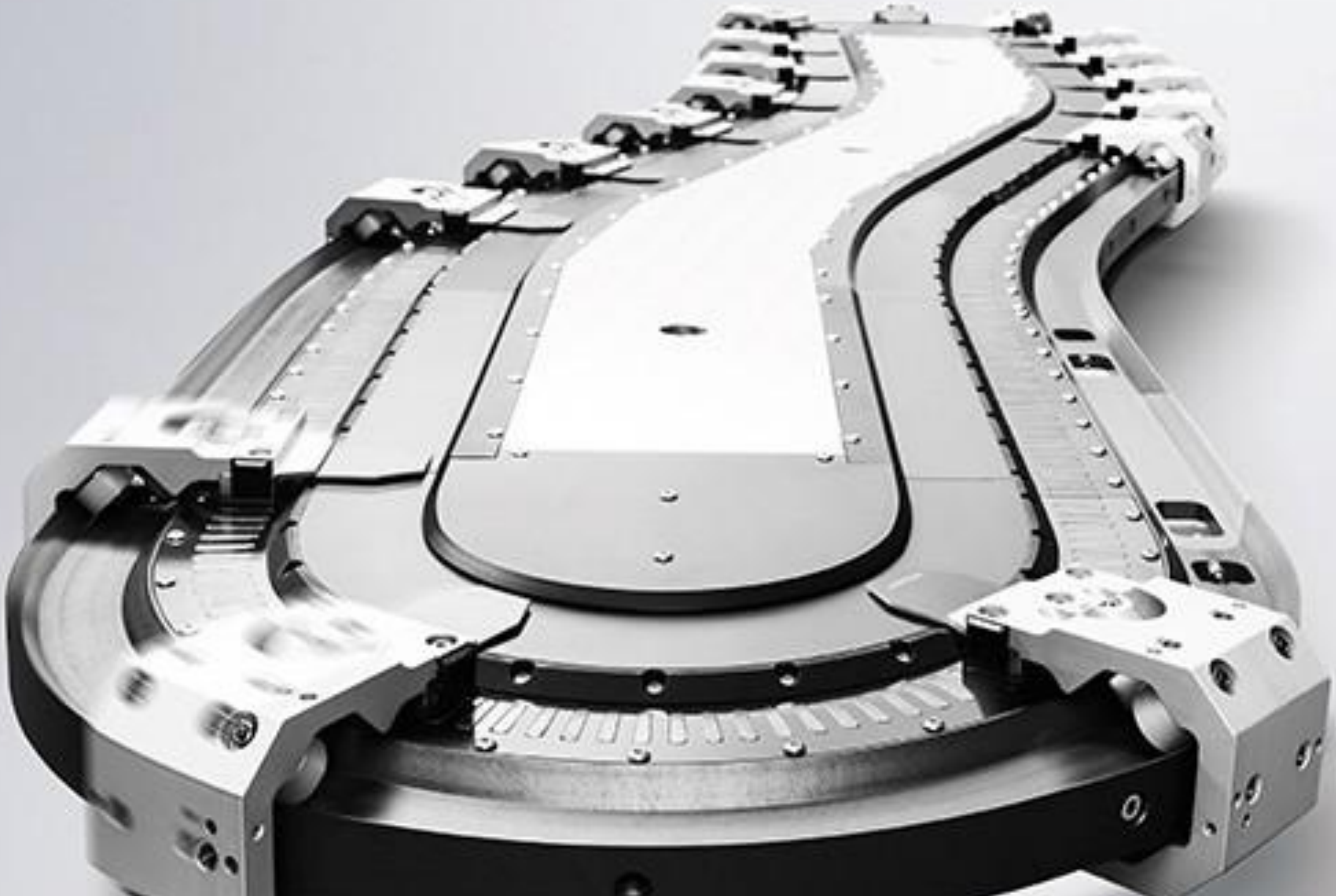
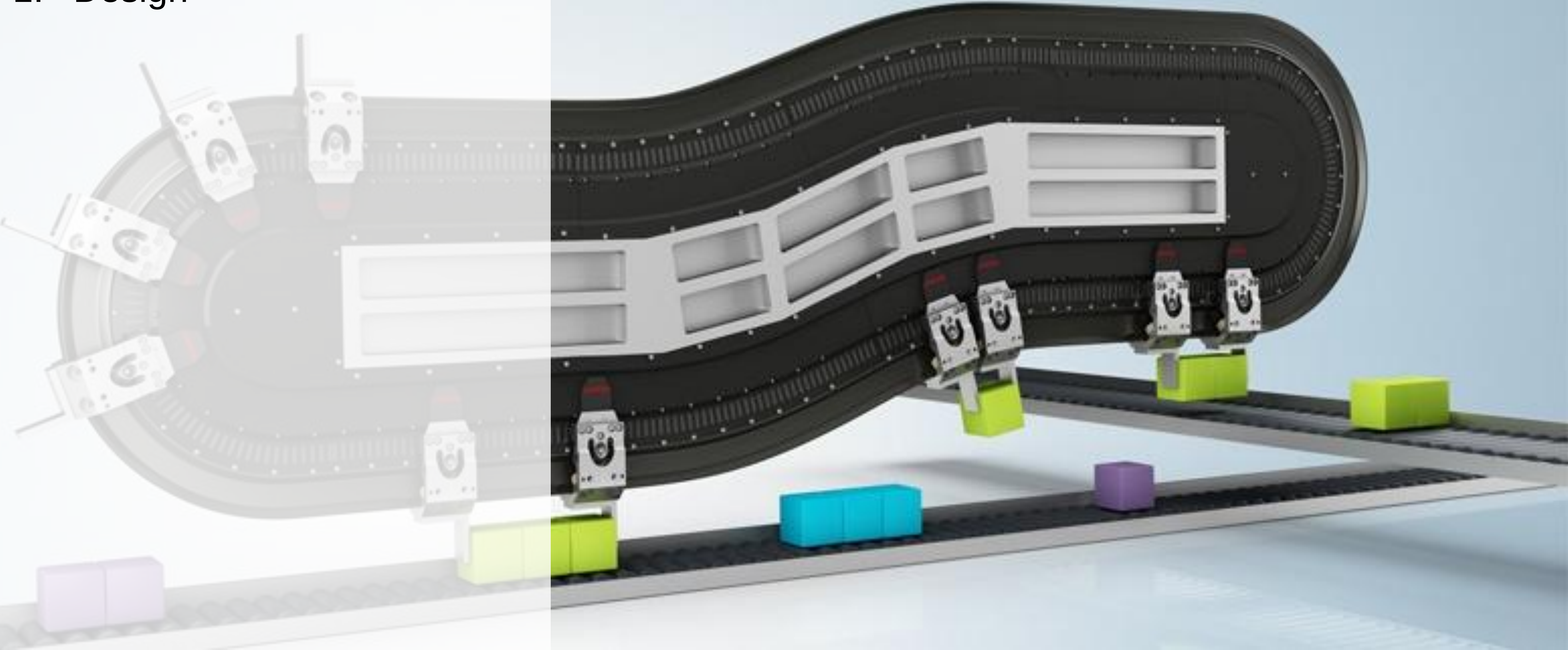
XTS TRANSPORT LAYER – Mover Class



