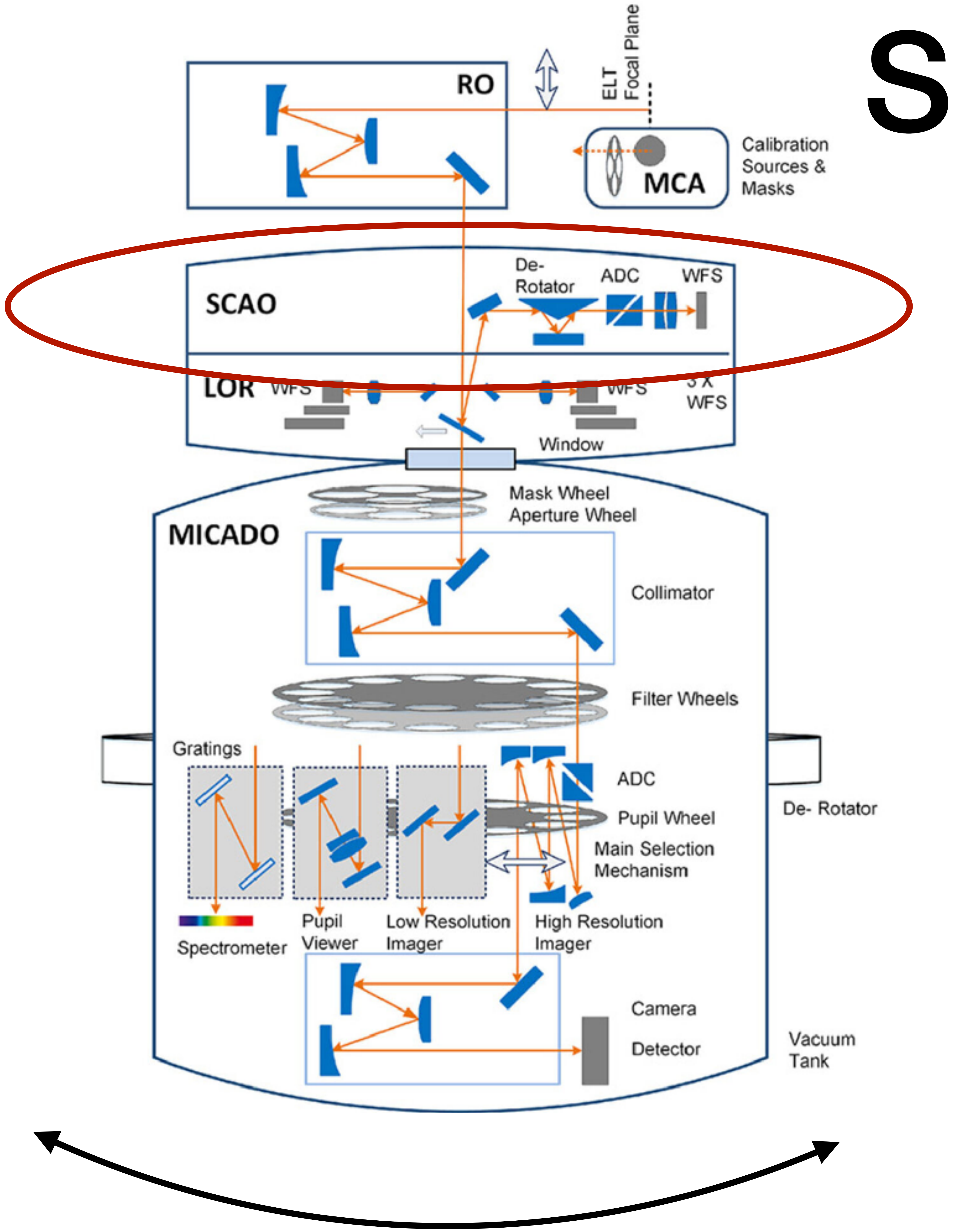
IFW Workshop 2024

MICADO / SCAO

Sylvain Guieu
Roderick Dembet, Gilles Fasola
Sylvestre Taburet , Jordan Raffael
EFISOFT

SCAO In a Nutshell



Single-Conjugated Adaptive Optics For MICADO

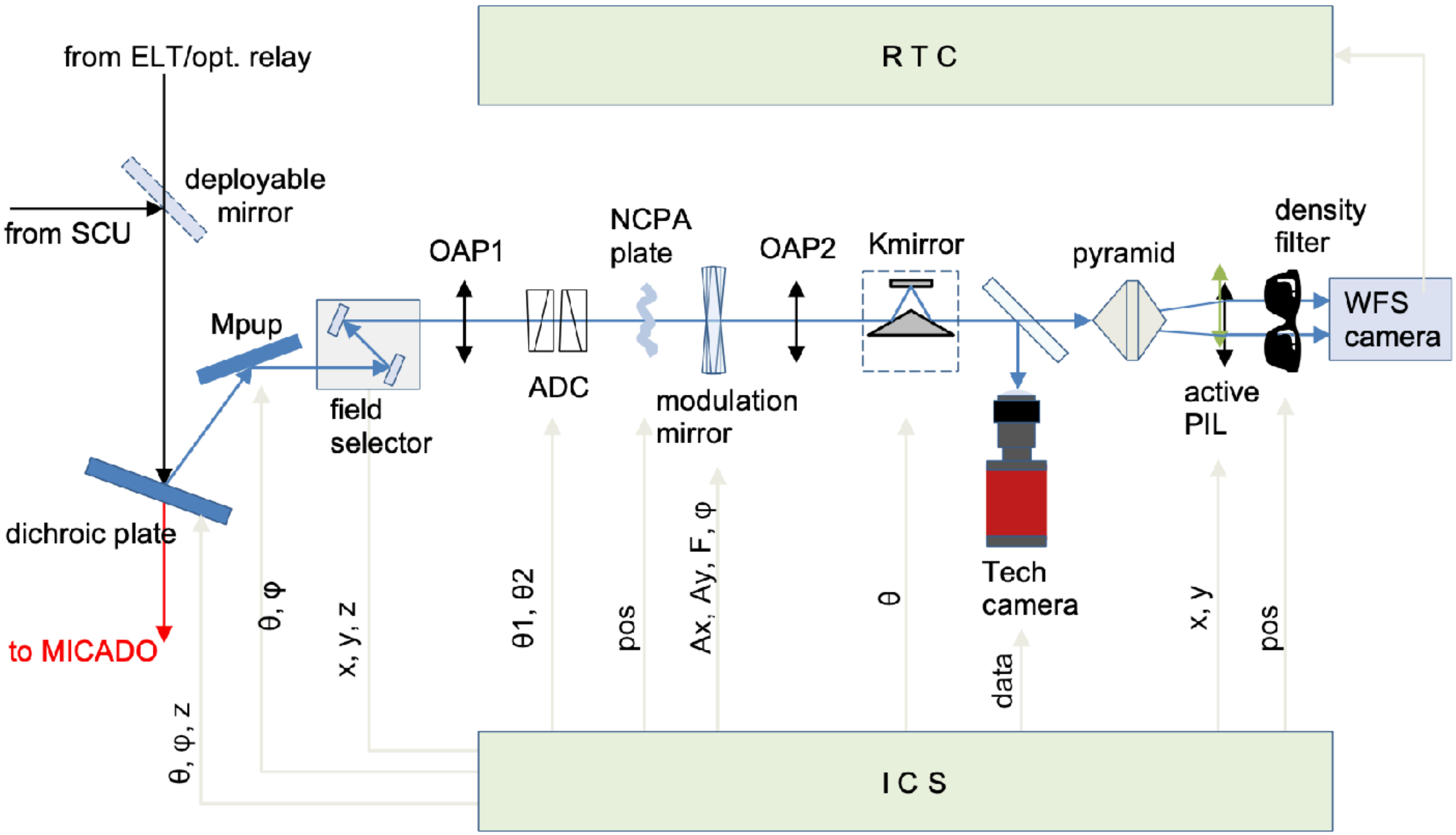
Done at LESIA. INSU / France

MICAO will work on “Standalone” with SCAO before MORFEO arrival.
SCAO SW will be “merged” into MORFEO SW

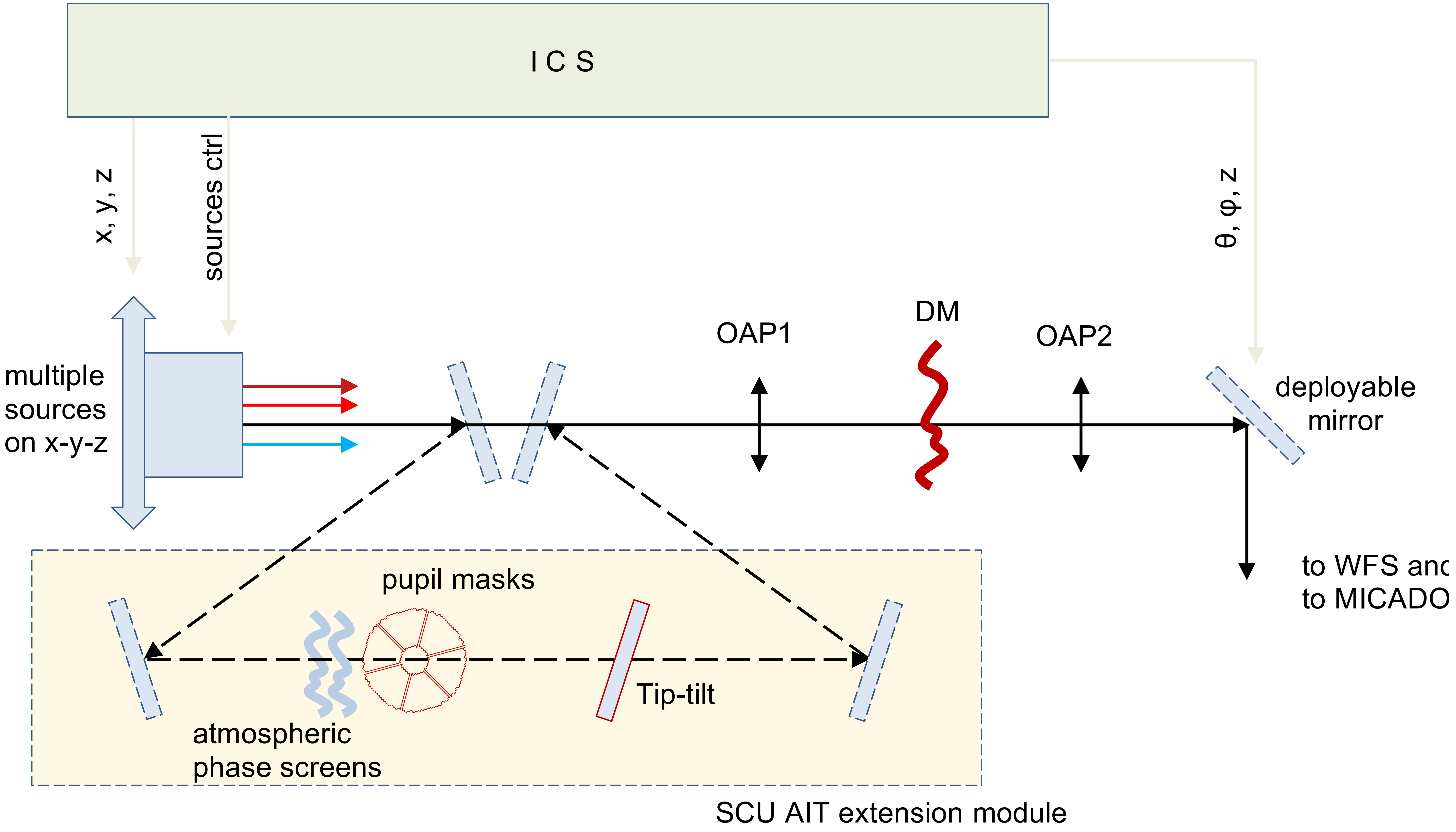
- **Pyramid Wave Front Sensor @ 0.71µm**
- **Patrol Field of 6” x 20”**

- **Interface with the CCS. MICADO send command to SCAO SW (MORFEO) Not CCS**

SCAO


R T C
from ELT/opt. relay
deployable mirror
from SCU
Mpup
field selector
OAP1
ADC
NCPA plate
modulation mirror
OAP2
Kmirror
pyramid
density filter
WFS camera
active PIL
Tech camera
dichroic plate
to MICADO
θ, φ, z
θ, φ
x, y, z
θ1, θ2
pos
Ax, Ay, F, φ
θ
data
x, y
pos
I C S

Calibration Unit

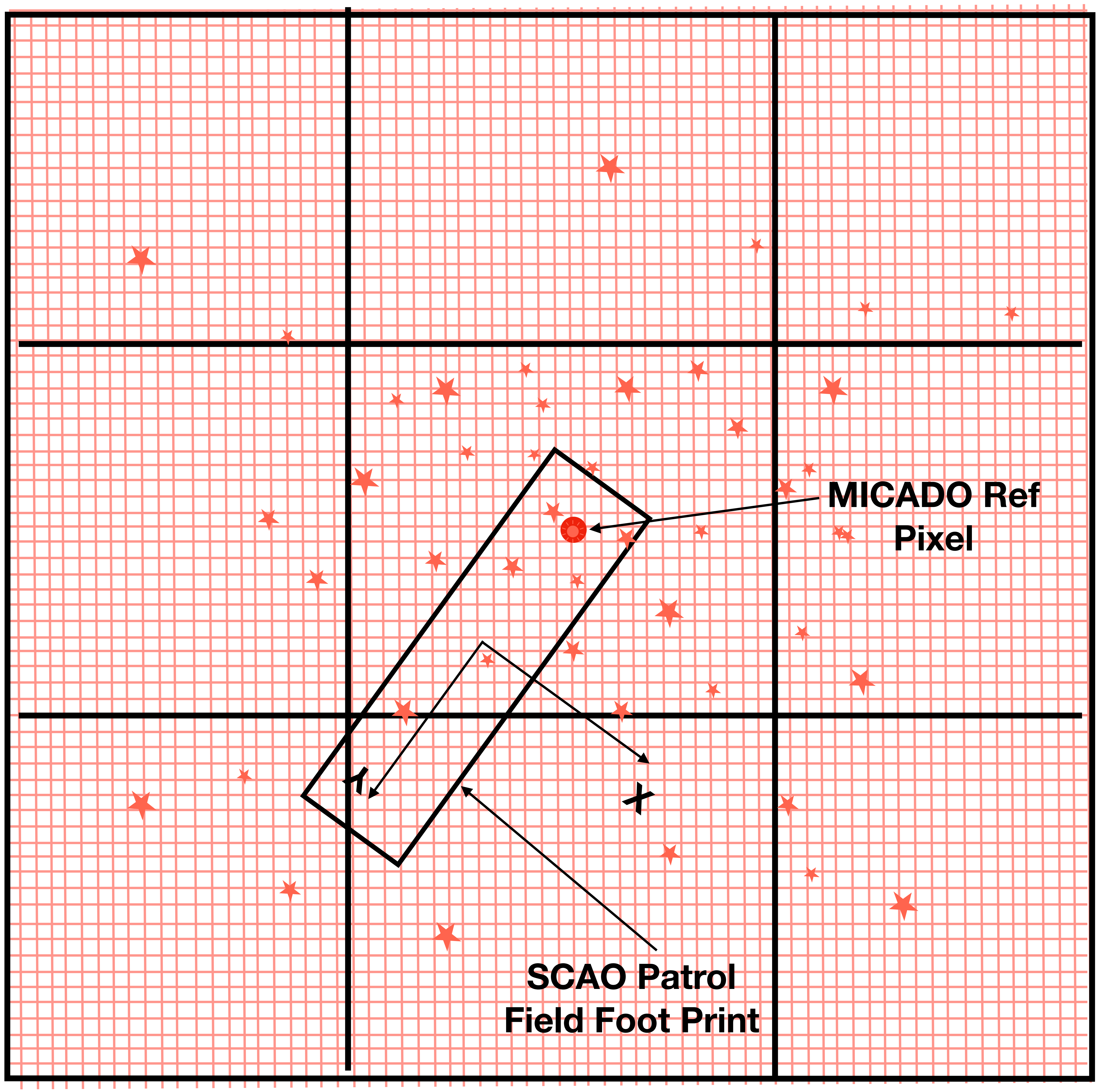


FCS 3 Instances

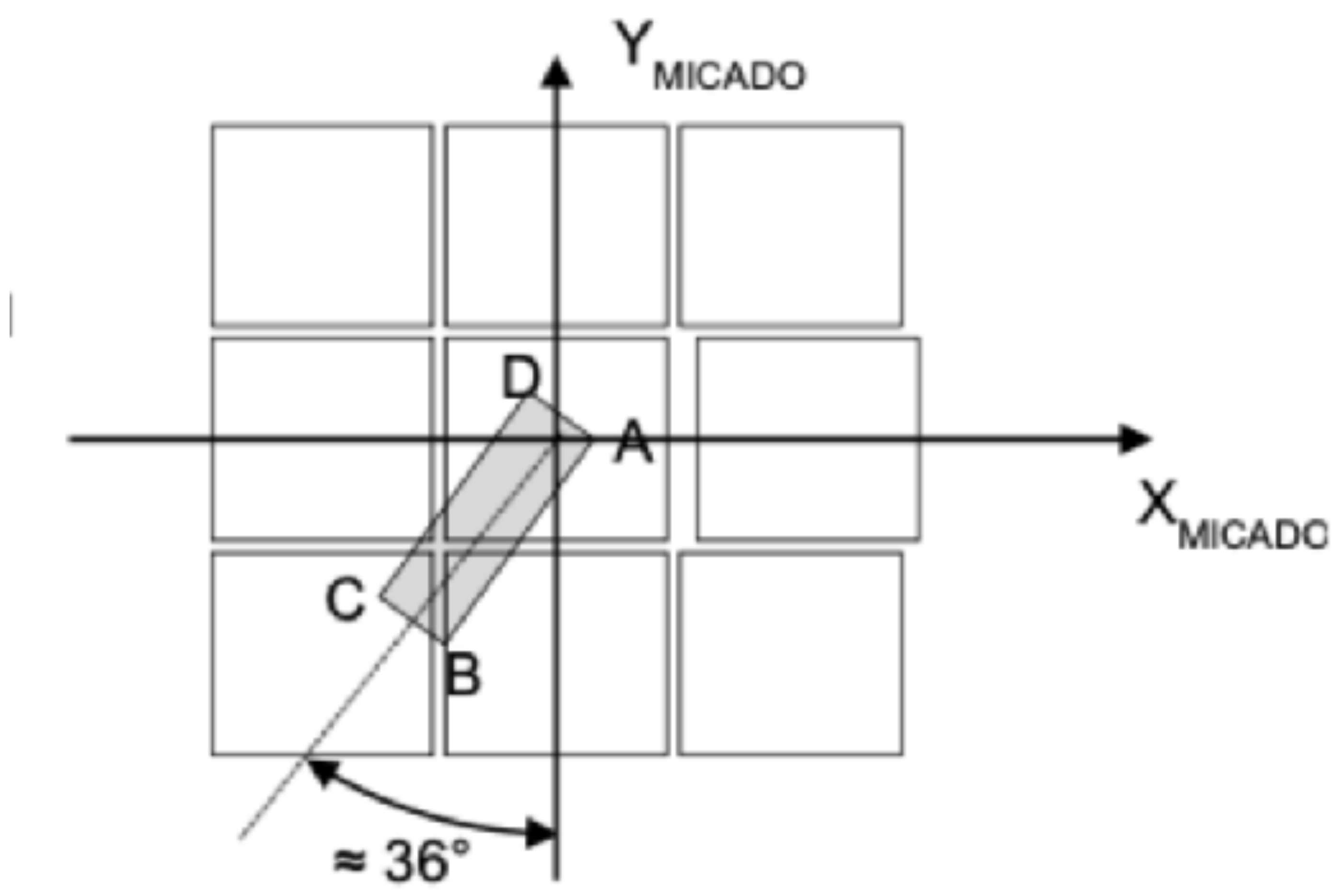
- **FCS1 SCAO Devices**
- **FCS2 SCU Calibration unit**
- **FCS3 devices on MICADO Optical Path**

