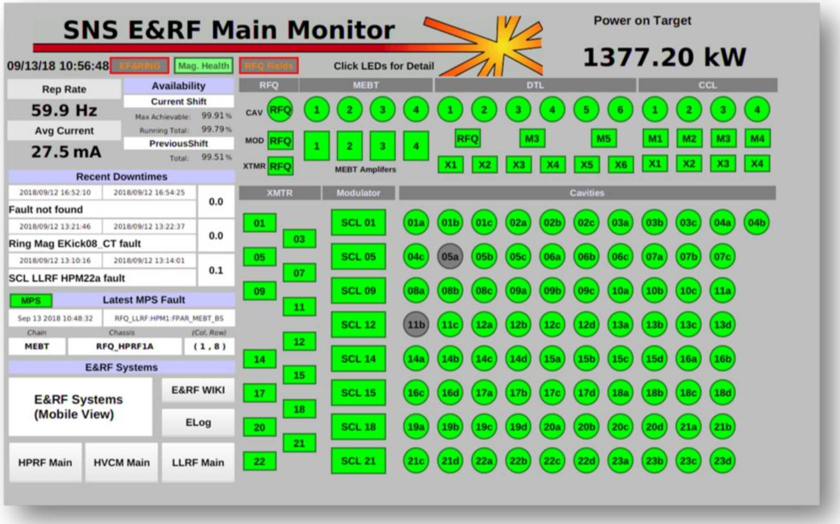
Paul Scherrer Institute

4.5 ★★★★★ 104 Google reviews

Research institute in Würenlingen, Switzerland

PCASpy Example

controlssoftware.sns.ornl.gov



SNS E&RF Main Monitor

Power on Target

1377.20 kW

09/13/18 10:56:48

Rep Rate: 59.9 Hz

Avg Current: 27.5 mA

Availability – Current Shift: Max Achievable: 99.91% · Running Total: 99.79%

PreviousShift: Total: 99.51%

Recent Downtimes

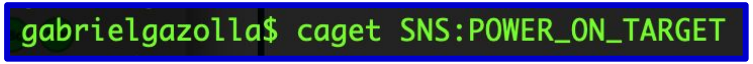
2018/09/12 16:52:10 – 2018/09/12 16:54:25 – Fault not found – 0.0

2018/09/12 13:21:46 – 2018/09/12 13:22:37 – Ring Mag EKick08_CT fault – 0.0

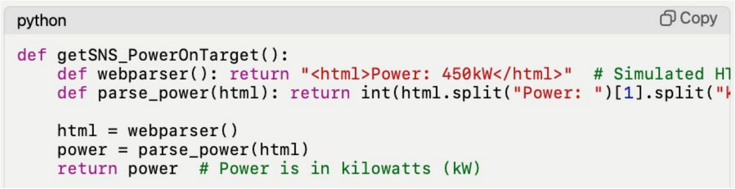
2018/09/12 13:10:16 – 2018/09/12 13:14:01 – SCL LLRF HPM22a fault – 0.1

Latest MPS Fault: Sep 13 2018 10:48:32 – RFQ_LLRF:HPM1:FPAR_MEBT_BS

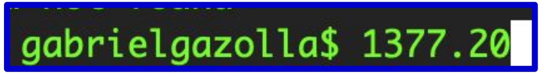
Chain: MEBT · Chassis: RFQ_HPRF1A · (Col, Row): (1 , 8)



```
gabrielgazolla$ caget SNS:POWER_ON_TARGET
```



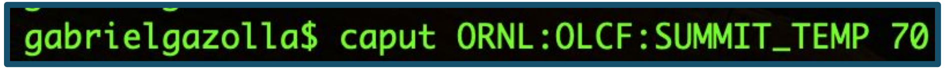
```python
def getSNS_PowerOnTarget():
    def webparser(): return "<html>Power: 450kW</html>"  # Simulated HT
    def parse_power(html): return int(html.split("Power: ")[1].split("k

    html = webparser()
    power = parse_power(html)
    return power  # Power is in kilowatts (kW)
```



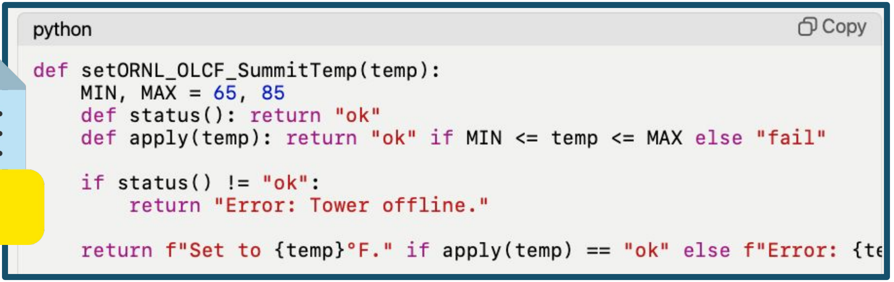
```
gabrielgazolla$ 1377.20
```

Problem is transformed into: “**Do I know how to implement a get/set for the function I need?**

PCASpy Example



```
gabrielgazolla$ caput ORNL:OLCF:SUMMIT_TEMP 70
```



```python
def setORNL_OLCF_SummitTemp(temp):
    MIN, MAX = 65, 85
    def status(): return "ok"
    def apply(temp): return "ok" if MIN <= temp <= MAX else "fail"

    if status() != "ok":
        return "Error: Tower offline."

    return f"Set to {temp}°F." if apply(temp) == "ok" else f"Error: {te
```

Problem is transformed into: “**Do I know how to implement a set for the function I need?**

Agenda & Objectives

Learn what PCASpy is capable of.

✓

How to install PCASpy ?

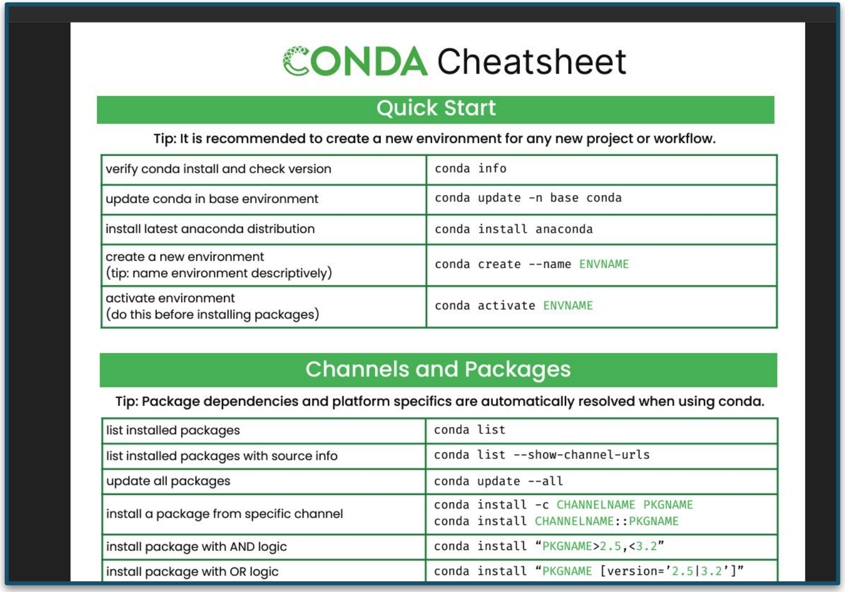
Installing Software...

79%

What is Anaconda ?

BioSciences

Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment.



CONDA Cheatsheet

Quick Start

Tip: It is recommended to create a new environment for any new project or workflow.

verify conda install and check version	`conda info`
update conda in base environment	`conda update -n base conda`
install latest anaconda distribution	`conda install anaconda`
create a new environment (tip: name environment descriptively)	`conda create --name ENVNAME`
activate environment (do this before installing packages)	`conda activate ENVNAME`

Channels and Packages

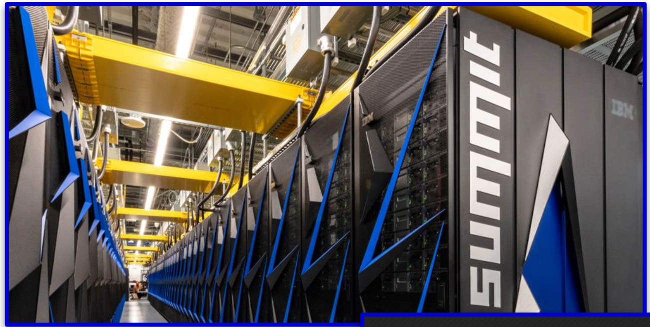
Tip: Package dependencies and platform specifics are automatically resolved when using conda.

list installed packages	`conda list`
list installed packages with source info	`conda list --show-channel-urls`
update all packages	`conda update --all`

install a package from specific channel	`conda install -c CHANNELNAME PKGNAME`
	`conda install CHANNELNAME::PKGNAME`

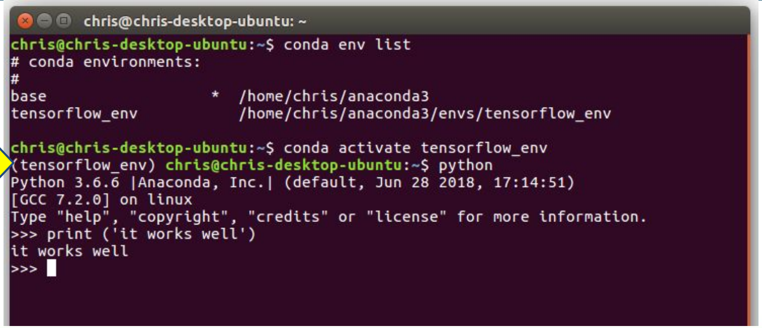
install package with AND logic	`conda install "PKGNAME>2.5,<3.2"`
install package with OR logic	`conda install "PKGNAME [version='2.5|3.2']"`

ANACONDA®



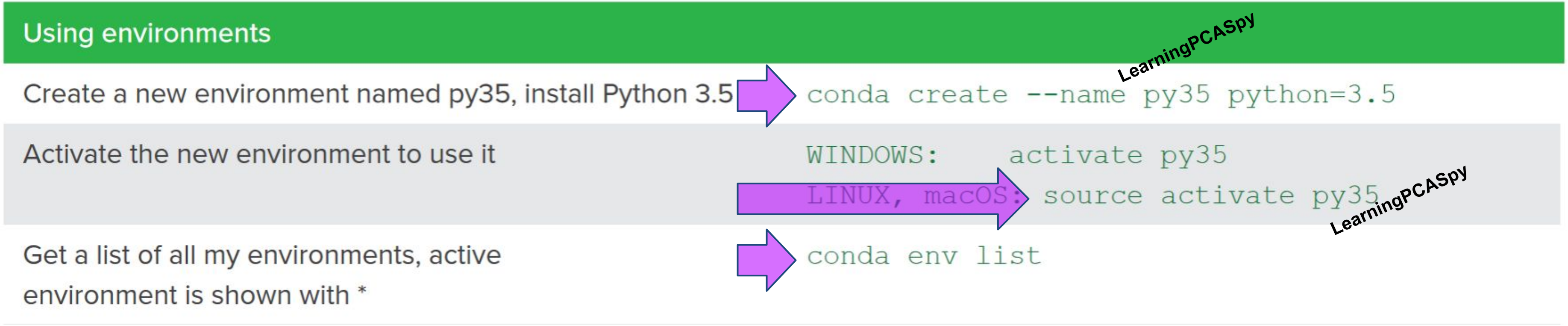
Using Conda

BioSciences



```
chris@chris-desktop-ubuntu:~$ conda env list
# conda environments:
#
base                  *  /home/chris/anaconda3
tensorflow_env           /home/chris/anaconda3/envs/tensorflow_env

chris@chris-desktop-ubuntu:~$ conda activate tensorflow_env
(tensorflow_env) chris@chris-desktop-ubuntu:~$ python
Python 3.6.6 |Anaconda, Inc.| (default, Jun 28 2018, 17:14:51)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ('it works well')
it works well
>>> 
```