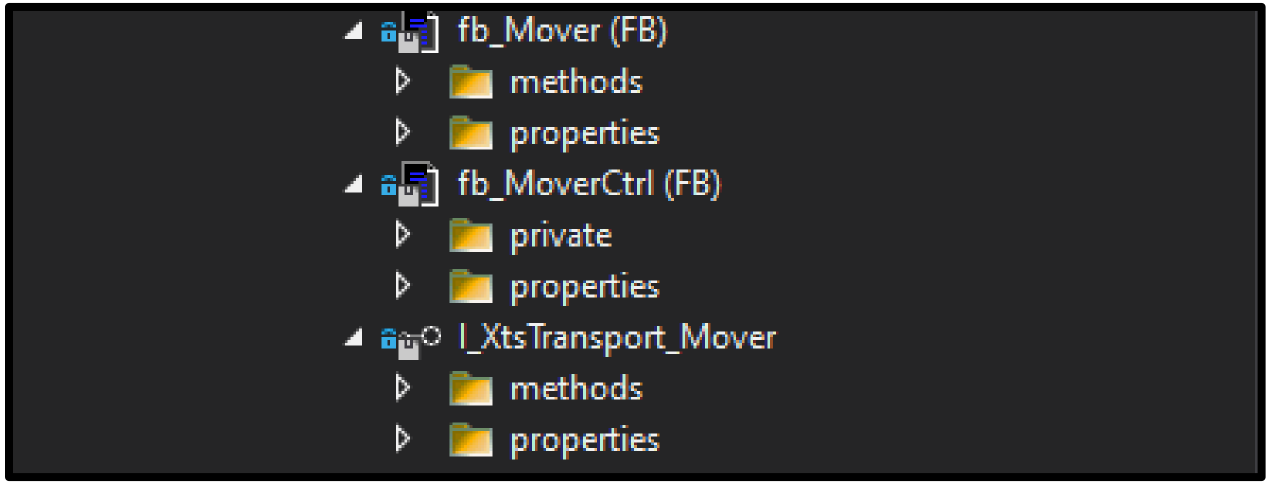
XTS TRANSPORT LAYER – Mover Class

1. Design



Design

- **GVL_XTS.Mover (fb_MoverCtrl)**
  - fb_Mover
    - Encapsulates Motion Control and Collision Avoidance function blocks
    - Message handling
    - Base class of fb_MoverCtrl
  - fb_MoverCtrl
    - Cyclic execution wrapper
    - OnChange check of commands
    - State feedback
  - I_XtsTransport_Mover
    - Interface for use in fb_CaGroup, fb_Station, fb_TransportUnit

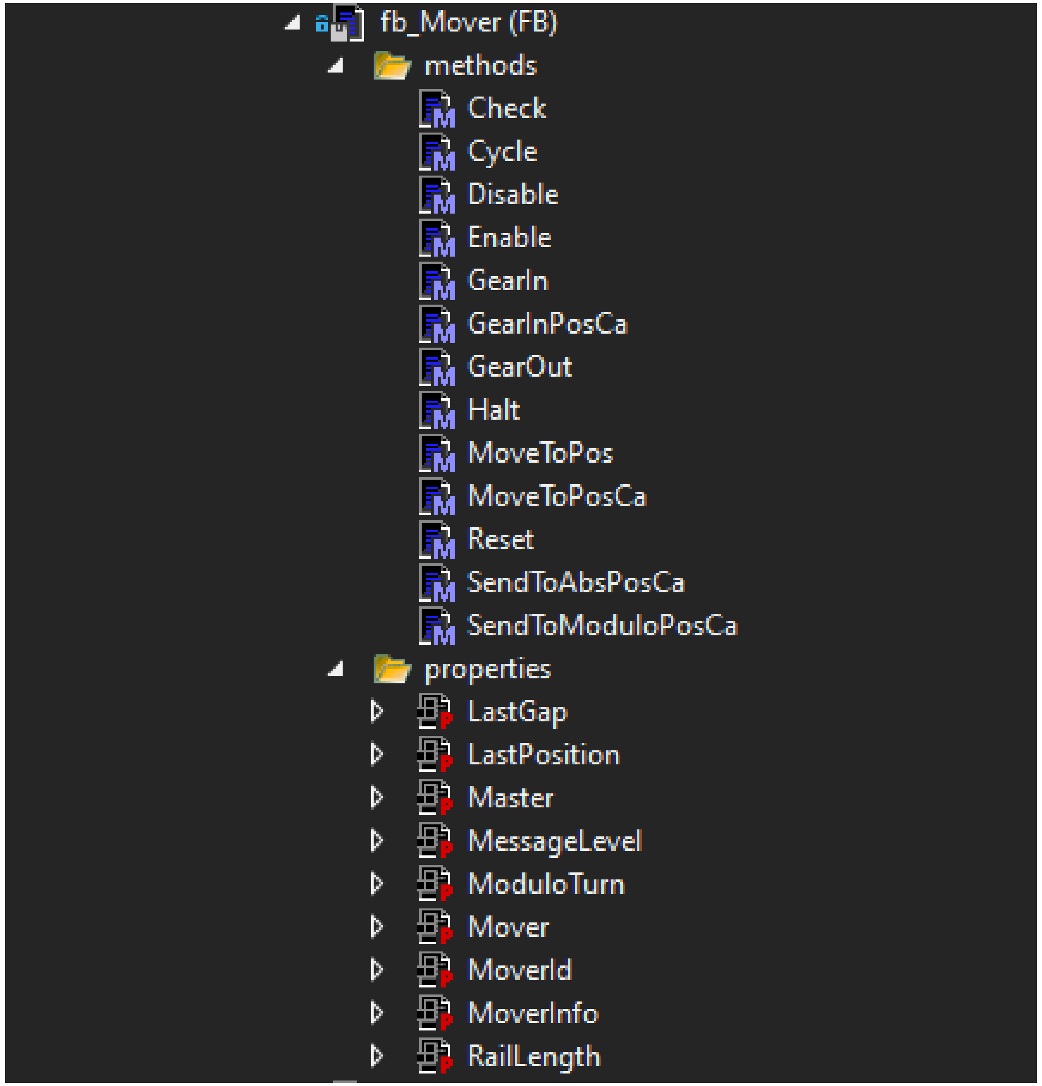


- fb_Mover (FB)
  - methods
  - properties
- fb_MoverCtrl (FB)
  - private
  - properties
- I_XtsTransport_Mover
  - methods
  - properties