MICADO Seeing
The Sky
(NIR light)

SCAO Seeing
The Sky
(R Band)

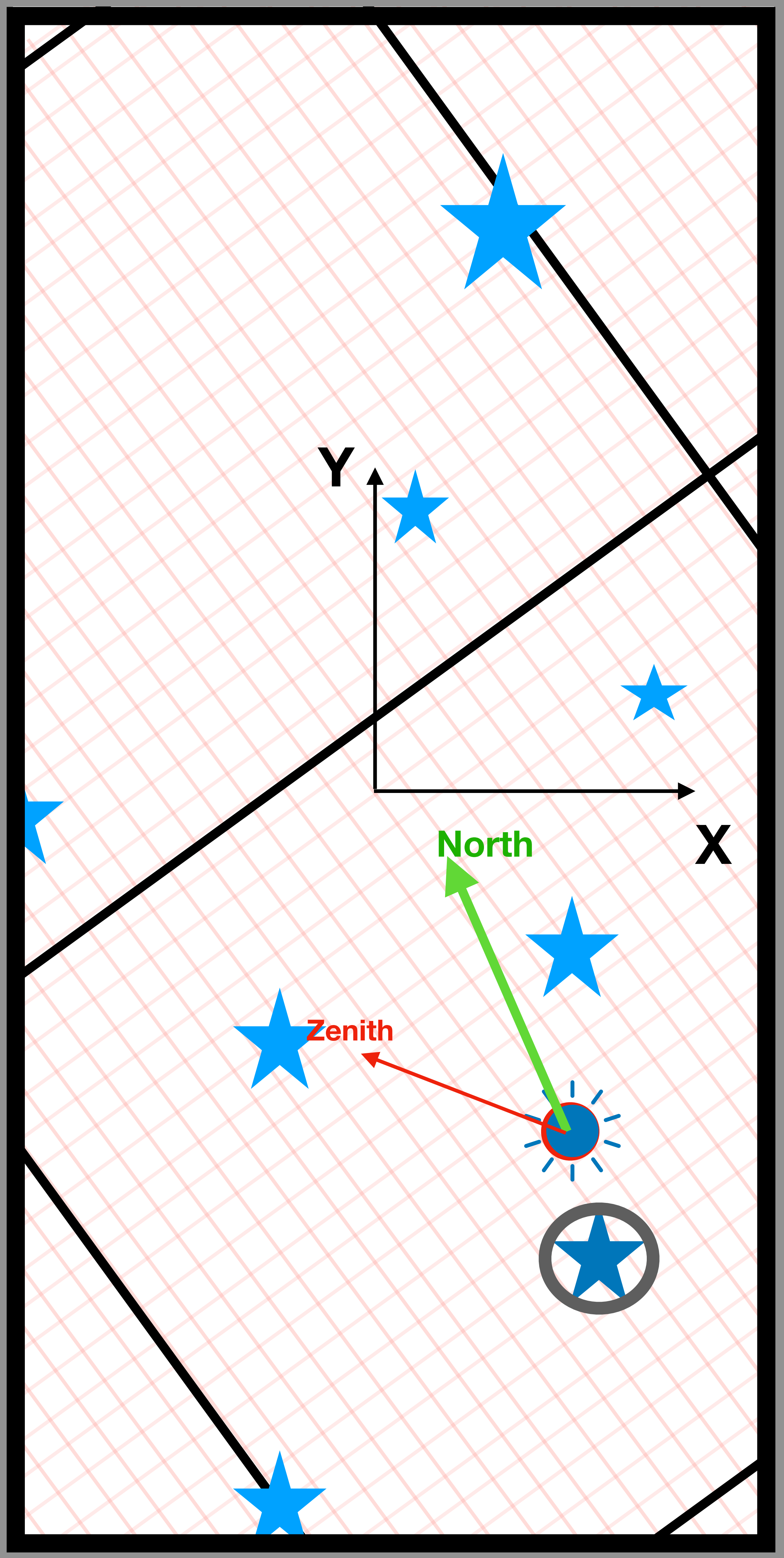


MICADO And SCAO
Are “Glued” to each
others



As Seen From SCAO

- Select Guide star in The field
- Handle MICADO User PA and rotation MODE (SKY & ELEV)

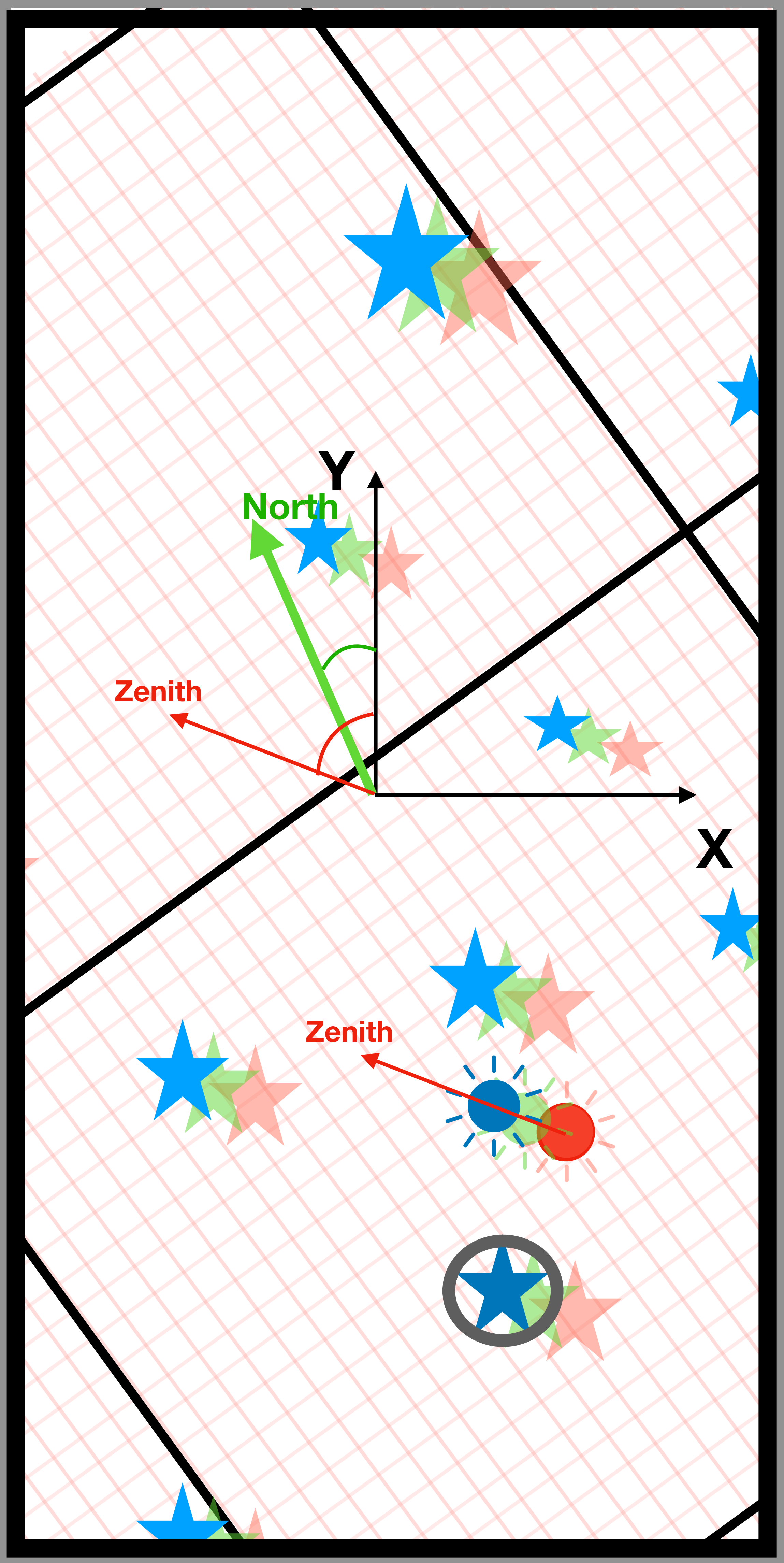


As Seen From SCAO

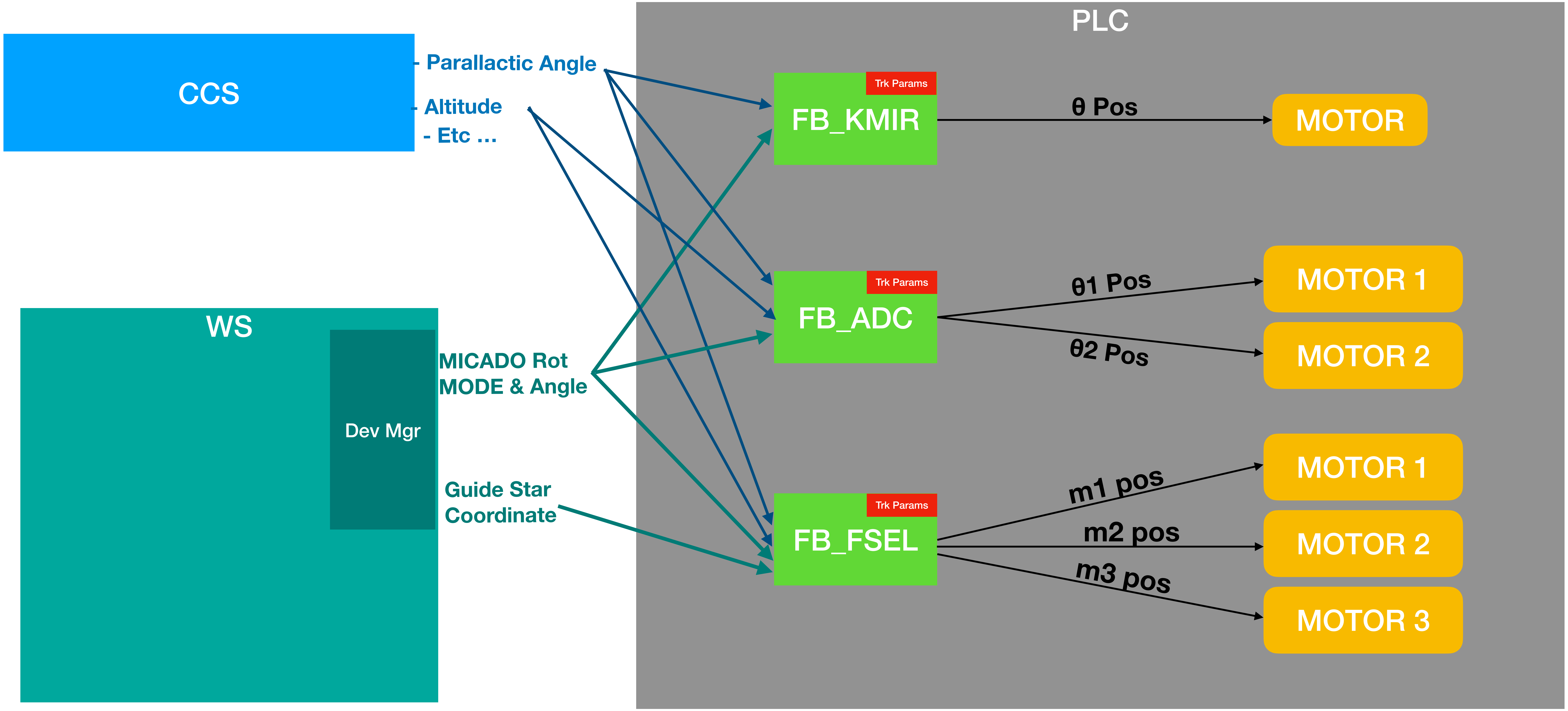
- Select Guide star in The field
- Handle MICADO User PA and rotation MODE (SKY & ELEV)

- Handle Differential atmospheric dispersion

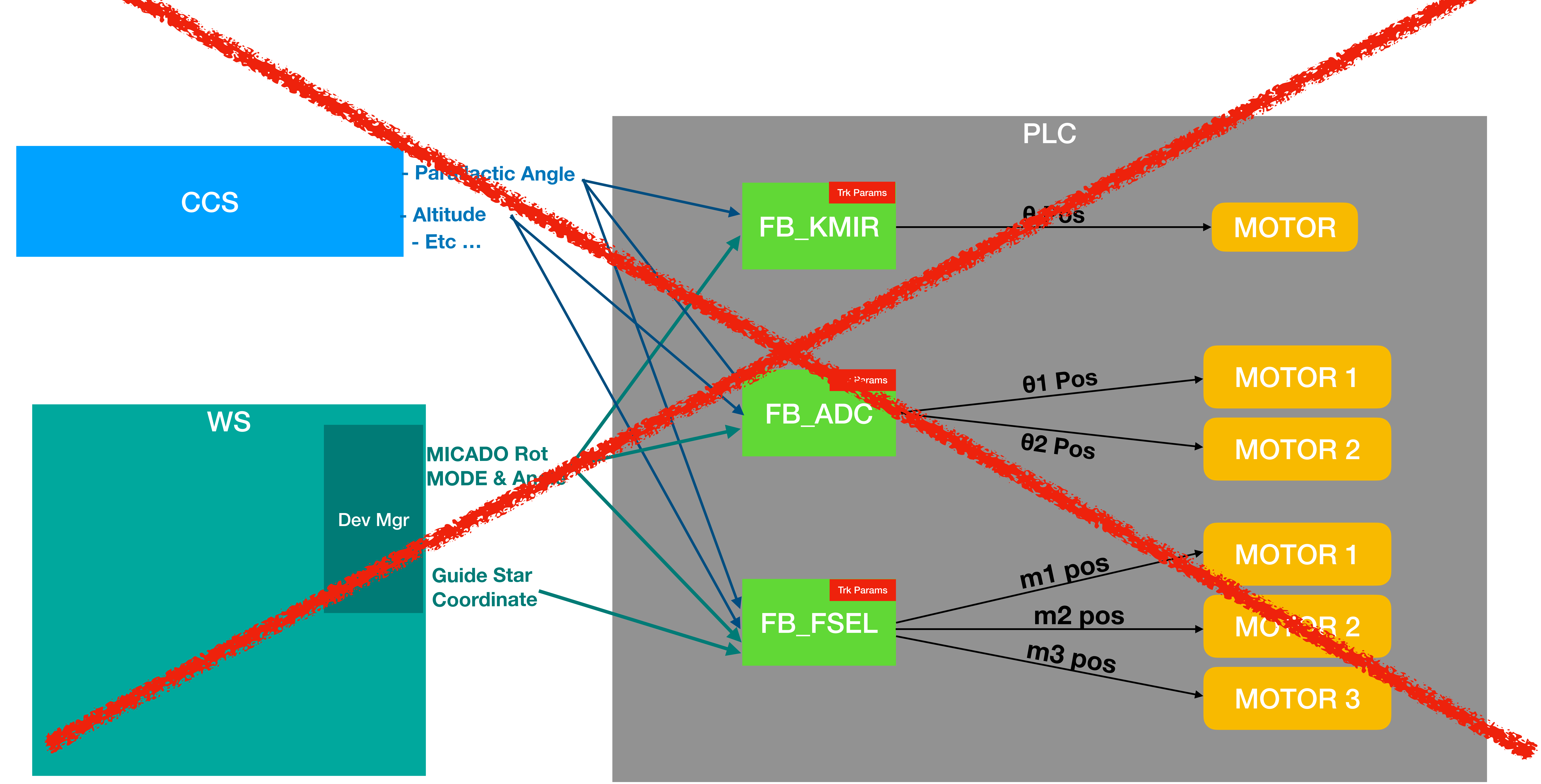
- Handle non sidereal tracking



Solution 1 : self-tracking devices (ESO standard like)

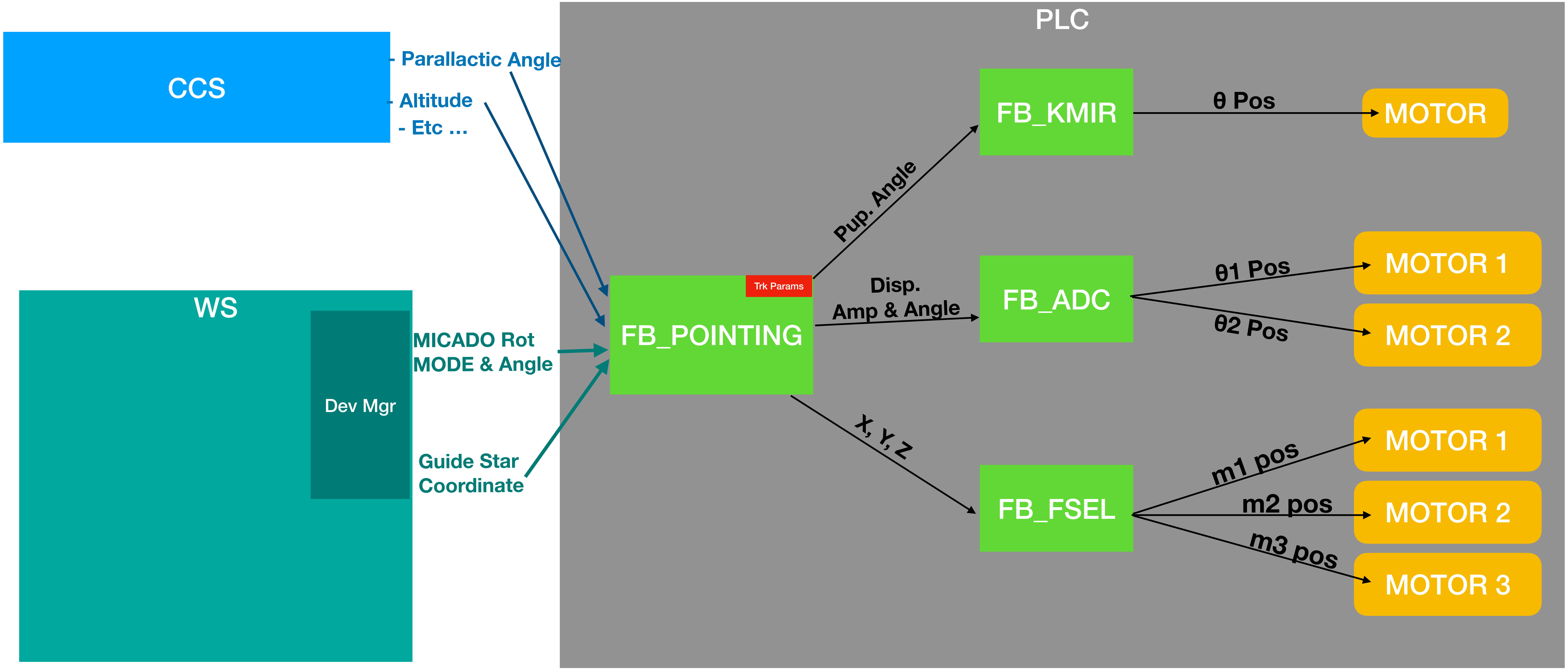


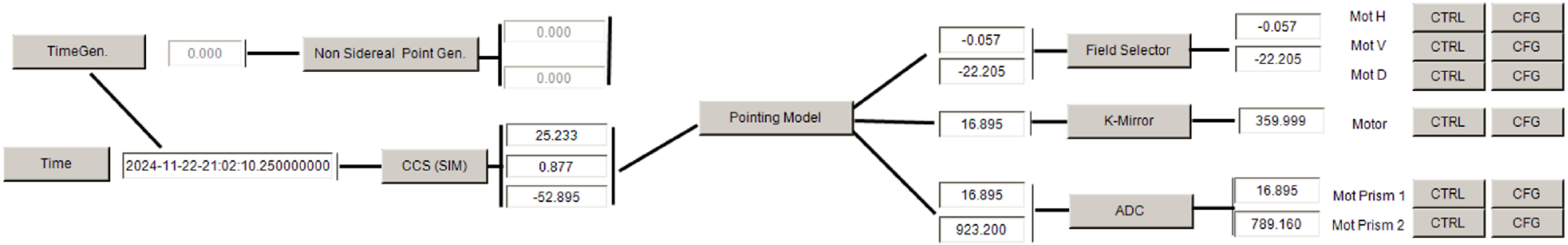
Solution 1 : self-tracking devices (ESO standard like)

