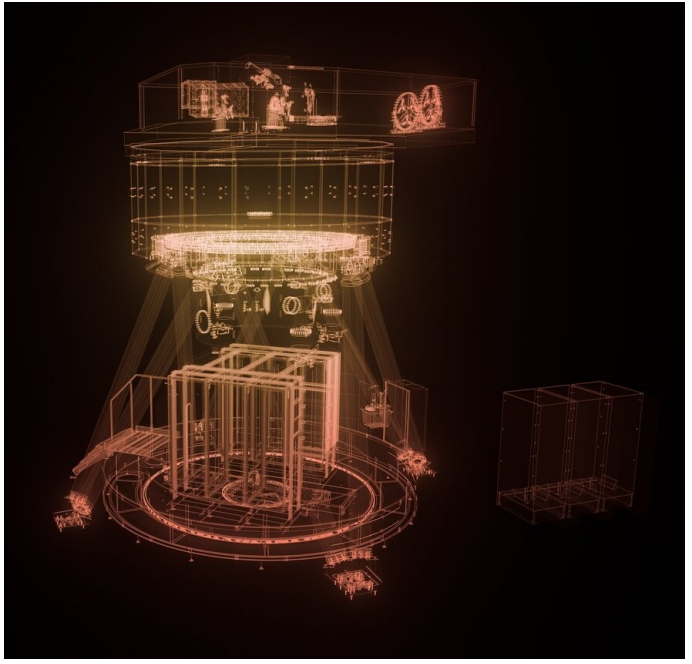
MICADO:
Instrument Control Software

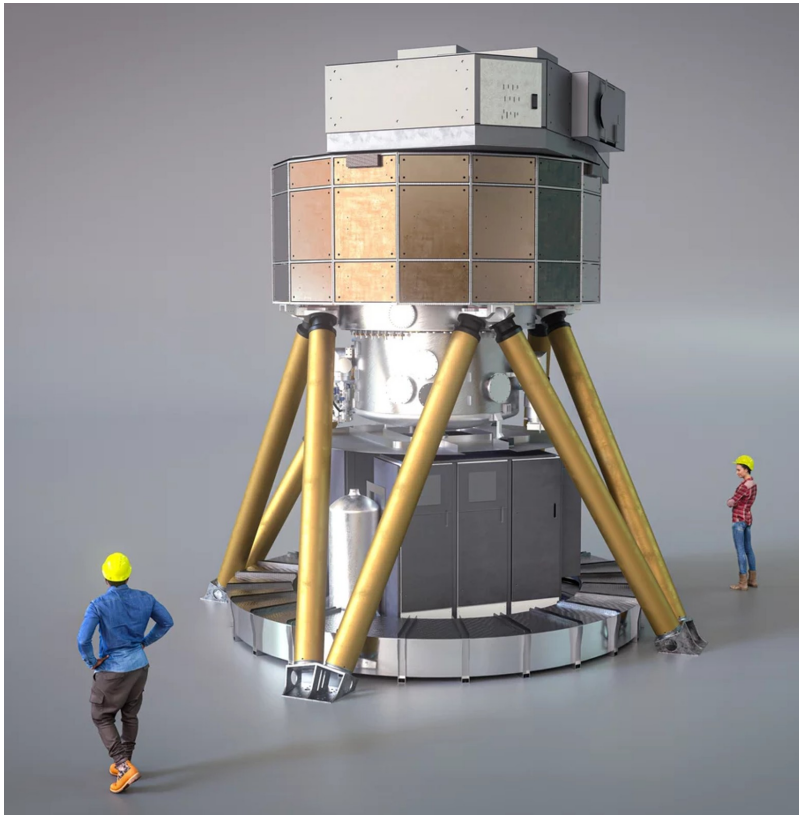
- Agenda:
  - The instrument
  - SW architecture / MICADO specials
  - Interaction MICADO / MORFEO
  - Function control SW (USM)
  - Function control SW (MPIA)
  - Continuous integration