Design

- **fb_Mover:**
  - **Cycle()** method must be called by extending class
  - **Check()** must be used in extending class (pointer checks)
  - Method flow charts, see doc folder
    - Execute behaviour:

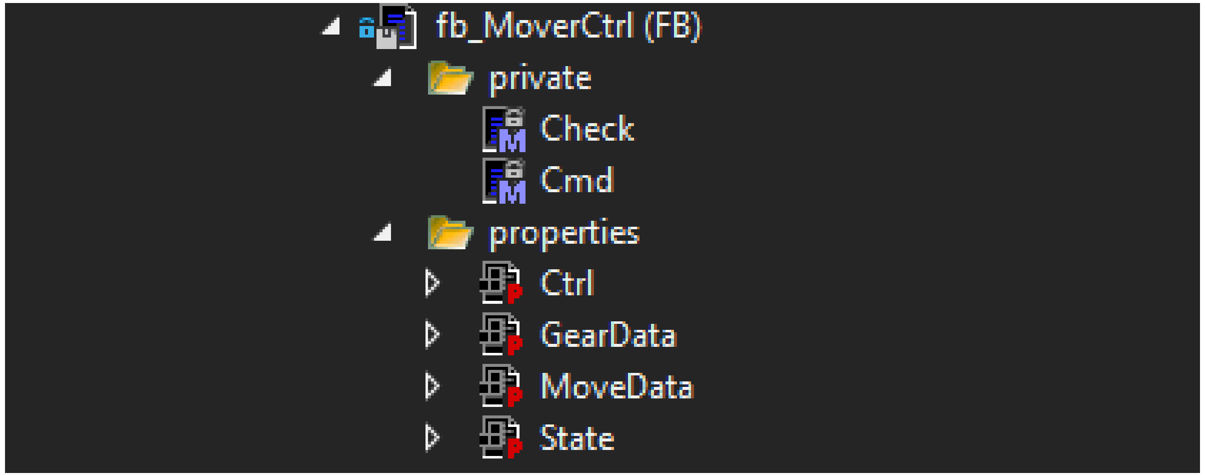
      → initialize call with ‘Execute:=FALSE’
    - Methods feedback: E_PROGRESS



- fb_Mover (FB)
  - methods
    - Check
    - Cycle
    - Disable
    - Enable
    - GearIn
    - GearInPosCa
    - GearOut
    - Halt
    - MoveToPos
    - MoveToPosCa
    - Reset
    - SendToAbsPosCa
    - SendToModuloPosCa
  - properties
    - LastGap
    - LastPosition
    - Master
    - MessageLevel
    - ModuloTurn
    - Mover
    - MoverId
    - MoverInfo
    - RailLength

Design

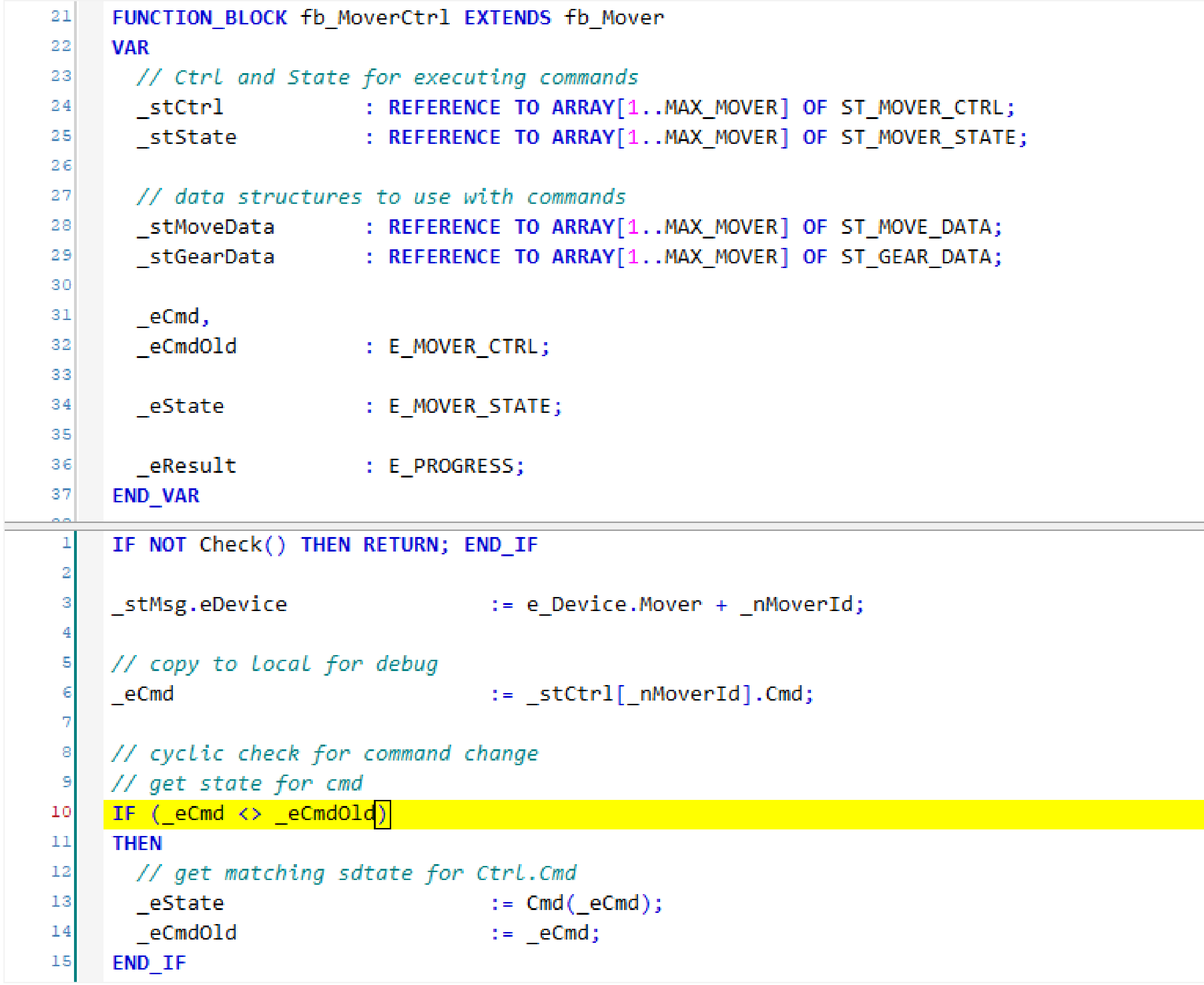
- **fb_MoverCtrl():**
  - Requires cyclic call of instance
    - See calls in MAIN() as example
  - OnChange check of Ctrl.Cmd
  - References to Ctrl/State structs
  - References to Mover motion parameter structs
  - State.State feedback:
    - Command enum equivalent + E_PROGRESS



