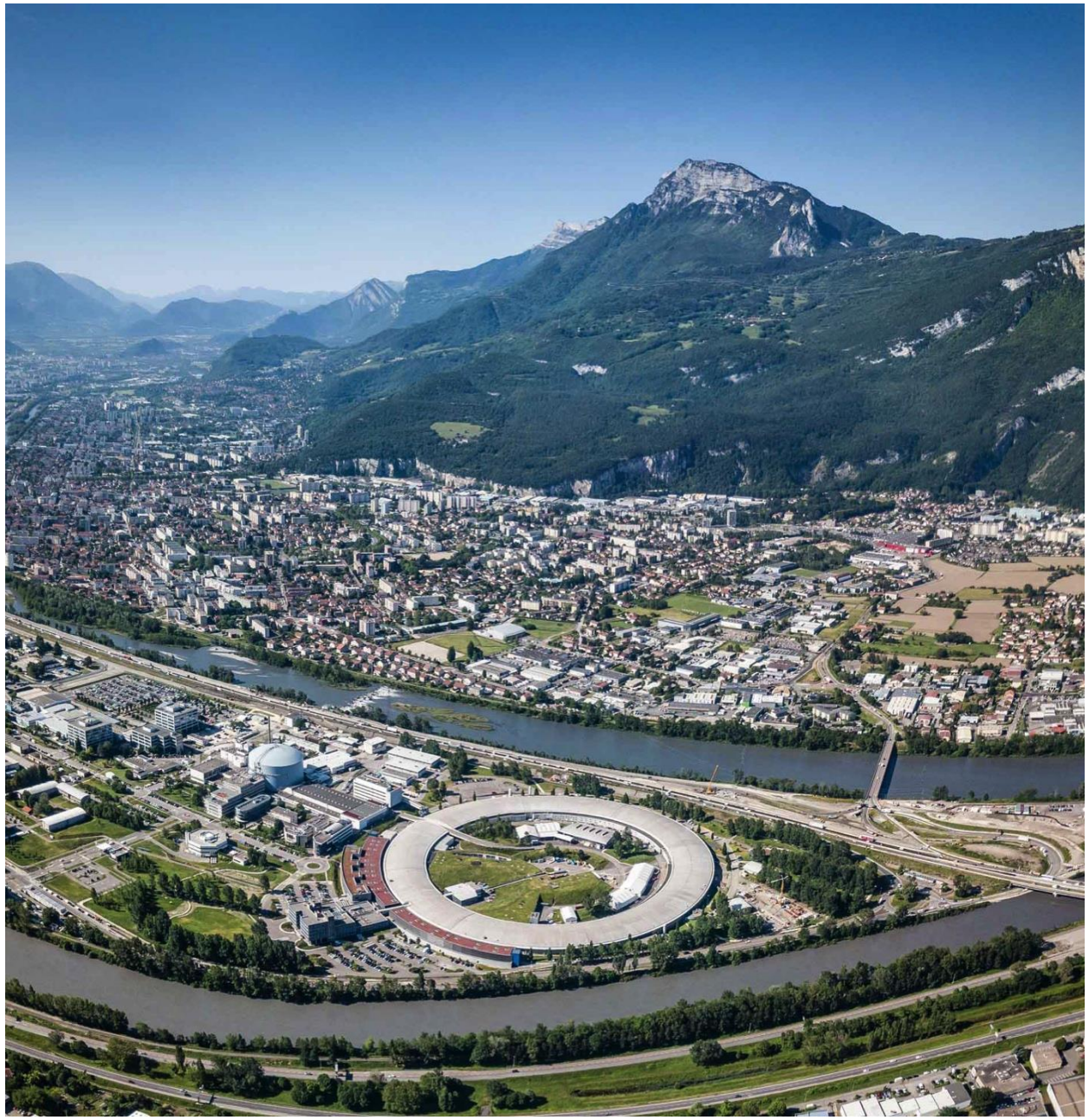


BLISS CONTINUOUS SCANS

PAPILLON Emmanuel
Beamline Control Unit

PIONEERING SYNCHROTRON SCIENCE

ESRF
The European Synchrotron

30 years of science

Continuous scans description

Hardware – triggering module

Questions & Answers

What’s next

A set of commonly used continuous scans:

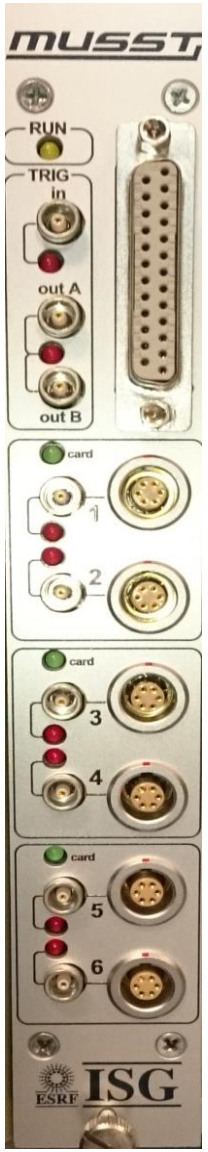
- Time base scans : ***ftimescan***, ***ftimescanlookup***
- One continuous motor: ***fscan***, ***fscan2d***, ***fscan3d***
  - > *Filling angular space :* ***finterlaced***, ***fsweep***
  - > *Repeating scans without stopping motor :* ***fscanloop***
- Two continuous motors: ***f2scan***
- Minimal configuration:
  - > *Available on any beamline with a* ***MUSST*** *board*

More advanced continuous scans:

- Tomography : package on top of fscans
- Diffraction Tomography
- Energy scans : link with monochromator framework

Triggering module : **MUSST board**

- Inputs : encoders, ADC, counters, trigger
- Outputs: trigger, gate
- Programmable:
  - > ***Targets*** *on encoder position, ADC value, timer, input signal*
  - > ***Actions*** *on targets: generates trigger, gate, store data*
  - > ***Circular buffer*** *to store timer, encoder positions, I/O*



Motor controllers:

- Any motor/encoder that can be connected to MUSST
- IcePAP (multiplexer), Aerotech, Elmo, Micos, Etel, …

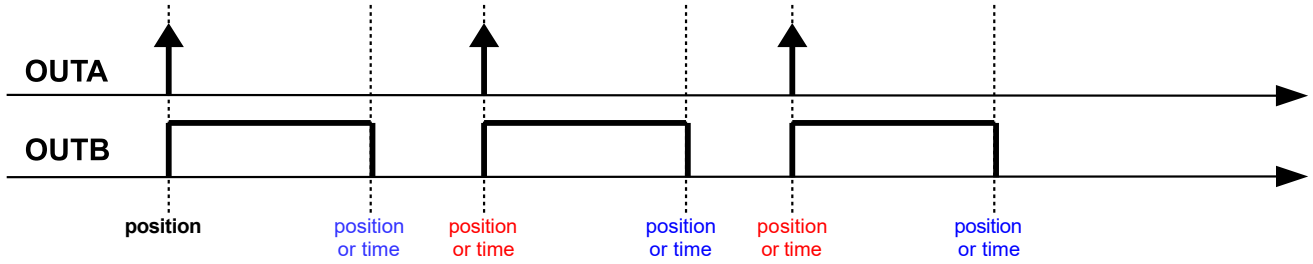
I/O signal multiplexer : **OPIOM board**

Counters / Detectors:

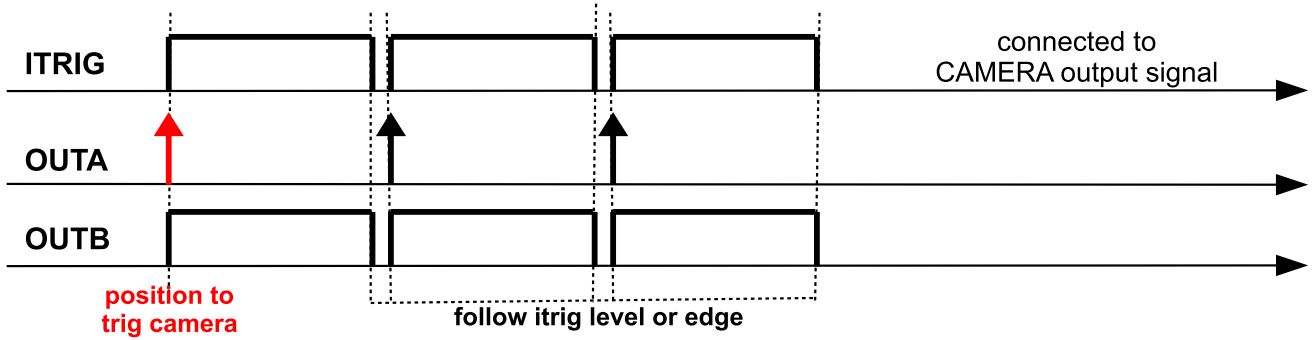
- Counter/timer board, LIMA devices, MCA devices, …