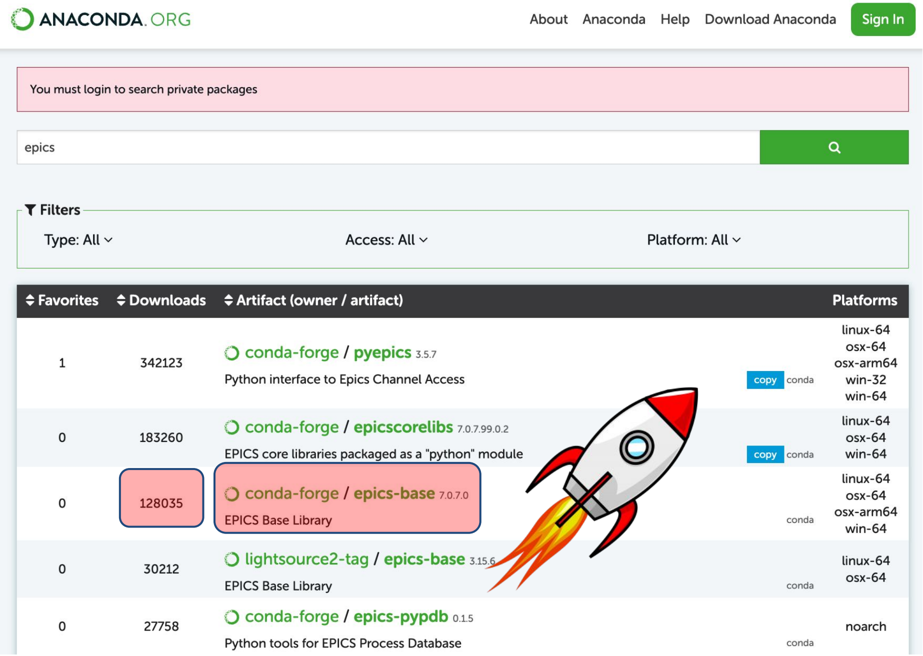
ANACONDA®


LearningPCASpy
LearningPCASpy

Using environments

Create a new environment named py35, install Python 3.5	`conda create --name py35 python=3.5`
Activate the new environment to use it	WINDOWS: `activate py35` LINUX, macOS: `source activate py35`
Get a list of all my environments, active environment is shown with *	`conda env list`

Installing EPICS Base with Conda



ANACONDA.ORG   About   Anaconda   Help   Download Anaconda   Sign In

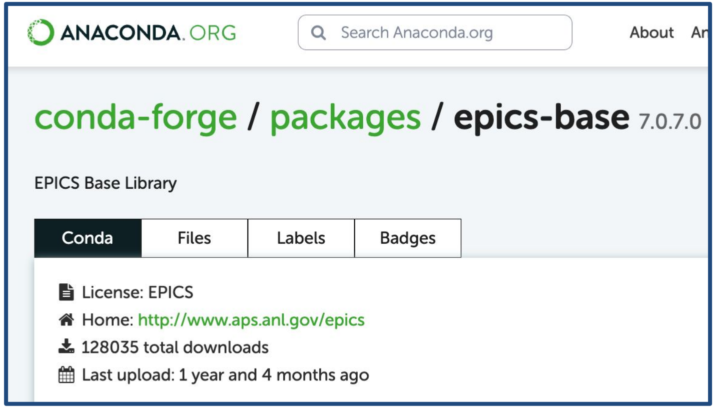
You must login to search private packages

epics

Filters

Type: All   Access: All   Platform: All

Favorites	Downloads	Artifact (owner / artifact)		Platforms
1	342123	conda-forge / pyepics 3.5.7 Python interface to Epics Channel Access	copy conda	linux-64 osx-64 osx-arm64 win-32 win-64
0	183260	conda-forge / epicscorelibs 7.0.7.99.0.2 EPICS core libraries packaged as a "python" module	copy conda	linux-64 osx-64 win-64
0	128035	conda-forge / epics-base 7.0.7.0 EPICS Base Library	conda	linux-64 osx-64 osx-arm64 win-64
0	30212	lightsource2-tag / epics-base 3.15.6 EPICS Base Library	conda	linux-64 osx-64
0	27758	conda-forge / epics-pypdb 0.1.5 Python tools for EPICS Process Database	conda	noarch



ANACONDA.ORG   Search Anaconda.org   About   An

conda-forge / packages / epics-base 7.0.7.0

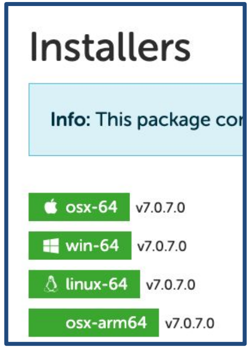
EPICS Base Library

License: EPICS

Home: http://www.aps.anl.gov/epics

128035 total downloads

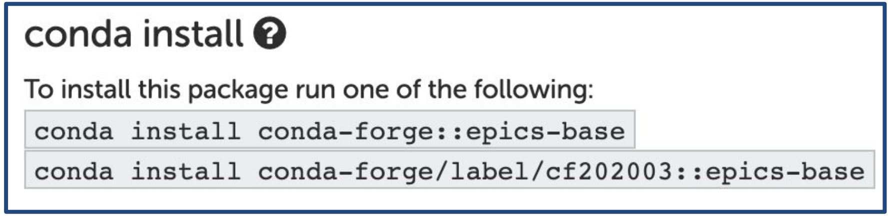
Last upload: 1 year and 4 months ago



Installers

Info: This package co

osx-64	v7.0.7.0
win-64	v7.0.7.0
linux-64	v7.0.7.0
osx-arm64	v7.0.7.0

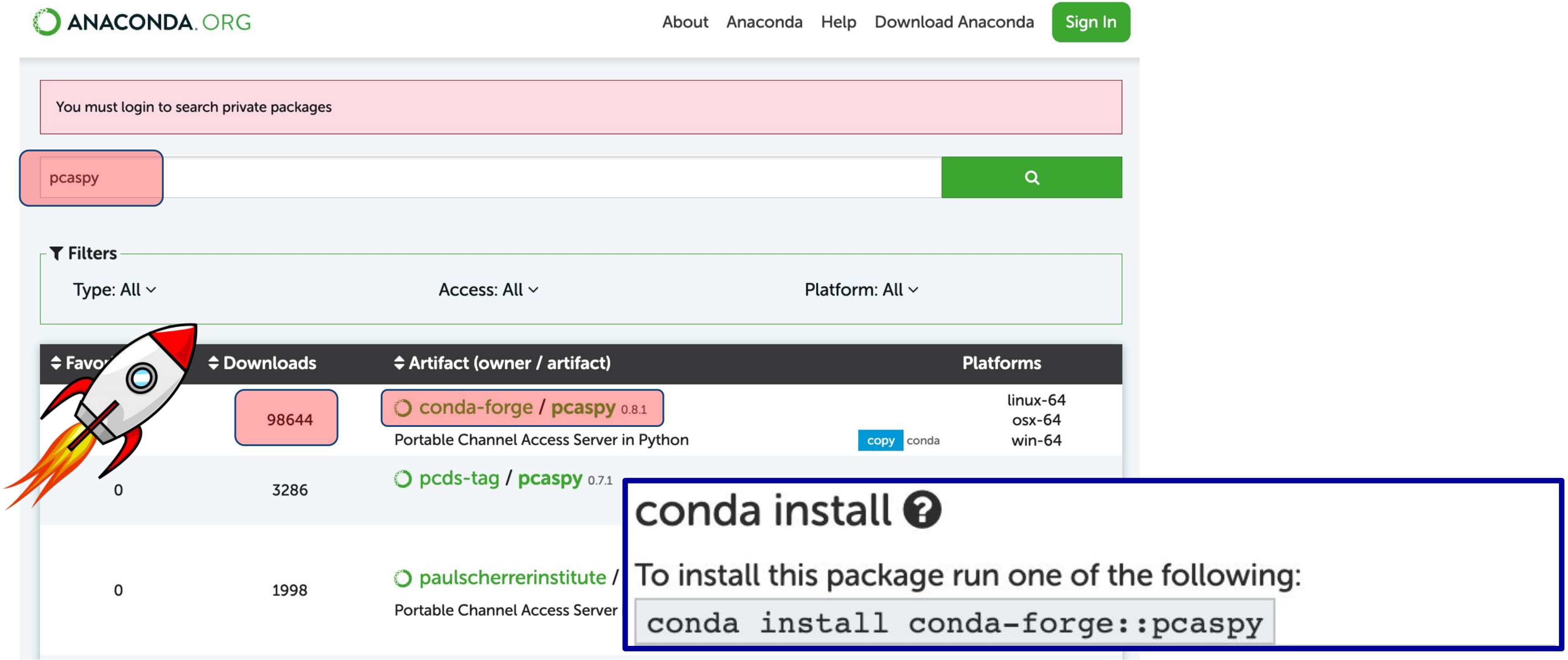


conda install ?

To install this package run one of the following:

```
conda install conda-forge::epics-base
conda install conda-forge/label/cf202003::epics-base
```

Installing PCASpy - Easy Installation



You must login to search private packages

pcaspy

Type: All
Access: All
Platform: All

Favo	Downloads	Artifact (owner / artifact)	Platforms
	98644	conda-forge / pcaspy 0.8.1 Portable Channel Access Server in Python	linux-64 osx-64 win-64
0	3286	pcds-tag / pcaspy 0.7.1	
0	1998	paulscherrerinstitute / Portable Channel Access Server	

conda install

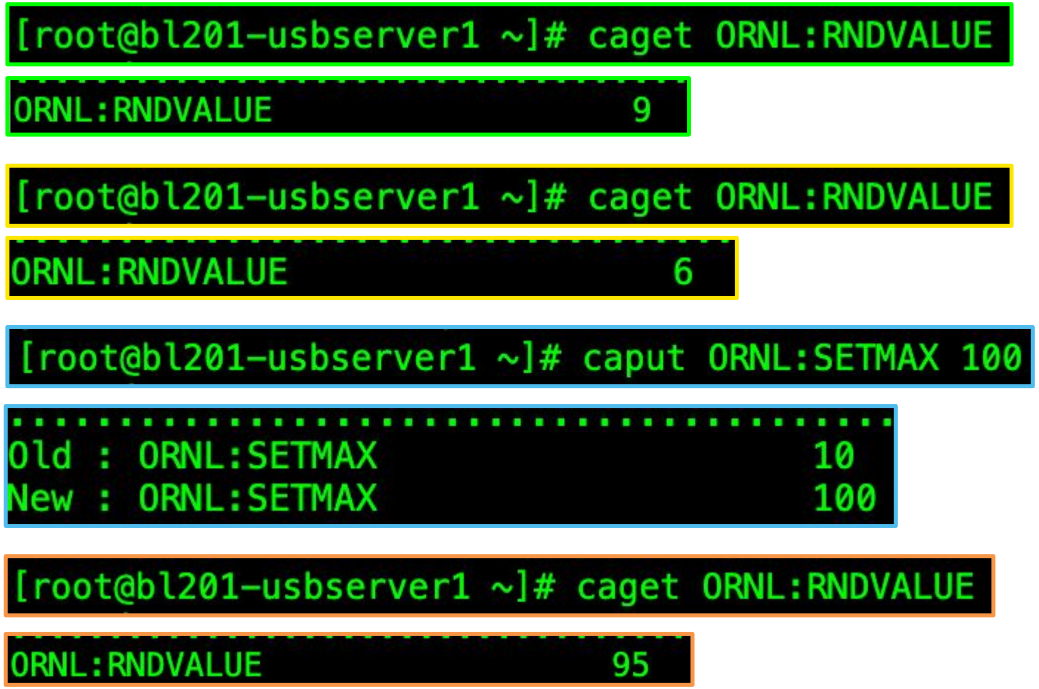
To install this package run one of the following:

```
conda install conda-forge::pcaspy
```

Hello World! example using PCASpy

PCASpy

BioSciences



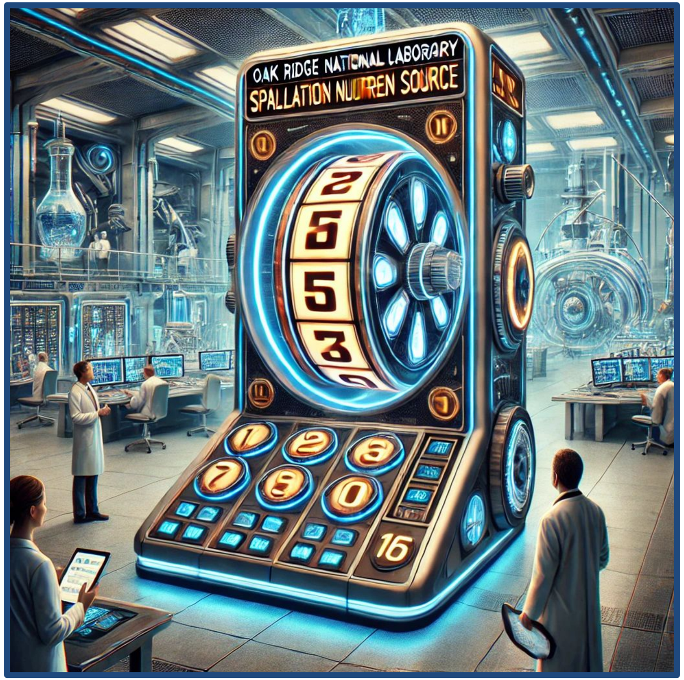
```
[root@bl201-usbserver1 ~]# caget ORNL:RNDVALUE
ORNL:RNDVALUE                     9
```

```
[root@bl201-usbserver1 ~]# caget ORNL:RNDVALUE
ORNL:RNDVALUE                     6
```

```
[root@bl201-usbserver1 ~]# caput ORNL:SETMAX 100
Old : ORNL:SETMAX                 10
New : ORNL:SETMAX                 100
```

```
[root@bl201-usbserver1 ~]# caget ORNL:RNDVALUE
ORNL:RNDVALUE                     95
```