- fb_MoverCtrl (FB)
  - private
    - Check
    - Cmd
  - properties
    - Ctrl
    - GearData
    - MoveData
    - State

Design

- **fb_MoverCtrl():**
  - Cyclic execution of movement commands
  - OnChange check of Ctrl.Cmd
  - References to Ctrl/State structs
  - References to Mover motion parameter structs

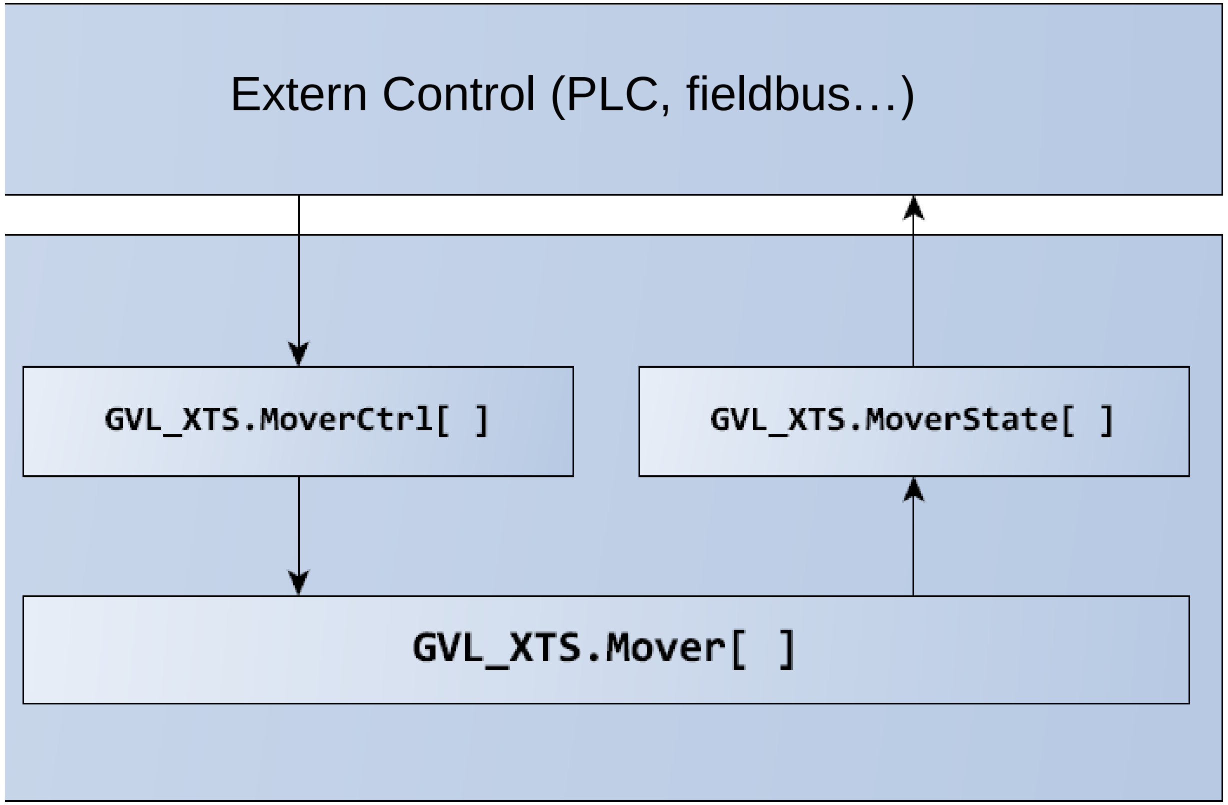


```
FUNCTION_BLOCK fb_MoverCtrl EXTENDS fb_Mover
VAR
  // Ctrl and State for executing commands
  _stCtrl           : REFERENCE TO ARRAY[1..MAX_MOVER] OF ST_MOVER_CTRL;
  _stState          : REFERENCE TO ARRAY[1..MAX_MOVER] OF ST_MOVER_STATE;

  // data structures to use with commands
  _stMoveData       : REFERENCE TO ARRAY[1..MAX_MOVER] OF ST_MOVE_DATA;
  _stGearData       : REFERENCE TO ARRAY[1..MAX_MOVER] OF ST_GEAR_DATA;

  _eCmd,
  _eCmdOld          : E_MOVER_CTRL;

  _eState           : E_MOVER_STATE;

  _eResult          : E_PROGRESS;
END_VAR
```

```
IF NOT Check() THEN RETURN; END_IF

_stMsg.eDevice                := e_Device.Mover + _nMoverId;

// copy to local for debug
_eCmd                         := _stCtrl[_nMoverId].Cmd;

// cyclic check for command change
// get state for cmd
IF (_eCmd <> _eCmdOld)
THEN
  // get matching sdtate for Ctrl.Cmd
  _eState                     := Cmd(_eCmd);
  _eCmdOld                    := _eCmd;
END_IF
```

