EUROPEAN SPALLATION SOURCE

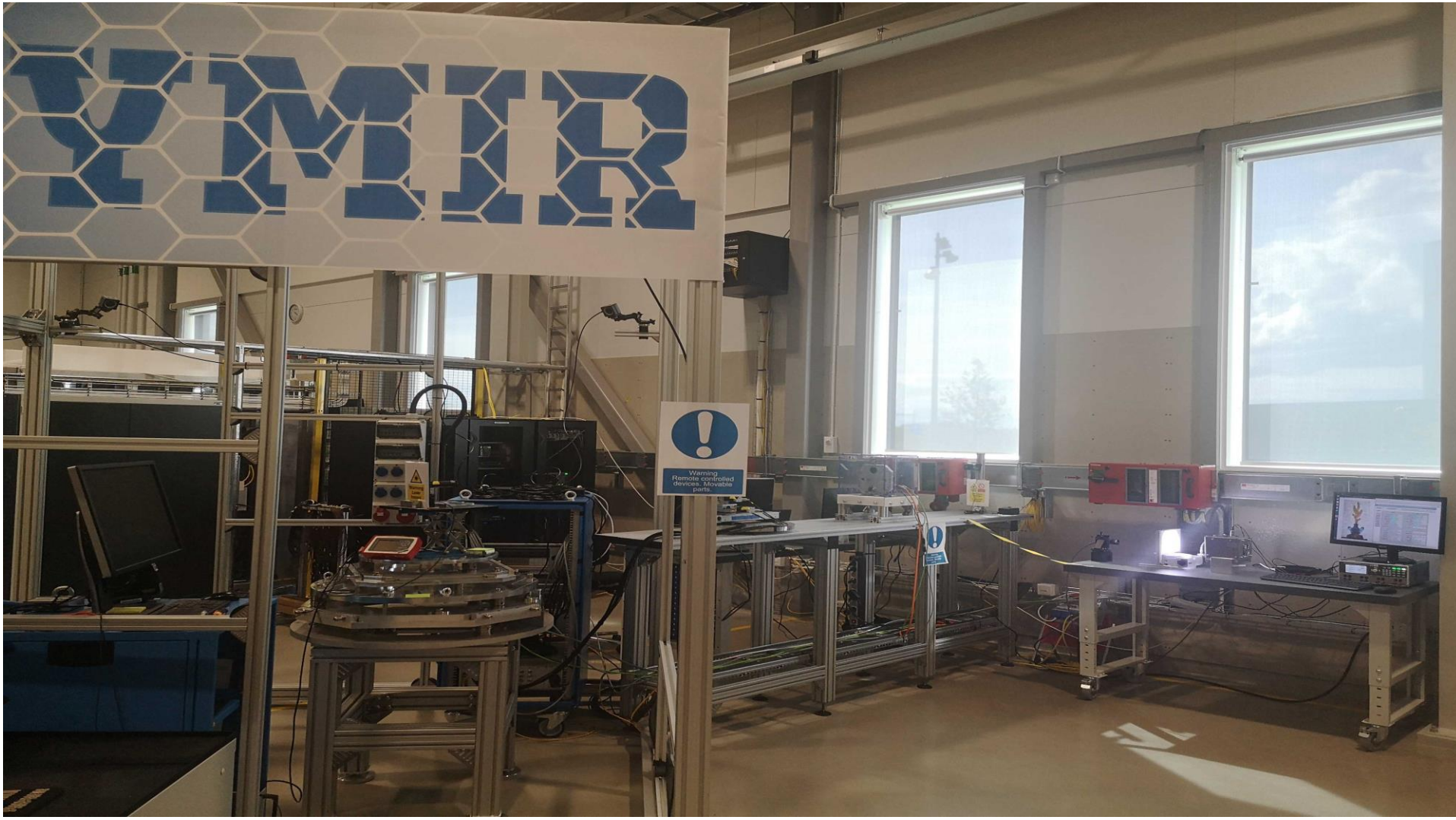
# TwinCAT – EPICS with motion timestamping

