HZB BESSY II Light Source

kiwi-scan: A Modular Scan Framework for Commissioning and Diagnostics in EPICS Environments

Andreas Balzer
Parvathi Sreelatha Devi

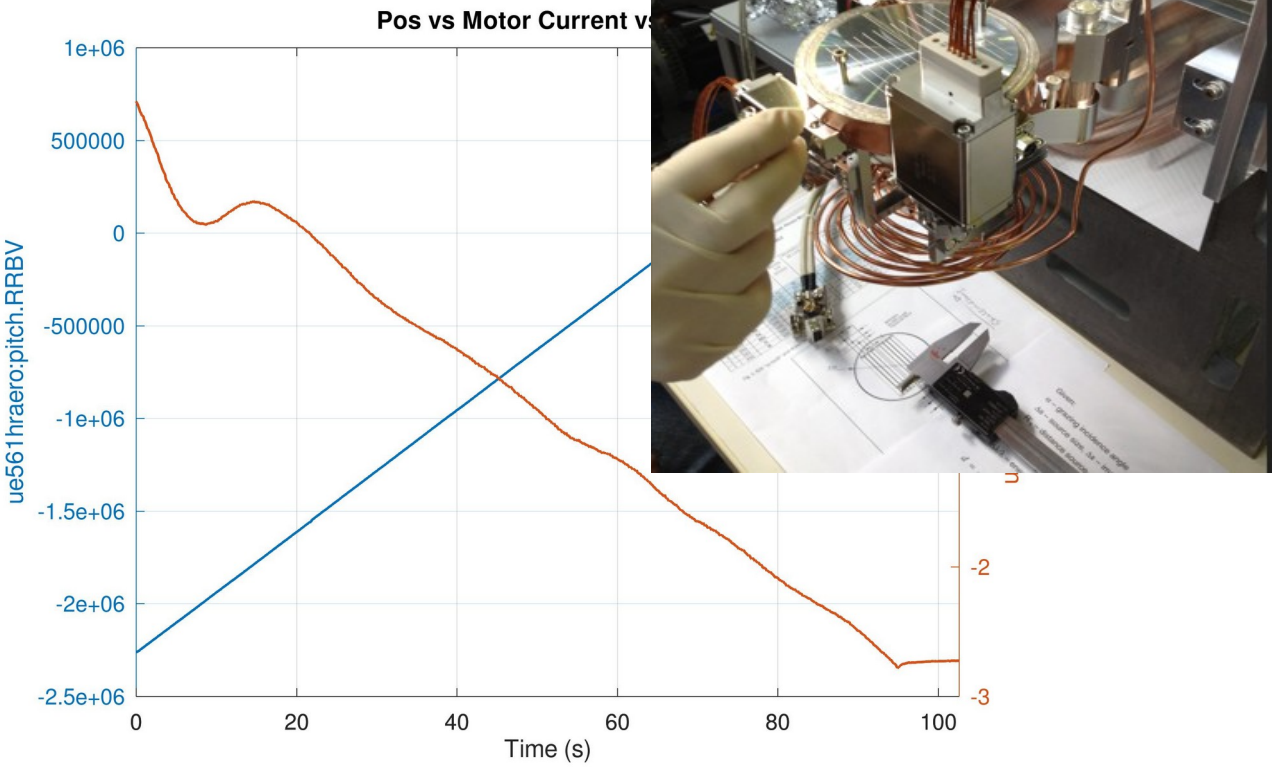
EPICS Spring Meeting 2026

Motivation & Background

- Unify legacy test scripts into one framework
- Flexible EPICS Wrapper
  (monitoring, logging, testing)
- Support IOC debugging
  and motion control
- Avoid string-based code
  (reduce device-specific implementations)
- A playground for testing design ideas
  before porting to C++
- Automate diagnostic scans
  and provide as a service

From scripts to framework

- Many legacy scripts:
  hard to reuse and maintain
- Growing complexity and requirements
- Move to a YAML-driven scan engine



Use Case: Initial motion control tests at FemtoSlicing FS HR Beamline at BESSY II
Photo: Karsten Holldack, Department Optics and Beamlines

kiwi-scan Features

HZB BESSY II Light Source

kiwi-scan

- Lightweight scan engine (everything in repo ~10k lines)
- Actuators + scan engine + YAML config
- Declarative scan definitions
- Continuously evolving
- Source: **`github.com/hz-b/kiwi-scan`**