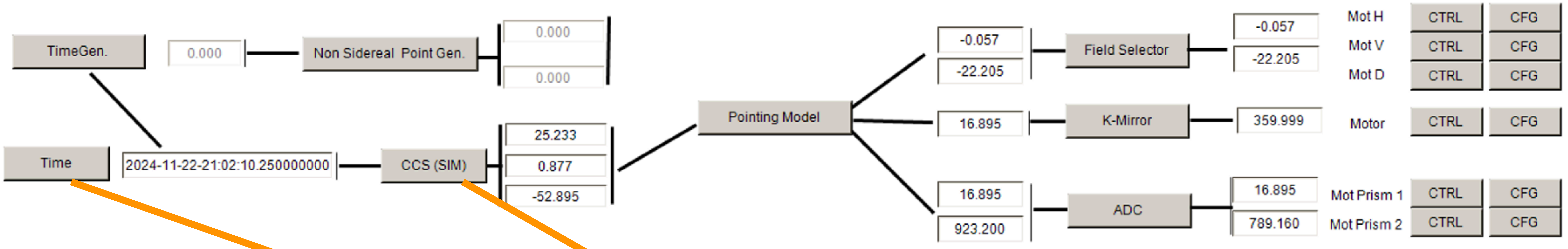


Solution 2 : pointing box

A Pointing FB is computing device target in the “subsystem” coordinate




TimeGen.
0.000
Non Sidereal Point Gen.
0.000
0.000
Time
2024-11-22-21:02:10.250000000
CCS (SIM)
25.233
0.877
-52.895
Pointing Model
-0.057
-22.205
Field Selector
-0.057
-22.205
Mot H
Mot V
Mot D
16.895
K-Mirror
359.999
Motor
16.895
923.200
ADC
16.895
789.160
Mot Prism 1
Mot Prism 2
CTRL
CFG



TimeGen. 0.000 — Non Sidereal Point Gen. 0.000 / 0.000

Time 2024-11-22-21:02:10.250000000 — CCS (SIM) 25.233 / 0.877 / -52.895

Pointing Model

-0.057 / -22.205 — Field Selector — -0.057 / -22.205

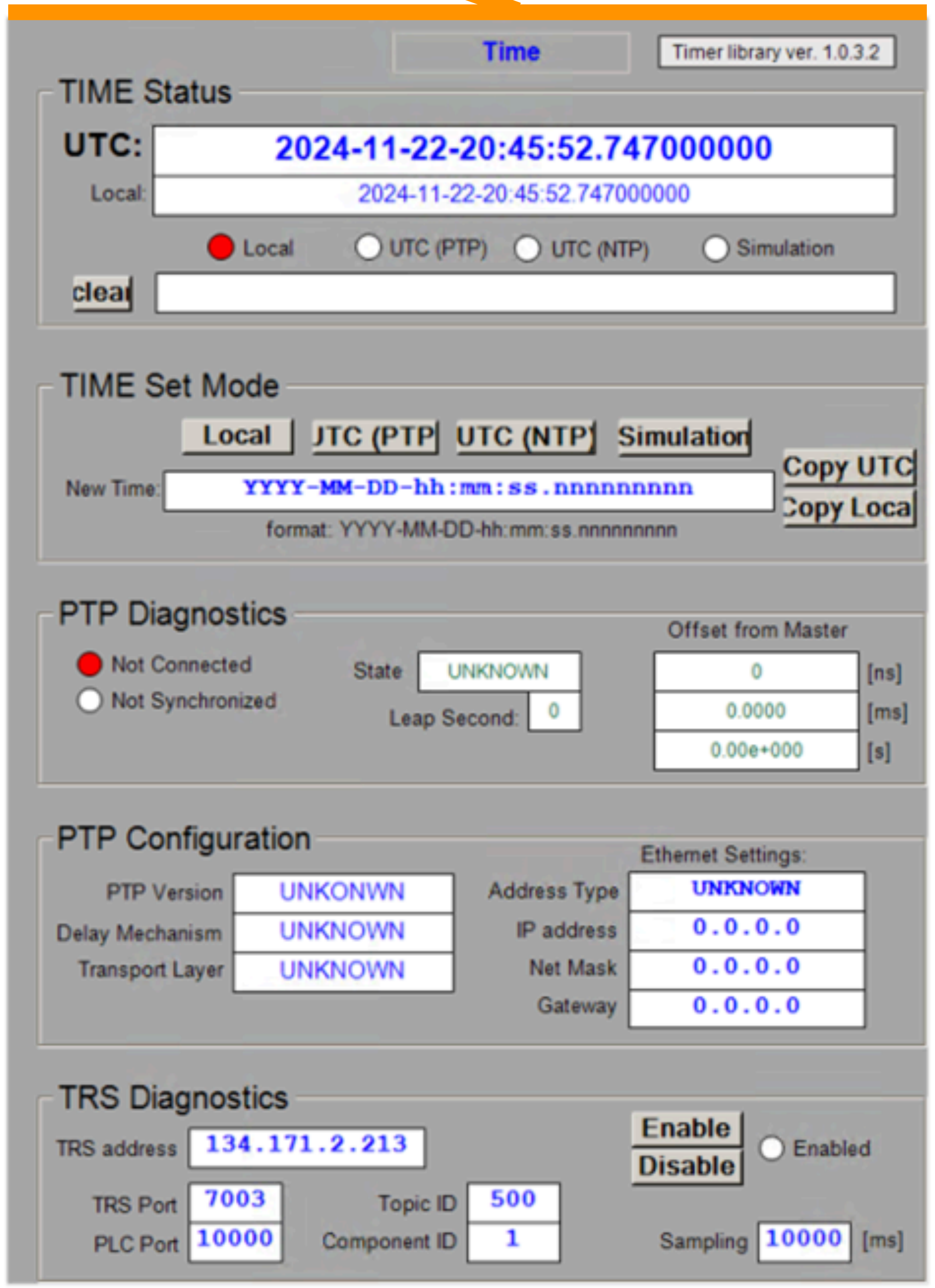
Mot H	CTRL	CFG
Mot V	CTRL	CFG
Mot D	CTRL	CFG

16.895 — K-Mirror — 359.999

Motor	CTRL	CFG

16.895 / 923.200 — ADC — 16.895 / 789.160

Mot Prism 1	CTRL	CFG
Mot Prism 2	CTRL	CFG



Time — Timer library ver. 1.0.3.2

TIME Status

UTC: 2024-11-22-20:45:52.747000000

Local: 2024-11-22-20:45:52.747000000

● Local ○ UTC (PTP) ○ UTC (NTP) ○ Simulation

clear

TIME Set Mode

Local | JTC (PTP | UTC (NTP) | Simulation

New Time: YYYY-MM-DD-hh:mm:ss.nnnnnnnnn

Copy UTC | Copy Loca

format: YYYY-MM-DD-hh:mm:ss.nnnnnnnnn

PTP Diagnostics

● Not Connected
○ Not Synchronized

State: UNKNOWN
Leap Second: 0

Offset from Master: 0 [ns] / 0.0000 [ms] / 0.00e+000 [s]

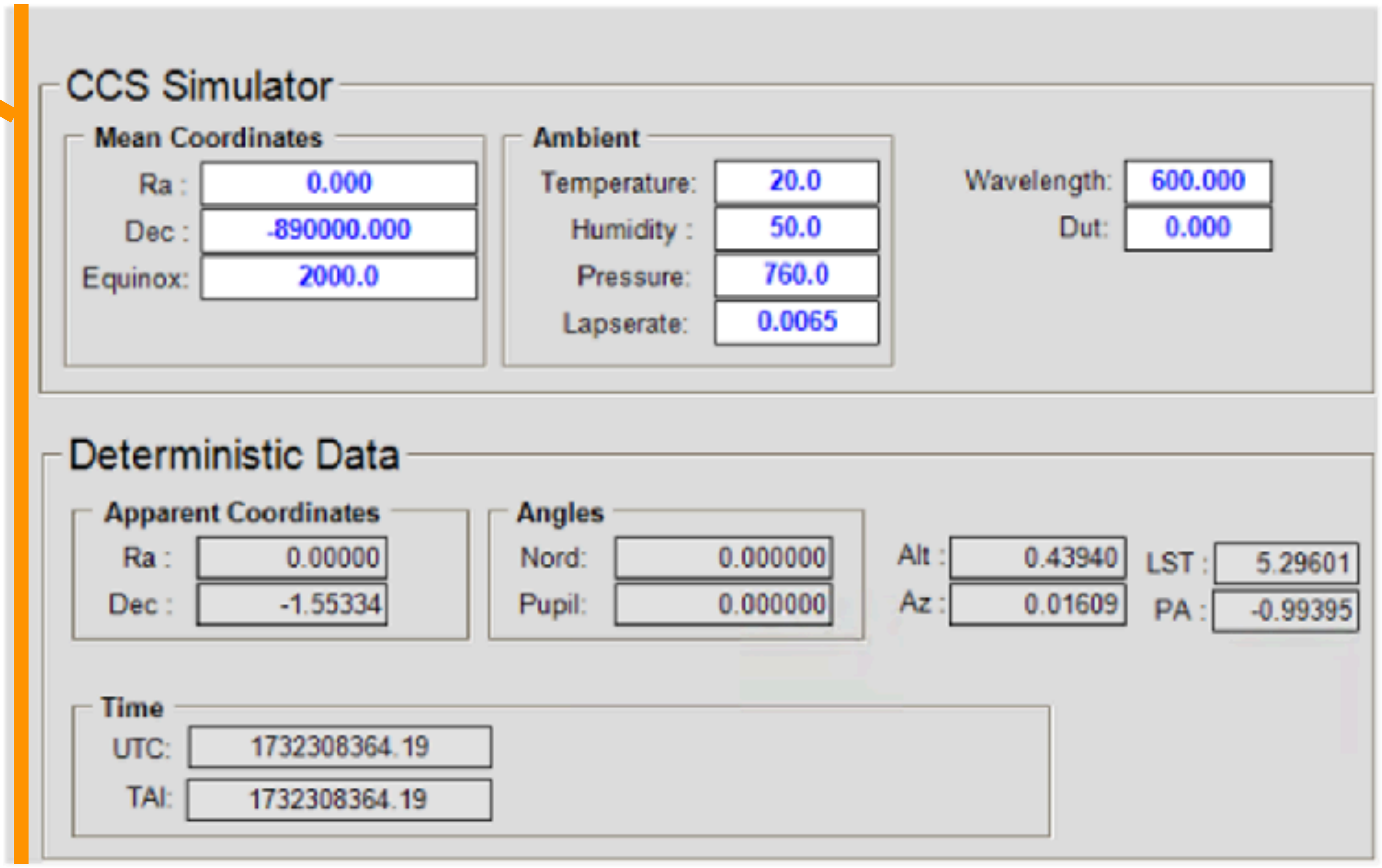
PTP Configuration

PTP Version: UNKONWN
Delay Mechanism: UNKNOWN
Transport Layer: UNKNOWN

Ethernet Settings:
Address Type: UNKNOWN
IP address: 0.0.0.0
Net Mask: 0.0.0.0
Gateway: 0.0.0.0

TRS Diagnostics

TRS address: 134.171.2.213
Enable | Disable ○ Enabled
TRS Port: 7003
PLC Port: 10000
Topic ID: 500
Component ID: 1
Sampling: 10000 [ms]



CCS Simulator

Mean Coordinates
Ra : 0.000
Dec : -890000.000
Equinox: 2000.0

Ambient
Temperature: 20.0
Humidity : 50.0
Pressure: 760.0
Lapserate: 0.0065

Wavelength: 600.000
Dut: 0.000

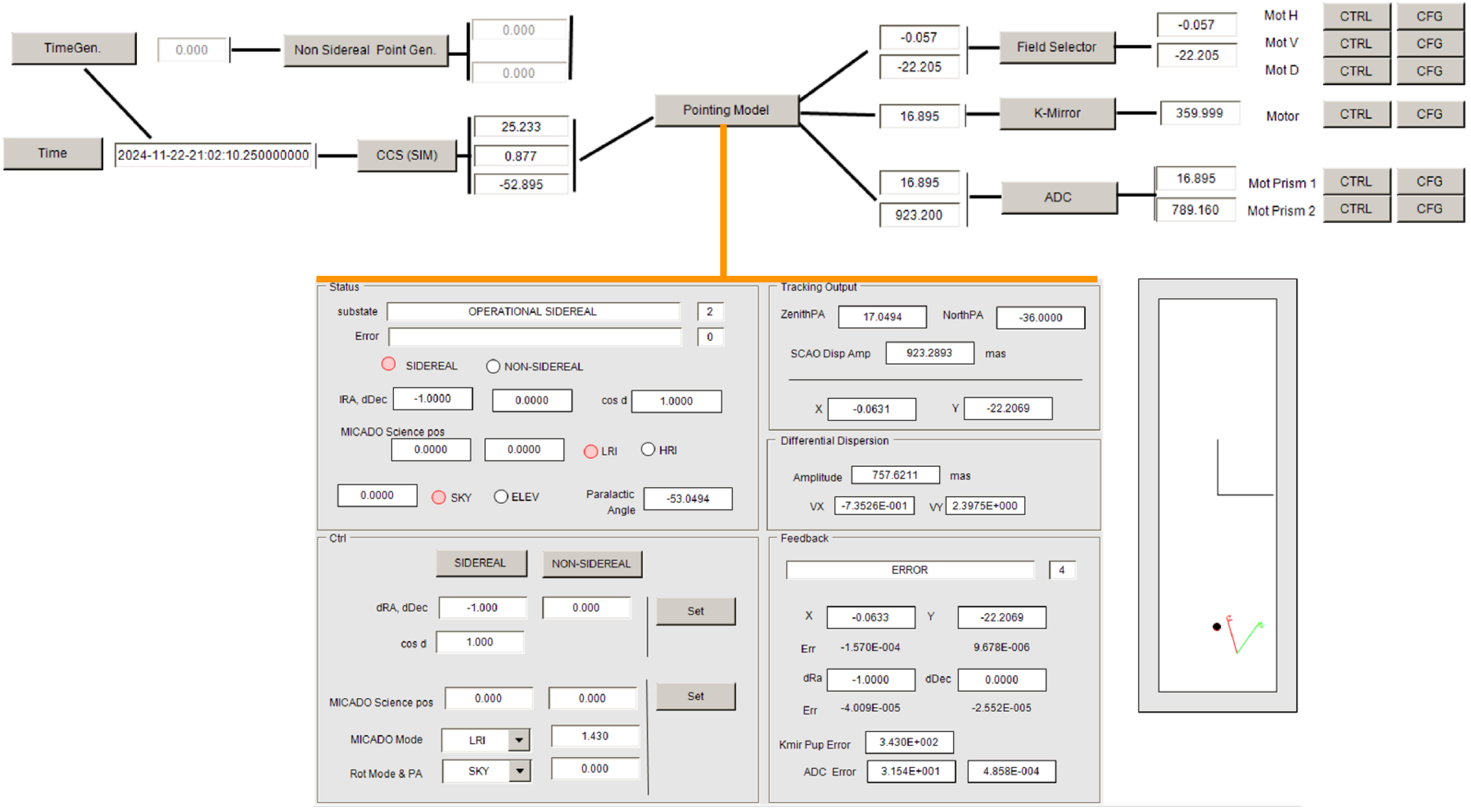
Deterministic Data

Apparent Coordinates
Ra : 0.00000
Dec : -1.55334

Angles
Nord: 0.000000
Pupil: 0.000000

Alt : 0.43940
Az : 0.01609
LST : 5.29601
PA : -0.99395

Time
UTC: 1732308364.19
TAI: 1732308364.19



TimeGen. 0.000

Non Sidereal Point Gen. 0.000 / 0.000

Time 2024-11-22-21:02:10.250000000

CCS (SIM) 25.233 / 0.877 / -52.895

Field Selector: -0.057 / -22.205 → -0.057 / -22.205
Mot H CTRL CFG
Mot V CTRL CFG
Mot D CTRL CFG

K-Mirror: 16.895 → 359.999
Motor CTRL CFG

ADC: 16.895 / 923.200 → 16.895 / 789.160
Mot Prism 1 CTRL CFG
Mot Prism 2 CTRL CFG

Status

substate OPERATIONAL SIDEREAL 2
Error 0
● SIDEREAL ○ NON-SIDEREAL
IRA, dDec -1.0000 0.0000 cos d 1.0000
MICADO Science pos 0.0000 0.0000 ● LRI ○ HRI
0.0000 ● SKY ○ ELEV Paralactic Angle -53.0494

Tracking Output

ZenithPA 17.0494 NorthPA -36.0000
SCAO Disp Amp 923.2893 mas
X -0.0631 Y -22.2069