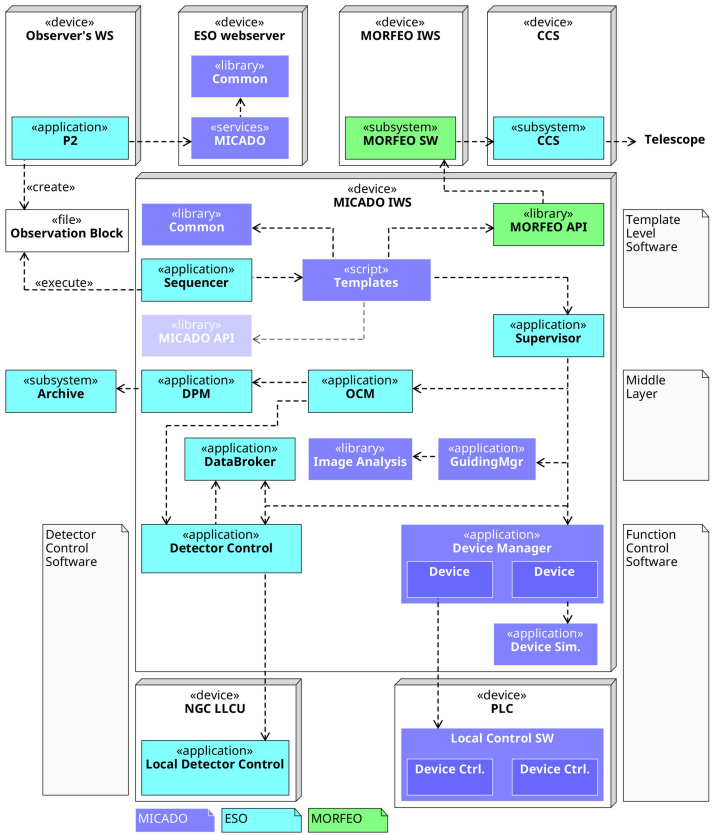
- MICADO ICS team:
  - Udo Neumann (MPIA)
  - Jörg Schlichter (USM)
  - Michael Wegner (USM)

The MICADO instrument

- First light instrument for the ELT
- FDR in summer 2024 :-)
- Cryogenic instrument / near infrared
- Observation modes:
  - Low and high resolution imaging
  - Spectroscopy
  - Coronagraphy
  - (Pupil viewing)
- Support for different AO systems:
  - SCAO (built by MICADO consortium)
  - MCAO (built by MORFEO consortium)
  - SCAO control system: Part of MORFEO



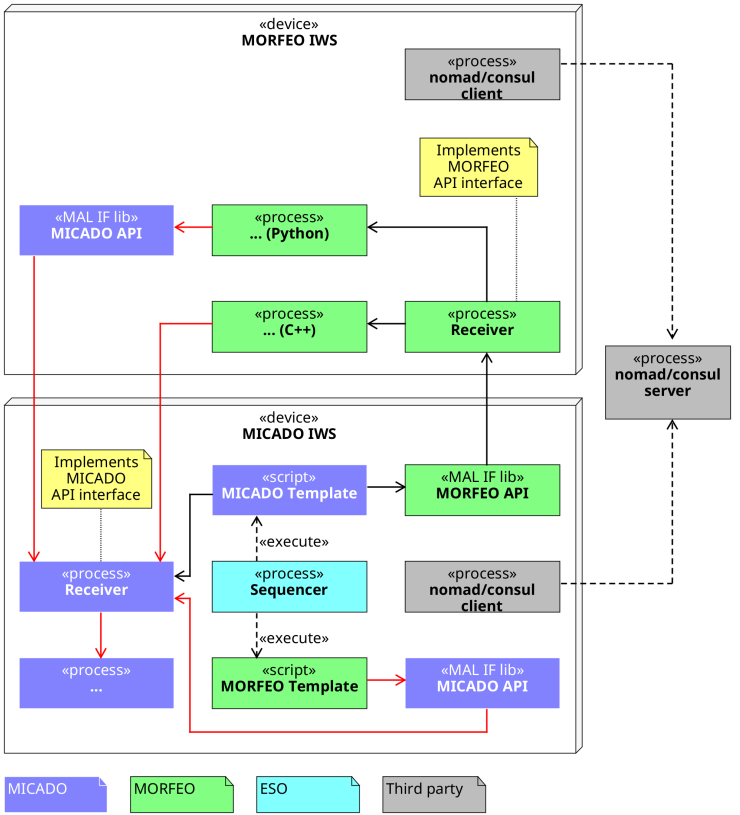
SW architecture



- MICADO SW specials:
  - Observation preparation SW:
    - MICADO micro services + common library
    - (GUI prototypes)
  - Template level SW:
    - Acquisition, observation, calibration, maintenance templates
    - MICADO API library
  - Middle layer (observation coordination, ...):
    - Guiding Manager + image analysis library
  - Function control SW:
    - Special devices + device simulators
  - User interfaces
- Support SW for hardware tests

