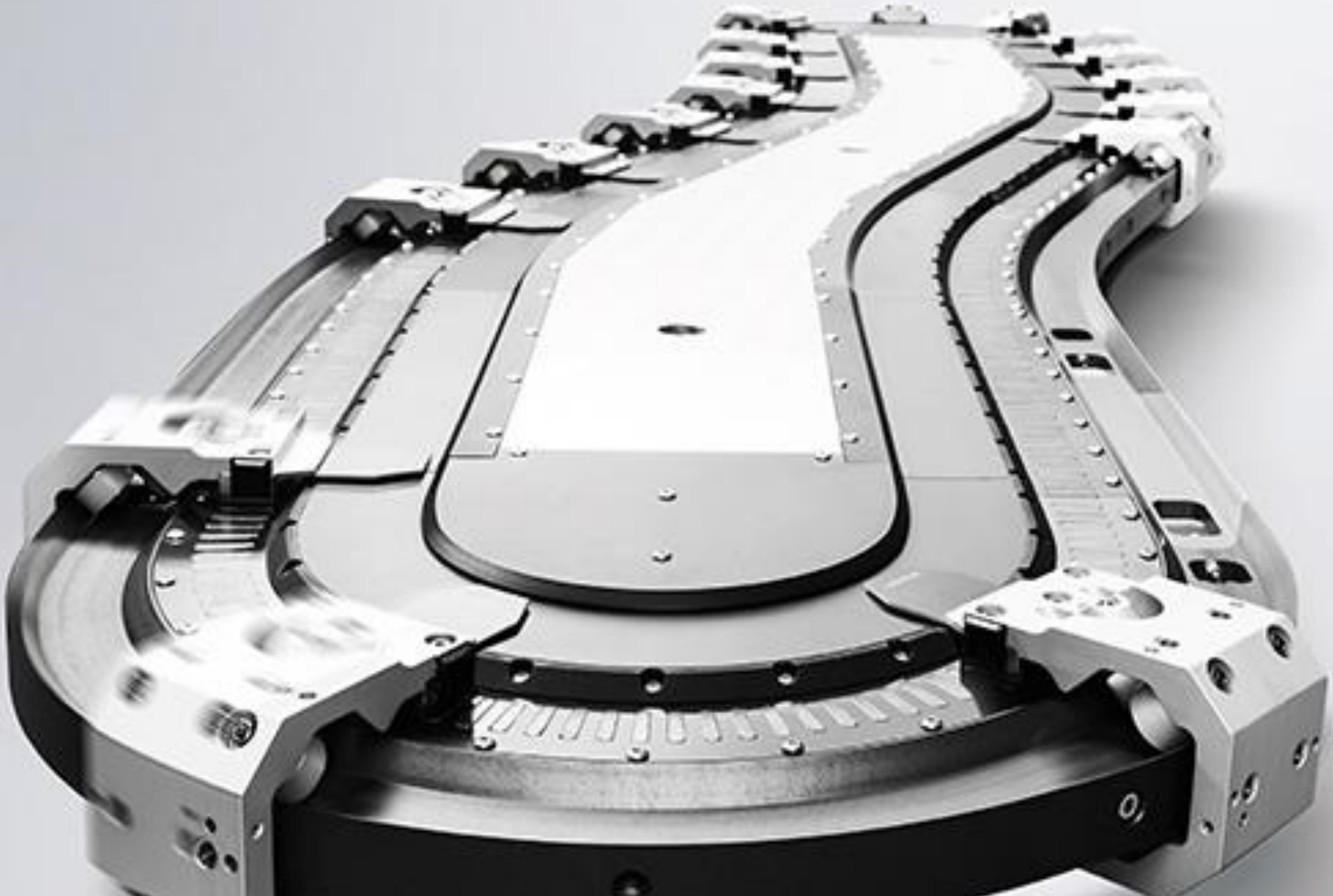
XTS TRANSPORT LAYER – a station based approach

BECKHOFF



XTS TRANSPORT LAYER

Daniel Hauer (HAUD)

BECKHOFF

XTS TRANSPORT LAYER (HAUD)

BECKHOFF

1. Introduction

- This project collection is intended to convey the idea of a stand alone XTS transport layer to use in heterogen environments / applications.
- The main idea is that for every process a corresponding position on the xts exists.
- In order to reduce the amount of repetitive work when implementing a XTS into a machine, this project collection may help to put a transport layer in place
- A transport layer shall have an interface for guiding a mover through a process station
- A transport layer shall have an interface to manipulate a mover within a station or for a certain task
- A transport layer shall have an interface for setting-up or clearing the CollisionAvoidance Group

XTS TRANSPORT LAYER (HAUD)