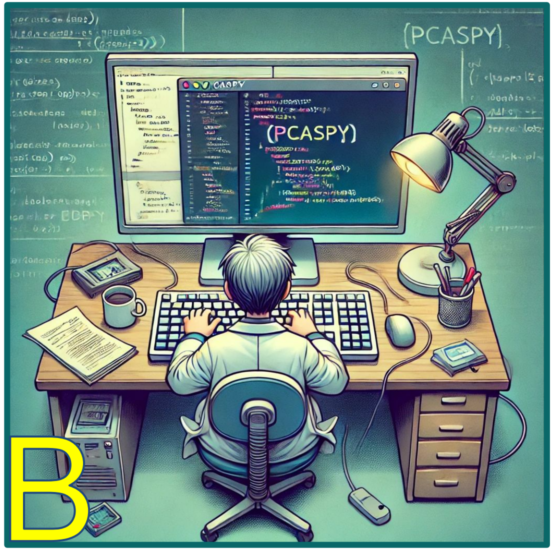
Random Number Generator [1,N]



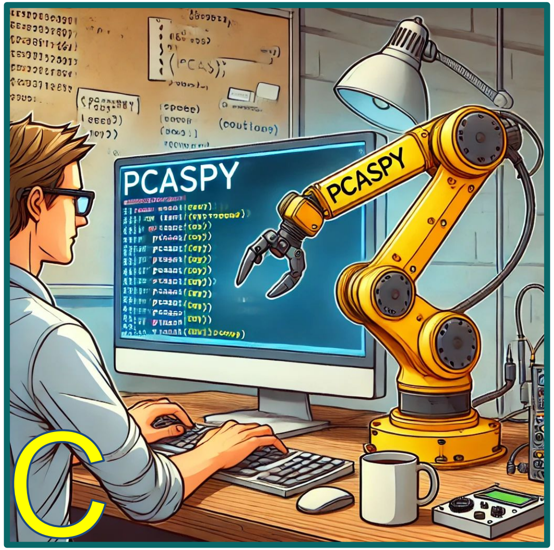
Agenda & Objectives

Learn what PCASpy is capable of.

✓

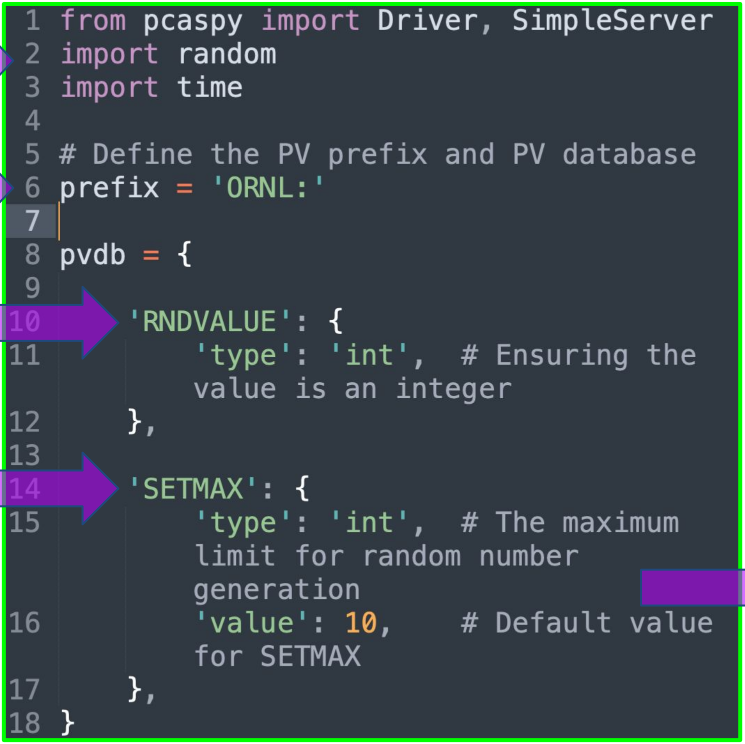


Be able to understand and read a PCASpy implementation.

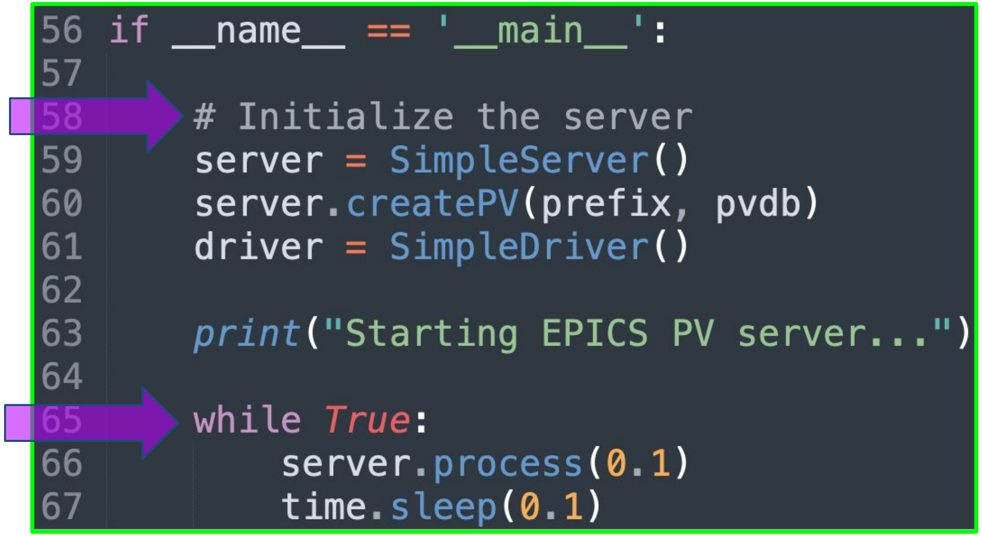


Implement a basic PCASpy driver.

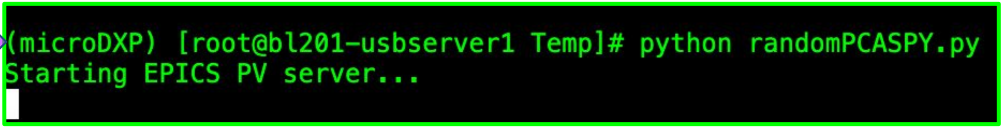
PCASPy - EPICS - Random Number Generator



```python
from pcaspy import Driver, SimpleServer
import random
import time

# Define the PV prefix and PV database
prefix = 'ORNL:'

pvdb = {

    'RNDVALUE': {
        'type': 'int',   # Ensuring the value is an integer
    },

    'SETMAX': {
        'type': 'int',   # The maximum limit for random number generation
        'value': 10,     # Default value for SETMAX
    },
}
```

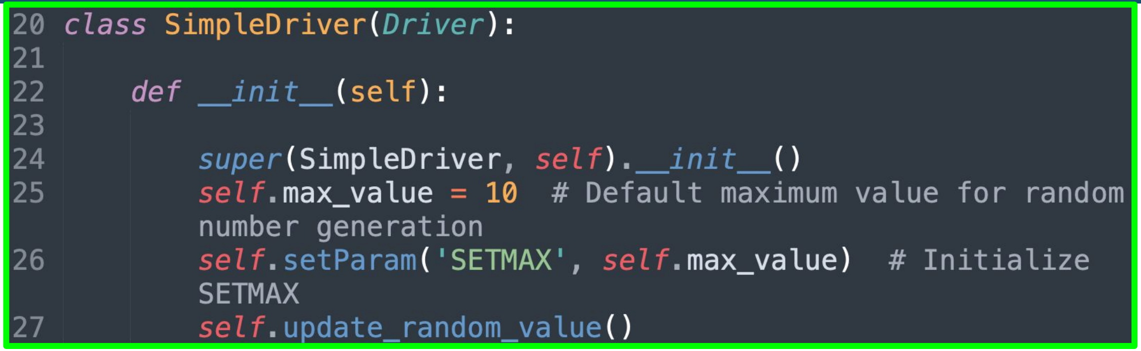


```python
if __name__ == '__main__':

    # Initialize the server
    server = SimpleServer()
    server.createPV(prefix, pvdb)
    driver = SimpleDriver()

    print("Starting EPICS PV server...")

    while True:
        server.process(0.1)
        time.sleep(0.1)
```

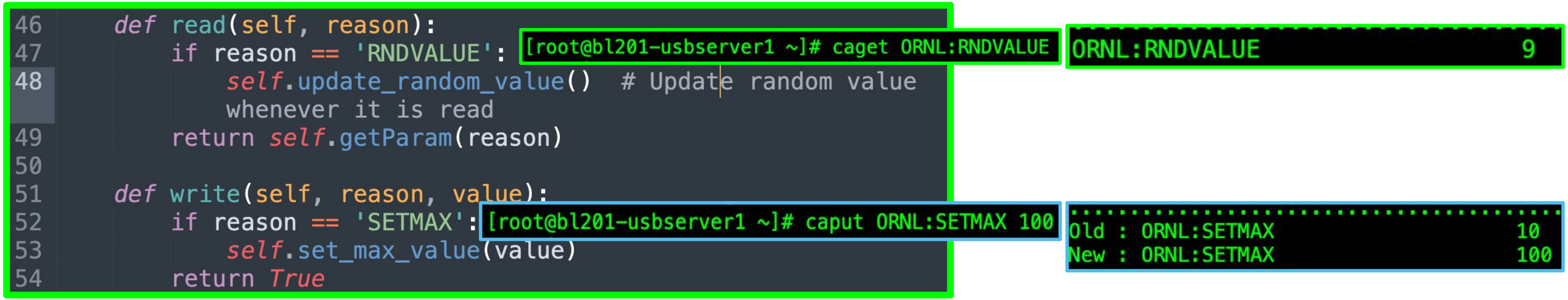


```
(microDXP) [root@bl201-usbserver1 Temp]# python randomPCASPY.py
Starting EPICS PV server...
```

PCASPy - EPICS - Random Number Generator



```python
class SimpleDriver(Driver):

    def __init__(self):

        super(SimpleDriver, self).__init__()
        self.max_value = 10  # Default maximum value for random number generation
        self.setParam('SETMAX', self.max_value)  # Initialize SETMAX
        self.update_random_value()
```

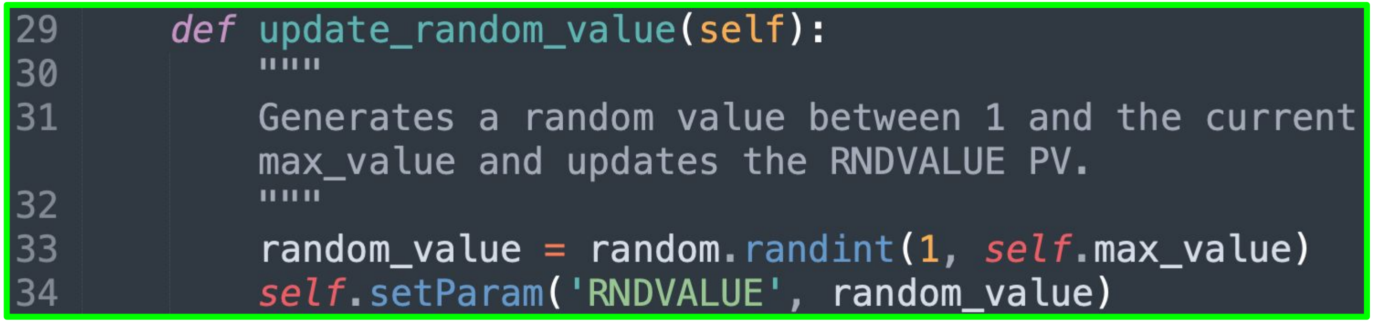


```python
    def read(self, reason):
        if reason == 'RNDVALUE':
            self.update_random_value()  # Update random value whenever it is read
        return self.getParam(reason)

    def write(self, reason, value):
        if reason == 'SETMAX':
            self.set_max_value(value)
        return True
```