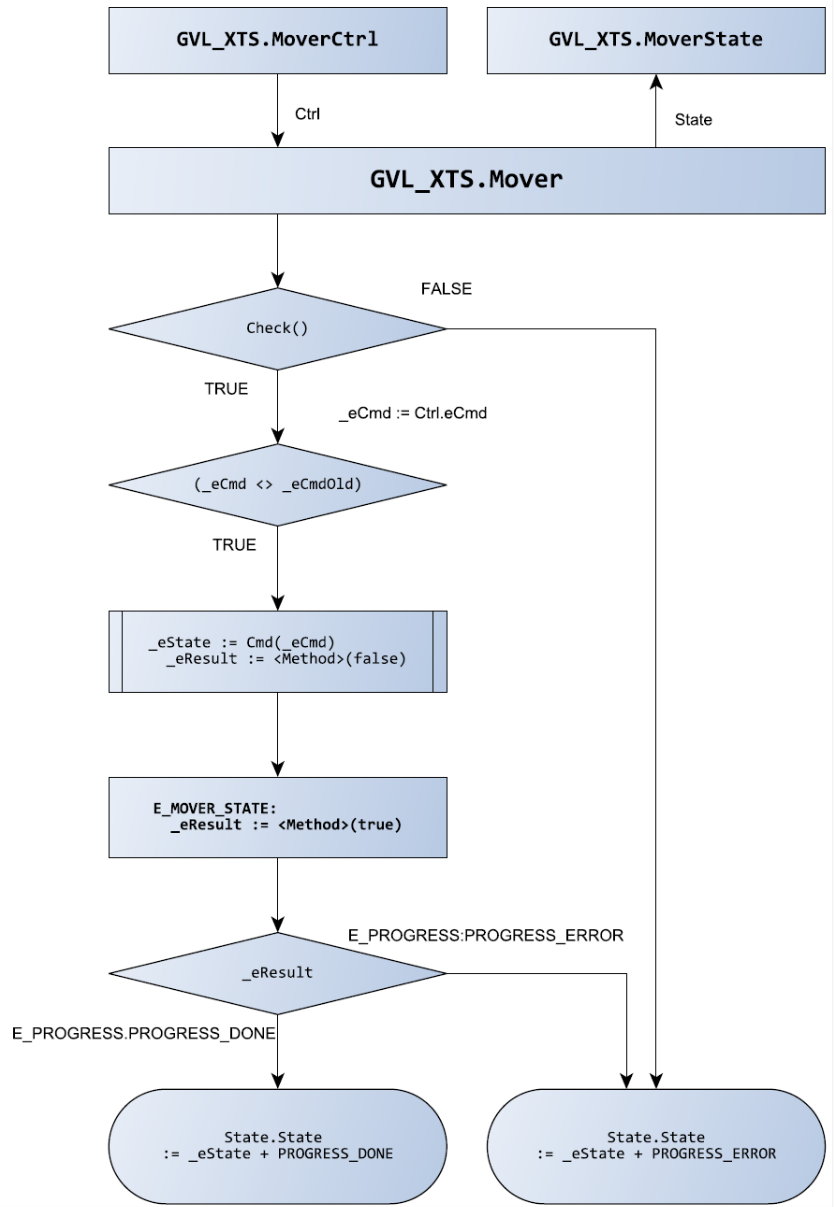
Design

- **fb_MoverCtrl():**
  - Control structure
    - ST_MOVER_CTRL
  - State structure
    - ST_MOVER_STATE
  - Ctrl/State structs are used to address mover
  - Parameter structures are to be used according to the job you want to give to the mover
    - ST_MOVE_DATA
    - ST_GEAR_DATA
  - Mover information:
    - ST_MOVER_INFO



Design

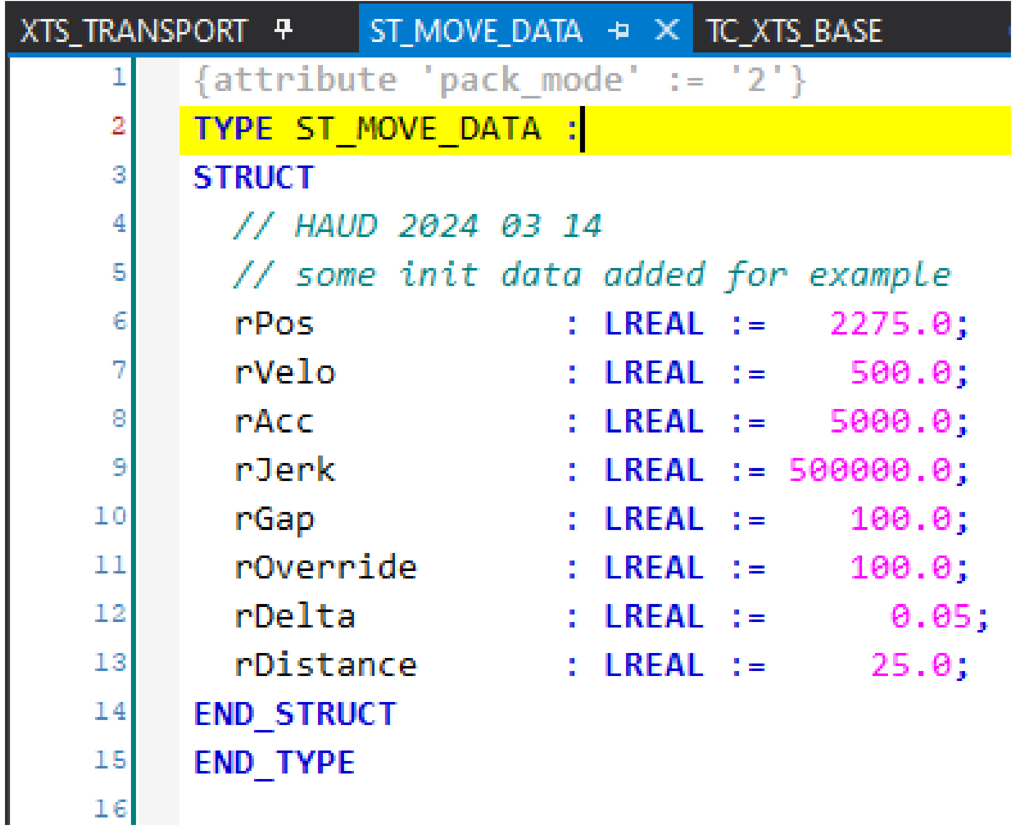
- **fb_MoverCtrl():**
  - **ST_MOVER_CTRL / ST_MOVER_STATE**
  - Command
  - State + Progress
    - See doc folder for flow chart