SUPPORTED SCANNING MODE

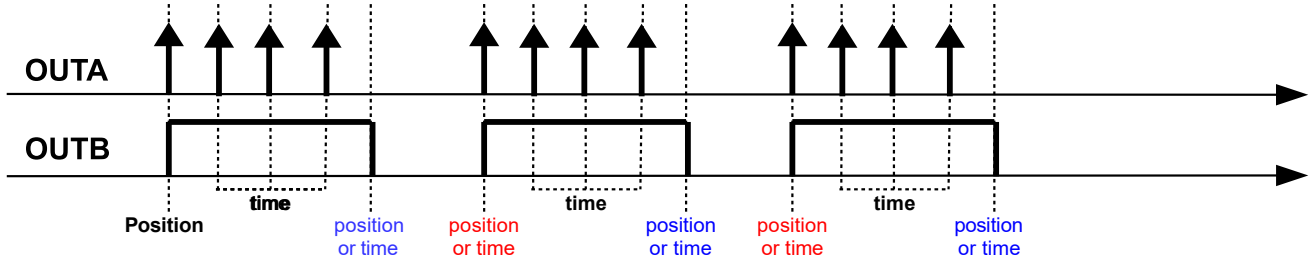
TIME or *POSITION* mode



CAMERA mode

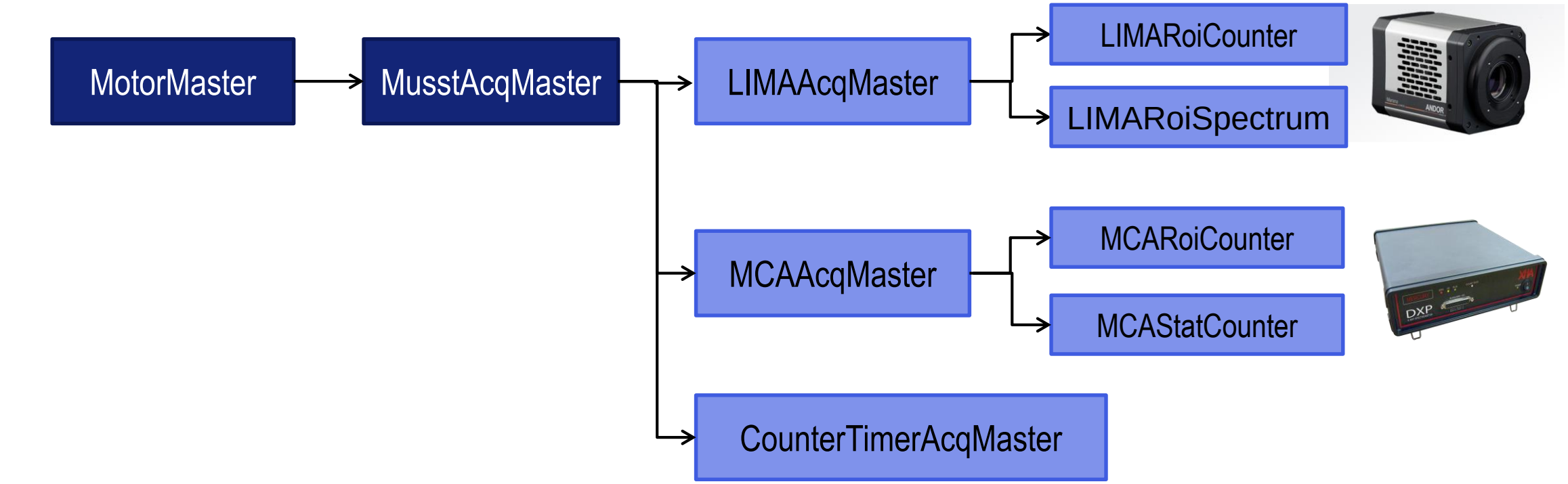


LIMA accumulation supported in all modes



Writing a scan is writing an AcquisitionChain

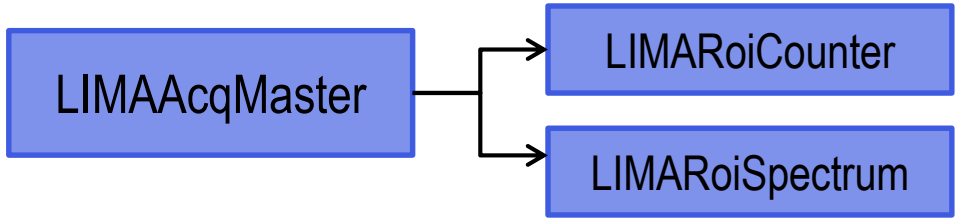
AcquisitionChain is a tree of **AcquisitionObject** with master/slave relations