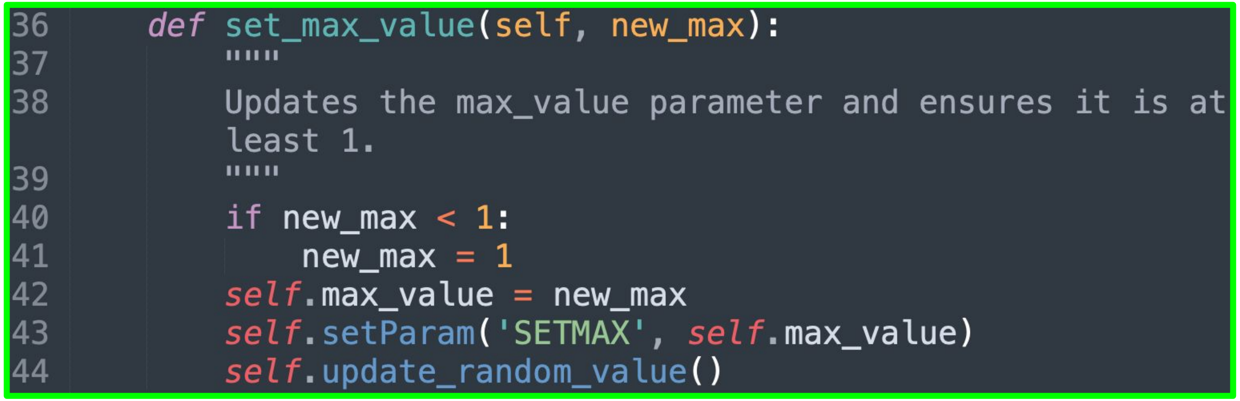
```
[root@bl201-usbserver1 ~]# caget ORNL:RNDVALUE
ORNL:RNDVALUE                      9
```

```
[root@bl201-usbserver1 ~]# caput ORNL:SETMAX 100
Old : ORNL:SETMAX                  10
New : ORNL:SETMAX                  100
```

PCASPy - EPICS - Random Number Generator



```python
    def update_random_value(self):
        """
        Generates a random value between 1 and the current max_value and updates the RNDVALUE PV.
        """
        random_value = random.randint(1, self.max_value)
        self.setParam('RNDVALUE', random_value)
```

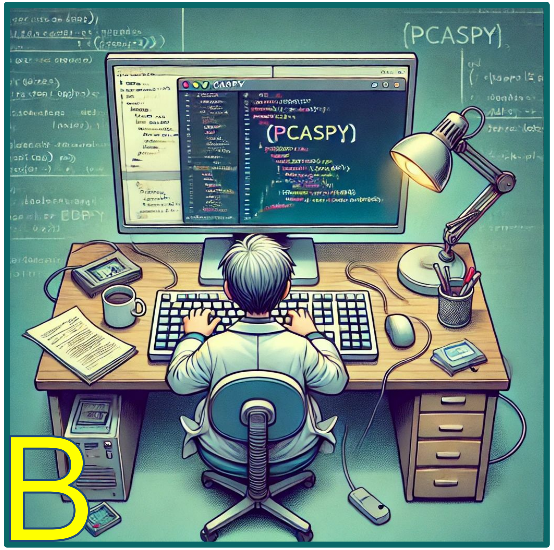


```python
    def set_max_value(self, new_max):
        """
        Updates the max_value parameter and ensures it is at least 1.
        """
        if new_max < 1:
            new_max = 1
        self.max_value = new_max
        self.setParam('SETMAX', self.max_value)
        self.update_random_value()
```

Agenda & Objectives

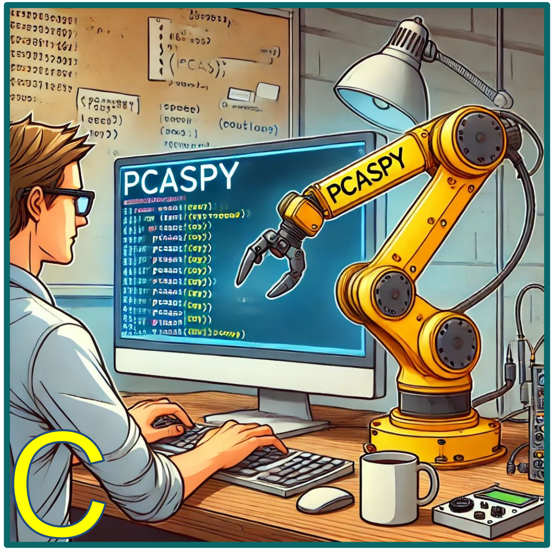
Learn what PCASpy is capable of.

✓



Be able to understand and read a PCASpy implementation.

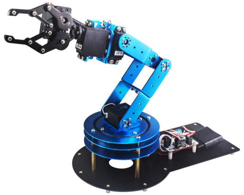
✓



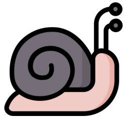
Implement a basic PCASpy driver.

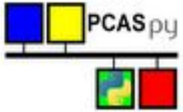
✓

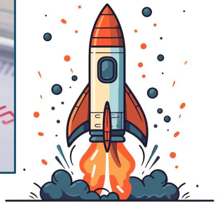
Limitations

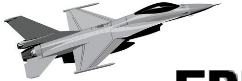


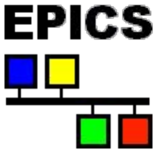
Speed | Trade Off | Real Time Requirements









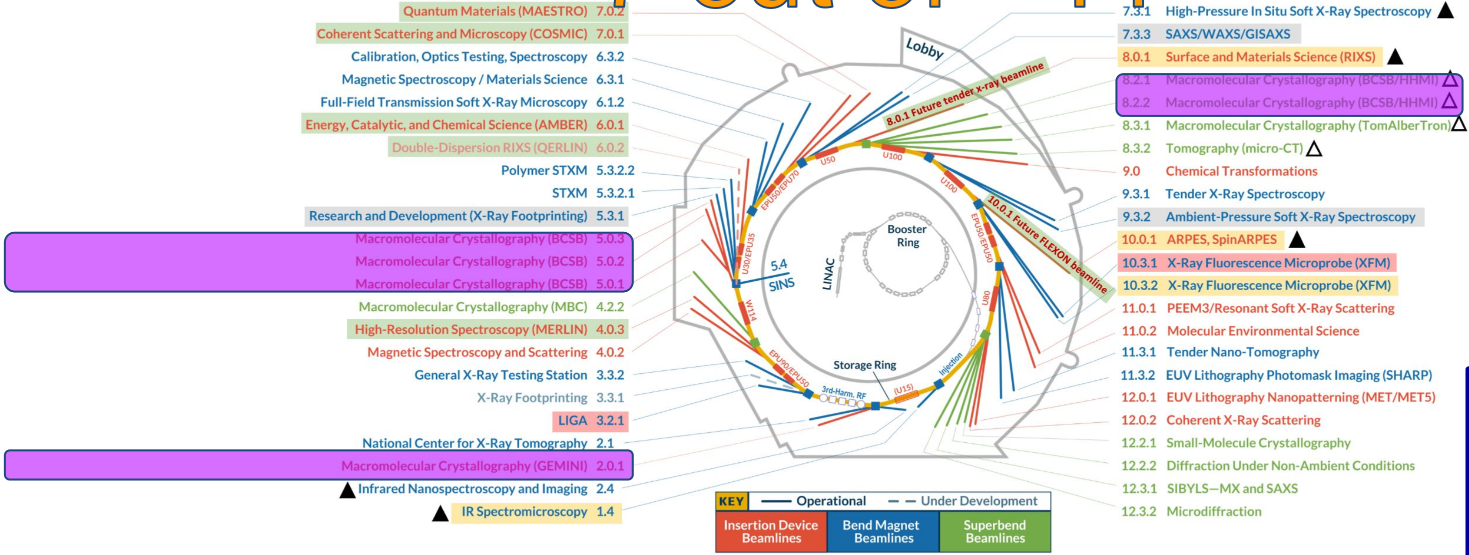


Real Case Scenario at BCSB (LBL)

BCSB

The **Berkeley Center for Structural Biology** brings over 20 years of experience to beamline management and innovation. We operate six high-throughput protein crystallography beamlines at the Advanced Light Source at Lawrence Berkeley National Laboratory.

7 out of ~44



Quantum Materials (MAESTRO) 7.0.2
Coherent Scattering and Microscopy (COSMIC) 7.0.1
Calibration, Optics Testing, Spectroscopy 6.3.2
Magnetic Spectroscopy / Materials Science 6.3.1
Full-Field Transmission Soft X-Ray Microscopy 6.1.2
Energy, Catalytic, and Chemical Science (AMBER) 6.0.1
Double-Dispersion RIXS (QERLIN) 6.0.2
Polymer STXM 5.3.2.2
STXM 5.3.2.1
Research and Development (X-Ray Footprinting) 5.3.1
Macromolecular Crystallography (BCSB) 5.0.3
Macromolecular Crystallography (BCSB) 5.0.2
Macromolecular Crystallography (BCSB) 5.0.1
Macromolecular Crystallography (MBC) 4.2.2
High-Resolution Spectroscopy (MERLIN) 4.0.3
Magnetic Spectroscopy and Scattering 4.0.2
General X-Ray Testing Station 3.3.2
X-Ray Footprinting 3.3.1
LIGA 3.2.1
National Center for X-Ray Tomography 2.1
Macromolecular Crystallography (GEMINI) 2.0.1
▲ Infrared Nanospectroscopy and Imaging 2.4
▲ IR Spectromicroscopy 1.4

7.3.1 High-Pressure In Situ Soft X-Ray Spectroscopy ▲
7.3.3 SAXS/WAXS/GISAXS
8.0.1 Surface and Materials Science (RIXS) ▲
8.2.1 Macromolecular Crystallography (BCSB/HHMI) △
8.2.2 Macromolecular Crystallography (BCSB/HHMI) △
8.3.1 Macromolecular Crystallography (TomAlberTron) △
8.3.2 Tomography (micro-CT) △
9.0 Chemical Transformations
9.3.1 Tender X-Ray Spectroscopy
9.3.2 Ambient-Pressure Soft X-Ray Spectroscopy
10.0.1 ARPES, SpinARPES ▲
10.3.1 X-Ray Fluorescence Microprobe (XFM)
10.3.2 X-Ray Fluorescence Microprobe (XFM)
11.0.1 PEEM3/Resonant Soft X-Ray Scattering
11.0.2 Molecular Environmental Science
11.3.1 Tender Nano-Tomography
11.3.2 EUV Lithography Photomask Imaging (SHARP)
12.0.1 EUV Lithography Nanopatterning (MET/MET5)
12.0.2 Coherent X-Ray Scattering
12.2.1 Small-Molecule Crystallography
12.2.2 Diffraction Under Non-Ambient Conditions
12.3.1 SIBYLS—MX and SAXS
12.3.2 Microdiffraction

KEY — Operational – – Under Development
Insertion Device Beamlines | Bend Magnet Beamlines | Superbend Beamlines

EPICS

microDXP

The microDXP is a complete, low power compact digital spectroscopy card design for a wide range of handheld, benchtop and other embedded applications, lowering cost and speeding time-to-market. Its small size allows for very compact assemblies and its low power consumption assures thermal stability and extended battery life.