Design

BECKHOFF

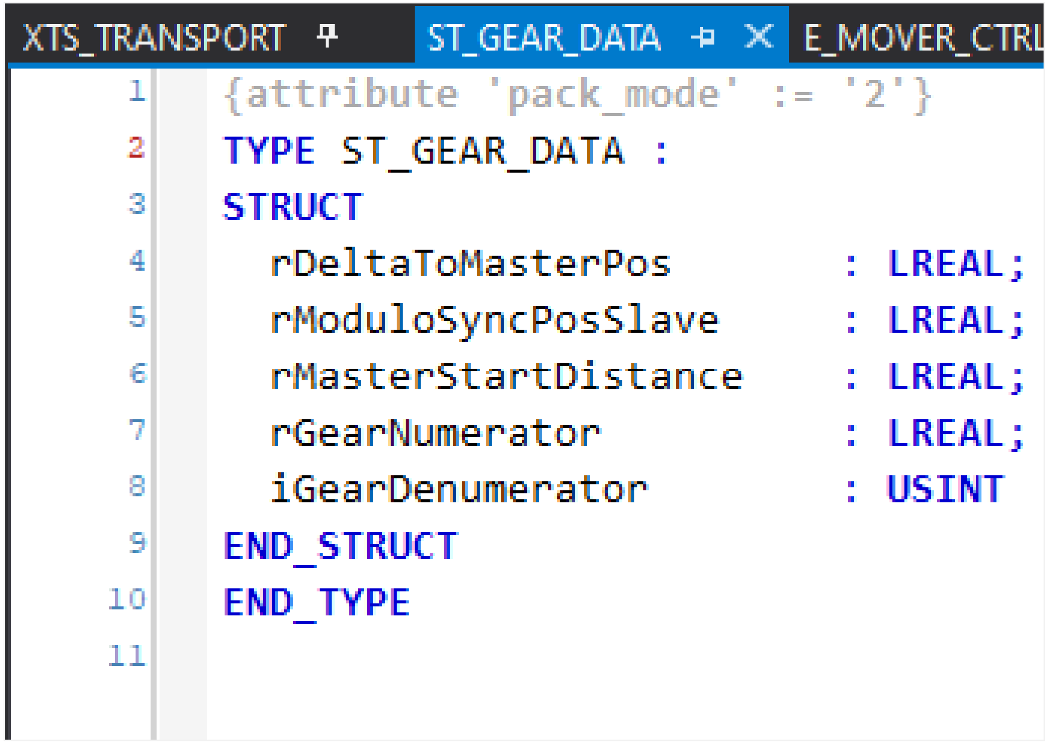
- **fb_MoverCtrl():**
  - **ST_MOVE_DATA**
  - Parameter struct for motion commands



```
{attribute 'pack_mode' := '2'}
TYPE ST_MOVE_DATA :
STRUCT
  // HAUD 2024 03 14
  // some init data added for example
  rPos            : LREAL :=   2275.0;
  rVelo           : LREAL :=    500.0;
  rAcc            : LREAL :=   5000.0;
  rJerk           : LREAL := 500000.0;
  rGap            : LREAL :=    100.0;
  rOverride       : LREAL :=    100.0;
  rDelta          : LREAL :=      0.05;
  rDistance       : LREAL :=     25.0;
END_STRUCT
END_TYPE
```

Design

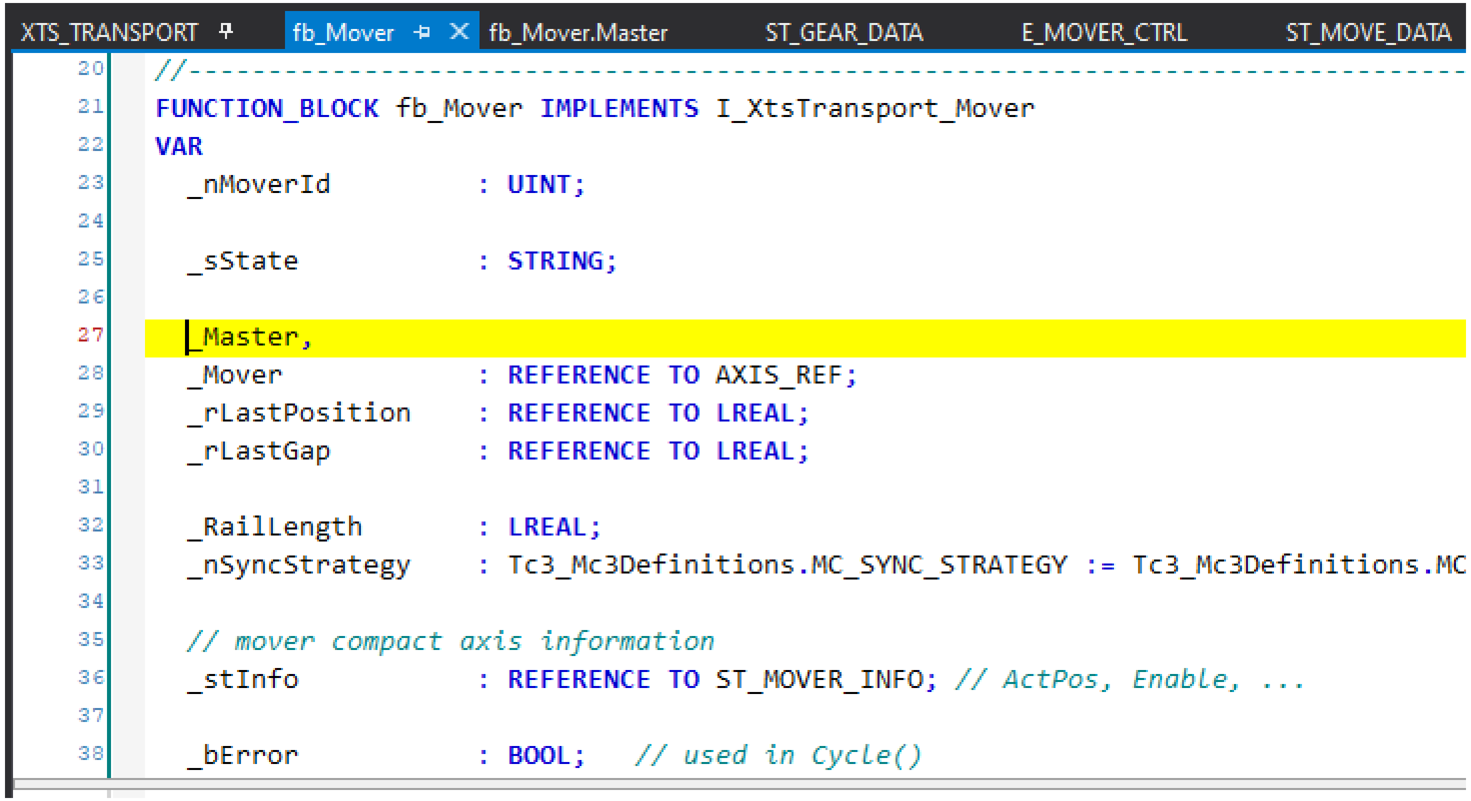
- **fb_MoverCtrl():**
  - **ST_GEAR_DATA**
  - Parameter struct for motion commands
    - E_MOVER_CTRL.MOVER_GEAR_IN
    - E_MOVER_CTRL.MOVER_GEAR_OUT
    - E_MOVER_CTRL.MOVER_GEAR_IN_POS_CA



```
{attribute 'pack_mode' := '2'}
TYPE ST_GEAR_DATA :
STRUCT
   rDeltaToMasterPos          : LREAL;
   rModuloSyncPosSlave        : LREAL;
   rMasterStartDistance       : LREAL;
   rGearNumerator             : LREAL;
   iGearDenumerator           : USINT
END_STRUCT
END_TYPE
```