Torsten Bögershausen

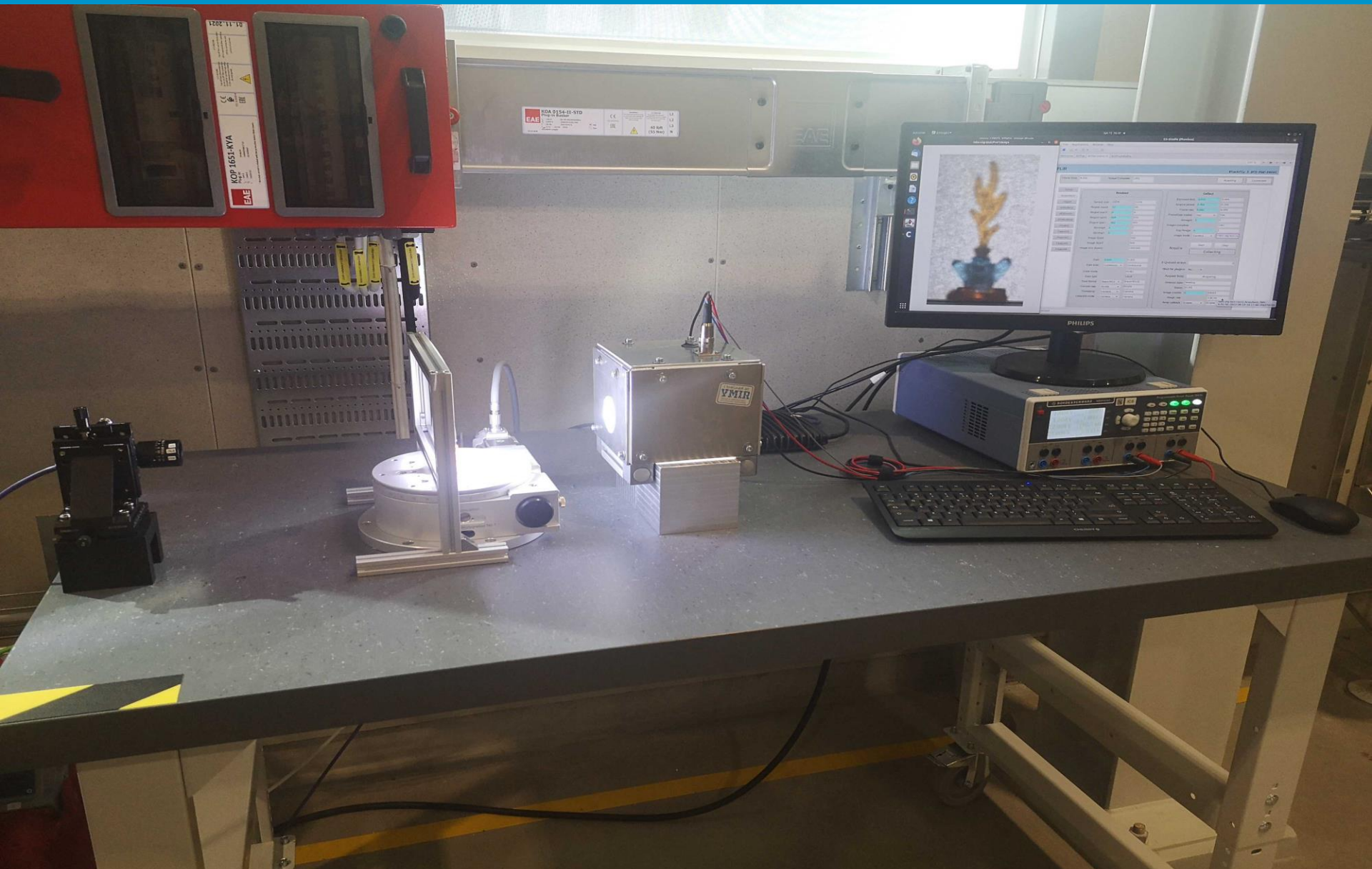
# Buzz words

- Imaging+scanning (tomography)
- All data in Apache Kafka
- Facility-wide timing system (”MRF”)
- Precise Timing Protocol
- TwinCAT
- Motor position timestamping in the controller
- EPICS IOC