Digitization Frequency (MHz) Choose an option

LabVIEW™

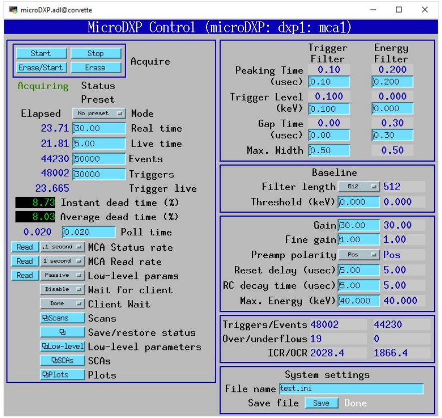
DXP - EPICS software for XIA Digital Signal Processing Systems

Release 6-1

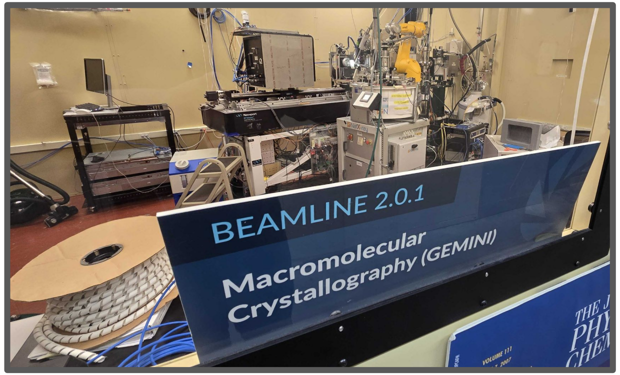
December 7, 2023

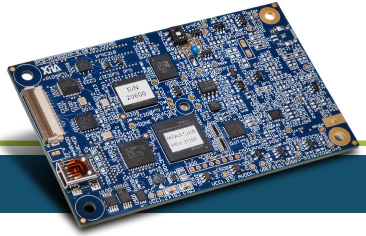
Mark Rivers

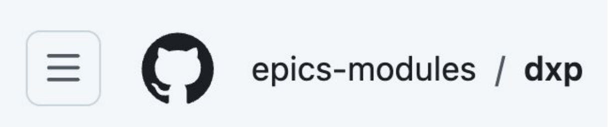
University of Chicago



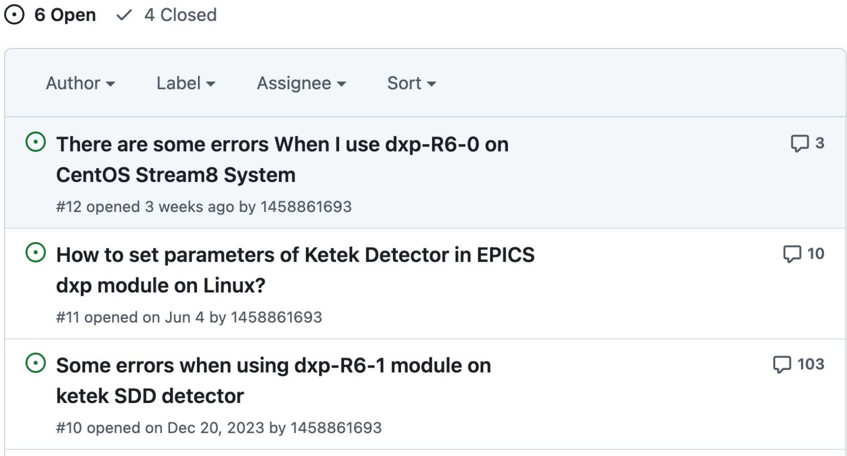
```
MicroDXP Control (microDXP: dxp1: mca1)

[Start] [Stop]               Acquire
[Erase/Start] [Erase]
Acquiring        Status
                 Preset
Elapsed  [No preset]  Mode
  23.71  30.00        Real time
  21.81  5.00         Live time
  44230  50000        Events
  48002  30000        Triggers
 23.665               Trigger live
   8.73  Instant dead time (%)
   8.03  Average dead time (%)
  0.020  0.020        Poll time
[Read] [.1 second]    MCA Status rate
[Read] [1 second]     MCA Read rate
[Read] [Passive]      Low-level params
       [Disable]      Wait for client
       [Done]         Client Wait
       [Scans]        Scans
       [ ]            Save/restore status
       [Low-level]    Low-level parameters
       [SCAs]         SCAs
       [Plots]        Plots

                      Trigger    Energy
                      Filter     Filter
Peaking Time           0.10       0.200
     (usec)            0.10       0.200
Trigger Level          0.100      0.000
      (keV)            0.100      0.000
   Gap Time            0.00       0.30
     (usec)            0.00       0.30
 Max. Width            0.50       0.50

                 Baseline
      Filter length  [512]     512
    Threshold (keV)  0.000     0.000

               Gain  30.00     30.00
          Fine gain  1.00      1.00
    Preamp polarity  [Pos]     Pos
 Reset delay (usec)  5.00      5.00
RC decay time (usec) 5.00      5.00
  Max. Energy (keV)  40.000    40.000

Triggers/Events 48002      44230
Over/underflows 19         0
        ICR/OCR 2028.4     1866.4

             System settings
File name test.ini
  Save file [Save] Done
```







epics-modules / dxp

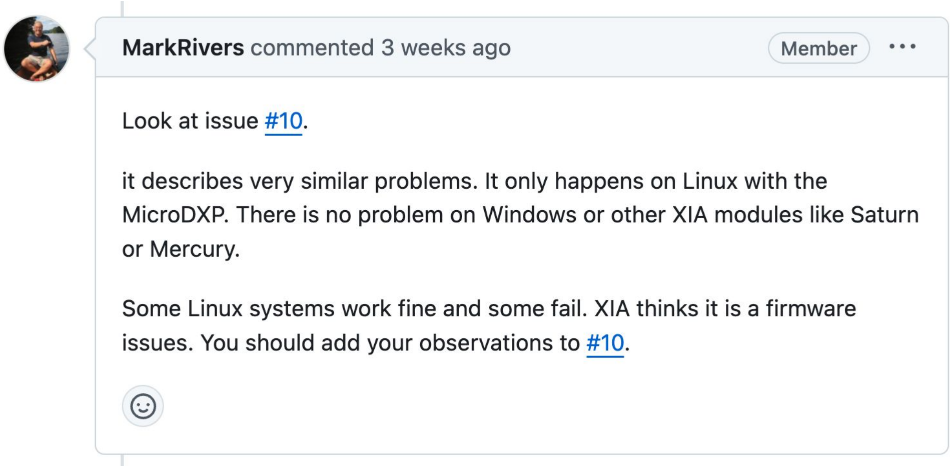


○ 6 Open ✓ 4 Closed

○ **There are some errors When I use dxp-R6-0 on CentOS Stream8 System** 💬 3
#12 opened 3 weeks ago by 1458861693

○ **How to set parameters of Ketek Detector in EPICS dxp module on Linux?** 💬 10
#11 opened on Jun 4 by 1458861693

○ **Some errors when using dxp-R6-1 module on ketek SDD detector** 💬 103
#10 opened on Dec 20, 2023 by 1458861693



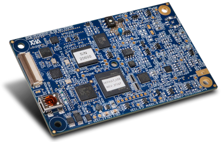
MarkRivers commented 3 weeks ago Member ···

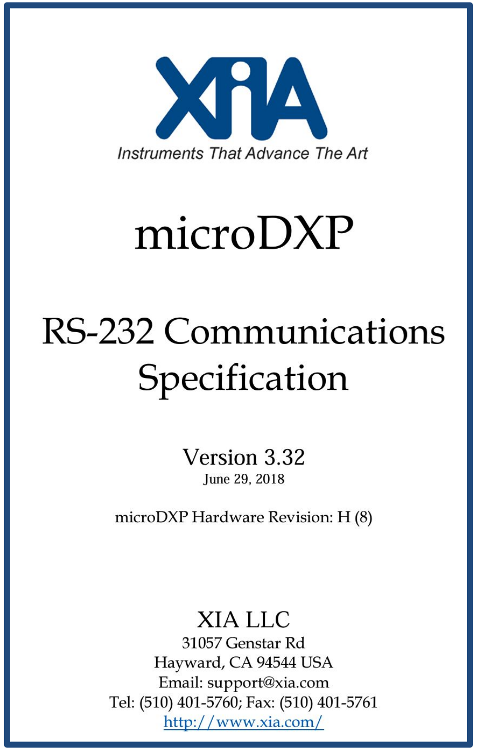
Look at issue #10.

it describes very similar problems. It only happens on Linux with the MicroDXP. There is no problem on Windows or other XIA modules like Saturn or Mercury.

Some Linux systems work fine and some fail. XIA thinks it is a firmware issues. You should add your observations to #10.

Using Serial Communication Over USB





XIA

Instruments That Advance The Art

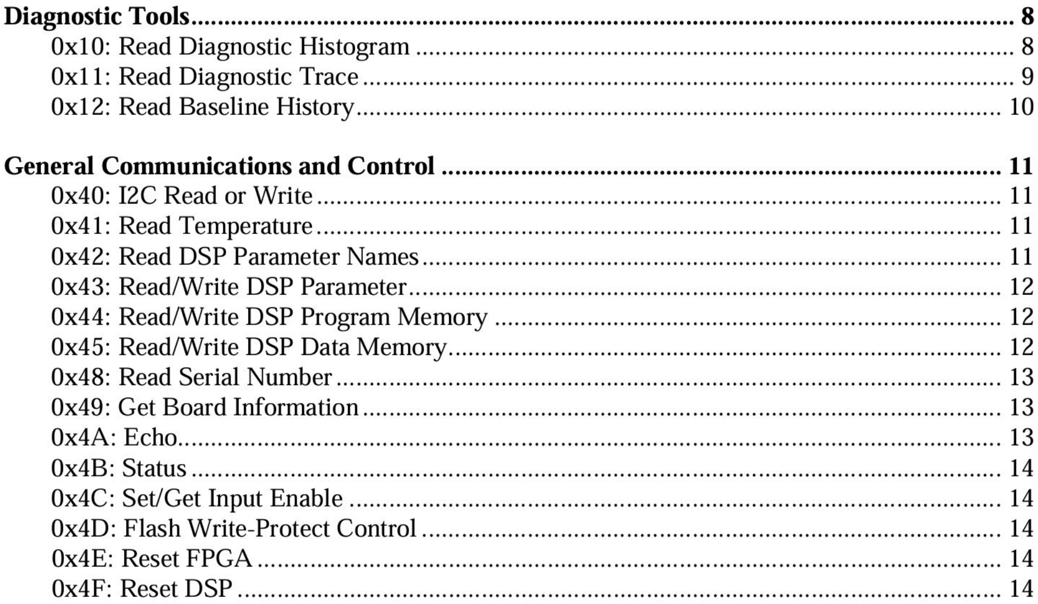
microDXP

RS-232 Communications Specification

Version 3.32
June 29, 2018

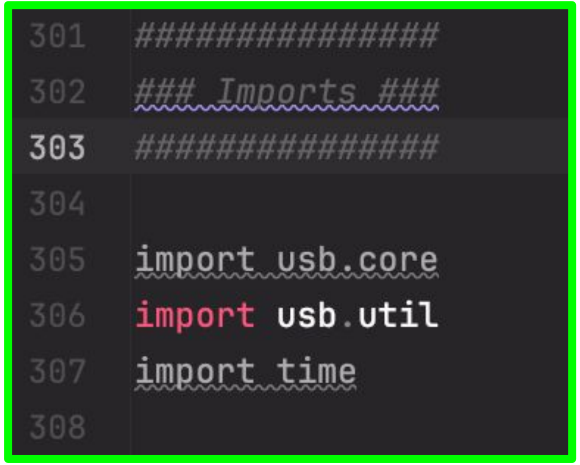
microDXP Hardware Revision: H (8)

XIA LLC
31057 Genstar Rd
Hayward, CA 94544 USA
Email: support@xia.com
Tel: (510) 401-5760; Fax: (510) 401-5761
http://www.xia.com/

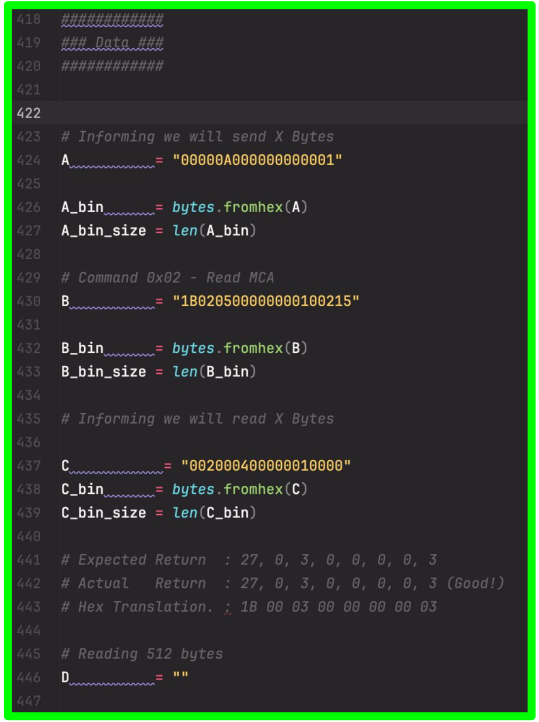


Using Serial Communication Over USB

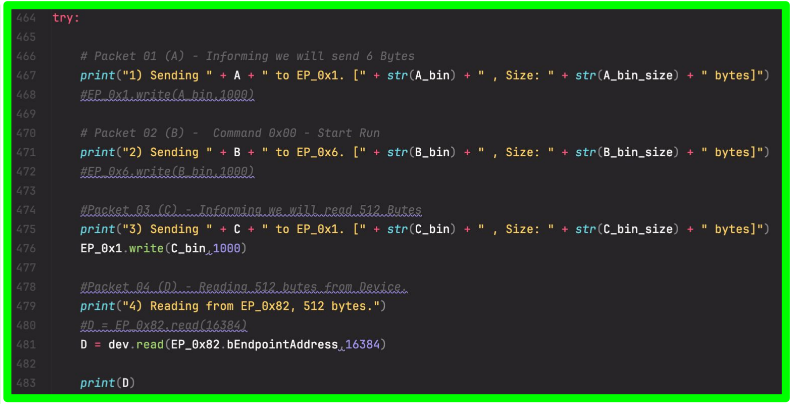
BioSciences



```
################
### Imports ###
################

import usb.core
import usb.util
import time
```