Differential Dispersion

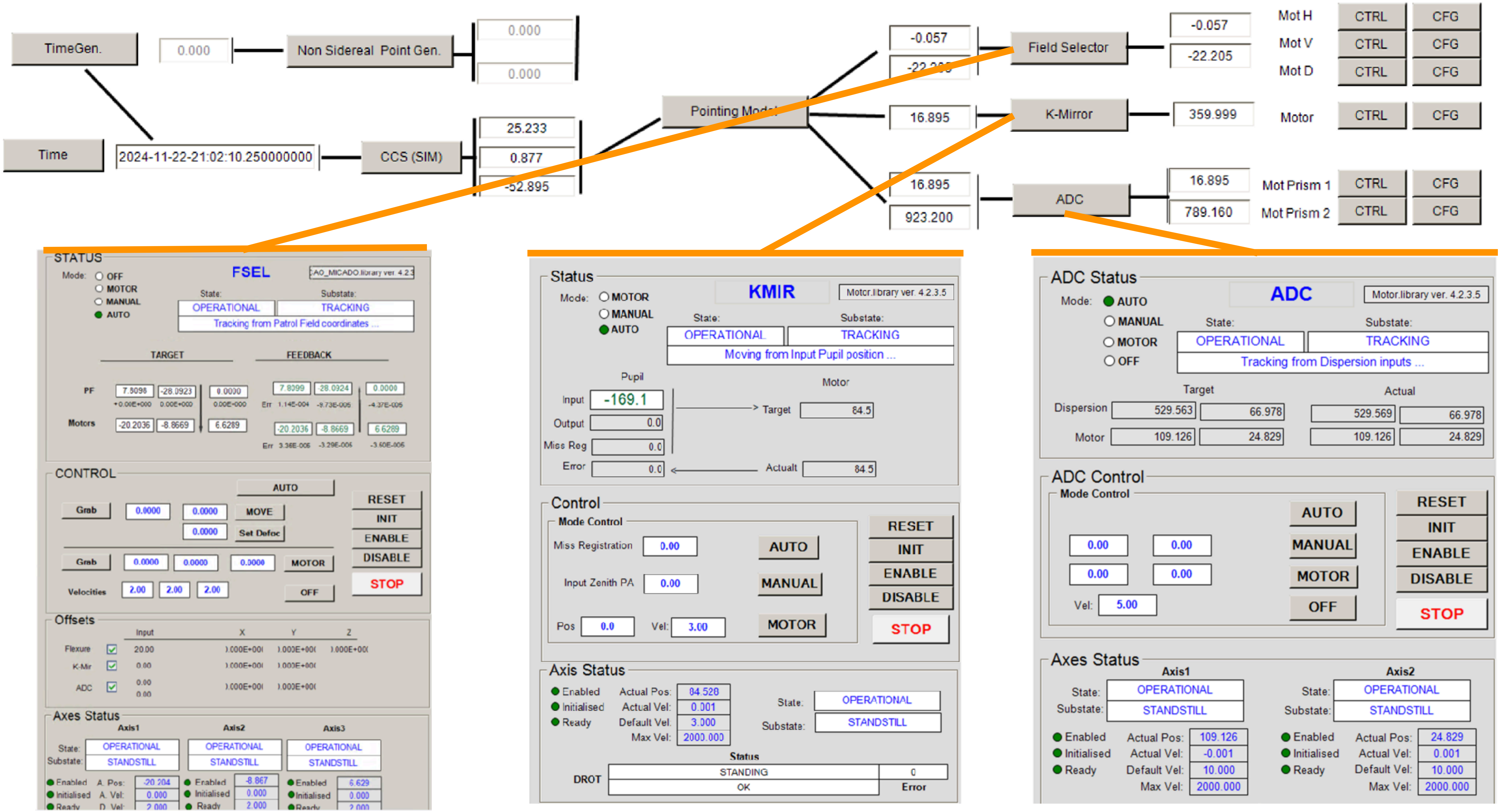
Amplitude 757.6211 mas
VX -7.3526E-001 VY 2.3975E+000

Ctrl

dRA, dDec -1.000 0.000
cos d 1.000
MICADO Science pos 0.000 0.000
MICADO Mode LRI 1.430
Rot Mode & PA SKY 0.000

Feedback

ERROR 4
X -0.0633 Y -22.2069
Err -1.570E-004 9.678E-006
dRa -1.0000 dDec 0.0000
Err -4.009E-005 -2.552E-005
Kmir Pup Error 3.430E+002
ADC Error 3.154E+001 4.858E-004



TimeGen. 0.000 — Non Sidereal Point Gen. 0.000 0.000

Time 2024-11-22-21:02:10.250000000 — CCS (SIM) 25.233 0.877 -52.895

Pointing Model — -0.057 -22.20[illegible] — Field Selector — -0.057 Mot H CTRL CFG; -22.205 Mot V CTRL CFG; Mot D CTRL CFG

Pointing Model — 16.895 — K-Mirror — 359.999 Motor CTRL CFG

Pointing Model — 16.895 923.200 — ADC — 16.895 Mot Prism 1 CTRL CFG; 789.160 Mot Prism 2 CTRL CFG

FSEL — AO_MICADO.library ver. 4.2.3

STATUS

Mode: ○ OFF ○ MOTOR ○ MANUAL ● AUTO

State: OPERATIONAL Substate: TRACKING

Tracking from Patrol Field coordinates ...

	TARGET			FEEDBACK		
PF	7.8098	-28.0923	0.0000	7.8099	-28.0924	0.0000
	+0.00E+000	0.00E+000	0.00E+000	Err 1.14E-004	-9.73E-006	-4.37E-005
Motors	-20.2036	-8.8669	6.6289	-20.2036	-8.8669	6.6289
				Err 3.36E-006	-3.29E-006	-3.60E-006

CONTROL

AUTO

Grab 0.0000 0.0000 MOVE; 0.0000 Set Defoc

Grab 0.0000 0.0000 0.0000 MOTOR

Velocities 2.00 2.00 2.00 OFF

RESET INIT ENABLE DISABLE STOP

Offsets

		Input	X	Y	Z
Flexure	☑	20.00	).000E+00(	).000E+00(	).000E+00(
K-Mir	☑	0.00	).000E+00(	).000E+00(	
ADC	☑	0.00 0.00	).000E+00(	).000E+00(	

Axes Status

	Axis1		Axis2		Axis3	
State:	OPERATIONAL		OPERATIONAL		OPERATIONAL	
Substate:	STANDSTILL		STANDSTILL		STANDSTILL	
● Enabled	A. Pos:	-20.204	● Enabled	-8.867	● Enabled	6.629
● Initialised	A. Vel:	0.000	● Initialised	0.000	● Initialised	0.000
● Ready	D. Vel:	2.000	● Ready	2.000	● Ready	2.000

KMIR — Motor.library ver. 4.2.3.5

Status

Mode: ○ MOTOR ○ MANUAL ● AUTO

State: OPERATIONAL Substate: TRACKING

Moving from Input Pupil position ...

Pupil — Input -169.1; Output 0.0; Miss Reg 0.0; Error 0.0

Motor — Target 84.5; Actualt 84.5

Control

Mode Control

Miss Registration 0.00 AUTO

Input Zenith PA 0.00 MANUAL

Pos 0.0 Vel: 3.00 MOTOR

RESET INIT ENABLE DISABLE STOP

Axis Status

● Enabled Actual Pos: 84.528

● Initialised Actual Vel: 0.001

● Ready Default Vel: 3.000

Max Vel: 2000.000

State: OPERATIONAL

Substate: STANDSTILL

DROT	Status	
	STANDING	0
	OK	Error

ADC — Motor.library ver. 4.2.3.5

ADC Status

Mode: ● AUTO ○ MANUAL ○ MOTOR ○ OFF

State: OPERATIONAL Substate: TRACKING

Tracking from Dispersion inputs ...

	Target		Actual	
Dispersion	529.563	66.978	529.569	66.978
Motor	109.126	24.829	109.126	24.829

ADC Control

Mode Control

0.00 0.00 AUTO

0.00 0.00 MANUAL

Vel: 5.00 MOTOR

OFF

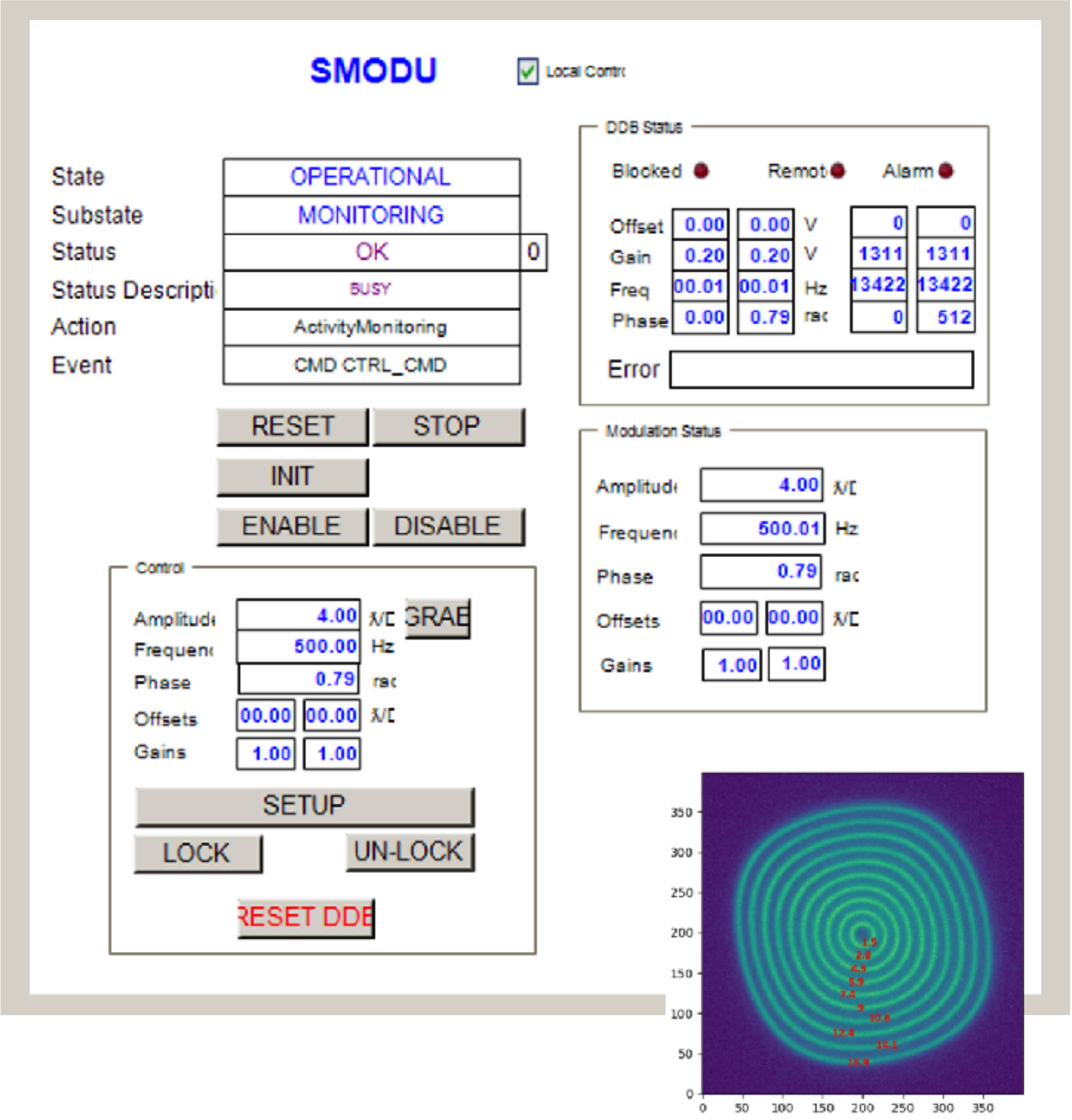
RESET INIT ENABLE DISABLE STOP

Axes Status

	Axis1		Axis2
State:	OPERATIONAL	State:	OPERATIONAL
Substate:	STANDSTILL	Substate:	STANDSTILL
● Enabled	Actual Pos: 109.126	● Enabled	Actual Pos: 24.829
● Initialised	Actual Vel: -0.001	● Initialised	Actual Vel: 0.001
● Ready	Default Vel: 10.000	● Ready	Default Vel: 10.000
	Max Vel: 2000.000		Max Vel: 2000.000

Other Special devices

Pyramid Modulator Serial Com



SMODU

☑ Local Contr

State: OPERATIONAL

Substate: MONITORING

Status: OK 0

Status Descripti: BUSY

Action: ActivityMonitoring

Event: CMD CTRL_CMD

DDB Status

Blocked ● Remot ● Alarm ●

Offset	0.00	0.00	V	0	0
Gain	0.20	0.20	V	1311	1311
Freq	00.01	00.01	Hz	13422	13422
Phase	0.00	0.79	rac	0	512

Error

Modulation Status

Amplitud: 4.00 λ/D

Frequen: 500.01 Hz

Phase: 0.79 rac

Offsets: 00.00 00.00 λ/D

Gains: 1.00 1.00

Control

Amplitud: 4.00 λ/D

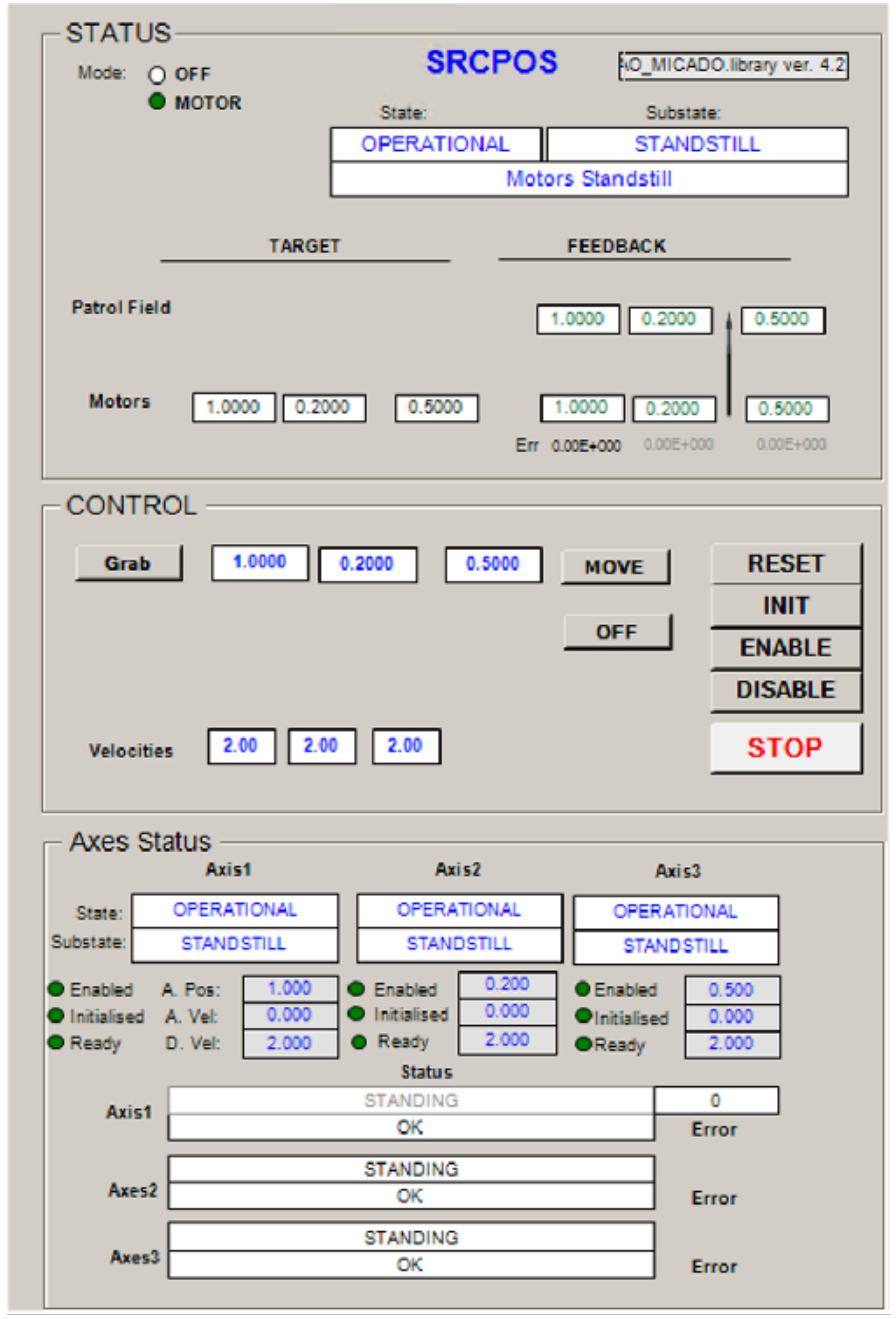
Frequen: 500.00 Hz

Phase: 0.79 rac

Offsets: 00.00 00.00 λ/D

Gains: 1.00 1.00

Source Positioner (3 axes)



STATUS

SRCPOS

AO_MICADO.library ver. 4.2

Mode: ○ OFF ● MOTOR

State: OPERATIONAL

Substate: STANDSTILL

Motors Standstill

	TARGET			FEEDBACK		
Patrol Field				1.0000	0.2000	0.5000
Motors	1.0000	0.2000	0.5000	1.0000	0.2000	0.5000
			Err	0.00E+000	0.00E+000	0.00E+000

CONTROL

Grab 1.0000 0.2000 0.5000 MOVE

Velocities 2.00 2.00 2.00

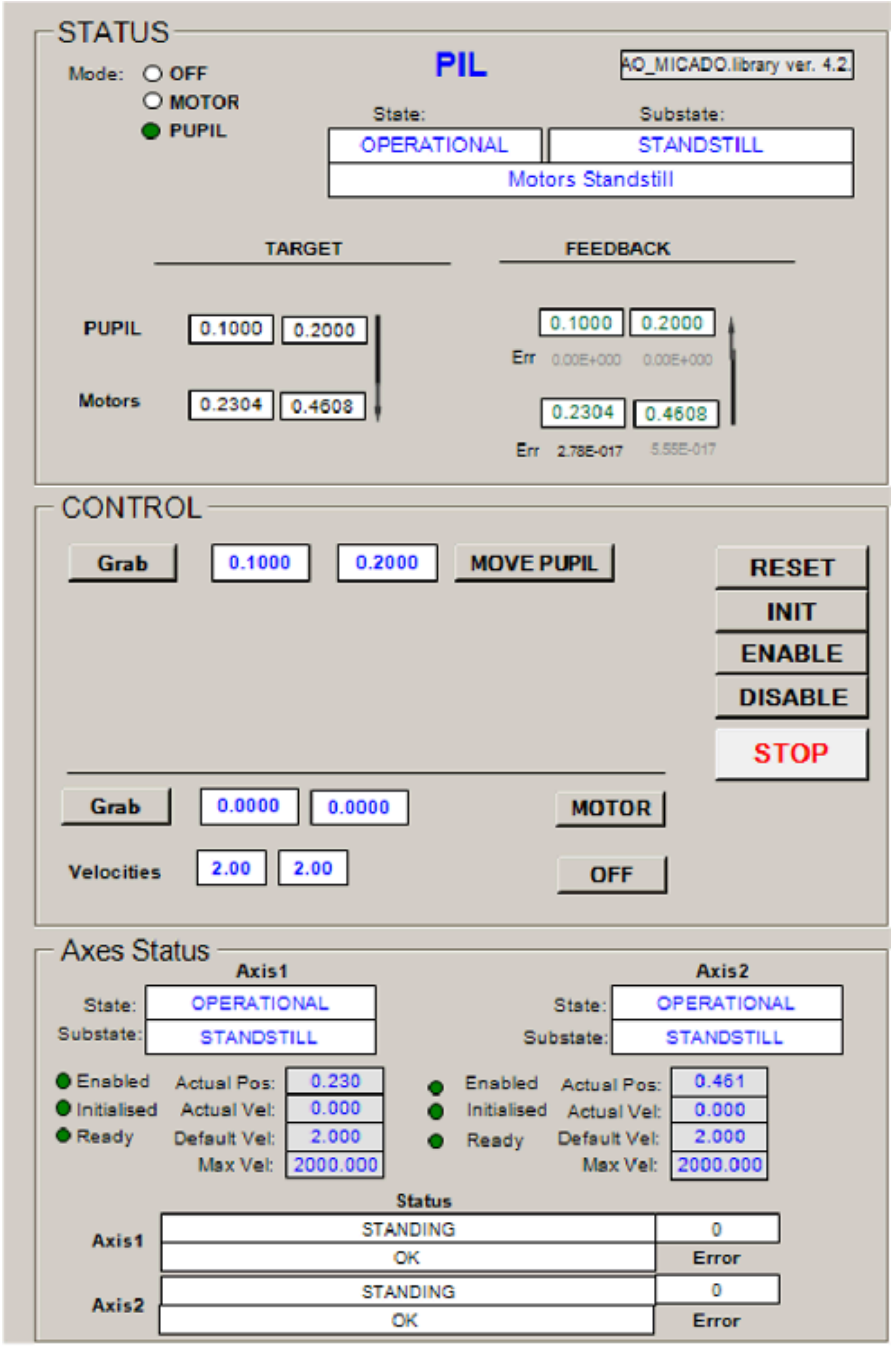
Axes Status

	Axis1		Axis2	Axis3
State:	OPERATIONAL		OPERATIONAL	OPERATIONAL
Substate:	STANDSTILL		STANDSTILL	STANDSTILL
● Enabled	A. Pos:	1.000	● Enabled 0.200	● Enabled 0.500
● Initialised	A. Vel:	0.000	● Initialised 0.000	● Initialised 0.000
● Ready	D. Vel:	2.000	● Ready 2.000	● Ready 2.000

Status

Axis1	STANDING OK	0 Error
Axes2	STANDING OK	Error
Axes3	STANDING OK	Error

PIL (2 axes)



STATUS

PIL

AO_MICADO.library ver. 4.2.