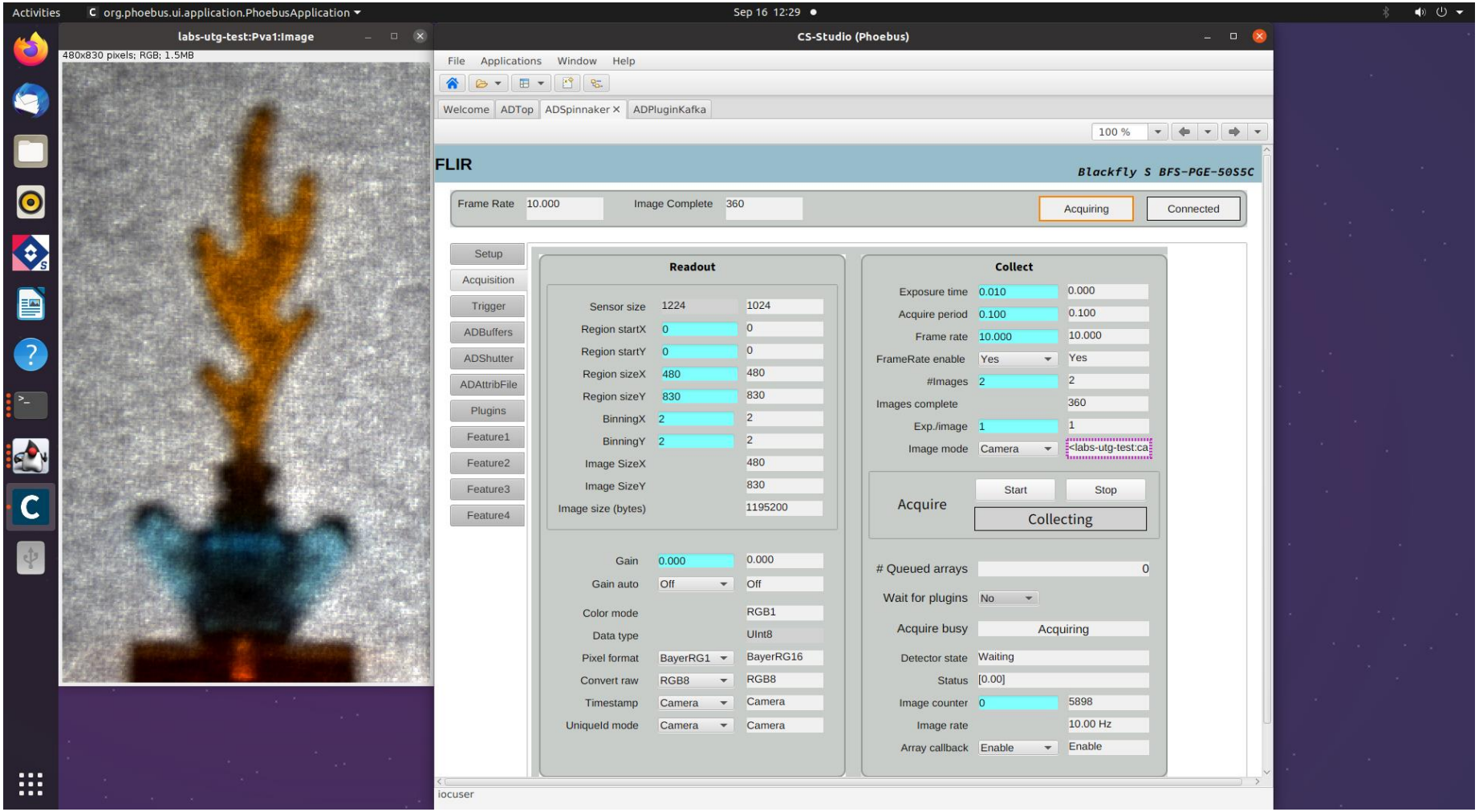
# YMIR



# Light tomography



# Control screen



# Timestamping – what, how

- Motor positions and EPICS:
  - Motor positions are “acquired” in MCU
- Visible in EPICS motorRecord (“.RBV”)
- EPICS records are “timestamped” in the IOC, when they are processed.
- Roundtrip times of a poll:
  - 10..20 msec/axis (1st gen: EPICScom module)
  - 5..10 msec/axis (2nd gen: twincat-ads)
  - 5..15 msec all axes (3rd gen: PILS)

# Old communication MCU-EPICS

- 10 msec roundtrip / axis:
- We run into a limitation:

8 axes: 80 msec poll time
16 axes 160 msec poll time

- Motor positions of different axes
are ”someone somewhere somewhen”

# Timestamping – not accurate enough

- For positioning: 100 msec is good enough (many axes)
- E.g. tomography needs something better: specs ???
- specNotYetFullyDefinedException @ ESS
- Our assumption:
  <= 1 msec is good enough for scanning