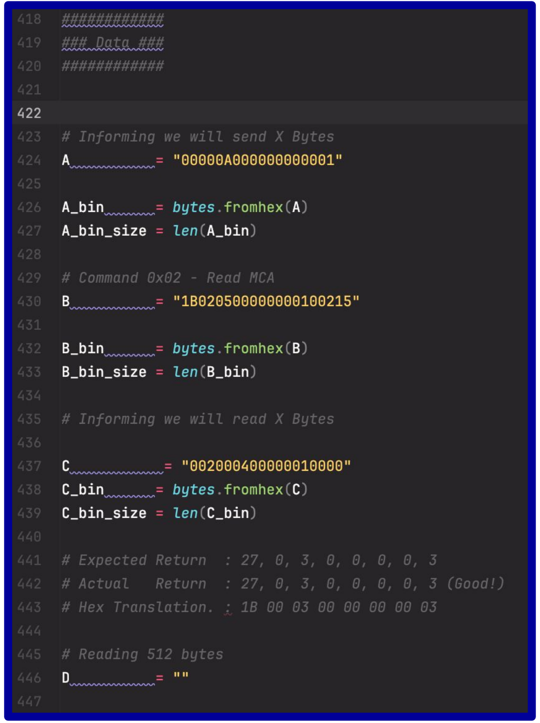


```
############
### Data ###
############


# Informing we will send X Bytes
A          = "00000A000000000001"

A_bin      = bytes.fromhex(A)
A_bin_size = len(A_bin)

# Command 0x02 - Read MCA
B          = "1B020500000000100215"

B_bin      = bytes.fromhex(B)
B_bin_size = len(B_bin)

# Informing we will read X Bytes

C           = "002000400000010000"
C_bin      = bytes.fromhex(C)
C_bin_size = len(C_bin)

# Expected Return  : 27, 0, 3, 0, 0, 0, 0, 3
# Actual   Return  : 27, 0, 3, 0, 0, 0, 0, 3 (Good!)
# Hex Translation. : 1B 00 03 00 00 00 00 03

# Reading 512 bytes
D          = ""

```

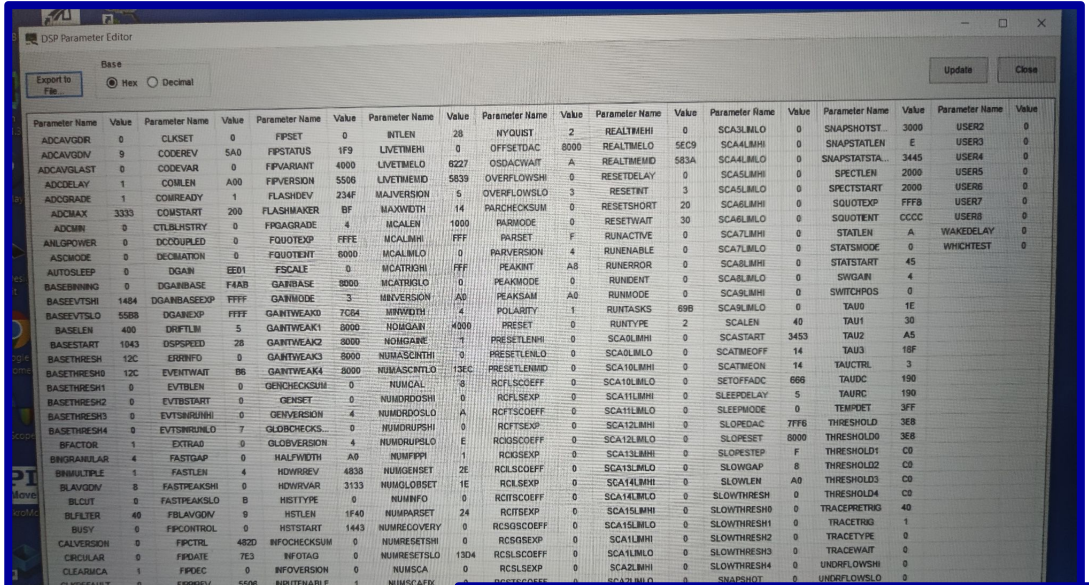


```
try:

    # Packet 01 (A) - Informing we will send 6 Bytes
    print("1) Sending " + A + " to EP_0x1. [" + str(A_bin) + " , Size: " + str(A_bin_size) + " bytes]")
    #EP_0x1.write(A_bin,1000)

    # Packet 02 (B) -  Command 0x00 - Start Run
    print("2) Sending " + B + " to EP_0x6. [" + str(B_bin) + " , Size: " + str(B_bin_size) + " bytes]")
    #EP_0x6.write(B_bin,1000)

    #Packet 03 (C) - Informing we will read 512 Bytes
    print("3) Sending " + C + " to EP_0x1. [" + str(C_bin) + " , Size: " + str(C_bin_size) + " bytes]")
    EP_0x1.write(C_bin,1000)

    #Packet 04 (D) - Reading 512 bytes from Device.
    print("4) Reading from EP_0x82, 512 bytes.")
    #D = EP_0x82.read(16384)
    D = dev.read(EP_0x82.bEndpointAddress,16384)

    print(D)
```

Using Serial Communication Over USB

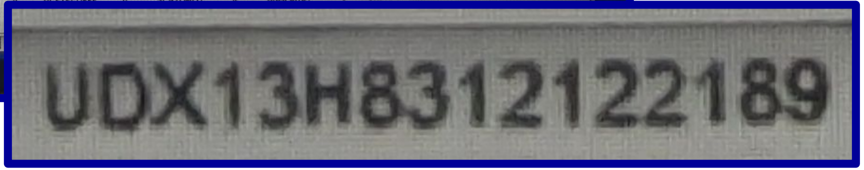


```
############
### Data ###
############


# Informing we will send X Bytes
A          = "00000A000000000001"

A_bin      = bytes.fromhex(A)
A_bin_size = len(A_bin)

# Command 0x02 - Read MCA
B          = "1B020500000000100215"

B_bin      = bytes.fromhex(B)
B_bin_size = len(B_bin)

# Informing we will read X Bytes

C           = "002000400000010000"
C_bin      = bytes.fromhex(C)
C_bin_size = len(C_bin)

# Expected Return  : 27, 0, 3, 0, 0, 0, 0, 3
# Actual   Return  : 27, 0, 3, 0, 0, 0, 0, 3 (Good!)
# Hex Translation. : 1B 00 03 00 00 00 00 03

# Reading 512 bytes
D          = ""

```



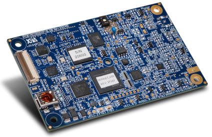
Base: ● Hex ○ Decimal

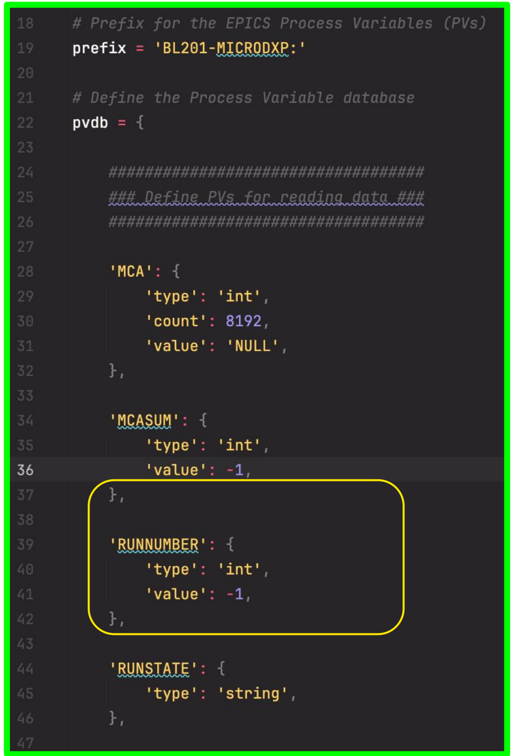
Parameter Name	Value	Parameter Name	Value	Parameter Name	Value	Parameter Name	Value	Parameter Name	Value	Parameter Name	Value	Parameter Name	Value	Parameter Name	Value	Parameter Name	Value
ADCAVGDIR	0	CLKSET	0	FIPSET	0	INTLEN	28	NYQUIST	2	REALTIMEHI	0	SCA3LIMLO	0	SNAPSHOTST...	3000	USER2	0
ADCAVGDIV	9	CODEREV	5A0	FIPSTATUS	1F9	LIVETIMEHI	0	OFFSETDAC	8000	REALTIMELO	5EC9	SCA4LIMHI	0	SNAPSTATLEN	E	USER3	0
ADCAVGLAST	0	CODEVAR	0	FIPVARIANT	4000	LIVETIMELO	6227	OSDACWAIT	A	REALTIMEMID	583A	SCA4LIMLO	0	SNAPSTATSTA...	3445	USER4	0
ADCDELAY	1	COMLEN	A00	FIPVERSION	5506	LIVETIMEMID	5839	OVERFLOWSHI	0	RESETDELAY	0	SCA5LIMHI	0	SPECTLEN	2000	USER5	0
ADCGRADE	1	COMREADY	1	FLASHDEV	234F	MAJVERSION	5	OVERFLOWSLO	3	RESETINT	3	SCA5LIMLO	0	SPECTSTART	2000	USER6	0
ADCMAX	3333	COMSTART	200	FLASHMAKER	BF	MAXWIDTH	14	PARCHECKSUM	0	RESETSHORT	20	SCA6LIMHI	0	SQUOTEXP	FFF8	USER7	0
ADCMIN	0	CTLBLHSTRY	0	FPGAGRADE	4	MCALEN	1000	PARMODE	0	RESETWAIT	30	SCA6LIMLO	0	SQUOTIENT	CCCC	USER8	0
ANLGPOWER	0	DCCOUPLED	0	FQUOTEXP	FFFE	MCALIMHI	FFF	PARSET	F	RUNACTIVE	0	SCA7LIMHI	0	STATLEN	A	WAKEDELAY	0
ASCMODE	0	DECIMATION	0	FQUOTIENT	8000	MCALIMLO	0	PARVERSION	4	RUNENABLE	0	SCA7LIMLO	0	STATSMODE	0	WHICHTEST	0
AUTOSLEEP	0	DGAIN	EE01	FSCALE	0	MCATRIGHI	FFF	PEAKINT	A8	RUNERROR	0	SCA8LIMHI	0	STATSTART	45		
BASEBINNING	0	DGAINBASE	F4AB	GAINBASE	8000	MCATRIGLO	0	PEAKMODE	0	RUNIDENT	0	SCA8LIMLO	0	SWGAIN	4		
BASEEVTSHI	1484	DGAINBASEEXP	FFFF	GAINMODE	3	MINVERSION	A0	PEAKSAM	A0	RUNMODE	0	SCA9LIMHI	0	SWITCHPOS	0		
BASEEVTSLO	55B8	DGAINEXP	FFFF	GAINTWEAK0	7C84	MINWIDTH	4	POLARITY	1	RUNTASKS	69B	SCA9LIMLO	0	TAU0	1E		
BASELEN	400	DRIFTLIM	5	GAINTWEAK1	8000	NOMGAIN	4000	PRESET	0	RUNTYPE	2	SCALEN	40	TAU1	30		
BASESTART	1043	DSPSPEED	28	GAINTWEAK2	8000	NOMGAINE	1	PRESETLENHI	0	SCA0LIMHI	0	SCASTART	3453	TAU2	A5		
BASETHRESH	12C	ERRINFO	0	GAINTWEAK3	8000	NUMASCINTHI	0	PRESETLENLO	0	SCA0LIMLO	0	SCATIMEOFF	14	TAU3	18F		
BASETHRESH0	12C	EVENTWAIT	B6	GAINTWEAK4	8000	NUMASCINTLO	13EC	PRESETLENMID	0	SCA10LIMHI	0	SCATIMEON	14	TAUCTRL	3		
BASETHRESH1	0	EVTBLEN	0	GENCHECKSUM	0	NUMCAL	8	RCFLSCOEFF	0	SCA10LIMLO	0	SETOFFADC	666	TAUDC	190		
BASETHRESH2	0	EVTBSTART	0	GENSET	0	NUMDRDOSHI	0	RCFLSEXP	0	SCA11LIMHI	0	SLEEPDELAY	5	TAURC	190		
BASETHRESH3	0	EVTSINRUNHI	0	GENVERSION	4	NUMDRDOSLO	A	RCFTSCOEFF	0	SCA11LIMLO	0	SLEEPMODE	0	TEMPDET	3FF		
BASETHRESH4	0	EVTSINRUNLO	7	GLOBCHECKS...	0	NUMDRUPSHI	0	RCFTSEXP	0	SCA12LIMHI	0	SLOPEDAC	7FF6	THRESHOLD	3E8		
BFACTOR	1	EXTRA0	0	GLOBVERSION	4	NUMDRUPSLO	E	RCIGSCOEFF	0	SCA12LIMLO	0	SLOPESET	8000	THRESHOLD0	3E8		
BINGRANULAR	4	FASTGAP	0	HALFWIDTH	A0	NUMFIPPI	1	RCIGSEXP	0	SCA13LIMHI	0	SLOPESTEP	F	THRESHOLD1	C0		
BINMULTIPLE	1	FASTLEN	4	HDWRREV	4838	NUMGENSET	2E	RCILSCOEFF	0	SCA13LIMLO	0	SLOWGAP	8	THRESHOLD2	C0		
BLAVGDIV	8	FASTPEAKSHI	0	HDWRVAR	3133	NUMGLOBSET	1E	RCILSEXP	0	SCA14LIMHI	0	SLOWLEN	A0	THRESHOLD3	C0		
BLCUT	0	FASTPEAKSLO	B	HISTTYPE	0	NUMINFO	0	RCITSCOEFF	0	SCA14LIMLO	0	SLOWTHRESH	0	THRESHOLD4	C0		
BLFILTER	40	FBLAVGDIV	9	HSTLEN	1F40	NUMPARSET	24	RCITSEXP	0	SCA15LIMHI	0	SLOWTHRESH0	0	TRACEPRETRIG	40		
BUSY	0	FIPCONTROL	0	HSTSTART	1443	NUMRECOVERY	0	RCSGSCOEFF	0	SCA15LIMLO	0	SLOWTHRESH1	0	TRACETRIG	1		
CALVERSION	0	FIPCTRL	482D	INFOCHECKSUM	0	NUMRESETSHI	0	RCSGSEXP	0	SCA1LIMHI	0	SLOWTHRESH2	0	TRACETYPE	0		
CIRCULAR	0	FIPDATE	7E3	INFOTAG	0	NUMRESETSLO	13D4	RCSLSCOEFF	0	SCA1LIMLO	0	SLOWTHRESH3	0	TRACEWAIT	0		
CLEARMCA	1	FIPDEC	0	INFOVERSION	0	NUMSCA	0	RCSLSEXP	0	SCA2LIMHI	0	SLOWTHRESH4	0	UNDRFLOWSHI	0		
CLKDEFAULT	0	FIPPIREV	5506	INPUTENABLE	1	NUMSCAFIX	0	RCSTSCOEFF	0	SCA2LIMLO	0	SNAPSHOT	0	UNDRFLOWSLO	0		
CLKENABLE	1	FIPPIVAR	4000	INTGAP	2	NUMSNAPSHO...	[illegible]										