Features

- Stable, modular, reusable core
- Flexible and extensible
- Real-time monitoring & events
- CLI tools, logging, tests
- Multi-actuator & custom scans
- Triggers, continuous mode
- Metadata, waveforms, plotting

Scan Loop Example Flows

HZB BESSY II Light Source

Continuous Event-Driven Stream

Scan Engine *(pluggable)*

start motion (velocity)

Actuators **Continuous Move**

Subscription Manager

sync

SyncController **→ wait for n PVs**

→ Parallel event processing
→ TriggerManager
→ Plugin Hooks
→ DAQ
→ Meta Data Monitor

Step Scan

Scan Engine *(pluggable)*

for each point

Move actuators to setpoint

Parallel wait until done

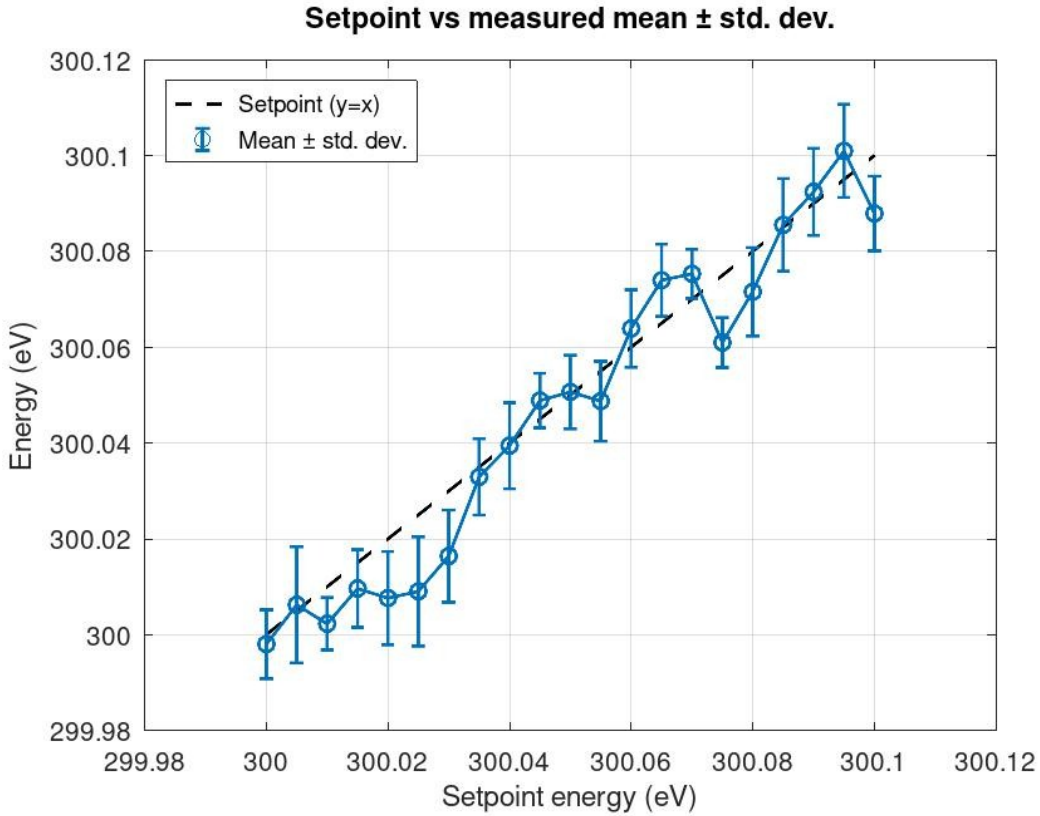
in position/delay

Triggers

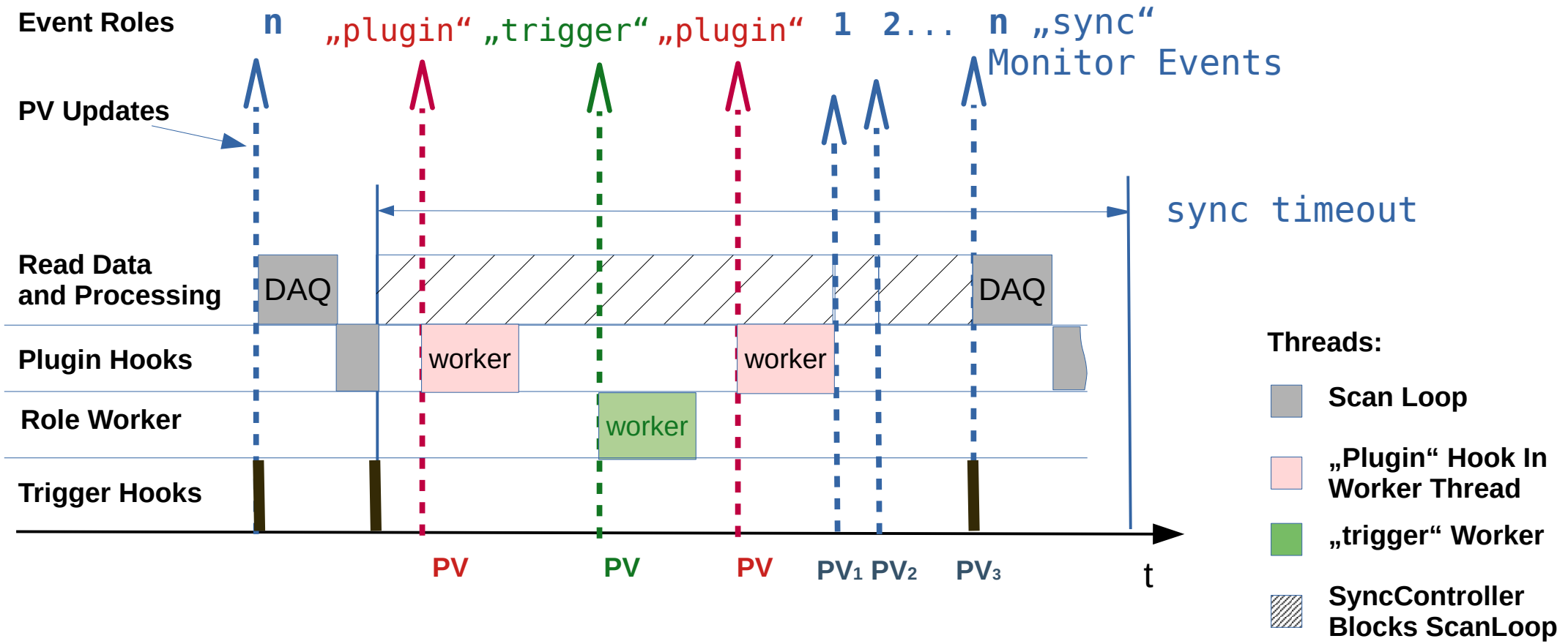
trigger delay

DAQ

→ Event processed data during integration time
→ TriggerManager
→ Plugin Hooks
→ Meta Data Monitor