Measurements groups are used by users to select counters/detectors

ChainBuilder helps creating **AcquisitionObject** from *MeasurementGroup*

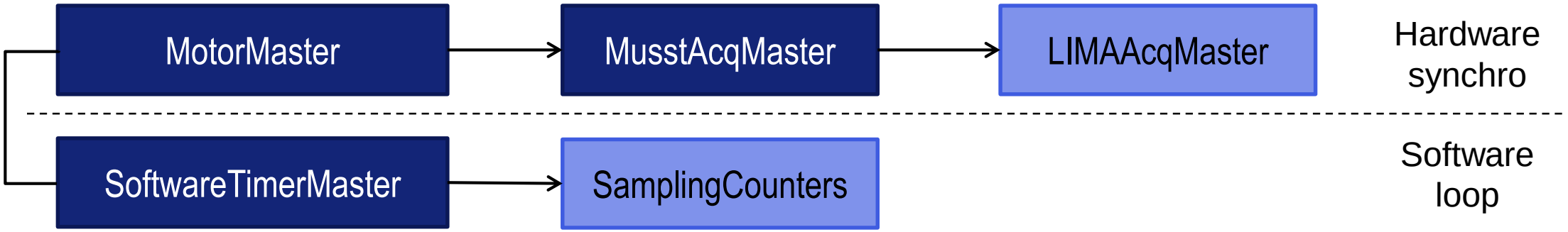
Creates known master / slave relation



CalculationCounters **:** added from *MeasurementGroup* or from scan

SoftwareAcquisitionMaster **:** hold counters that cannot be synchronized

> *Epoch counter provided to link acquisition of the 2 masters*



Shutter control and timing:

- Open/close shutter at begin/end of scans
  - > *Using a digital output on MUSST board*
  - > *Using a software **ScanPreset***
- Open/Close shutter at each acquisition
  - > *Delegate to detector (shutter interface in LIMA)*

Some scans supports *EXTSTART* mode:

- > *time-based scans and fscanloop*

Interaction with machine timing system

- Dedicated hardware : CITY, WHIST board
- Used for time-resolved experiment

Trajectory in bliss core:

- Common trajectory objects and methods on motor controller
- ***CalcController*** (pseudo-motor controller) can computes trajectory
- For icepap controller, we can also create a dedicated ***TrajectoryAxis***

Trajectory in continuous scans:

- From ***CalcAxis****:*
  - Choice of motor for position synchronization based on resolution
- From icepap ***TrajectoryAxis****:*
  - Used as any other motor
- From **scan** itself:
  - scan can create its own trajectory : *MeshTrajectoryMaster* / *DiffTomoTrajectoryMaster*

Position calculated in scans:

- Record the real(s) motor(s) position(s) on ***MUSST***
- Calculate position either from all real motors or estimate from trajectory table

BLISS data flow:

- Counters / spectrums published on **redis**
- Images managed by **LIMA** and referenced in **redis**

Performance:

- Reach detector performance:

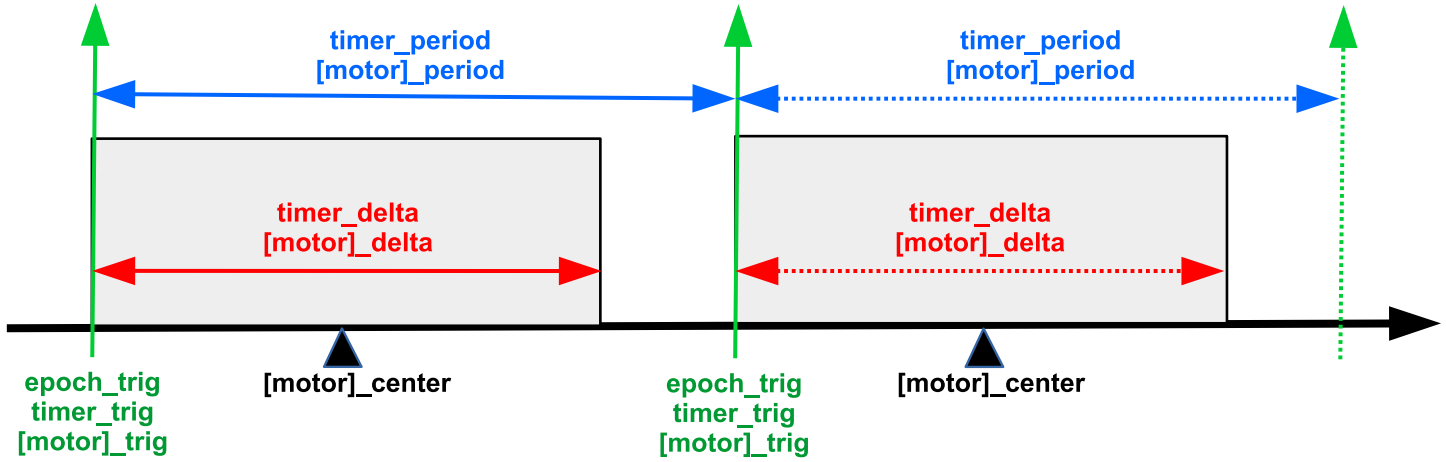
  Eiger2 4M, falconX: 1kHz; Maxipix, PCODiMax: 1.5 kHz; P201: 5kHz, …
- The slowest detector gives the scan maximum scan frequency

End-of-scan delay:

- All devices used in continuous scans has buffers
- Buffers are read continuously during scan
- **LIMA/MCA** devices are setup to follow maximum frequency of detectors

Calculated counters based on MUSST recordings:

Include clock frequency and motor units conversion

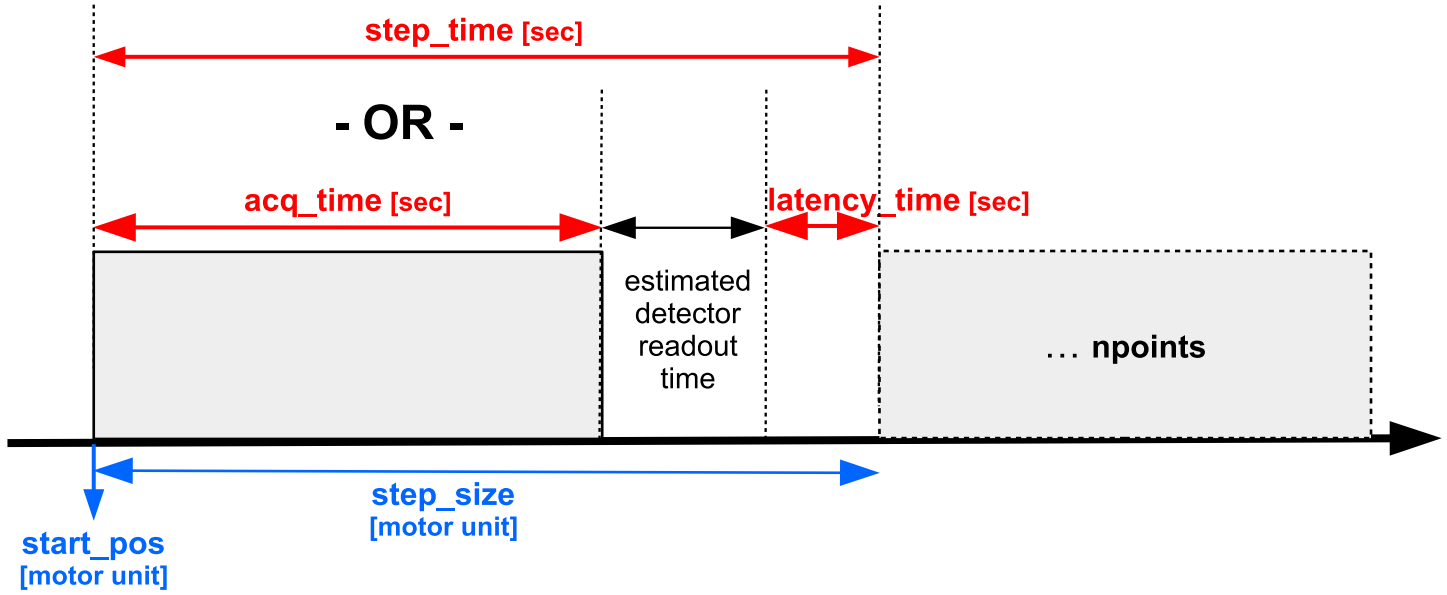


common MUSST calculated counters

Default assignment for plotting : center position

Q&A : LATENCY TIME

- Computes minimum period allowed for all detected in the scans
  - *adjust timing to fit the slowest one*
- Hardware trigger delays can be added per device in configuration
- *latency_time / step_time* can be defined by user if needed:

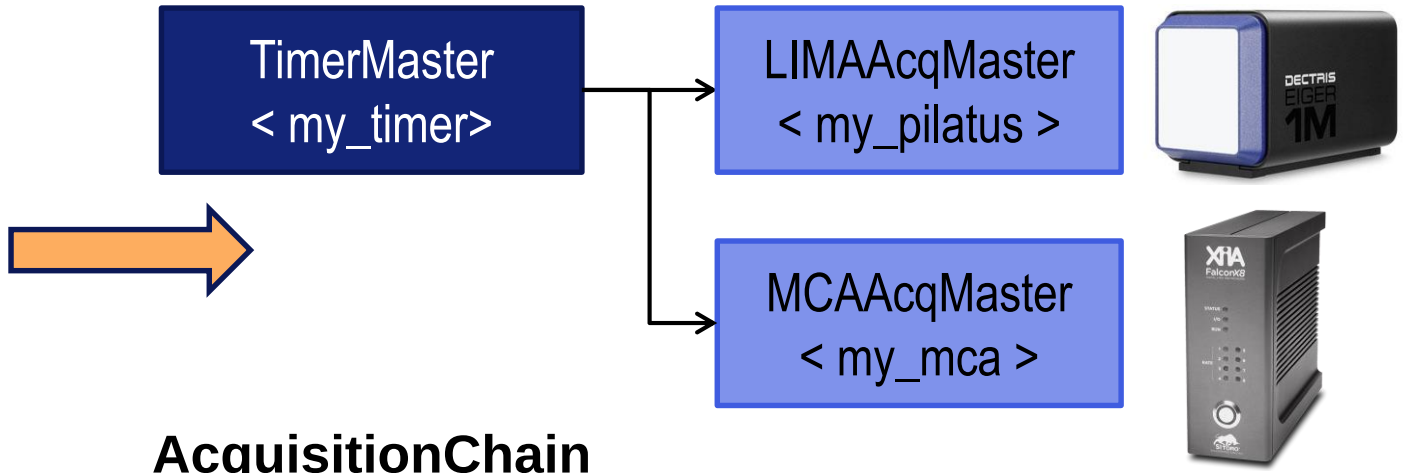


For *step scans*, acquisition chain can be configured:

```
chain_config:

- device: $my_pilatus
  acquisition_settings:
     - acq_trigger_mode: EXTERNAL_GATE
  master: $my_timer

- device: $my_mca
  acquisition_settings:
    - acquisition_settings: EXTERNAL_GATE
  master: $my_timer
```



For *continuous scans:*

The scan describes all master/slave relations in his ***AcquisitionChain***

Parameters of the ***AcquisitionObject*** in this chain can be locally configured

Bliss provides a *ScanWatchdog:*

- Receives start / stop scan event
- Receives last update on data channel
- Gets called on a timeout if no events comes in
- Allow to interrupt the scan properly
- To be done : flags “critical” and “non-critical” data

For user-feedback:

- on prompt: ***ScanProgress***

```
Running: mot1  1.4050 trig 92 pilatus 92 (S -2) mca 92
```

- Live-plot in ***flint*** : curve, image, scatter-plot

API separates controllers and user oriented objects:

- *Axis* are exposed to users, *MotorController* are not
- Trajectory and synchronization events are implemented at controller level
- Usage of those are delegated to motor *AcquisitionObject*, used to build the *AcquisitionChain* for the scan

Configuration belongs to unique object:

Axis holds steps_per_unit, acceleration, backlash, …

Musst calculation counters for example use this config. No duplication.

Optional features for motors:

Implemented in *MotionHook* (power-on, protection, …)

MAESTRIO will replace *MUSST* board

- More and faster inputs / outputs
- Efficient arrays for program inputs
- Multiple sequencer in parallel

Faster and faster detectors

➢ 2 or more synchronized masters in acquisition chain

PILATUS4 4M : up to 4kHz

User or external interaction during scans

Change acquisition rate between tomo without stopping rotation

Machine interaction

- Undulator synchronization
- White Rabbit network

THANK YOU FOR YOUR ATTENTION