Design

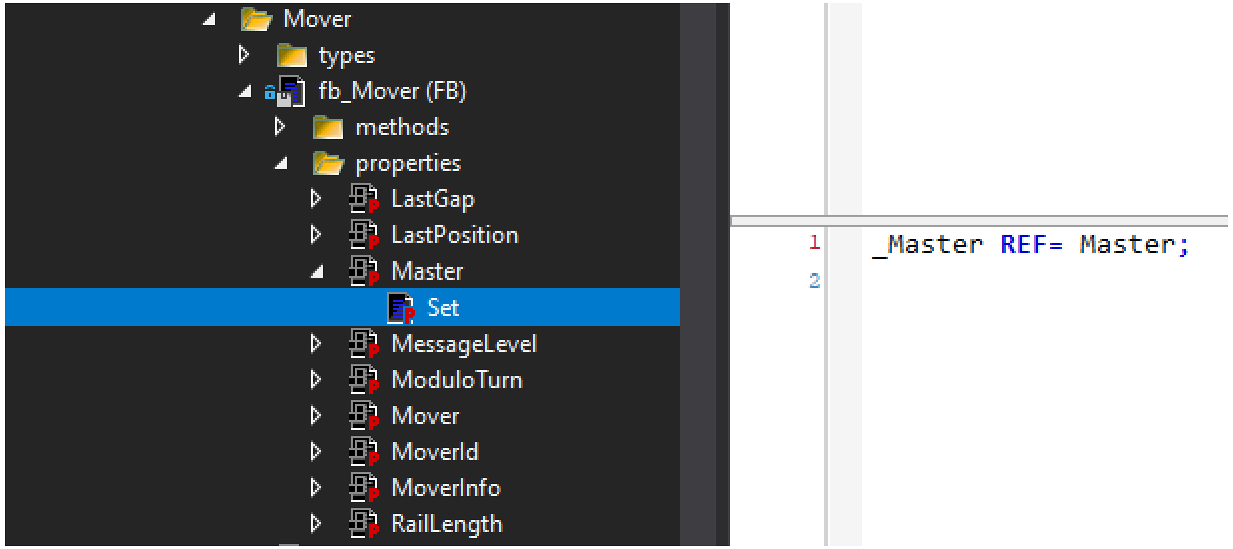
- **fb_MoverCtrl():**
  - **ST_GEAR_DATA**
  - Reference to MasterAxis
    - local var '_Master' must be valid reference
    - Base class contains property to link AXIS_REF of master axis to fb_Mover



```
//------------------------------------------------------------------------------------------------------
FUNCTION_BLOCK fb_Mover IMPLEMENTS I_XtsTransport_Mover
VAR
  _nMoverId          : UINT;

  _sState            : STRING;

  _Master,
  _Mover             : REFERENCE TO AXIS_REF;
  _rLastPosition     : REFERENCE TO LREAL;
  _rLastGap          : REFERENCE TO LREAL;

  _RailLength        : LREAL;
  _nSyncStrategy     : Tc3_Mc3Definitions.MC_SYNC_STRATEGY := Tc3_Mc3Definitions.MC

  // mover compact axis information
  _stInfo            : REFERENCE TO ST_MOVER_INFO; // ActPos, Enable, ...

  _bError            : BOOL;   // used in Cycle()
```

Design

- **fb_MoverCtrl():**
  - **ST_GEAR_DATA**
  - Reference to MasterAxis
    - local var '_Master' must be valid reference
    - Base class contains property to link AXIS_REF of master axis to Mover
    - Property you have to set before / on use of gearing motion function