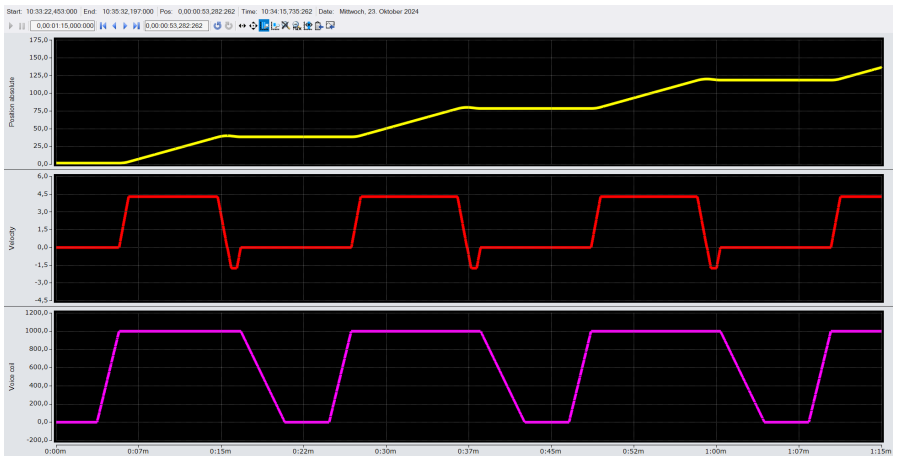
Interaction MICADO - MORFEO

- MICADO:
  - Command interface to MORFEO
  - No direct interaction with telescope
- MORFEO:
  - Command interface to MICADO
    (only during calibrations and acquisition)
- API libraries:
  - Hide details from other instrument
  - Decouple implementation process
  - Common interface for SCAO and MCAO
  - Procedures defined in ICD
- Implementation of libraries needs:
  - Careful handling of dependencies
  - Common nomad/consul instance

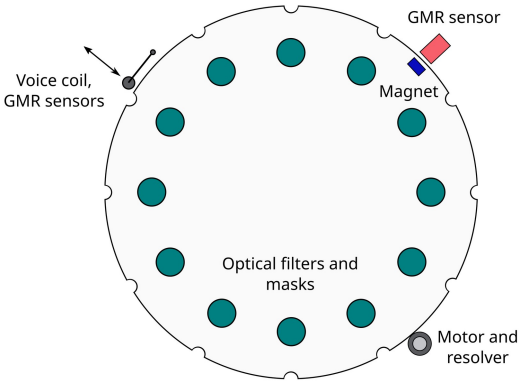


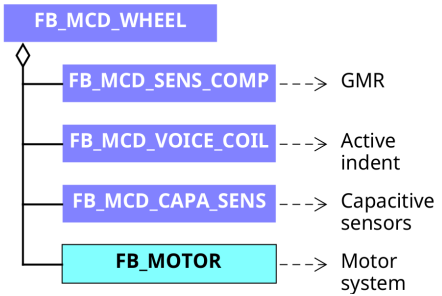
Cold wheel mechanisms

- Common functionality:
  - Named positions
  - GMR sensors as switches:
    - Compare values with tresholds
    - Warm and cold configurations
  - Active or passive indent mechanism
  - Similar sequence of movements



Mask wheel with indent mechanism



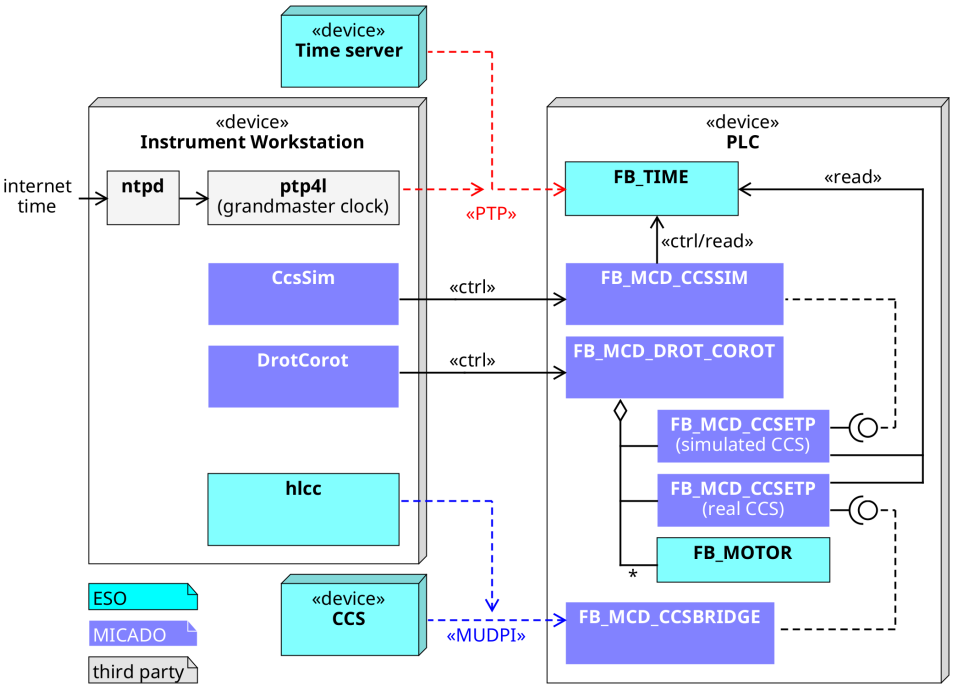


- Function block FB_MCD_WHEEL:
  - Can be adapted via config flags
  - “Standard” state machine and Rpc-interface
  - Organizes actions of child function blocks
- Successfully tested with:
  - Main selction mechanism prototype
  - Pupil wheel (warm and cold)