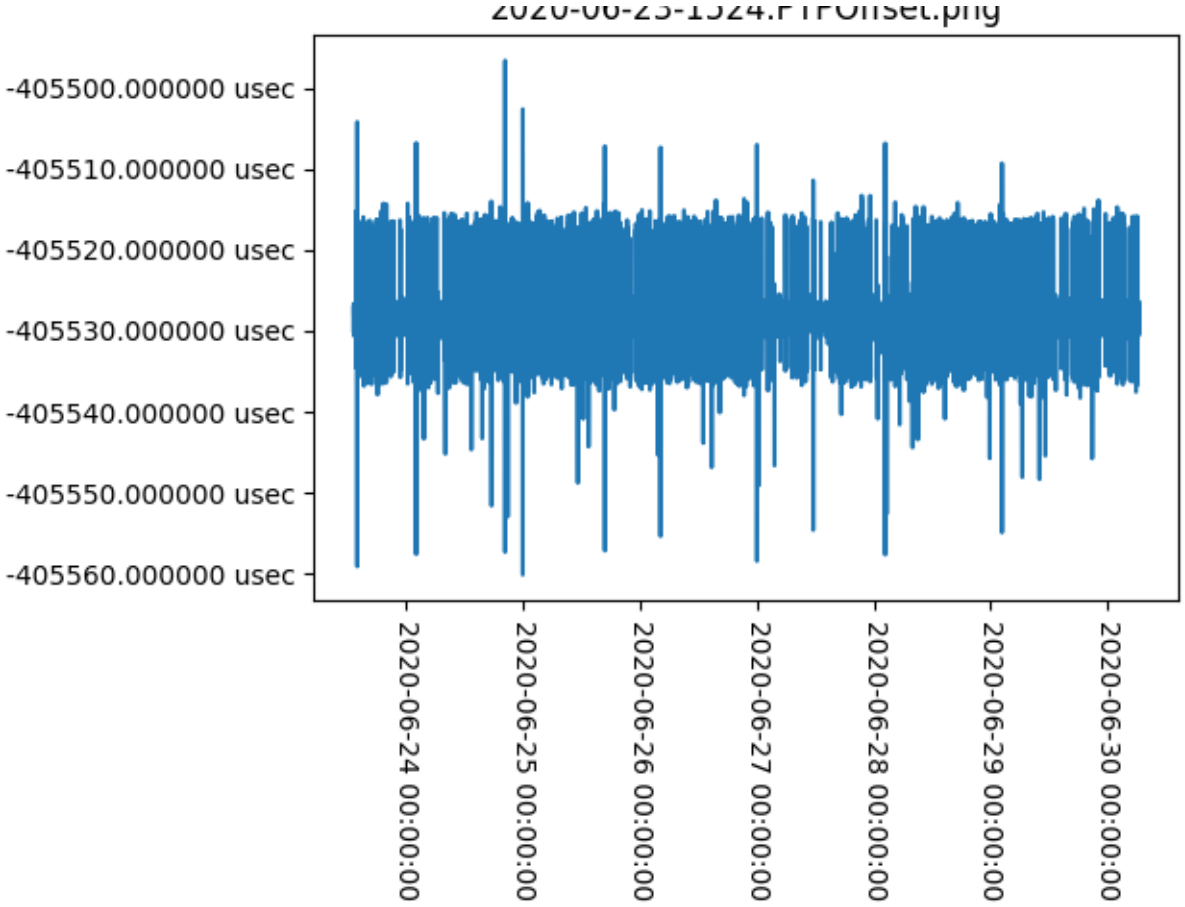
# Time stamping in MCU/PLC

- Each PLC cycle (every 10msec):
- Read all motorPositions
- Read status of all axes (error, busy, homed, LS)
- Read other sensors
- Gets UTC from timing system
- Put everything into continuous memory block
- EPICS IOC polls this every 10..1000 msec