Subscriptions, SyncController and Trigger Manager



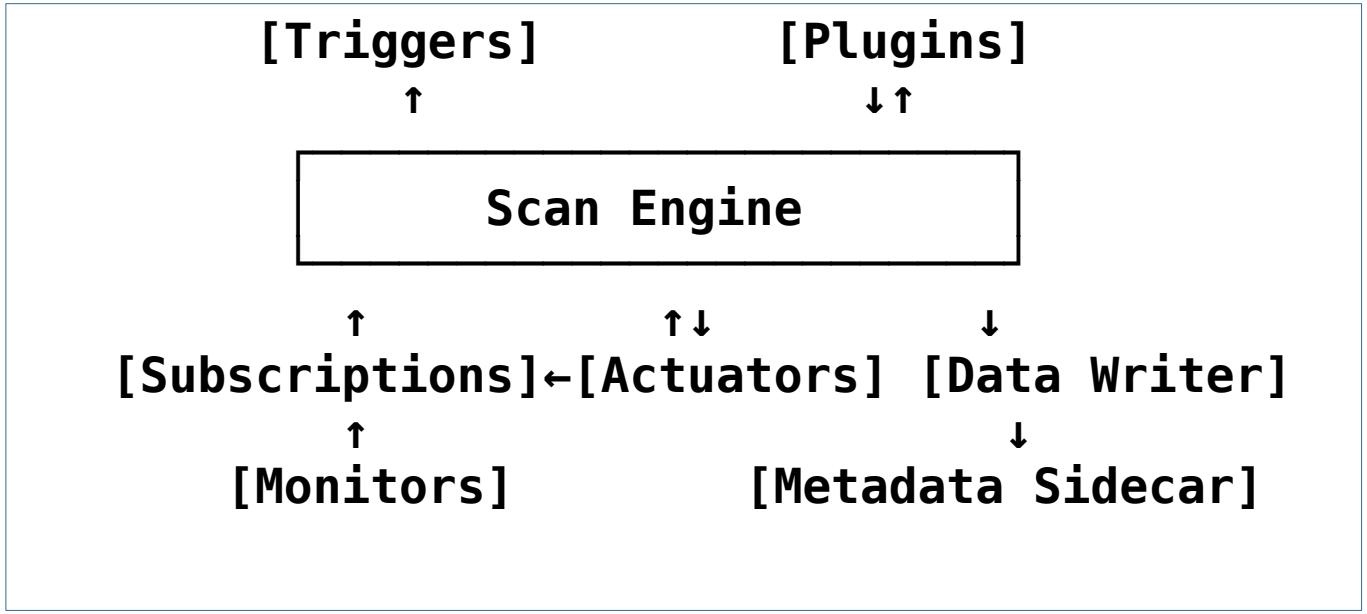
- Subscriptions route monitored events into defined roles.
- Triggers allow actions before, or after scan points.

Architecture Overview

HZB BESSY II Light Source

Event-driven, modular system, simple direct execution model

- **Scan engines**
  (linear, CM, poll, …, pluggable)
- **Actuators**
  (EPICS PVs / simulation)
- **Triggers**
  (before / on_point / after / monitor)
- **Subscriptions**
  (role-based event routing)
- **Plugins**
  (extend behavior, add data, hooks)
- **Data**
  (data writer + metadata sidecar)
- **Queue Plotter**



Event-Driven Scan Architecture

YAML Interface

HZB BESSY II Light Source

Python: engine
YAML: configuration

- Actuators
- Detector PVs
- Triggers
- Metadata (PVs + constants)
- Subscriptions / Role Events
- Plugins
- Plot

Goal:
- Reusable readable setups
- No beamline specific python

```
actuators:
  theta:
    type: "epics"
    pv: "${IOC_NAME}:theta.VAL"
    rb_pv: "${IOC_NAME}:theta.RBV"
    status_pv: "${IOC_NAME}:theta.MSTA"
    ready_value:  0x2
    ready_bitmask: 0x2
    dwell_time: 1.0
    queueing_delay: 0.01
    in_position_band: 0.000001
  beta:
    ...
  energy:
    ...
  slit:
    ...
detector_pvs:
  - "MDIZ3T5G:current"
  ...
metadata_pvs:
  - "${IOC_NAME}:cff"
metadata_constants:
  ...
```

```
triggers:
  before:
    ...
  on_point:
    - pv: ${IOC_NAME}:SomeAction.PROC 1
      value: 1
      delay: 0.01
  after:
    - pv: "UE52ID5R:DiagSpdSet"
      value: [1.0, 1.0]
plugin_configs:
  - type: UndulatorPlugin
    ...
subscriptions:
  - pv: "${IOC_NAME}:idGetGap"
    name: ioc_heartbeat
    role: plugin
  - actuator: energy
    source: rbv
    name: mono_energy
    role: sync
monitor_type: print
stop_pv: ${IOC_NAME}:scan:stop
data_dir: "scandata"
output_file: "example.txt"
include_timestamps: True
debug: False
performance_report: True
```

API - Integration into Python Frameworks

HZB BESSY II Light Source

```python
import kiwi_scan
from kiwi_scan.datamodels import ScanConfig, ActuatorConfig, ScanDimension
from kiwi_scan.scan.tools import create_scan_with_config
import logging

kiwi_scan.load_all_scan_types()
kiwi_scan.load_all_plugins()

def create_config():
    return ScanConfig(
        actuators={"x": ActuatorConfig(type="sim", pv="X")},
        detector_pvs=[],
        scan_dimensions=[ScanDimension("x", 0, 10, 5)],
    )

scan = create_scan_with_config("linear", create_config())
scan.execute()
```