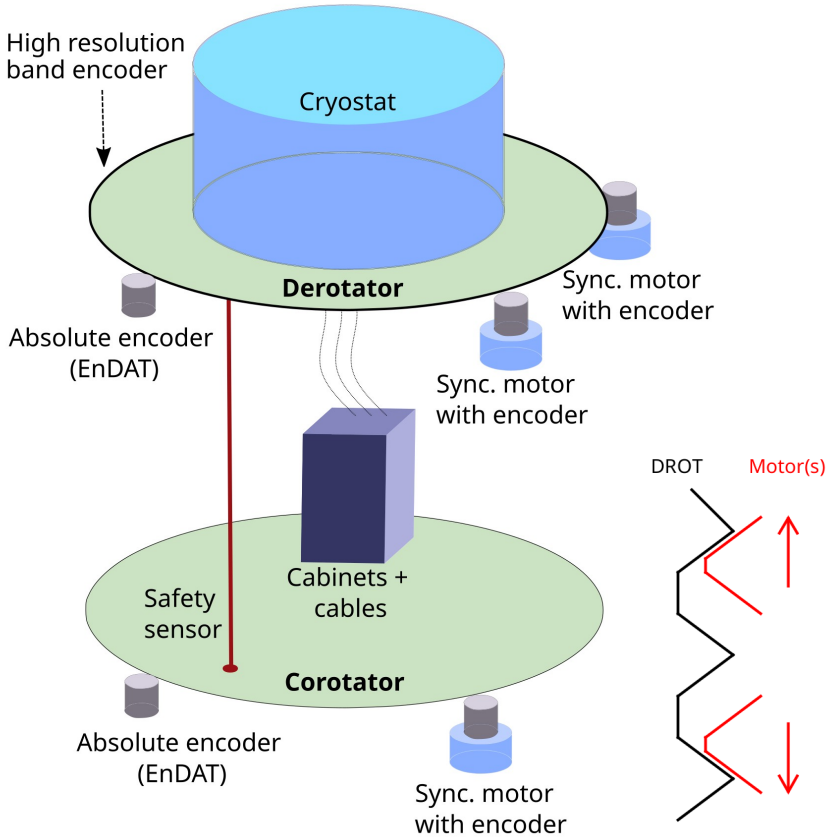
Tracking devices

- Tracking devices:
  - Atmospheric Dispersion Corrector (ADC)
  - MICADO derotator/corotator system
- Very similar functionality:
  - ADC needs to be “de-derotated”
  - Interface CCS and time server
  - Extrapolation of CCS data
  - ...

    → several common function blocks
- MICADO CCS simulator (PLC)
  - Independent from SLA library
  - Same data interface as CCS
  - Provides environmental data + noise
  - “Device” → easy to use in templates