Mode: ○ OFF ○ MOTOR ● PUPIL

State: OPERATIONAL

Substate: STANDSTILL

Motors Standstill

	TARGET			FEEDBACK	
PUPIL	0.1000	0.2000		0.1000	0.2000
			Err	0.00E+000	0.00E+000
Motors	0.2304	0.4608		0.2304	0.4608
			Err	2.78E-017	5.55E-017

CONTROL

Grab 0.1000 0.2000 MOVE PUPIL

Grab 0.0000 0.0000 MOTOR

Velocities 2.00 2.00

Axes Status

	Axis1			Axis2
State:	OPERATIONAL		State:	OPERATIONAL
Substate:	STANDSTILL		Substate:	STANDSTILL
● Enabled	Actual Pos: 0.230	● Enabled	Actual Pos:	0.461
● Initialised	Actual Vel: 0.000	● Initialised	Actual Vel:	0.000
● Ready	Default Vel: 2.000	● Ready	Default Vel:	2.000
	Max Vel: 2000.000		Max Vel:	2000.000

Status

Axis1	STANDING OK	0 Error
Axis2	STANDING OK	0 Error

Secondary Processes

- Telescope Interface. For Non Sidereal Tracking. Instrument will be in charge of The ephemeris file (change in CSS/Instrument ICD)

- RTC to ICS offsets: receive miss-registration and offset measured by the RTC and apply correction to hardware. (Pupil rotation and translation)

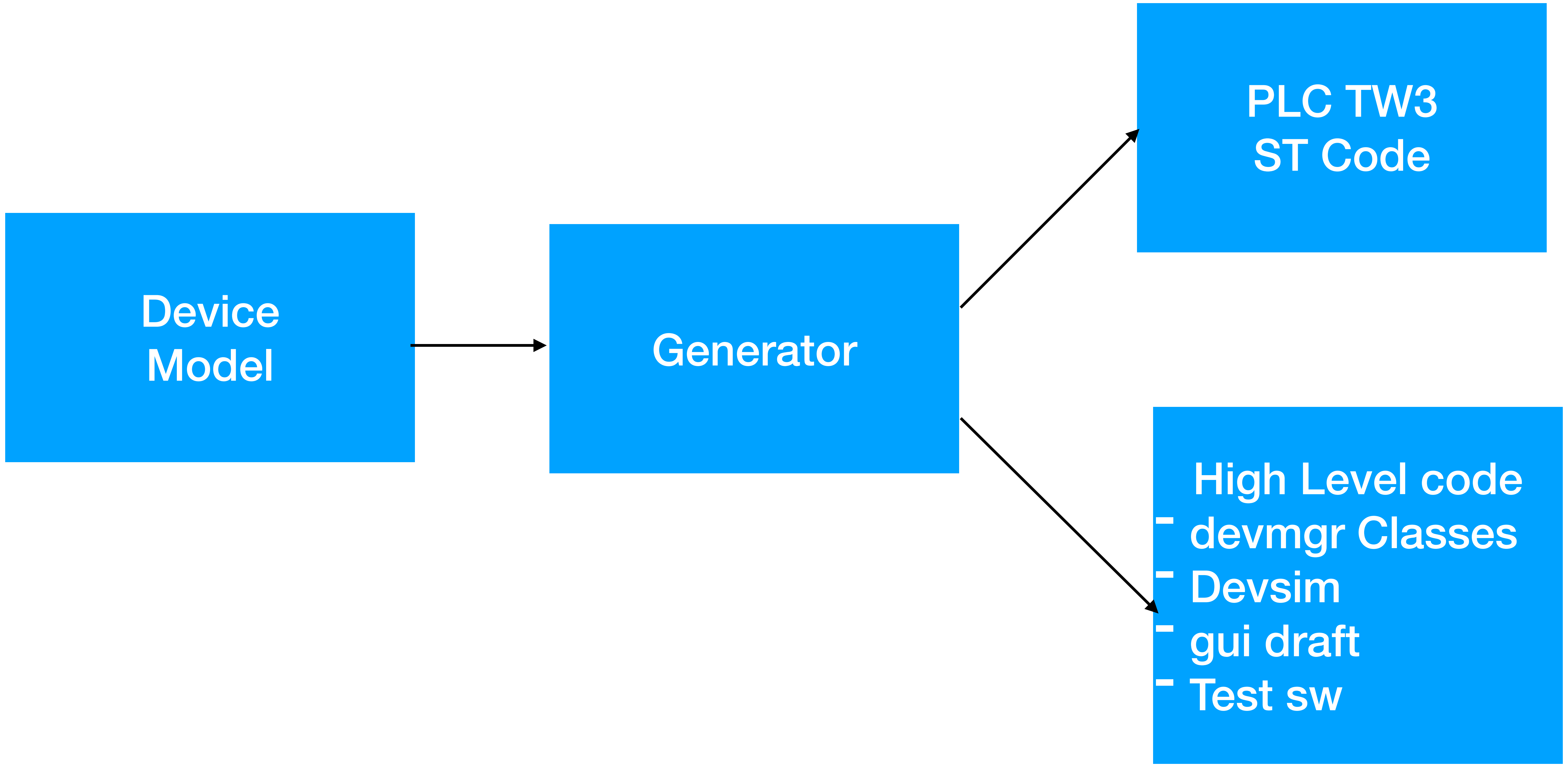
- Front-end process: receive commands from MICADO to setup SCAO

Other features

- **SCAO Calibration Templates**
- **API (python library) for MICADO + MICADO / MORFEO ICD**
- **SCAO RTC to SCAO ICS API (Python Library)**
- **CCF, 2 instances: One wave-front sensor (ALICE) and one technical camera (pupil imager)**
- **Deformable Mirror on the calibration unit. Housekeeping by ICS and control by RTC.**
- **AIT tools (mainly python scripts).**
- **High level Code generation tool. Please see HARMONI presentation (or now ?).**

Device Generator

Ideally: We would to have a model definition of a Device and generate everything but logic business



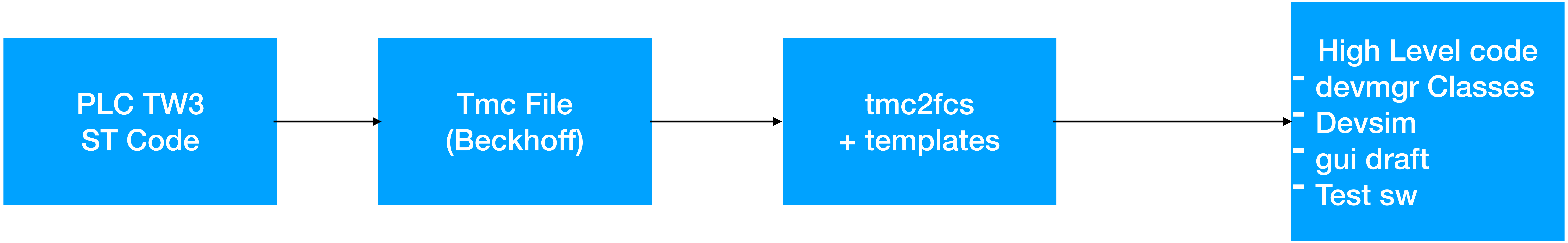
High Level code
- devmgr Classes
- Devsim
- gui draft
- Test sw

But:

- **A lot of development**
- **Risk of doing a model as complicated as the real implementation**
- **Model to Implementation code hard to maintain.**

Device Generator

Compromise: Use PLC tmc file to generate high level



PLC TW3
ST Code

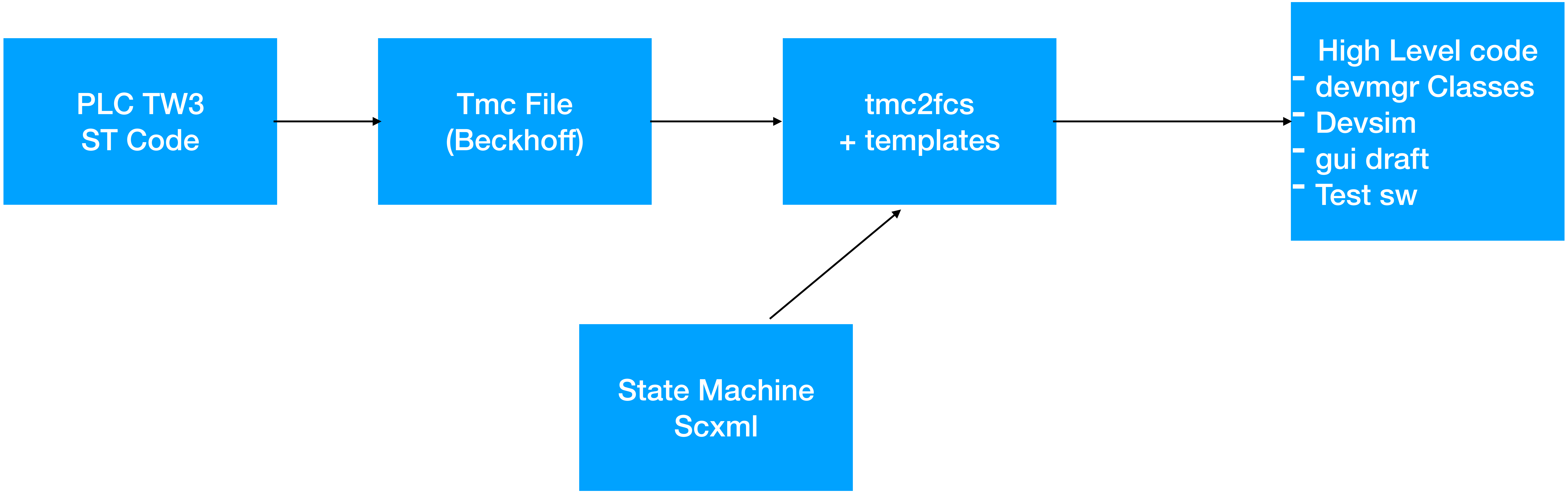
Tmc File
(Beckhoff)