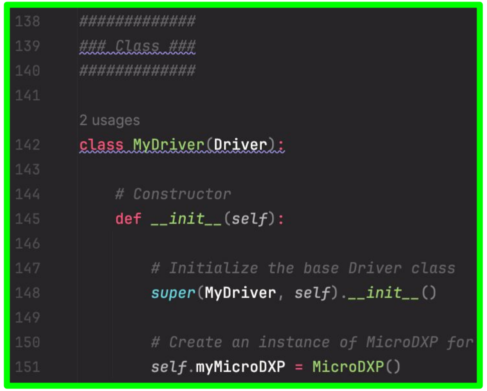
UDX13H83121222189

caget BL201-MICRODXP:RUNNUMBER

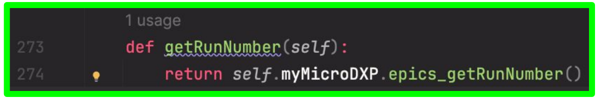




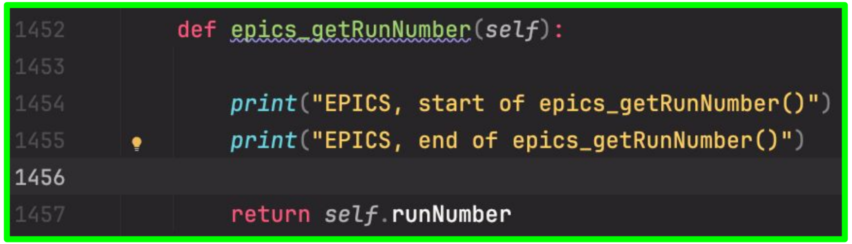
```python
# Prefix for the EPICS Process Variables (PVs)
prefix = 'BL201-MICRODXP:'

# Define the Process Variable database
pvdb = {

    ###################################
    ### Define PVs for reading data ###
    ###################################

    'MCA': {
        'type': 'int',
        'count': 8192,
        'value': 'NULL',
    },

    'MCASUM': {
        'type': 'int',
        'value': -1,
    },

    'RUNNUMBER': {
        'type': 'int',
        'value': -1,
    },

    'RUNSTATE': {
        'type': 'string',
    },

```



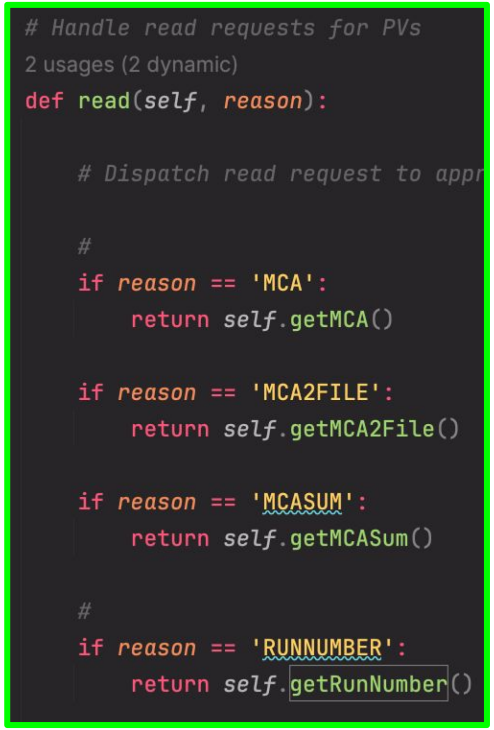
```python
#############
### Class ###
#############

2 usages
class MyDriver(Driver):

    # Constructor
    def __init__(self):

        # Initialize the base Driver class
        super(MyDriver, self).__init__()

        # Create an instance of MicroDXP for
        self.myMicroDXP = MicroDXP()
```



```python
    1 usage
    def getRunNumber(self):
        return self.myMicroDXP.epics_getRunNumber()
```

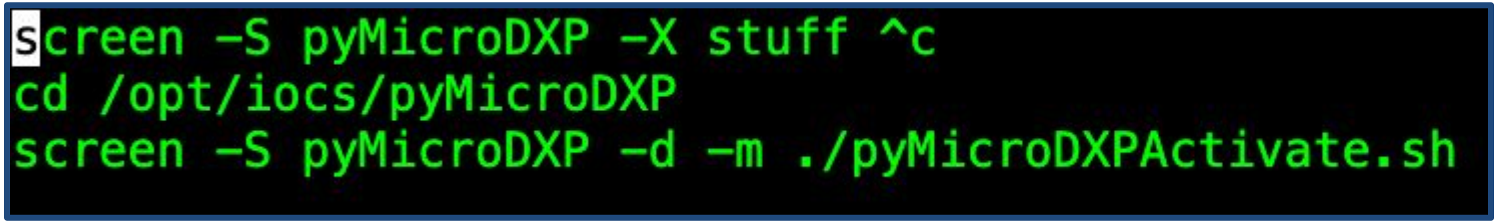


```python
    def epics_getRunNumber(self):

        print("EPICS, start of epics_getRunNumber()")
        print("EPICS, end of epics_getRunNumber()")

        return self.runNumber
```



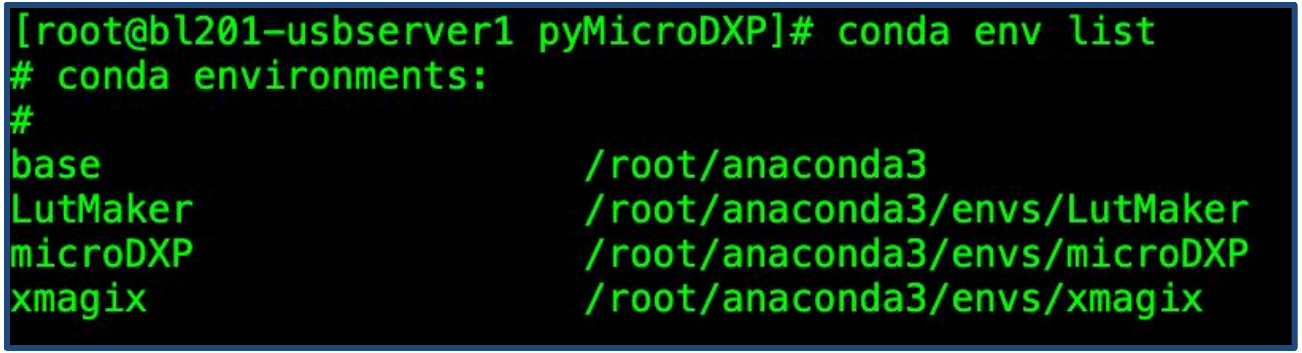
```python
# Handle read requests for PVs
2 usages (2 dynamic)
def read(self, reason):

    # Dispatch read request to appr

    #
    if reason == 'MCA':
        return self.getMCA()

    if reason == 'MCA2FILE':
        return self.getMCA2File()

    if reason == 'MCASUM':
        return self.getMCASum()

    #
    if reason == 'RUNNUMBER':
        return self.getRunNumber()
```

Tip: Running the IOC inside a Conda Environment



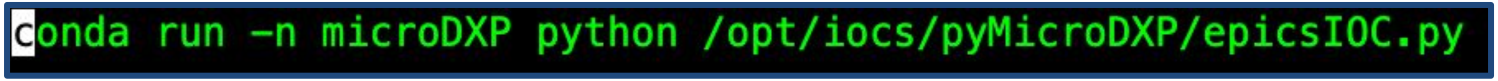
```
screen -S pyMicroDXP -X stuff ^c
cd /opt/iocs/pyMicroDXP
screen -S pyMicroDXP -d -m ./pyMicroDXPActivate.sh
```

Virtual Session



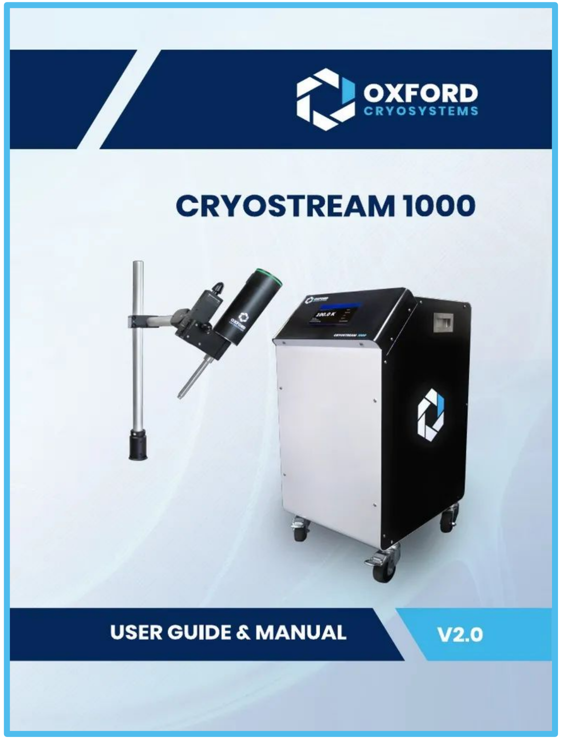
```
[root@bl201-usbserver1 pyMicroDXP]# conda env list
# conda environments:
#
base                     /root/anaconda3
LutMaker                 /root/anaconda3/envs/LutMaker
microDXP                 /root/anaconda3/envs/microDXP
xmagix                   /root/anaconda3/envs/xmagix
```