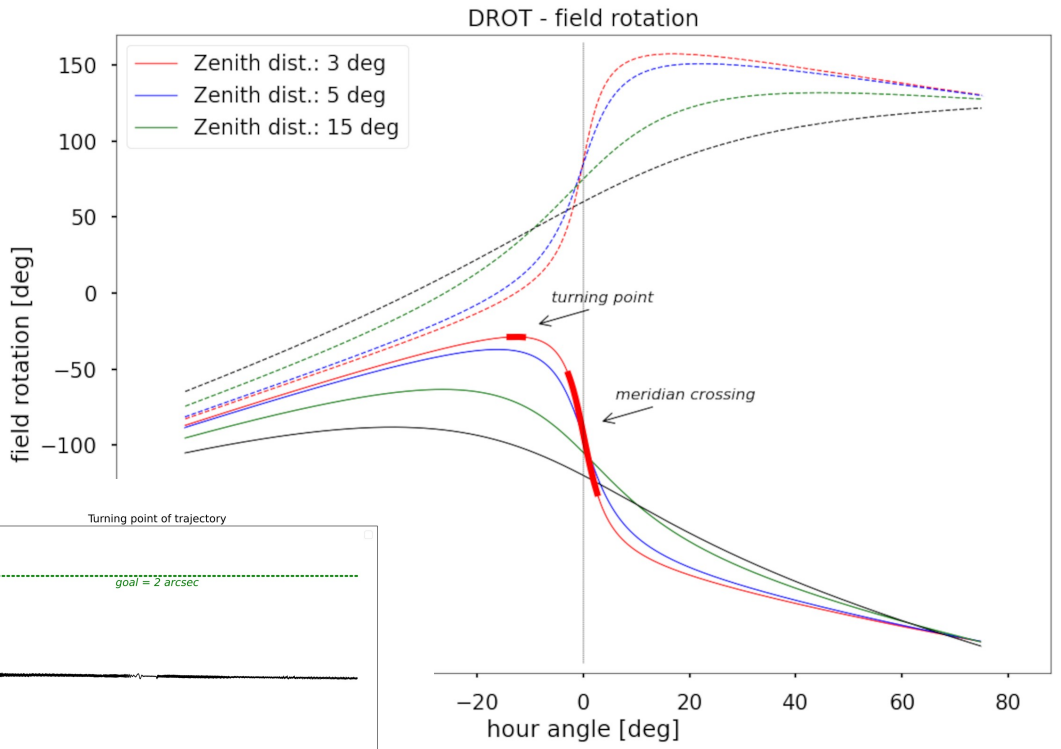


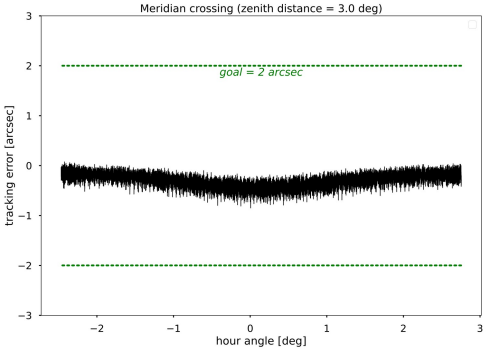
Derotator/Corotator system

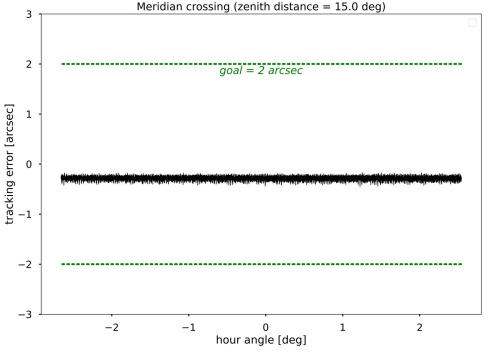


- Requirements:
  - High accuracy tracking
  - Synchronization only in SW (“master-slave coupling”)
- Derotator:
  - Uses absolute high resolution band encoder (EnDAT)
  - Needs to sample reference marks at startup
    → NC-Axis: Two encoders/controllers
  - Beckhoff EL5032 terminal: Bugs in firmware :-(
- Motor control options:
  - One motor (+ two spring-loaded gears)
  - Two counter-acting motors
  - Motor control configuration: Luis Neumeier (MPE)
- Tests: DROT on-going (MPE/USM), COROT summer 2025

Derotator tracking accuracy

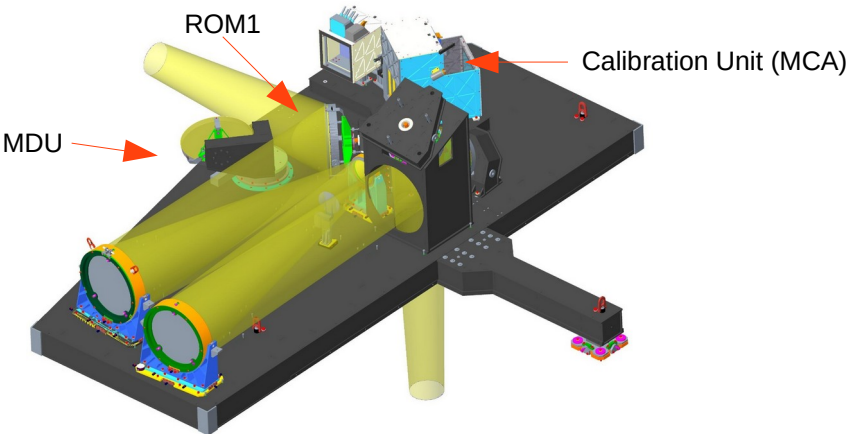






Relay Optics Mirrors (RO) + Calibration Assembly (MCA)

Bench



- Relay Optics
  - needed for beam without MCAO MORPHEO
  - MCA Deployment Unit (standard circular motor)
  - software for 3x MIRRORs, motorized for tip/tilt/piston

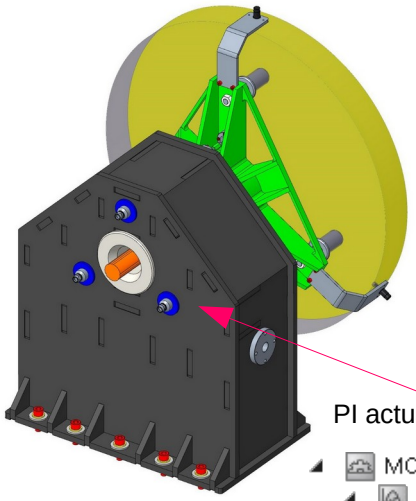
    ⇒ PLC special device

- Calibration Unit
  - well ... calibration.
  - up to now all devices are standard devices (motors, lamps and shutters)
  - execept: Hexapod for moving/rotating the Warm Astrometric Mask