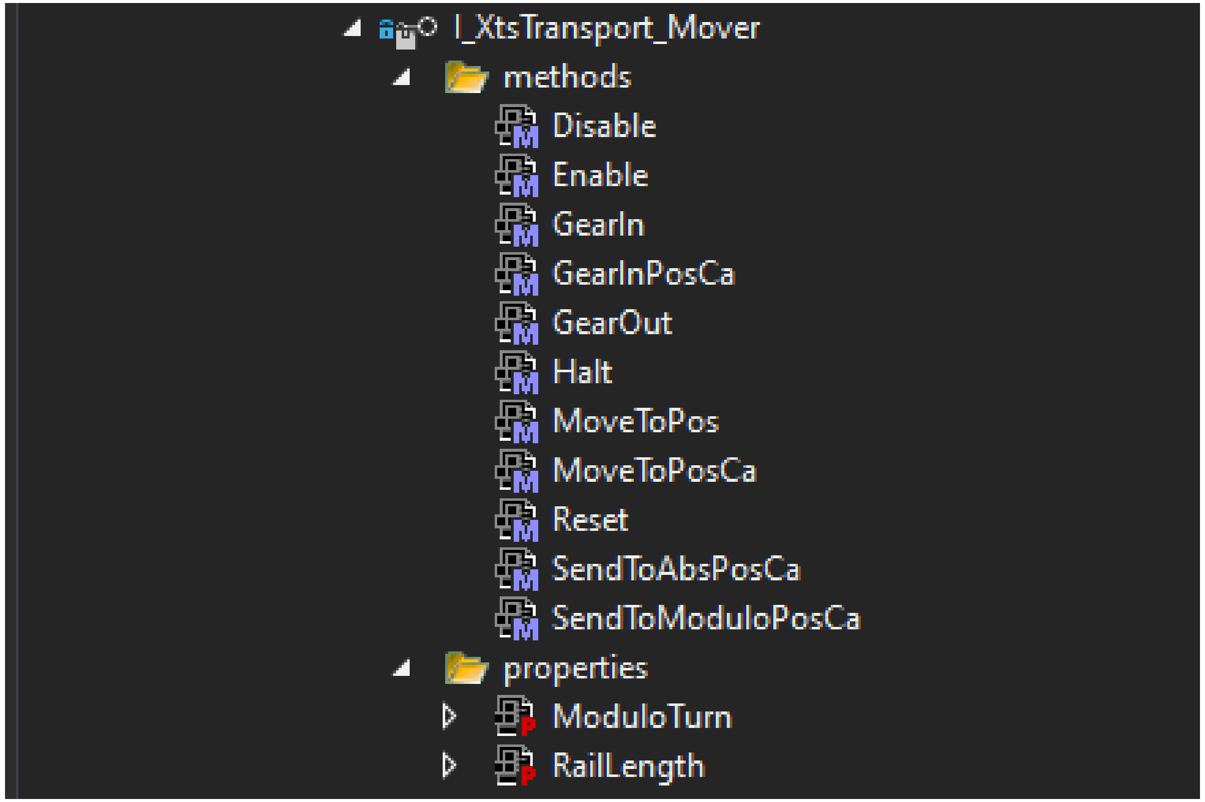


```
_Master REF= Master;
```

Design

BECKHOFF

- **I_Transport_Mover:**
  - Interface for use in other classes
    - Fb_TransportUnit
    - Fb_CaGroup
    - Fb_Station
  - Interface is implemented by fb_Mover

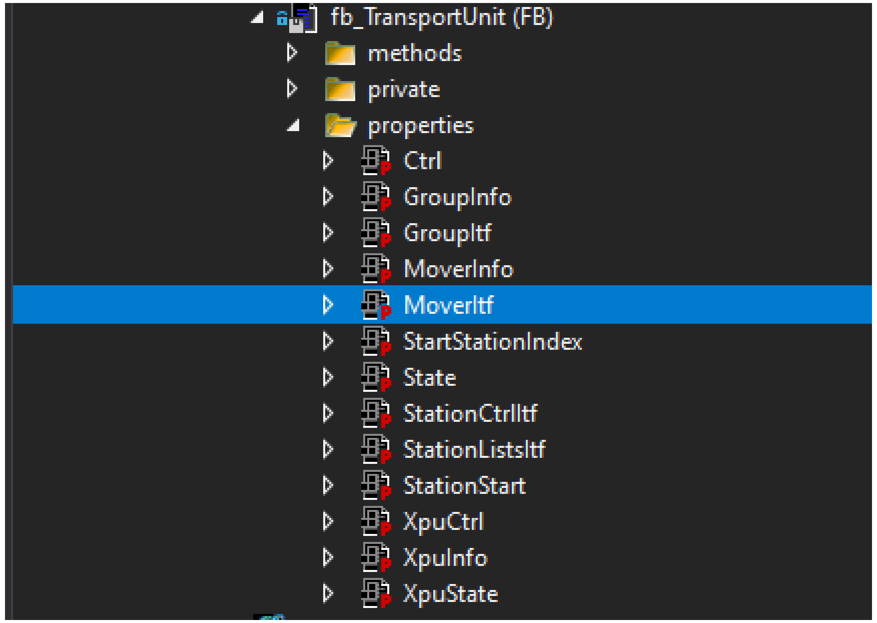


```
I_XtsTransport_Mover
  methods
    Disable
    Enable
    GearIn
    GearInPosCa
    GearOut
    Halt
    MoveToPos
    MoveToPosCa
    Reset
    SendToAbsPosCa
    SendToModuloPosCa
  properties
    ModuloTurn
    RailLength
```

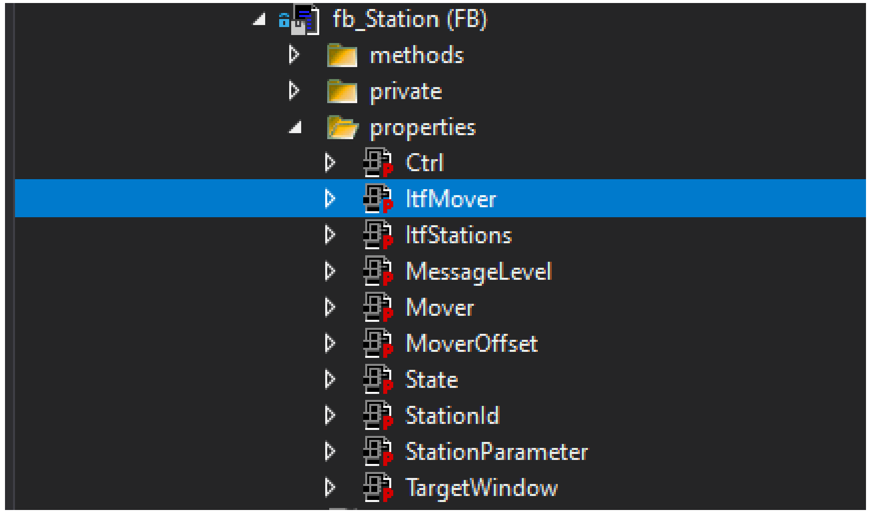
Design

BECKHOFF

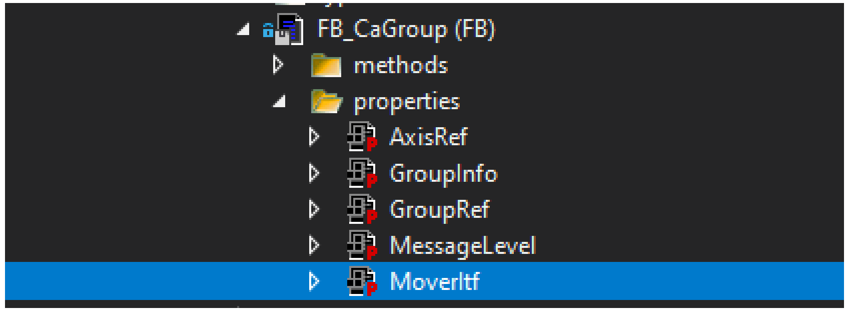
- **I_Transport_Mover:**
  - Used in Property by other classes



```
fb_TransportUnit (FB)
  methods
  private
  properties
    Ctrl
    GroupInfo
    GroupItf
    MoverInfo
    MoverItf
    StartStationIndex
    State
    StationCtrlItf
    StationListsItf
    StationStart
    XpuCtrl
    XpuInfo
    XpuState
```



```
fb_Station (FB)
  methods
  private
  properties
    Ctrl
    ItfMover
    ItfStations
    MessageLevel
    Mover
    MoverOffset
    State
    StationId
    StationParameter
    TargetWindow
```



```
FB_CaGroup (FB)
  methods
  properties
    AxisRef
    GroupInfo
    GroupRef
    MessageLevel
    MoverItf
```

LICENSE

XTS_TRANSPORT_LAYER project