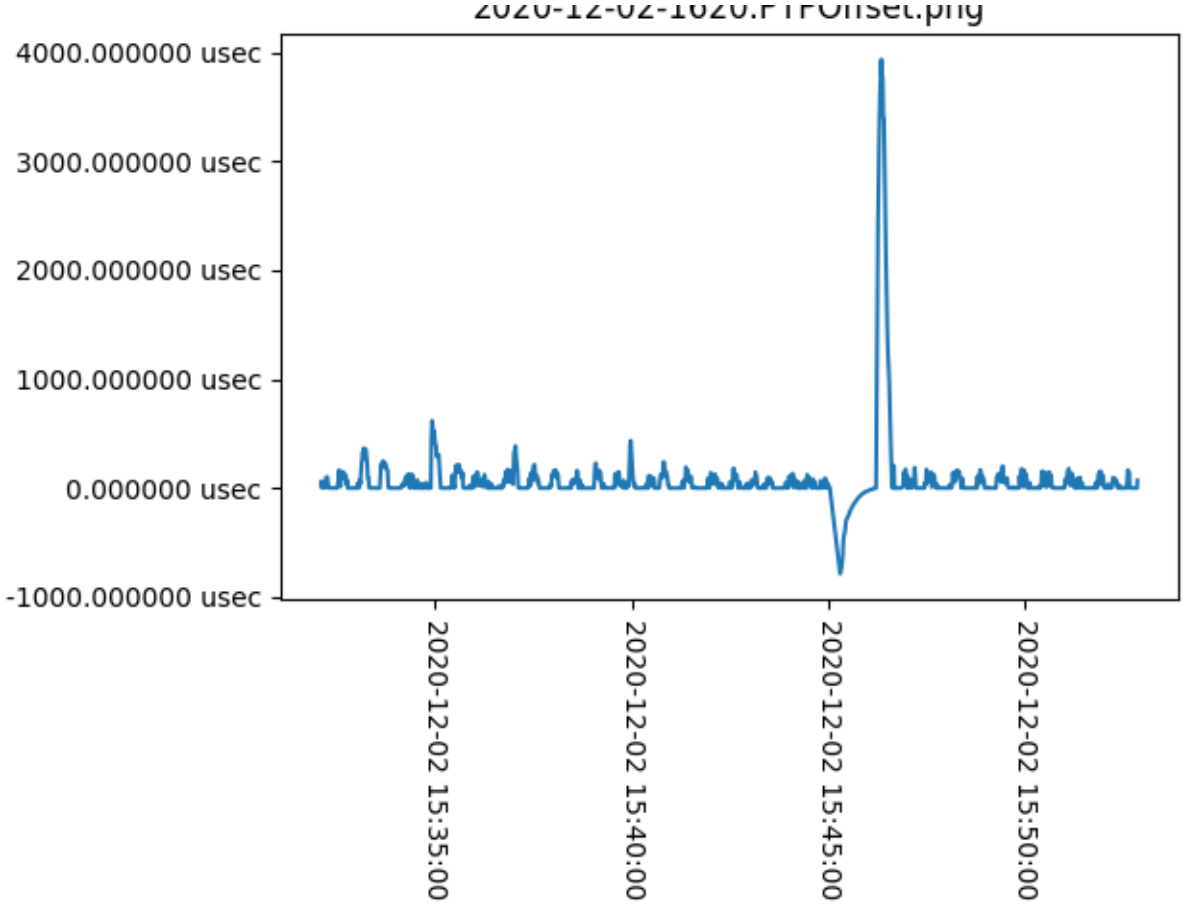
# Timing system; PTP

- An EVR acts as a PTP master (chrony on eth2)
- TwinCAT needs a PTP terminal: EL6688
- Needs to be configured (in TwinCAT)
- Needs to be evaluated

# PTP Offset master - EL6688



# PTP Offset master - glitch



# Transport protocol TwinCAT - EPICS

- Raw TCP Socket ? Or: ADS
  (locked against PLC cycle)
- ADS_READ (start, length),
  ADS_WRITE (start, length)
- Application protocol ”on top” : ”PILS”
  (the 3rd generation)

# Memory layout – Runtime data

- Poll data, example:

```
Date/Time (UTC)
Additional IO
Motor1    (position, status)
Motor2
Motor3
```

- PILS gives us a description

# IOC::asynPortDriver::poll()

- IOC reads all data with every poll()
- Distributes it into the different Records (asyn parameters)
- Calls asyn::setTimeStamp()
- We have additional ai PVs:
  ”$P$R-RBV-TSE.VAL” for the motor position
  ”$P$R-RBV-TSE.TIME” derived from MRF->PTP

# PILS

- Credits to:
- PLC Interface Layout Specification
- https://forge.frm2.tum.de/public/doc/plc/master/singlehtml/#device-address
- (Introduction/Lessons on request)

# Next steps

- Verification that everything works
  (“kafka has good data”)
  (motors, detectors, cameras)
- Ongoing project:
  Light tomography with a camera

# Questions

- hvala
- thanks
- tack
- danke
- Merci

# Links

- https://kafka.apache.org/
- https://en.wikipedia.org/wiki/Precision_Time_Protocol
- https://www.beckhoff.com/en-en/products/i-o/ethercat-terminals/el6xxx-communication/el6688.html
- https://forge.frm2.tum.de/public/doc/plc/master/singlehtml/#device-address
- https://www.ijs.si/time/