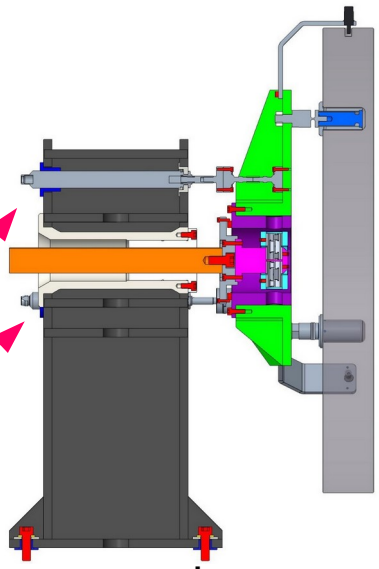
    ⇒ PLC special device

Relay Optics MIRRORs: Design

Example Mirror: ROM1

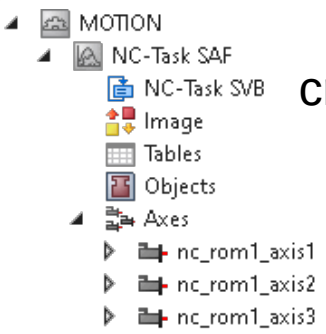


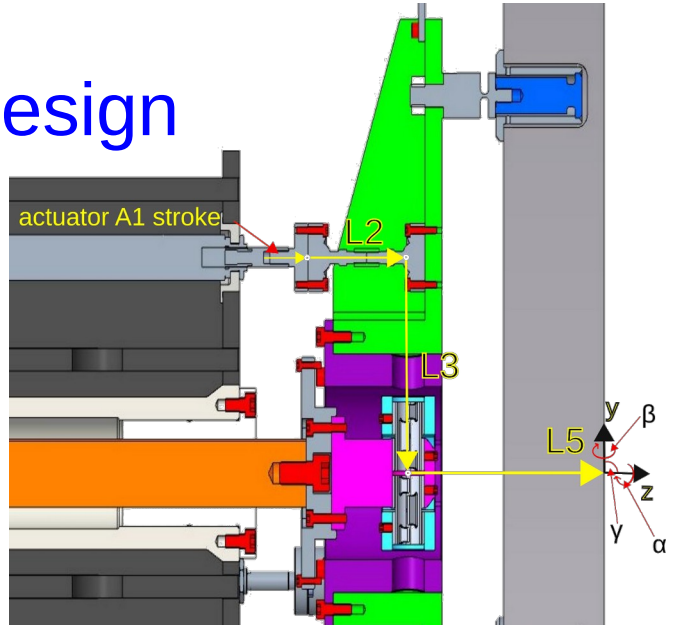
backside



cross section

PI actuators (axis[1-3])





Math by Robert J. Harris:

$$L2(z)+L3(z)+L5(z)+actuator=const$$

$$R=\begin{bmatrix}\cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma-\sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma+\sin\alpha\sin\gamma\\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma+\cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma-\cos\alpha\sin\gamma\\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma\end{bmatrix}$$

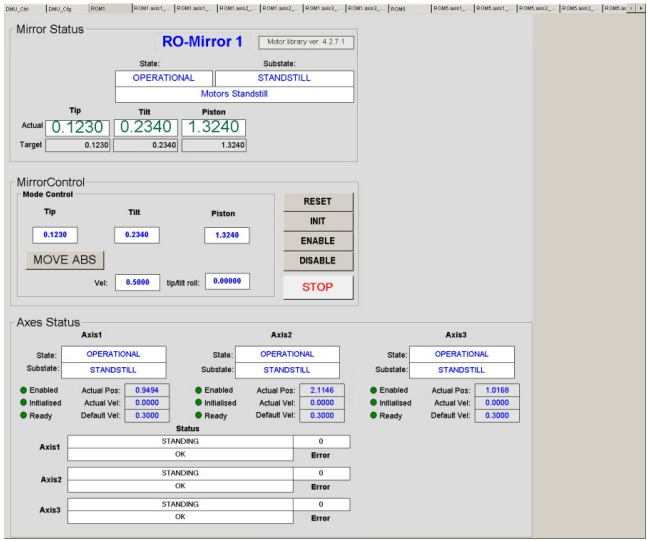
$$R(tip,tilt,\gamma=0)\longrightarrow \overrightarrow{L5}_{rot}=R\times\overrightarrow{L5} \quad \text{and} \quad \overrightarrow{L3}_{rot}=R\times\overrightarrow{L3}$$

$$\overrightarrow{L2}_{rot}=\begin{bmatrix} L3_{rot}[x]-L2_{start}[x] \\ L3_{rot}[y]-L2_{start}[y] \\ \sqrt{L2_{abs}^2-(L2_{rot}[x]^2+L2_{rot}[y]^2)}\end{bmatrix} \cdots\longrightarrow \overrightarrow{A1}=const-(\overrightarrow{L2}_{rot}+\overrightarrow{L3}_{rot}+\overrightarrow{L5}_{rot})$$