BECKHOFF

1. Introduction

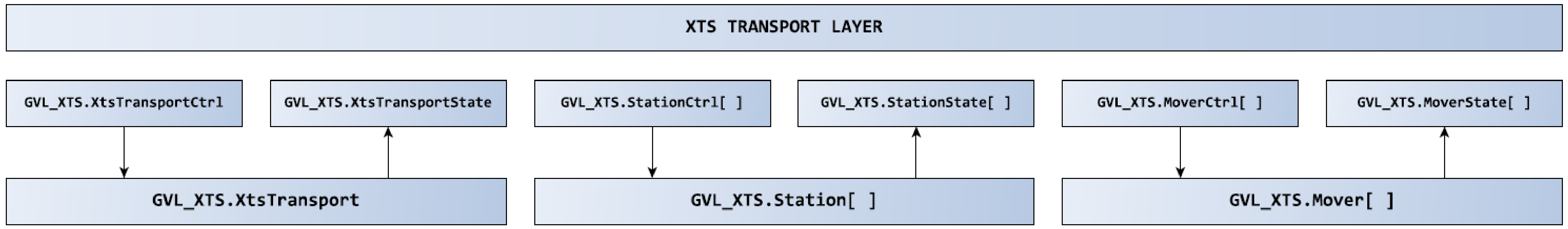
- The XTS transport system enables a flexible product transport for various proccesses.
- In combination with the Collision Avoidance library positioning of movers does not require extra monitoring of the axis
- Can be used for a station based approach, in which a station class is available for interaction with your process control
- Can be used for a mover based approach, so your process control has a direct connection to every mover
- Can be used as a combination of station based and mover based approach
- The use of predefined datafields enables you to control XTS Transport Layer through fieldbus or network.
- Use of interface pointers for cross communication between classes.

XTS TRANSPORT LAYER (HAUD)

BECKHOFF

1. Introduction

- designed for use with extern cyclic or non cyclic flow control
- station based approach with individual targeting of mover
- handshake in station with extern process flow (ST_STATION_CTRL / ST_STATION_STATE)
- individual cyclic mover interface with given set of movement functionalities (ST_MOVER_CTRL / ST_MOVER_STATE)



1. Introduction

Planning requirements for use of fb_Station:

- Put the Modulo turn anywhere, **BUT NOT** within WaitPos, StopPos, ReleaseDistance of a station. The code does not support crossing the modulo turn within a station.

- The Use of LinkedList methods (AddTail, GetHead) requires thought about when the mover is entered into the target station.
- a. parallel stations for a process:
    - P1 uses XTS_STN[1] to XTS_STN[4] → rReleaseDistance of STN[4] shall be shortest, all other stations follow accordingly.
- b. using stations sparsely:
    - In this case it is easiest to always handshake the stations and use the forwarding command if a station shall be skipped: STATION_MOVER_SEND.
- c. deactivating stations:
    - Make sure the queue is empty before deactivating, since the waiting mover will hold up all the others in case of required deactivation while movers are in the queue:
    - handshake mover with E_STATION_CTRL.STATION_MOVER_SEND to new target station
    - Do not send any new mover to the station in question
        - disable station
        - preceeding stations continue workflow with changed ST_STATION_CTRL.nTargetStation

XTS TRANSPORT LAYER (HAUD)

BECKHOFF

1. Introduction

Planning requirements for use of fb_Station:

- know thyself
  - all coordinates are modulo values, from station to station only forward,
    within station limits backward movement by use of negative nest offset
    or use of ST_MOVER_CTRL.
    IF move backwards you have to make sure that there is room for it
    --> distance between PosWait and PosStop

2. Requirements

- XtsTransport
  - Access to CA group function blocks (interface pointer)
  - Access to Stations (interface pointer)
  - Access to Movers (interface pointer)
  - Cyclic interface for access from extern control
    - Ctrl (write): command
    - State (read): response to command
      - information from Xpu
      - Information from CA Group

2. Requirements

- Xpu (XTS Processing Unit)
  - Check Init Parameter
  - Check Online Parameter
  - Get Module Info Data
  - Connect TcCOM Objects to instances from XTS_Utility.lib function blocks
  - Cyclic plausibility checks
    - Mover ID detection after init
  - Cyclic interface for access from main control
    - Ctrl (write): command
    - State (read): response to command
    - Info (read): details from cyclic checks

2. Requirements

BECKHOFF

- CaGroup
  - Access to group function blocks
  - Access to movers for group commands
  - Get Group Info Data
  - Implements interface pointer

2. Requirements

- Mover
  - Access to MC function blocks
  - Access to CA function blocks
  - Cyclic interface for access from extern control
    - Ctrl (write): command
    - Data (write): command parameter
    - State (read): response to command
  - Interface pointer for access from:
    - TransportUnit
    - Station