tmc2fcs
+ templates

High Level code
- devmgr Classes
- Devsim
- gui draft
- Test sw

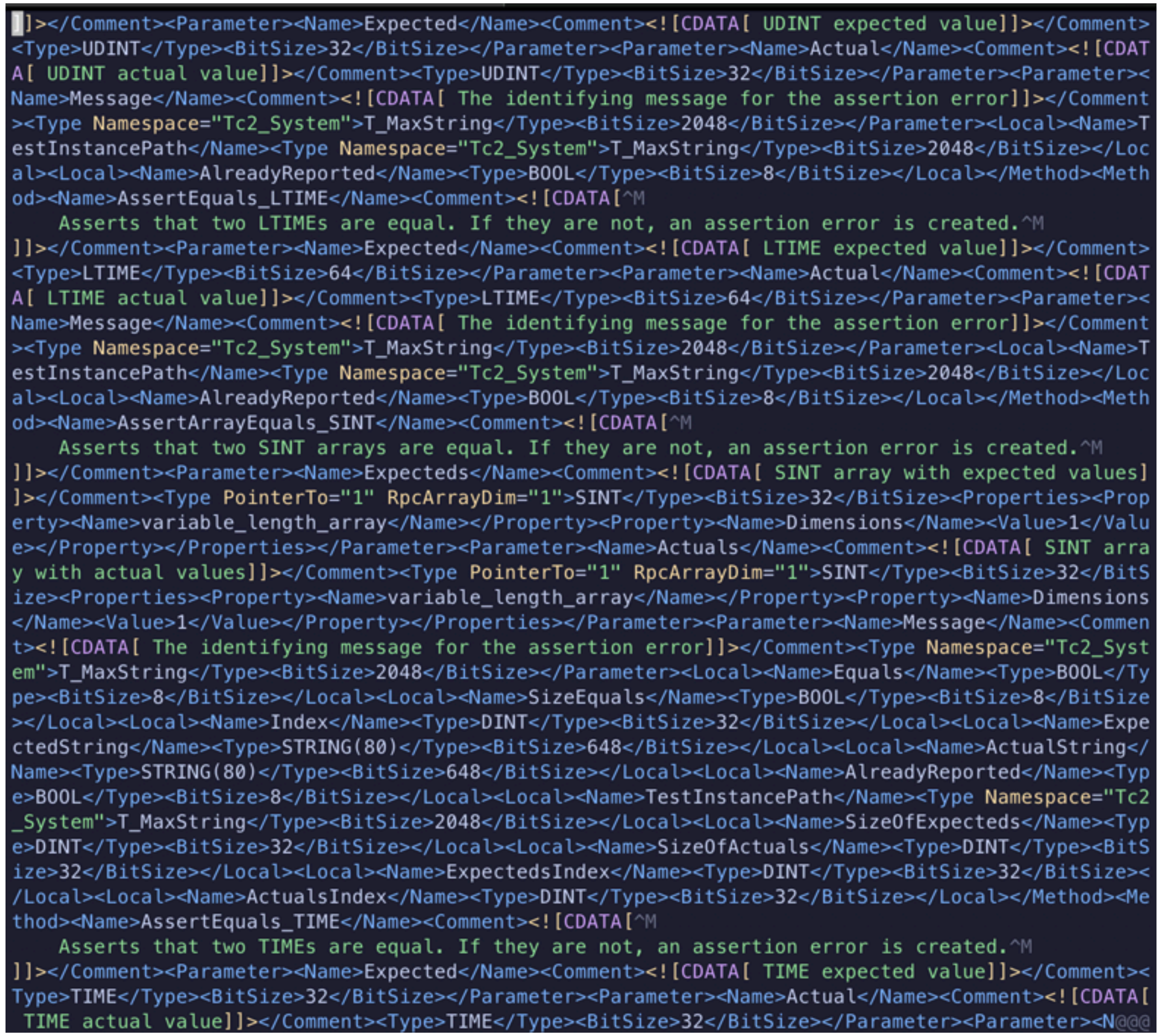
Device Generator

Compromise: Use PLC tmc file to generate high level



Device Generator

Compromise: Use PLC tmc file to generate high level

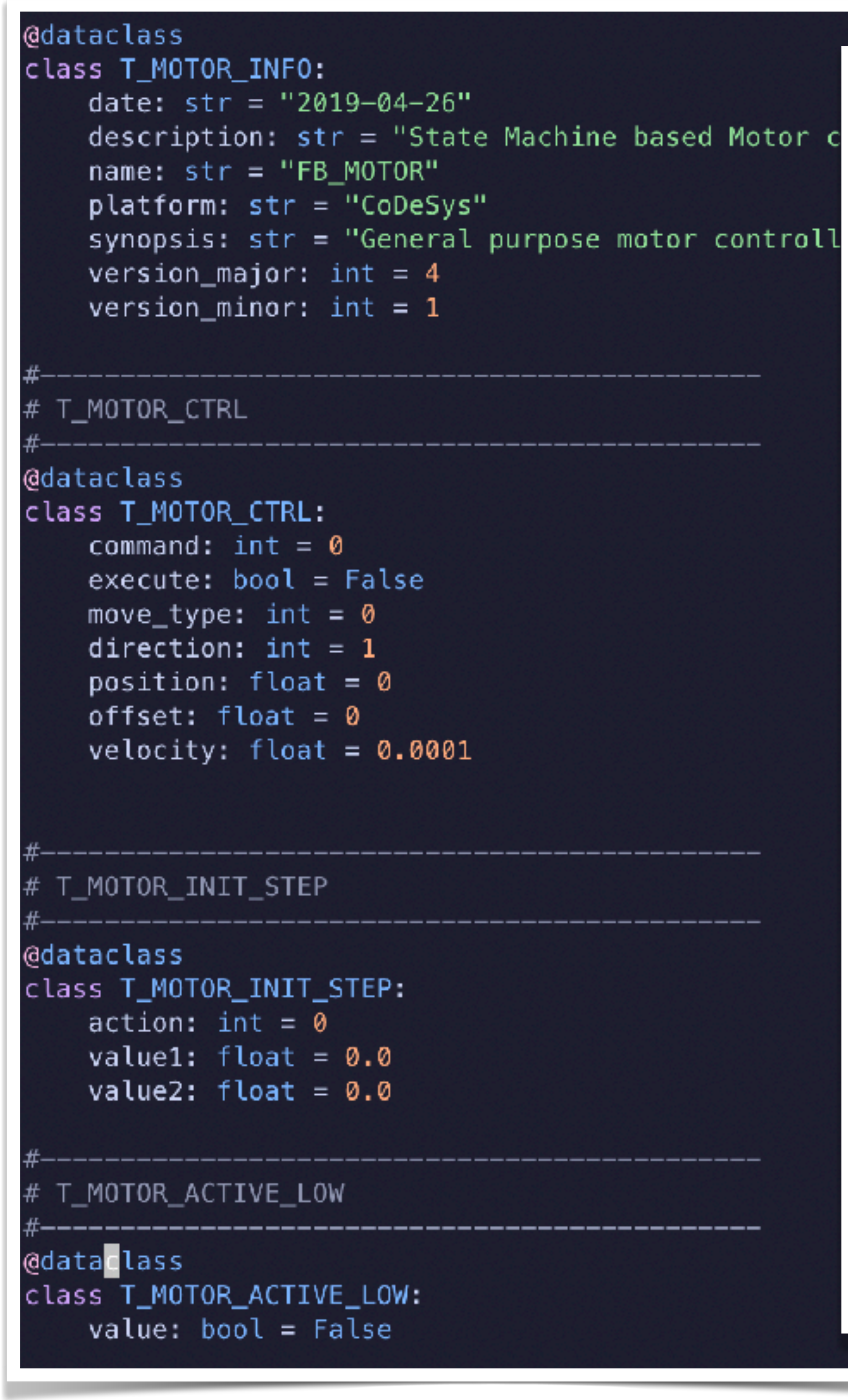


```
]]></Comment><Parameter><Name>Expected</Name><Comment><![CDATA[ UDINT expected value]]></Comment>
<Type>UDINT</Type><BitSize>32</BitSize></Parameter><Parameter><Name>Actual</Name><Comment><![CDAT
A[ UDINT actual value]]></Comment><Type>UDINT</Type><BitSize>32</BitSize></Parameter><Parameter><
Name>Message</Name><Comment><![CDATA[ The identifying message for the assertion error]]></Comment
><Type Namespace="Tc2_System">T_MaxString</Type><BitSize>2048</BitSize></Parameter><Local><Name>T
estInstancePath</Name><Type Namespace="Tc2_System">T_MaxString</Type><BitSize>2048</BitSize></Loc
al><Local><Name>AlreadyReported</Name><Type>BOOL</Type><BitSize>8</BitSize></Local></Method><Meth
od><Name>AssertEquals_LTIME</Name><Comment><![CDATA[^M
    Asserts that two LTIMEs are equal. If they are not, an assertion error is created.^M
]]></Comment><Parameter><Name>Expected</Name><Comment><![CDATA[ LTIME expected value]]></Comment>
<Type>LTIME</Type><BitSize>64</BitSize></Parameter><Parameter><Name>Actual</Name><Comment><![CDAT
A[ LTIME actual value]]></Comment><Type>LTIME</Type><BitSize>64</BitSize></Parameter><Parameter><
Name>Message</Name><Comment><![CDATA[ The identifying message for the assertion error]]></Comment
><Type Namespace="Tc2_System">T_MaxString</Type><BitSize>2048</BitSize></Parameter><Local><Name>T
estInstancePath</Name><Type Namespace="Tc2_System">T_MaxString</Type><BitSize>2048</BitSize></Loc
al><Local><Name>AlreadyReported</Name><Type>BOOL</Type><BitSize>8</BitSize></Local></Method><Meth
od><Name>AssertArrayEquals_SINT</Name><Comment><![CDATA[^M
    Asserts that two SINT arrays are equal. If they are not, an assertion error is created.^M
]]></Comment><Parameter><Name>Expecteds</Name><Comment><![CDATA[ SINT array with expected values]
]></Comment><Type PointerTo="1" RpcArrayDim="1">SINT</Type><BitSize>32</BitSize><Properties><Prop
erty><Name>variable_length_array</Name></Property><Property><Name>Dimensions</Name><Value>1</Valu
e></Property></Properties></Parameter><Parameter><Name>Actuals</Name><Comment><![CDATA[ SINT arra
y with actual values]]></Comment><Type PointerTo="1" RpcArrayDim="1">SINT</Type><BitSize>32</BitS
ize><Properties><Property><Name>variable_length_array</Name></Property><Property><Name>Dimensions
</Name><Value>1</Value></Property></Properties></Parameter><Parameter><Name>Message</Name><Commen
t><![CDATA[ The identifying message for the assertion error]]></Comment><Type Namespace="Tc2_Syst
em">T_MaxString</Type><BitSize>2048</BitSize></Parameter><Local><Name>Equals</Name><Type>BOOL</Ty
pe><BitSize>8</BitSize></Local><Local><Name>SizeEquals</Name><Type>BOOL</Type><BitSize>8</BitSize
></Local><Local><Name>Index</Name><Type>DINT</Type><BitSize>32</BitSize></Local><Local><Name>Expe
ctedString</Name><Type>STRING(80)</Type><BitSize>648</BitSize></Local><Local><Name>ActualString</
Name><Type>STRING(80)</Type><BitSize>648</BitSize></Local><Local><Name>AlreadyReported</Name><Typ
e>BOOL</Type><BitSize>8</BitSize></Local><Local><Name>TestInstancePath</Name><Type Namespace="Tc2
_System">T_MaxString</Type><BitSize>2048</BitSize></Local><Local><Name>SizeOfExpecteds</Name><Typ
e>DINT</Type><BitSize>32</BitSize></Local><Local><Name>SizeOfActuals</Name><Type>DINT</Type><BitS
ize>32</BitSize></Local><Local><Name>ExpectedsIndex</Name><Type>DINT</Type><BitSize>32</BitSize><
/Local><Local><Name>ActualsIndex</Name><Type>DINT</Type><BitSize>32</BitSize></Local></Method><Me
thod><Name>AssertEquals_TIME</Name><Comment><![CDATA[^M
    Asserts that two TIMEs are equal. If they are not, an assertion error is created.^M
]]></Comment><Parameter><Name>Expected</Name><Comment><![CDATA[ TIME expected value]]></Comment><
Type>TIME</Type><BitSize>32</BitSize></Parameter><Parameter><Name>Actual</Name><Comment><![CDATA[
 TIME actual value]]></Comment><Type>TIME</Type><BitSize>32</BitSize></Parameter><Parameter><N@@@
```

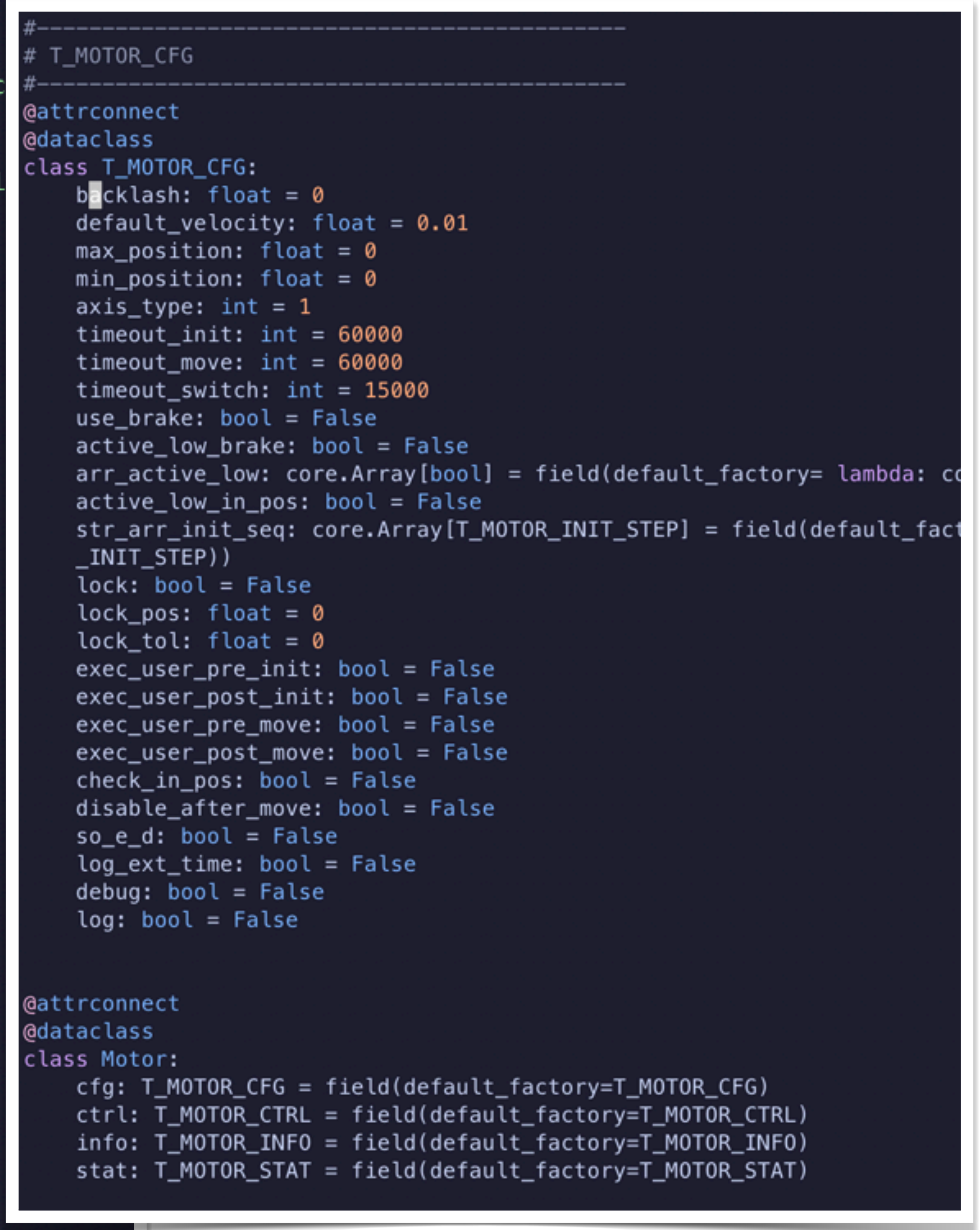
Tmc file

Device Generator

tmc file to generate high level



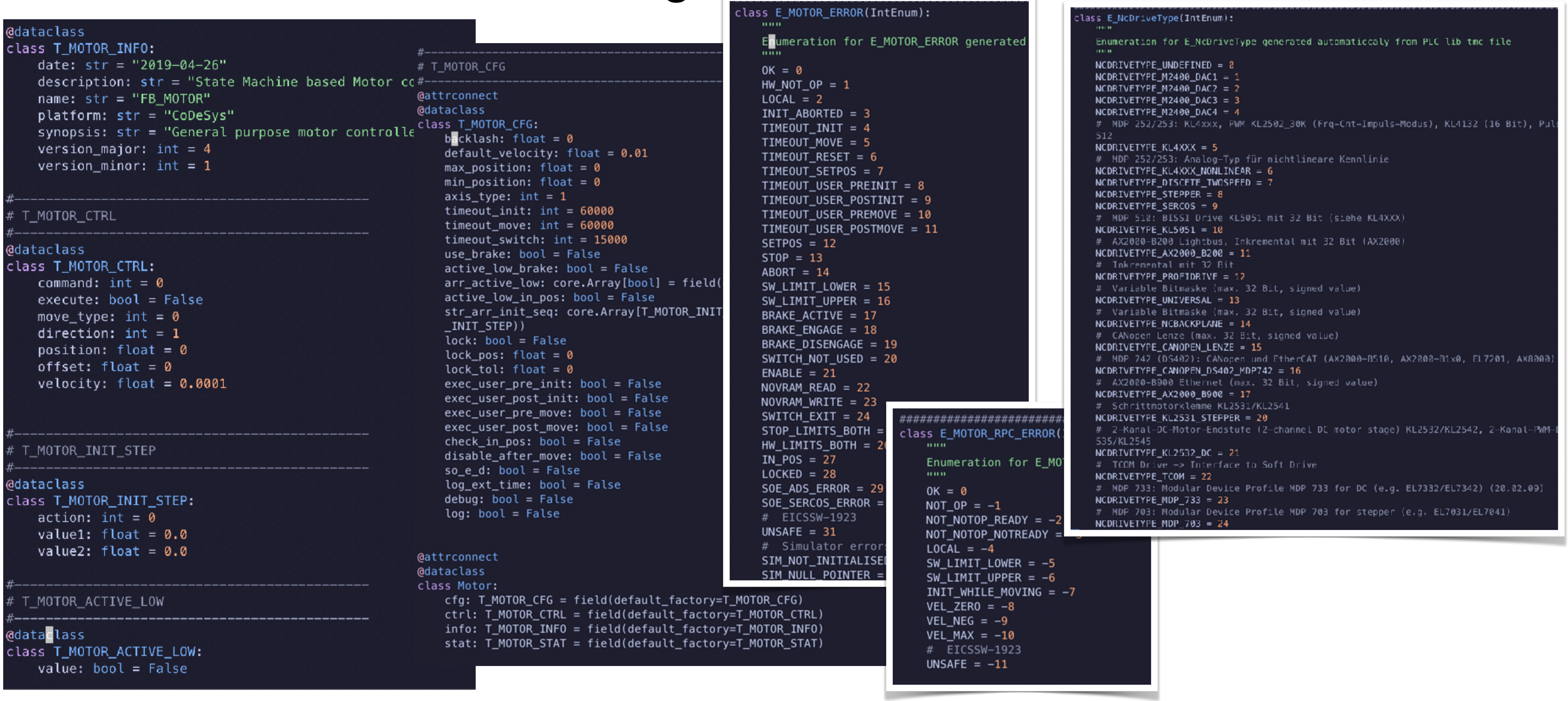
```
@dataclass
class T_MOTOR_INFO:
    date: str = "2019-04-26"
    description: str = "State Machine based Motor c
    name: str = "FB_MOTOR"
    platform: str = "CoDeSys"
    synopsis: str = "General purpose motor controll
    version_major: int = 4
    version_minor: int = 1

#-------------------------------------------
# T_MOTOR_CTRL
#-------------------------------------------
@dataclass
class T_MOTOR_CTRL:
    command: int = 0
    execute: bool = False
    move_type: int = 0
    direction: int = 1
    position: float = 0
    offset: float = 0
    velocity: float = 0.0001


#-------------------------------------------
# T_MOTOR_INIT_STEP
#-------------------------------------------
@dataclass
class T_MOTOR_INIT_STEP:
    action: int = 0
    value1: float = 0.0
    value2: float = 0.0

#-------------------------------------------
# T_MOTOR_ACTIVE_LOW
#-------------------------------------------
@dataclass
class T_MOTOR_ACTIVE_LOW:
    value: bool = False
```



```
#-------------------------------------------
# T_MOTOR_CFG
#-------------------------------------------
@attrconnect
@dataclass
class T_MOTOR_CFG:
    backlash: float = 0
    default_velocity: float = 0.01
    max_position: float = 0
    min_position: float = 0
    axis_type: int = 1
    timeout_init: int = 60000
    timeout_move: int = 60000
    timeout_switch: int = 15000
    use_brake: bool = False
    active_low_brake: bool = False
    arr_active_low: core.Array[bool] = field(default_factory= lambda: co
    active_low_in_pos: bool = False
    str_arr_init_seq: core.Array[T_MOTOR_INIT_STEP] = field(default_fact
    _INIT_STEP))
    lock: bool = False
    lock_pos: float = 0
    lock_tol: float = 0
    exec_user_pre_init: bool = False
    exec_user_post_init: bool = False
    exec_user_pre_move: bool = False
    exec_user_post_move: bool = False
    check_in_pos: bool = False
    disable_after_move: bool = False
    so_e_d: bool = False
    log_ext_time: bool = False
    debug: bool = False
    log: bool = False


@attrconnect
@dataclass
class Motor:
    cfg: T_MOTOR_CFG = field(default_factory=T_MOTOR_CFG)
    ctrl: T_MOTOR_CTRL = field(default_factory=T_MOTOR_CTRL)
    info: T_MOTOR_INFO = field(default_factory=T_MOTOR_INFO)
    stat: T_MOTOR_STAT = field(default_factory=T_MOTOR_STAT)
```

Device Generator

tmc file to generate high level



```
@dataclass
class T_MOTOR_INFO:
    date: str = "2019-04-26"
    description: str = "State Machine based Motor cc
    name: str = "FB_MOTOR"
    platform: str = "CoDeSys"
    synopsis: str = "General purpose motor controlle
    version_major: int = 4
    version_minor: int = 1

#------------------------------------------------------
# T_MOTOR_CTRL
#------------------------------------------------------
@dataclass
class T_MOTOR_CTRL:
    command: int = 0
    execute: bool = False
    move_type: int = 0
    direction: int = 1
    position: float = 0
    offset: float = 0
    velocity: float = 0.0001


#------------------------------------------------------
# T_MOTOR_INIT_STEP
#------------------------------------------------------
@dataclass
class T_MOTOR_INIT_STEP:
    action: int = 0
    value1: float = 0.0
    value2: float = 0.0

#------------------------------------------------------
# T_MOTOR_ACTIVE_LOW
#------------------------------------------------------
@dataclass
class T_MOTOR_ACTIVE_LOW:
    value: bool = False
```

```
#----------------------------------------------------
# T_MOTOR_CFG
#----------------------------------------------------
@attrconnect
@dataclass
class T_MOTOR_CFG:
    backlash: float = 0
    default_velocity: float = 0.01
    max_position: float = 0
    min_position: float = 0
    axis_type: int = 1
    timeout_init: int = 60000
    timeout_move: int = 60000
    timeout_switch: int = 15000
    use_brake: bool = False
    active_low_brake: bool = False
    arr_active_low: core.Array[bool] = field(
    active_low_in_pos: bool = False
    str_arr_init_seq: core.Array[T_MOTOR_INIT
    _INIT_STEP))
    lock: bool = False
    lock_pos: float = 0
    lock_tol: float = 0
    exec_user_pre_init: bool = False
    exec_user_post_init: bool = False
    exec_user_pre_move: bool = False
    exec_user_post_move: bool = False
    check_in_pos: bool = False
    disable_after_move: bool = False
    so_e_d: bool = False
    log_ext_time: bool = False
    debug: bool = False
    log: bool = False


@attrconnect
@dataclass
class Motor:
    cfg: T_MOTOR_CFG = field(default_factory=T_MOTOR_CFG)
    ctrl: T_MOTOR_CTRL = field(default_factory=T_MOTOR_CTRL)
    info: T_MOTOR_INFO = field(default_factory=T_MOTOR_INFO)
    stat: T_MOTOR_STAT = field(default_factory=T_MOTOR_STAT)
```