Relay Optics Mirrors: taken/adapted from ADC@ESO

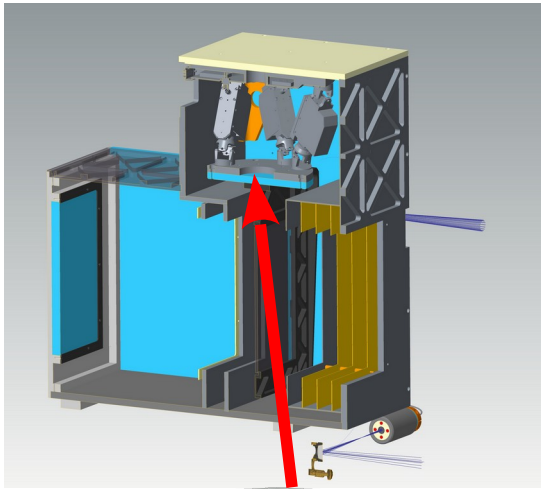
- MultiAxis device:
  - on PLC: engineers might control via PLC HMI
  - in principle only implementation/changes for:
    - ✗ `T_MIRROR_CFG EXTENDS T_MA_CFG:`
      `lrl2_abs, lrl3_y, lrl3_z, lrl5_z, lractuator_angle`
    - ✗ transform needed to be implemented:

```
METHOD User_ComputeNextPos : BOOL
…

(* Calculate the motor position with the coordinate transform *)
CoordinateTransform();

User_ComputeNextPos := TRUE;
```



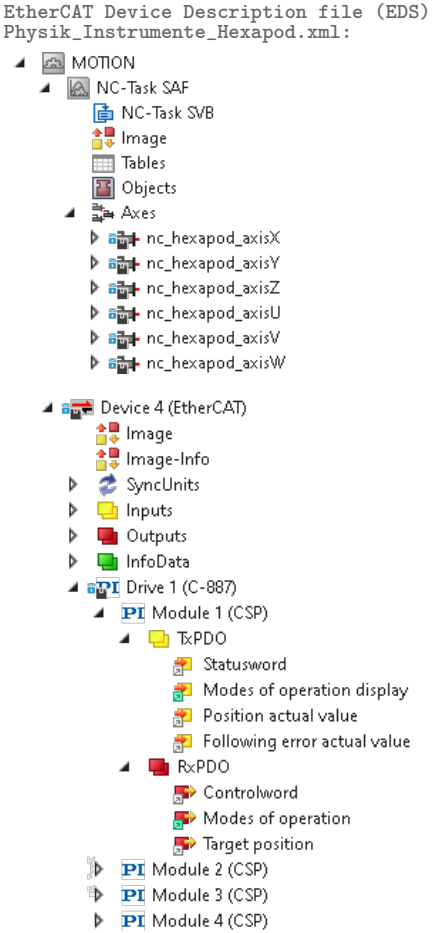
    - ✗ `RPC_Move()` calls accepting tip/tilt/piston.
    - ✗ additionally copy + adapt the ADC code that is not provided with the `FB_MA_BASE` function block.
  - WS: copy of ADC + extend to have 3 axes – could be replaced with MultiAxis @ ESO (`!cfg.type:Maxis`)
  - validation of transformation by comparing with numerical calculated values from CAD model for 3 instances having different parameters for the geometry: ROM1, ROM5, ROM6
    - ➔ automatic etr/robot tests