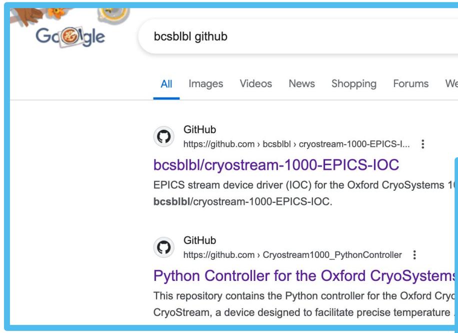
Listing Virtual Environments



```
conda run -n microDXP python /opt/iocs/pyMicroDXP/epicsIOC.py
```

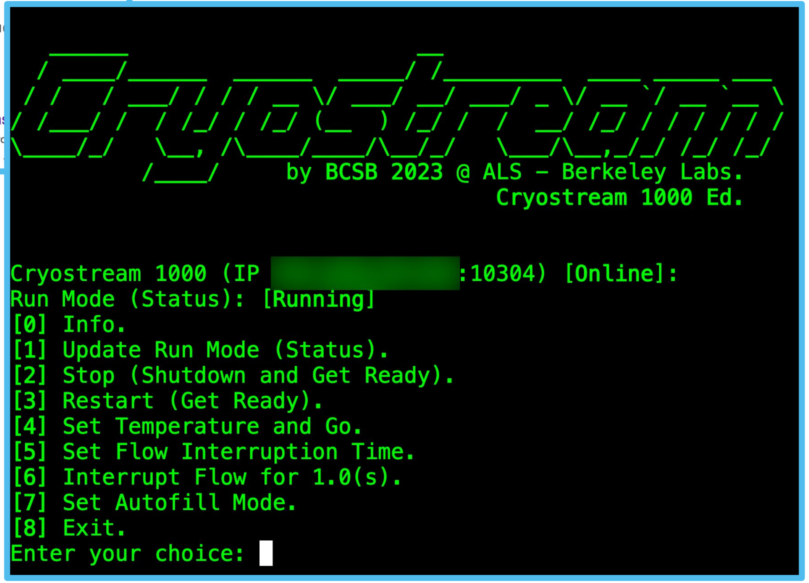
Running in the background inside a Conda Environment





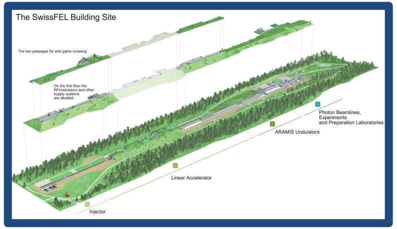
GitHub
https://github.com › bcsblbl › cryostream-1000-EPICS-I...
bcsblbl/cryostream-1000-EPICS-IOC
EPICS stream device driver (IOC) for the Oxford CryoSystems 1
bcsblbl/cryostream-1000-EPICS-IOC.

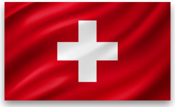
GitHub
https://github.com › Cryostream1000_PythonController
Python Controller for the Oxford CryoSystems
This repository contains the Python controller for the Oxford Cryo
CryoStream, a device designed to facilitate precise temperature

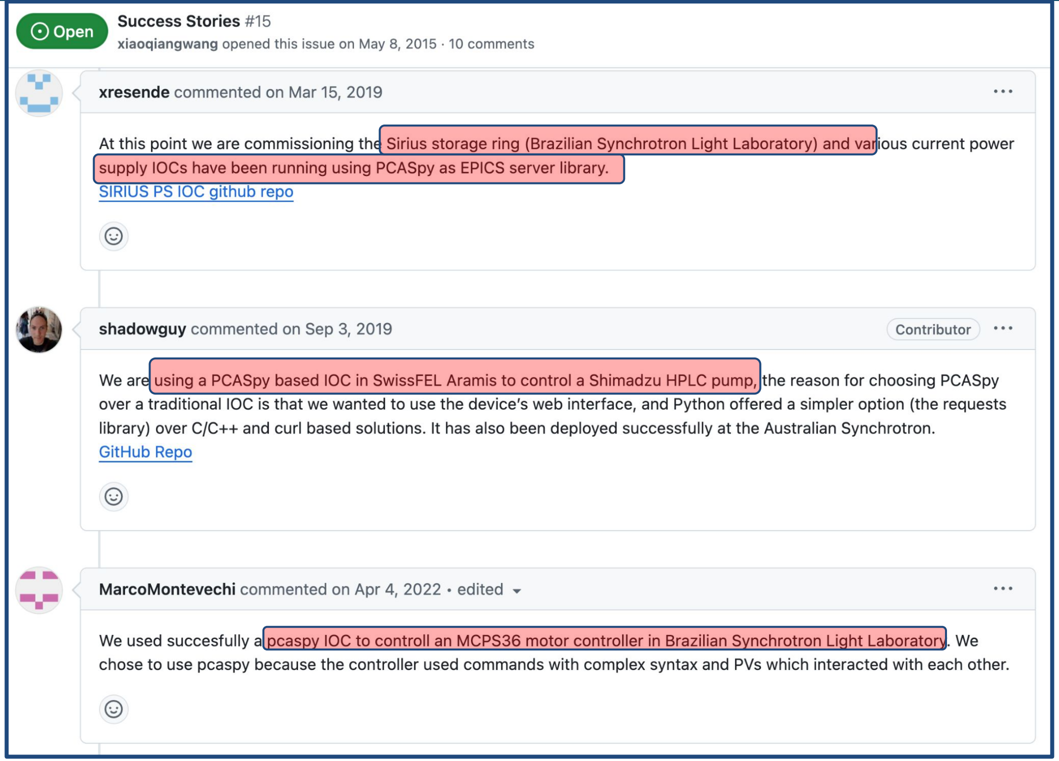


```
   ______                     __
  / ____/______  ______  _____/ /_________  ____ _____ ___
 / /   / ___/ / / / __ \/ ___/ __/ ___/ _ \/ __ `/ __ `__ \
/ /___/ /  / /_/ / /_/ (__  ) /_/ /  /  __/ /_/ / / / / / /
\____/_/   \__, /\____/____/\__/_/   \___/\__,_/_/ /_/ /_/
          /____/      by BCSB 2023 @ ALS - Berkeley Labs.
                                   Cryostream 1000 Ed.


Cryostream 1000 (IP [redacted]:10304) [Online]:
Run Mode (Status): [Running]
[0] Info.
[1] Update Run Mode (Status).
[2] Stop (Shutdown and Get Ready).
[3] Restart (Get Ready).
[4] Set Temperature and Go.
[5] Set Flow Interruption Time.
[6] Interrupt Flow for 1.0(s).
[7] Set Autofill Mode.
[8] Exit.
Enter your choice: 
```

Some Success Stories from GitHub


The SwissFEL Building Site
The two passages for wild game crossing
On the first floor the RF modulators and other supply systems are situated
Photon Beamlines, Experiments and Preparation Laboratories
ARAMIS Undulators
Linear Accelerator
Injector





Success Stories #15

xiaoqiangwang opened this issue on May 8, 2015 · 10 comments

xresende commented on Mar 15, 2019

At this point we are commissioning the Sirius storage ring (Brazilian Synchrotron Light Laboratory) and various current power supply IOCs have been running using PCASpy as EPICS server library.

SIRIUS PS IOC github repo

shadowguy commented on Sep 3, 2019 · Contributor

We are using a PCASpy based IOC in SwissFEL Aramis to control a Shimadzu HPLC pump, the reason for choosing PCASpy over a traditional IOC is that we wanted to use the device's web interface, and Python offered a simpler option (the requests library) over C/C++ and curl based solutions. It has also been deployed successfully at the Australian Synchrotron.

GitHub Repo

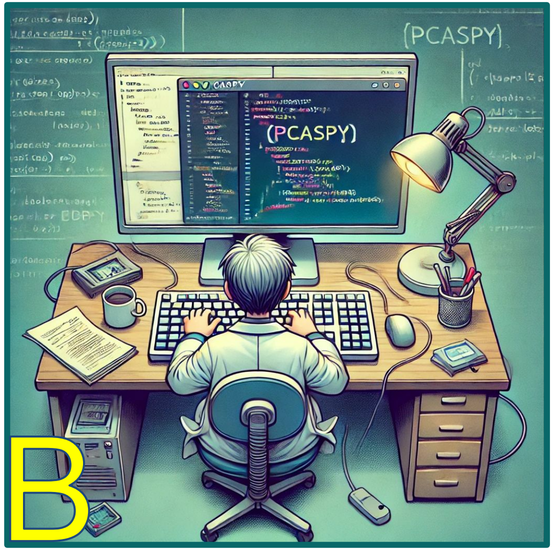
MarcoMontevechi commented on Apr 4, 2022 • edited

We used succesfully a pcaspy IOC to controll an MCPS36 motor controller in Brazilian Synchrotron Light Laboratory. We chose to use pcaspy because the controller used commands with complex syntax and PVs which interacted with each other.

Agenda & Objectives

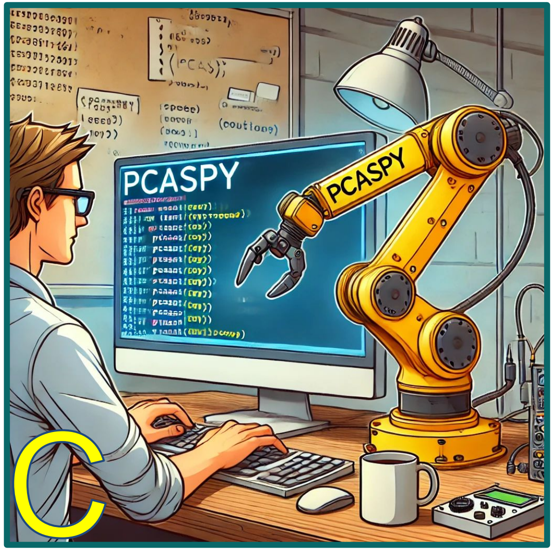
Learn what PCASpy is capable of.

✓



Be able to understand and read a PCASpy implementation.

✓



Implement a basic PCASpy driver.

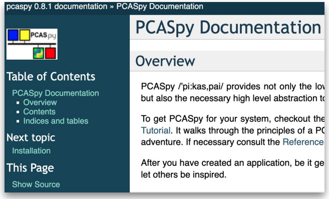
✓

Closing Remarks (1/2)

- PCASPy simplifies certain aspects of working with EPICS but introduces its own set of challenges in the process.
- Additional effort is required to establish Python code.
- Best suited for non-real-time critical applications due to its Python foundation.

Closing Remarks (2/2)

- Not a universal solution but useful as a tool for specific cases requiring new driver implementation.



pcaspy 0.8.1 documentation » PCASpy Documentation

PCASpy Documentation

Overview

PCASpy /'pi:kas,pai/ provides not only the low
but also the necessary high level abstraction to

To get PCASpy for your system, checkout the
Tutorial. It walks through the principles of a PC
adventure. If necessary consult the Reference

After you have created an application, be it ge
let others be inspired.

Table of Contents

PCASpy Documentation

- Overview
- Contents
- Indices and tables

Next topic

Installation

This Page

Show Source

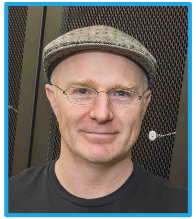
- This presentation covered the basics; there are many more advanced features to explore and learn.

Acknowledgments

The Berkeley Center for Structural Biology (BCSB) is supported by the Howard Hughes Medical Institute, Participating Research Team members, and the National Institutes of Health, National Institute of General Medical Sciences, ALS-ENABLE grant P30 GM124169. The Advanced Light Source is a Department of Energy Office of Science User Facility under Contract No. DE-AC02-05CH11231.

Questions and Discussion

Software and Instrumentation (SWI) Group at BCSB.



John Taylor

Head of SWI

jrtaylor@lbl.gov

Gabriel Gazolla

Computer Systems Engineer

gabrielgazolla@lbl.gov