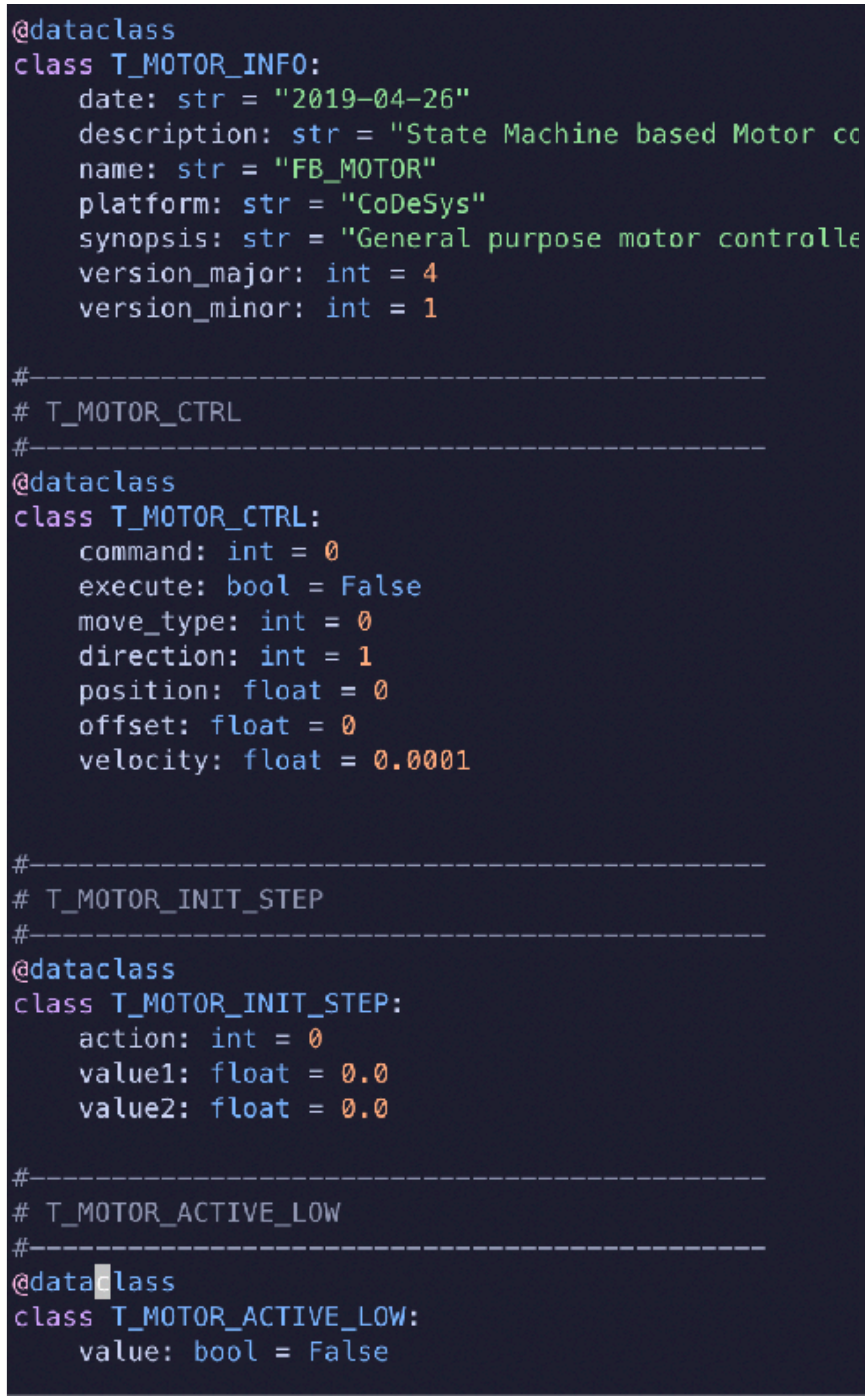
```
class E_MOTOR_ERROR(IntEnum):
    """
    Enumeration for E_MOTOR_ERROR generated
    """
    OK = 0
    HW_NOT_OP = 1
    LOCAL = 2
    INIT_ABORTED = 3
    TIMEOUT_INIT = 4
    TIMEOUT_MOVE = 5
    TIMEOUT_RESET = 6
    TIMEOUT_SETPOS = 7
    TIMEOUT_USER_PREINIT = 8
    TIMEOUT_USER_POSTINIT = 9
    TIMEOUT_USER_PREMOVE = 10
    TIMEOUT_USER_POSTMOVE = 11
    SETPOS = 12
    STOP = 13
    ABORT = 14
    SW_LIMIT_LOWER = 15
    SW_LIMIT_UPPER = 16
    BRAKE_ACTIVE = 17
    BRAKE_ENGAGE = 18
    BRAKE_DISENGAGE = 19
    SWITCH_NOT_USED = 20
    ENABLE = 21
    NOVRAM_READ = 22
    NOVRAM_WRITE = 23
    SWITCH_EXIT = 24
    STOP_LIMITS_BOTH =
    HW_LIMITS_BOTH = 2
    IN_POS = 27
    LOCKED = 28
    SOE_ADS_ERROR = 29
    SOE_SERCOS_ERROR =
    #  EICSSW-1923
    UNSAFE = 31
    #  Simulator errors
    SIM_NOT_INITIALISED
    SIM_NULL_POINTER =
```

```
##########################
class E_MOTOR_RPC_ERROR(
    """
    Enumeration for E_MOT
    """
    OK = 0
    NOT_OP = -1
    NOT_NOTOP_READY = -2
    NOT_NOTOP_NOTREADY =
    LOCAL = -4
    SW_LIMIT_LOWER = -5
    SW_LIMIT_UPPER = -6
    INIT_WHILE_MOVING = -7
    VEL_ZERO = -8
    VEL_NEG = -9
    VEL_MAX = -10
    #  EICSSW-1923
    UNSAFE = -11
```

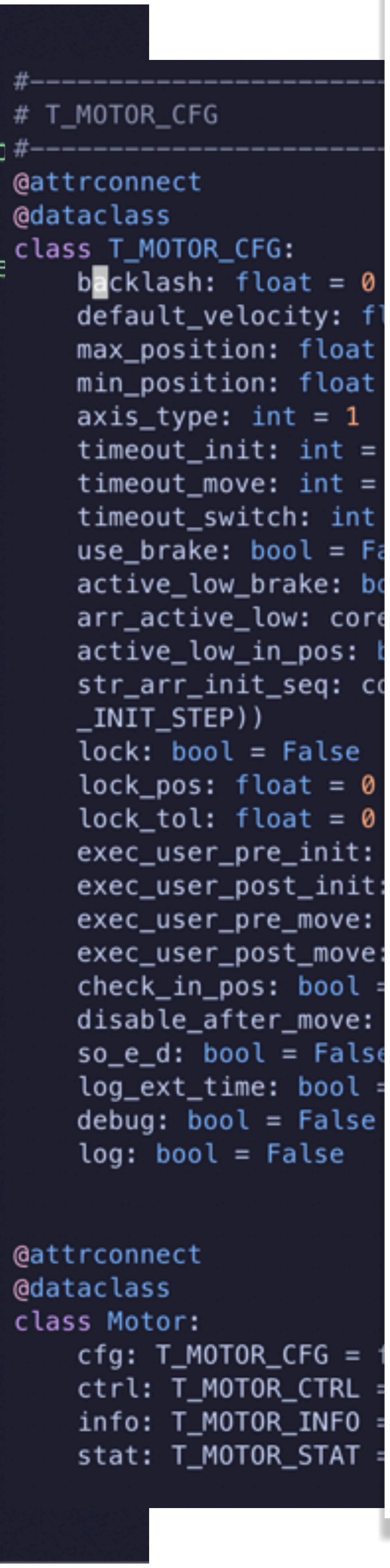
```
class E_NcDriveType(IntEnum):
    """
    Enumeration for E_NcDriveType generated automaticcaly from PLC lib tmc file
    """
    NCDRIVETYPE_UNDEFINED = 0
    NCDRIVETYPE_M2400_DAC1 = 1
    NCDRIVETYPE_M2400_DAC2 = 2
    NCDRIVETYPE_M2400_DAC3 = 3
    NCDRIVETYPE_M2400_DAC4 = 4
    #  MDP 252/253: KL4xxx, PWM KL2502_30K (Frq-Cnt-Impuls-Modus), KL4132 (16 Bit), Pul
    512
    NCDRIVETYPE_KL4XXX = 5
    #  MDP 252/253: Analog-Typ für nichtlineare Kennlinie
    NCDRIVETYPE_KL4XXX_NONLINEAR = 6
    NCDRIVETYPE_DISCETE_TWOSPEED = 7
    NCDRIVETYPE_STEPPER = 8
    NCDRIVETYPE_SERCOS = 9
    #  MDP 510: BISSI Drive KL5051 mit 32 Bit (siehe KL4XXX)
    NCDRIVETYPE_KL5051 = 10
    #  AX2000-B200 Lightbus, Inkremental mit 32 Bit (AX2000)
    NCDRIVETYPE_AX2000_B200 = 11
    #  Inkremental mit 32 Bit
    NCDRIVETYPE_PROFIDRIVE = 12
    #  Variable Bitmaske (max. 32 Bit, signed value)
    NCDRIVETYPE_UNIVERSAL = 13
    #  Variable Bitmaske (max. 32 Bit, signed value)
    NCDRIVETYPE_NCBACKPLANE = 14
    #  CANopen Lenze (max. 32 Bit, signed value)
    NCDRIVETYPE_CANOPEN_LENZE = 15
    #  MDP 742 (DS402): CANopen und EtherCAT (AX2000-B510, AX2000-B1x0, EL7201, AX8000)
    NCDRIVETYPE_CANOPEN_DS402_MDP742 = 16
    #  AX2000-B900 Ethernet (max. 32 Bit, signed value)
    NCDRIVETYPE_AX2000_B900 = 17
    #  Schrittmotorklemme KL2531/KL2541
    NCDRIVETYPE_KL2531_STEPPER = 20
    #  2-Kanal-DC-Motor-Endstufe (2-channel DC motor stage) KL2532/KL2542, 2-Kanal-PWM-
    535/KL2545
    NCDRIVETYPE_KL2532_DC = 21
    #  TCOM Drive -> Interface to Soft Drive
    NCDRIVETYPE_TCOM = 22
    #  MDP 733: Modular Device Profile MDP 733 for DC (e.g. EL7332/EL7342) (20.02.09)
    NCDRIVETYPE_MDP_733 = 23
    #  MDP 703: Modular Device Profile MDP 703 for stepper (e.g. EL7031/EL7041)
    NCDRIVETYPE_MDP_703 = 24
```

Device Generator

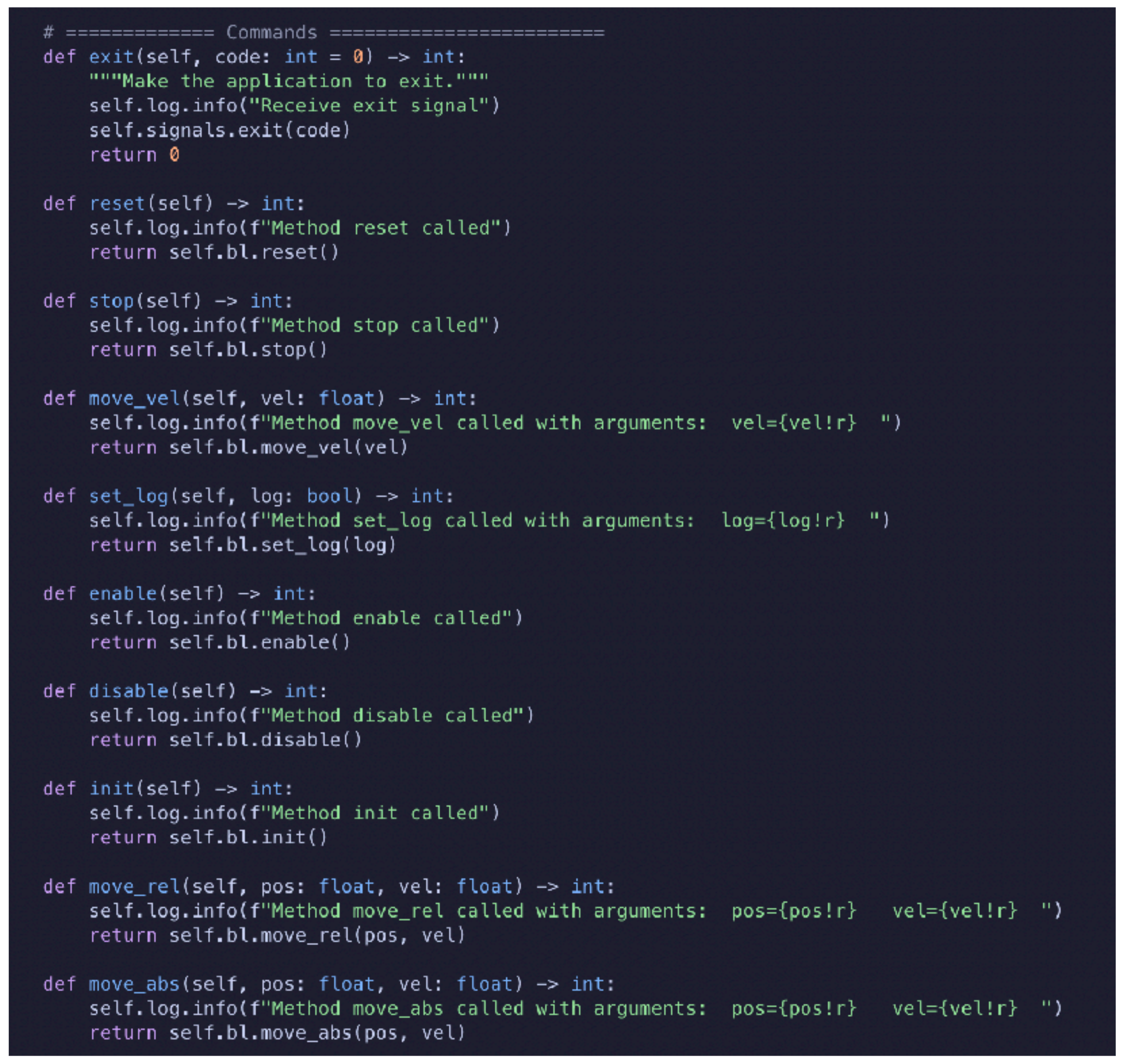
tmc file to generate high level



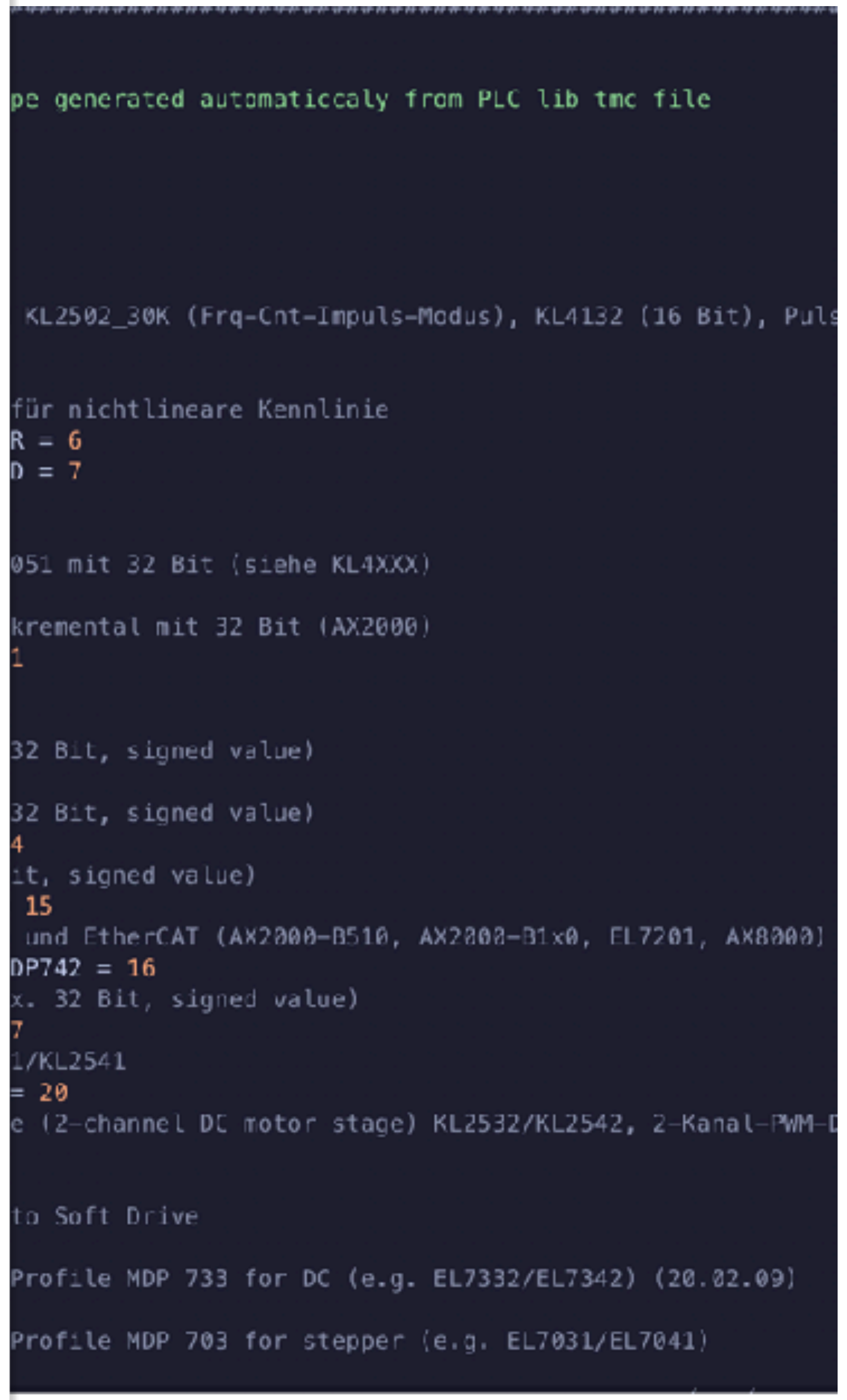
```
@dataclass
class T_MOTOR_INFO:
    date: str = "2019-04-26"
    description: str = "State Machine based Motor co
    name: str = "FB_MOTOR"
    platform: str = "CoDeSys"
    synopsis: str = "General purpose motor controlle
    version_major: int = 4
    version_minor: int = 1

#----------------------------------------------
# T_MOTOR_CTRL
#----------------------------------------------
@dataclass
class T_MOTOR_CTRL:
    command: int = 0
    execute: bool = False
    move_type: int = 0
    direction: int = 1
    position: float = 0
    offset: float = 0
    velocity: float = 0.0001


#----------------------------------------------
# T_MOTOR_INIT_STEP
#----------------------------------------------
@dataclass
class T_MOTOR_INIT_STEP:
    action: int = 0
    value1: float = 0.0
    value2: float = 0.0

#----------------------------------------------
# T_MOTOR_ACTIVE_LOW
#----------------------------------------------
@dataclass
class T_MOTOR_ACTIVE_LOW:
    value: bool = False
```



```
#-------------------
# T_MOTOR_CFG
#-------------------
@attrconnect
@dataclass
class T_MOTOR_CFG:
    backlash: float = 0
    default_velocity: fl
    max_position: float
    min_position: float
    axis_type: int = 1
    timeout_init: int =
    timeout_move: int =
    timeout_switch: int
    use_brake: bool = Fa
    active_low_brake: bo
    arr_active_low: core
    active_low_in_pos: b
    str_arr_init_seq: co
    _INIT_STEP))
    lock: bool = False
    lock_pos: float = 0
    lock_tol: float = 0
    exec_user_pre_init:
    exec_user_post_init:
    exec_user_pre_move:
    exec_user_post_move:
    check_in_pos: bool =
    disable_after_move:
    so_e_d: bool = False
    log_ext_time: bool =
    debug: bool = False
    log: bool = False


@attrconnect
@dataclass
class Motor:
    cfg: T_MOTOR_CFG = f
    ctrl: T_MOTOR_CTRL =
    info: T_MOTOR_INFO =
    stat: T_MOTOR_STAT =
```