Calibration Unit (MCA): Hexapod

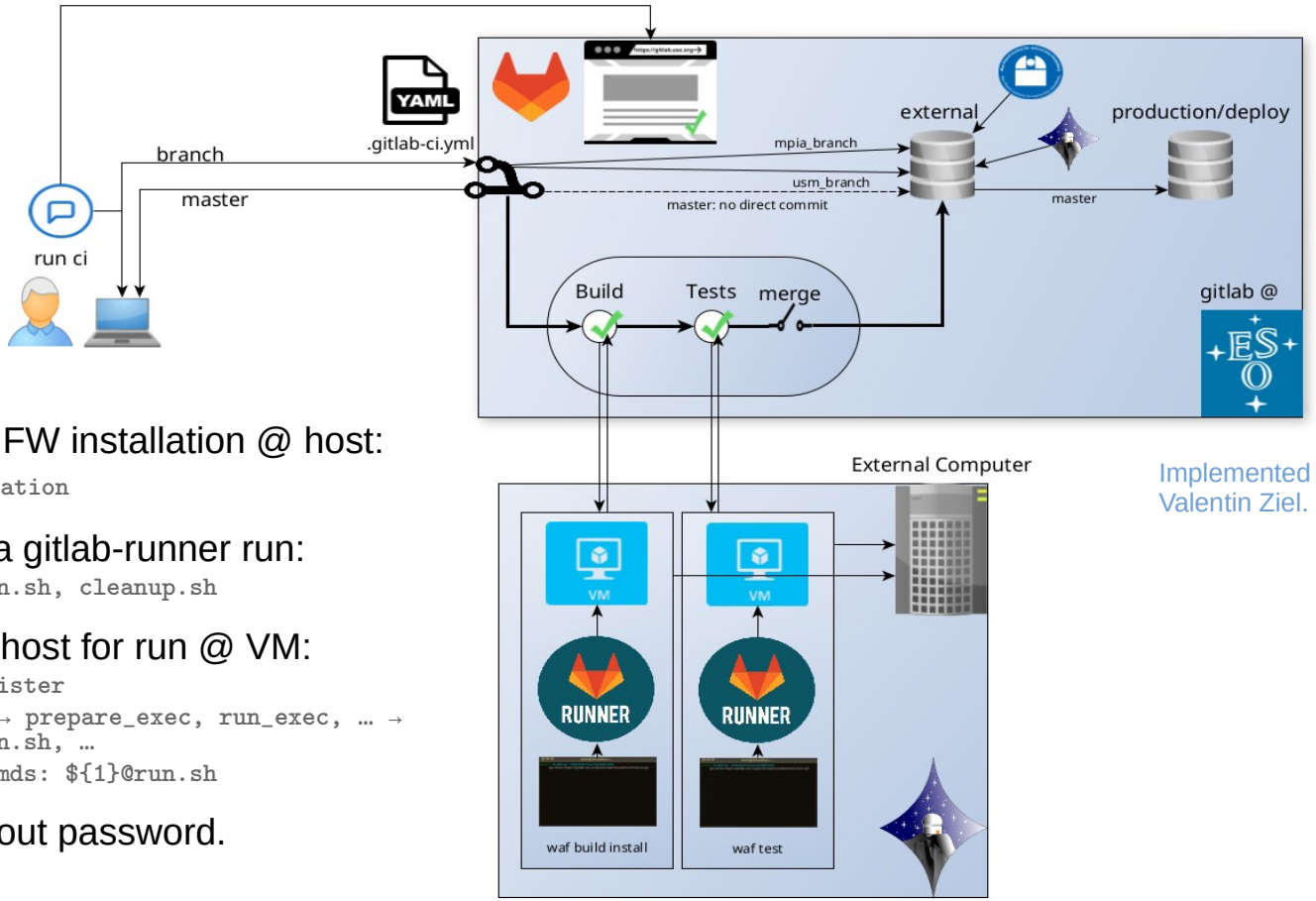


- EtherCAT Interface:
  - Controller does internal device handling:
    - Movement control: collision, max. range + coordinate transform
  - Provides:
    - 6x nc-axes: X, Y, Z, U, V, W
    - Control/status IO: mode of operation, statusword
    - for each requested position max. travel range via SDOs
  - TwinCAT examples provided PI.
- Seems simple, but:
  - enabling/poweron nc-axes does only succeed after `MC_Power() + MC_Reset()`
  - PI methods/procedures to be adapted to motor @ ESO
  - `ctrl.nModeOfOperation = 8 → stat.nStatusWord.bit#10 = 0←→1`
    - ↳ `RPC_Reset()→ ctrl.nModeOfOperation = 0 only successful if stat.nStatusWord.bit#10 = 0` else fails and is only recoverable by resetting the controller.
- WS: `!cfg.type:Maxis`, DevSim missing up to now

EtherCAT Device Description file (EDS) Physik_Instrumente_Hexapod.xml:



Continuous Integration (CI) Pipeline



- VM creation with ELT IFW installation @ host:
  - + `gitlab-runner installation`
- VM setup @ host for a gitlab-runner run:
  - `base.sh, prepare.sh, run.sh, cleanup.sh`
- gitlab-runner setup @ host for run @ VM:
  - `host: gitlab-runner register`
  - `host: /etc/config.toml → prepare_exec, run_exec, … → VM setup prepare.sh, run.sh, …`
  - `execution of pipeline cmds: ${1}@run.sh`
- ssh keys for login without password.

Implemented by former colleague Valentin Ziel.