- Embedded in Python applications:
  - `execute()`

- Usage with asyncio (IOCs / state machines, …):
  - `run_in_executor()` or
  - `to_thread()`

```python
async def run_scan_task(scan):
    # run blocking scan in background thread
    # await asyncio.to_thread(scan.execute)
    await asyncio.get_running_loop().run_in_executor(None, scan.execute)

async def main():
    kiwi_scan.load_all_plugins()
    kiwi_scan.load_all_scan_types()

    scan = create_scan_with_config("linear", create_config())

    task = asyncio.create_task(run_scan_task(scan))

    while not task.done():
        print("position:", scan.position)
        await asyncio.sleep(0.5)

    await task
    print("Scan finished cleanly.")
```

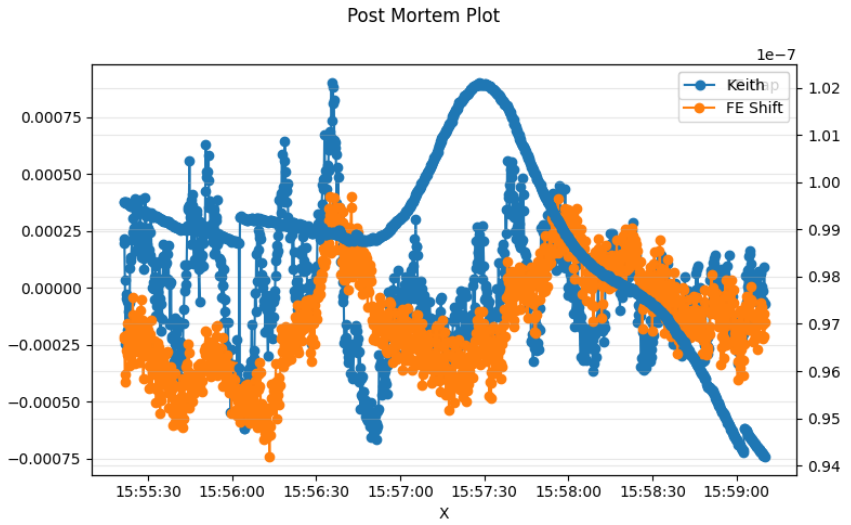
Command Line Tools

HZB BESSY II Light Source

- scan_runner → execute scans from YAML
- actuator_runner → direct actuator control
- scanplotter_cli → post-mortem analysis

```
scan_runner --config-file power_pmac.yaml \
            --scan_type linear \
            --replace IOC_NAME=PGMO21U15L \
            --dim actuator=theta,start=3.0,stop=3.4,steps=1001 \
            --dim actuator=beta,start=3.4,stop=3.0,steps=1001
```

```
actuator_runner --config-file mono.yaml \
                --monitor phi:rbv \
                --monitor psi:rbv \
                --move phi=6000 \
                --move psi=800
```




```
scanplotter_cli \
          --x_column="${X_COL}" \
          --series file="${META_FILE}",column="${Y0_COL}",axis=0,label=Kd,type=meta \
          --series file="${SCAN_FILE}",column="${Y1_COL}",axis=1,label="FE Gap"
            …
```

Conclusions and Outlook

HZB BESSY II Light Source

Conclusions

- Flexible lightweight scan framework for commissioning and diagnostics
- Combines simple scripting with modular, event-driven design
- Experimentation with control concepts via pluggable scan engines and plugins.
- Easily integrates into larger frameworks and heterogeneous environments
- Usable both as CLI tool and Python API
- kiwi-scan README (GitHub): **`https://github.com/hz-b/kiwi-scan`**

Outlook

- Generic IOC
  (with custom EPICS interface)
- Data conversion tools and Integration with PyMCA
  (standard export formats for import into established workflows)
- Advanced event synchronization
  (multi-event coordination and conditional logic)
- Enhanced YAML interface for multi-actuators
  (clearer, more expressive configuration for complex moves)

HZB BESSY II Light Source

Thank you for your attention.

Old scans, new flow - no more stop-and-go.