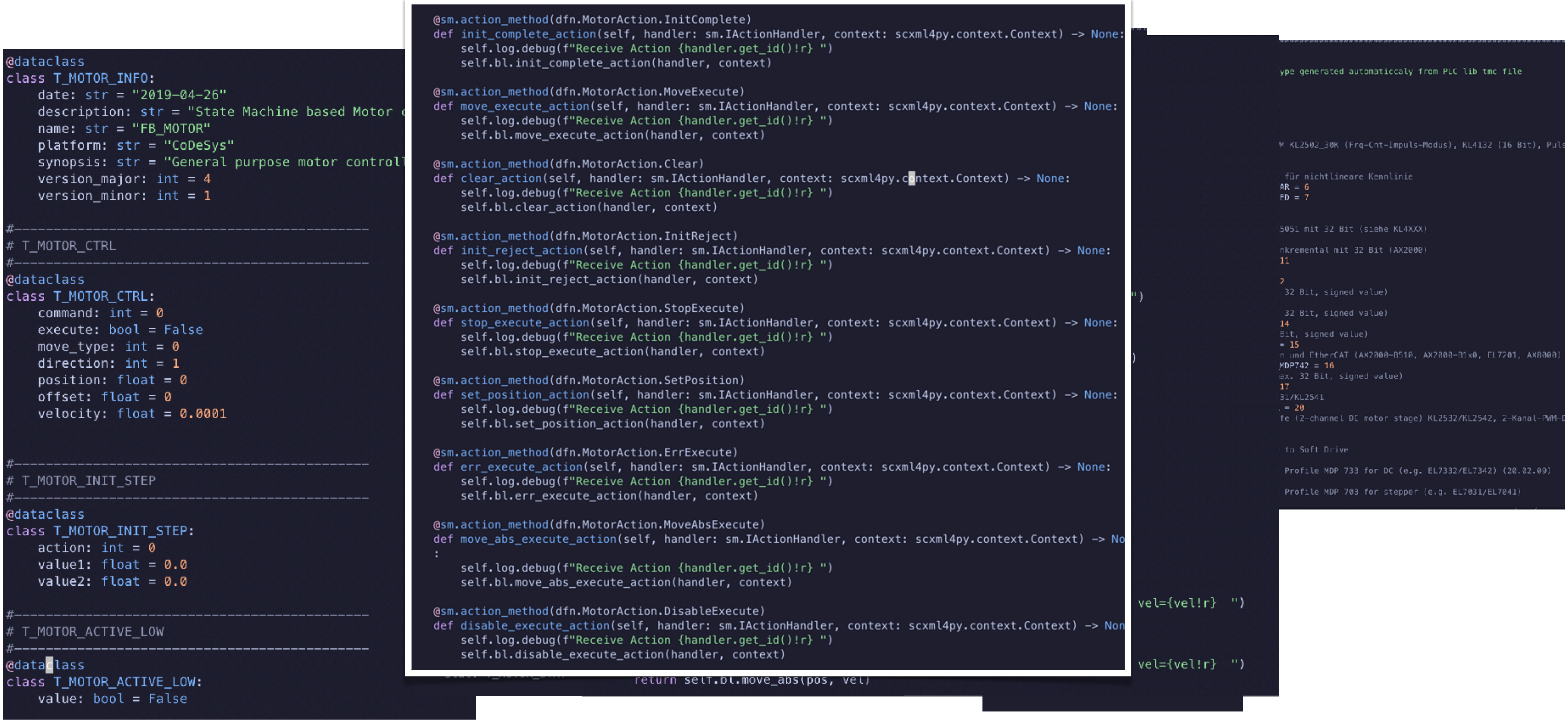
```
# ============= Commands ========================
def exit(self, code: int = 0) -> int:
    """Make the application to exit."""
    self.log.info("Receive exit signal")
    self.signals.exit(code)
    return 0

def reset(self) -> int:
    self.log.info(f"Method reset called")
    return self.bl.reset()

def stop(self) -> int:
    self.log.info(f"Method stop called")
    return self.bl.stop()

def move_vel(self, vel: float) -> int:
    self.log.info(f"Method move_vel called with arguments:  vel={vel!r}  ")
    return self.bl.move_vel(vel)

def set_log(self, log: bool) -> int:
    self.log.info(f"Method set_log called with arguments:  log={log!r}  ")
    return self.bl.set_log(log)

def enable(self) -> int:
    self.log.info(f"Method enable called")
    return self.bl.enable()

def disable(self) -> int:
    self.log.info(f"Method disable called")
    return self.bl.disable()

def init(self) -> int:
    self.log.info(f"Method init called")
    return self.bl.init()

def move_rel(self, pos: float, vel: float) -> int:
    self.log.info(f"Method move_rel called with arguments:  pos={pos!r}   vel={vel!r}  ")
    return self.bl.move_rel(pos, vel)

def move_abs(self, pos: float, vel: float) -> int:
    self.log.info(f"Method move_abs called with arguments:  pos={pos!r}   vel={vel!r}  ")
    return self.bl.move_abs(pos, vel)
```



```
#################################################

pe generated automaticcaly from PLC lib tmc file





 KL2502_30K (Frq-Cnt-Impuls-Modus), KL4132 (16 Bit), Puls

für nichtlineare Kennlinie
R = 6
D = 7

051 mit 32 Bit (siehe KL4XXX)

krementaL mit 32 Bit (AX2000)
1

32 Bit, signed value)

32 Bit, signed value)
4
it, signed value)
 15
 und EtherCAT (AX2000-B510, AX2000-B1x0, EL7201, AX8000)
DP742 = 16
x. 32 Bit, signed value)
7
1/KL2541
= 20
e (2-channel DC motor stage) KL2532/KL2542, 2-Kanal-PWM-D

to Soft Drive

Profile MDP 733 for DC (e.g. EL7332/EL7342) (20.02.09)

Profile MDP 703 for stepper (e.g. EL7031/EL7041)
```

Device Generator

tmc file to generate high level



```
@dataclass
class T_MOTOR_INFO:
    date: str = "2019-04-26"
    description: str = "State Machine based Motor c
    name: str = "FB_MOTOR"
    platform: str = "CoDeSys"
    synopsis: str = "General purpose motor controll
    version_major: int = 4
    version_minor: int = 1

#----------------------------------------------
# T_MOTOR_CTRL
#----------------------------------------------
@dataclass
class T_MOTOR_CTRL:
    command: int = 0
    execute: bool = False
    move_type: int = 0
    direction: int = 1
    position: float = 0
    offset: float = 0
    velocity: float = 0.0001


#----------------------------------------------
# T_MOTOR_INIT_STEP
#----------------------------------------------
@dataclass
class T_MOTOR_INIT_STEP:
    action: int = 0
    value1: float = 0.0
    value2: float = 0.0

#----------------------------------------------
# T_MOTOR_ACTIVE_LOW
#----------------------------------------------
@dataclass
class T_MOTOR_ACTIVE_LOW:
    value: bool = False
```

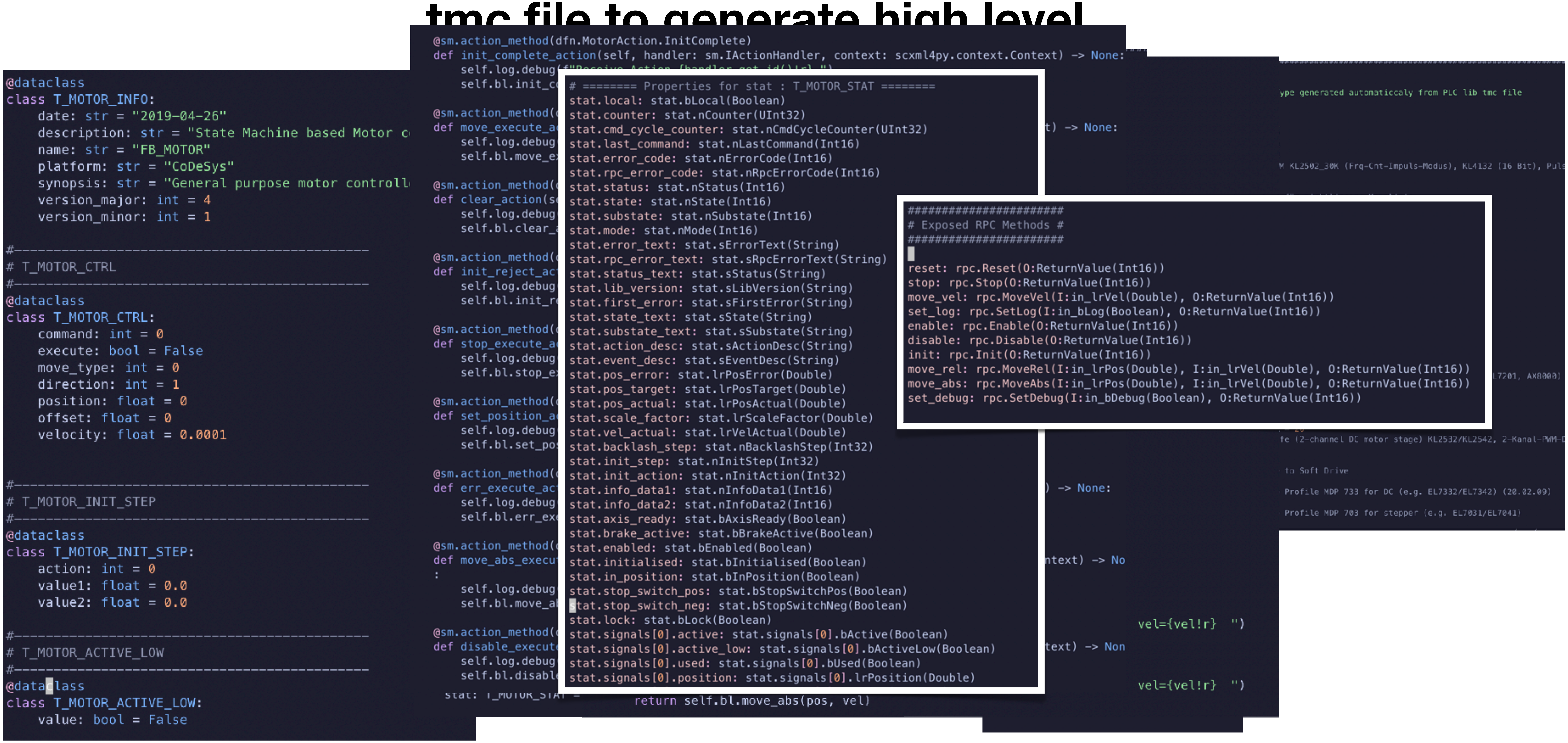
```
@sm.action_method(dfn.MotorAction.InitComplete)
def init_complete_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.init_complete_action(handler, context)

@sm.action_method(dfn.MotorAction.MoveExecute)
def move_execute_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.move_execute_action(handler, context)

@sm.action_method(dfn.MotorAction.Clear)
def clear_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.clear_action(handler, context)

@sm.action_method(dfn.MotorAction.InitReject)
def init_reject_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.init_reject_action(handler, context)

@sm.action_method(dfn.MotorAction.StopExecute)
def stop_execute_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.stop_execute_action(handler, context)

@sm.action_method(dfn.MotorAction.SetPosition)
def set_position_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.set_position_action(handler, context)

@sm.action_method(dfn.MotorAction.ErrExecute)
def err_execute_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.err_execute_action(handler, context)

@sm.action_method(dfn.MotorAction.MoveAbsExecute)
def move_abs_execute_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> No
:
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.move_abs_execute_action(handler, context)

@sm.action_method(dfn.MotorAction.DisableExecute)
def disable_execute_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> Non
    self.log.debug(f"Receive Action {handler.get_id()!r} ")
    self.bl.disable_execute_action(handler, context)
```

```
")
)
vel={vel!r}  ")
vel={vel!r}  ")
return self.bl.move_abs(pos, vel)
```

```
ype generated automaticcaly from PLC lib tmc file

M KL2502_30K (Frq-Cnt-Impuls-Modus), KL4132 (16 Bit), Puls

 für nichtlineare Kennlinie
AR = 6
FD = 7

5051 mit 32 Bit (siehe KL4XXX)

nkremental mit 32 Bit (AX2000)
11

2
 32 Bit, signed value)

 32 Bit, signed value)
14
Bit, signed value)
= 15
n und EtherCAT (AX2000-B510, AX2000-B1x0, EL7201, AX8000)
MDP742 = 16
ax. 32 Bit, signed value)
17
31/KL2541
 = 20
fe (2-channel DC motor stage) KL2532/KL2542, 2-Kanal-PWM-D

 to Soft Drive

 Profile MDP 733 for DC (e.g. EL7332/EL7342) (20.02.09)

 Profile MDP 703 for stepper (e.g. EL7031/EL7041)
```

Device Generator

tmc file to generate high level



```
@dataclass
class T_MOTOR_INFO:
    date: str = "2019-04-26"
    description: str = "State Machine based Motor c
    name: str = "FB_MOTOR"
    platform: str = "CoDeSys"
    synopsis: str = "General purpose motor controll
    version_major: int = 4
    version_minor: int = 1

#-------------------------------------------------
# T_MOTOR_CTRL
#-------------------------------------------------
@dataclass
class T_MOTOR_CTRL:
    command: int = 0
    execute: bool = False
    move_type: int = 0
    direction: int = 1
    position: float = 0
    offset: float = 0
    velocity: float = 0.0001


#-------------------------------------------------
# T_MOTOR_INIT_STEP
#-------------------------------------------------
@dataclass
class T_MOTOR_INIT_STEP:
    action: int = 0
    value1: float = 0.0
    value2: float = 0.0

#-------------------------------------------------
# T_MOTOR_ACTIVE_LOW
#-------------------------------------------------
@dataclass
class T_MOTOR_ACTIVE_LOW:
    value: bool = False
```

```
@sm.action_method(dfn.MotorAction.InitComplete)
def init_complete_action(self, handler: sm.IActionHandler, context: scxml4py.context.Context) -> None:
    self.log.debug(
    self.bl.init_c

@sm.action_method(                                                                   t) -> None:
def move_execute_a
    self.log.debug
    self.bl.move_e

@sm.action_method(
def clear_action(s
    self.log.debug
    self.bl.clear_

@sm.action_method(
def init_reject_ac
    self.log.debug
    self.bl.init_r

@sm.action_method(
def stop_execute_a
    self.log.debug
    self.bl.stop_e

@sm.action_method(
def set_position_a
    self.log.debug
    self.bl.set_po

@sm.action_method(                                                                   ) -> None:
def err_execute_ac
    self.log.debug
    self.bl.err_ex

@sm.action_method(
def move_abs_execu                                                                   ntext) -> No
:
    self.log.debug
    self.bl.move_a
                                                                                     vel={vel!r}  ")
@sm.action_method(
def disable_execut                                                                   text) -> Non
    self.log.debug
    self.bl.disabl
                                                                                     vel={vel!r}  ")
  stat: T_MOTOR_STAT =
                 return self.bl.move_abs(pos, vel)
```

```
# ======== Properties for stat : T_MOTOR_STAT ========
stat.local: stat.bLocal(Boolean)
stat.counter: stat.nCounter(UInt32)
stat.cmd_cycle_counter: stat.nCmdCycleCounter(UInt32)
stat.last_command: stat.nLastCommand(Int16)
stat.error_code: stat.nErrorCode(Int16)
stat.rpc_error_code: stat.nRpcErrorCode(Int16)
stat.status: stat.nStatus(Int16)
stat.state: stat.nState(Int16)
stat.substate: stat.nSubstate(Int16)
stat.mode: stat.nMode(Int16)
stat.error_text: stat.sErrorText(String)
stat.rpc_error_text: stat.sRpcErrorText(String)
stat.status_text: stat.sStatus(String)
stat.lib_version: stat.sLibVersion(String)
stat.first_error: stat.sFirstError(String)
stat.state_text: stat.sState(String)
stat.substate_text: stat.sSubstate(String)
stat.action_desc: stat.sActionDesc(String)
stat.event_desc: stat.sEventDesc(String)
stat.pos_error: stat.lrPosError(Double)
stat.pos_target: stat.lrPosTarget(Double)
stat.pos_actual: stat.lrPosActual(Double)
stat.scale_factor: stat.lrScaleFactor(Double)
stat.vel_actual: stat.lrVelActual(Double)
stat.backlash_step: stat.nBacklashStep(Int32)
stat.init_step: stat.nInitStep(Int32)
stat.init_action: stat.nInitAction(Int32)
stat.info_data1: stat.nInfoData1(Int16)
stat.info_data2: stat.nInfoData2(Int16)
stat.axis_ready: stat.bAxisReady(Boolean)
stat.brake_active: stat.bBrakeActive(Boolean)
stat.enabled: stat.bEnabled(Boolean)
stat.initialised: stat.bInitialised(Boolean)
stat.in_position: stat.bInPosition(Boolean)
stat.stop_switch_pos: stat.bStopSwitchPos(Boolean)
stat.stop_switch_neg: stat.bStopSwitchNeg(Boolean)
stat.lock: stat.bLock(Boolean)
stat.signals[0].active: stat.signals[0].bActive(Boolean)
stat.signals[0].active_low: stat.signals[0].bActiveLow(Boolean)
stat.signals[0].used: stat.signals[0].bUsed(Boolean)
stat.signals[0].position: stat.signals[0].lrPosition(Double)
```

```
######################
# Exposed RPC Methods #
######################

reset: rpc.Reset(O:ReturnValue(Int16))
stop: rpc.Stop(O:ReturnValue(Int16))
move_vel: rpc.MoveVel(I:in_lrVel(Double), O:ReturnValue(Int16))
set_log: rpc.SetLog(I:in_bLog(Boolean), O:ReturnValue(Int16))
enable: rpc.Enable(O:ReturnValue(Int16))
disable: rpc.Disable(O:ReturnValue(Int16))
init: rpc.Init(O:ReturnValue(Int16))
move_rel: rpc.MoveRel(I:in_lrPos(Double), I:in_lrVel(Double), O:ReturnValue(Int16))
move_abs: rpc.MoveAbs(I:in_lrPos(Double), I:in_lrVel(Double), O:ReturnValue(Int16))
set_debug: rpc.SetDebug(I:in_bDebug(Boolean), O:ReturnValue(Int16))
```

```
########################################################################
ype generated automaticcaly from PLC lib tmc file

M KL2502_30K (Frq-Cnt-Impuls-Modus), KL4132 (16 Bit), Puls

L7201, AX8000]

fe (2-channel DC motor stage) KL2532/KL2542, 2-Kanal-PWM-D

 to Soft Drive

 Profile MDP 733 for DC (e.g. EL7332/EL7342) (20.02.09)

 Profile MDP 703 for stepper (e.g. EL7031/EL7041)
```

Device Generator

Tmc2fcs features

- Export all Structures, Enumerator, tables, FB exposed on OPC-UA. With name, type, default value and comment !
- With State machine File (scxml). All state machine function are exported
- Template base (scriban) with a lot of functionalities.
- Possibility to preserve a file for deletion : {{PRESERVE}}my_file_of{{DeviceName}}.py

Meaning you can re-generate device if business logic is well preserved from OPC-UA “interface” business.

Status (still beta)

- Generator executable 95% done
- Templates for (new) Devsim 90% done
- Working on:
    - Devmgr classes
    - GUI (Draft of UI)
    - Cfg file schema (draft)
    - Test software
    - RPM packaging
    - Complete documentation