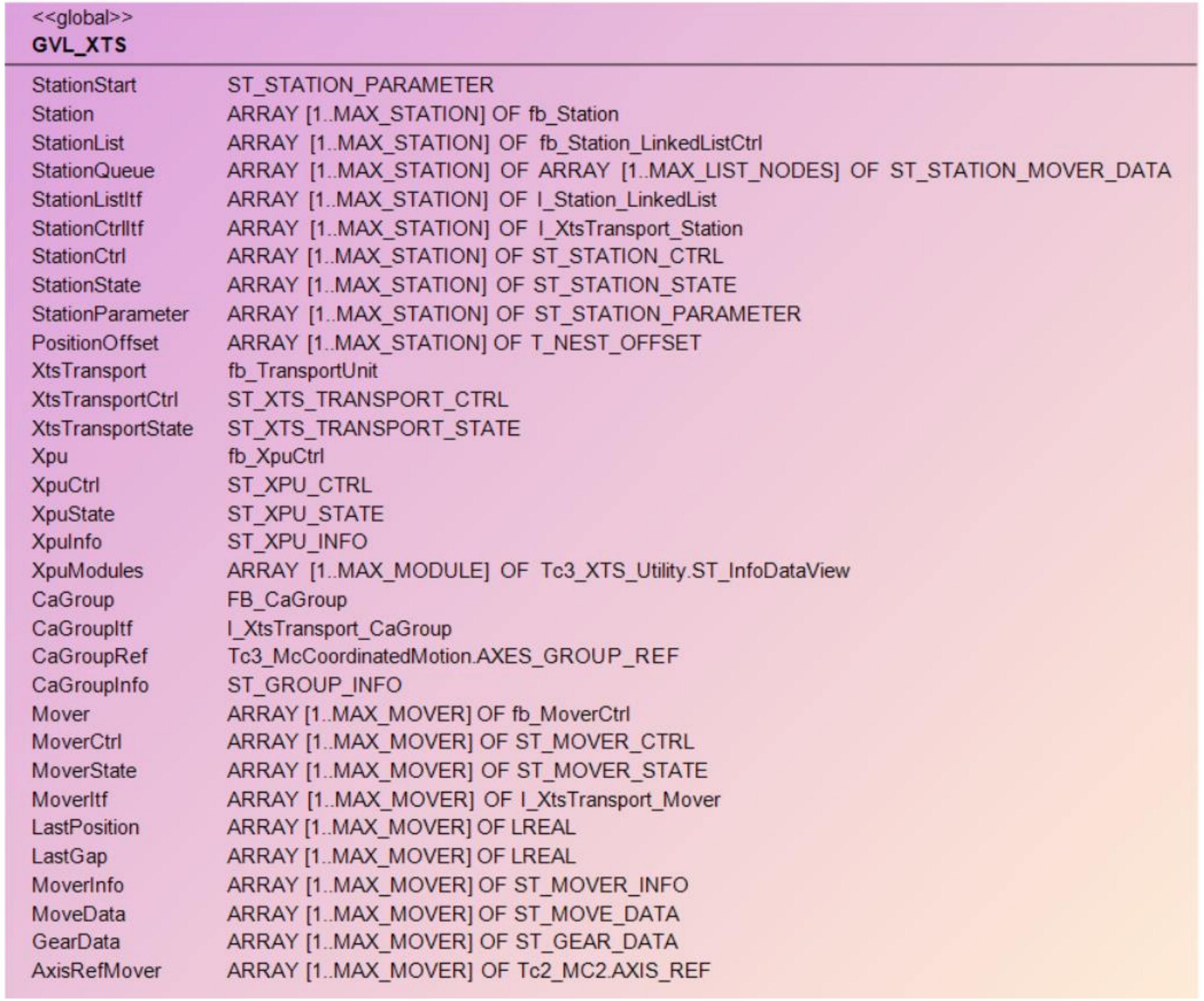
2. Requirements

- Station
  - Handshake mover transport with extern control
  - Close observation of movements with feedback to extern control
  - List for movers in queue
  - Cyclic interface for access from extern control
    - Ctrl (write): command and parameter
    - State (read): response to command and information about mover and queue
  - Uses Mover interface pointer

3. Design

BECKHOFF

- Namespace GVL_XTS
  - **Station**
    - Handshake with Process for mover transport
  - **XtsTransport**
    - Main command interface to extern control
  - **XpuCtrl**
    - Access to TcCOM Objects
    - Cyclic plausibility checks
  - **CaGroup**
    - Access to CA library
  - **MoverCtrl**
    - Access to MC and CA library

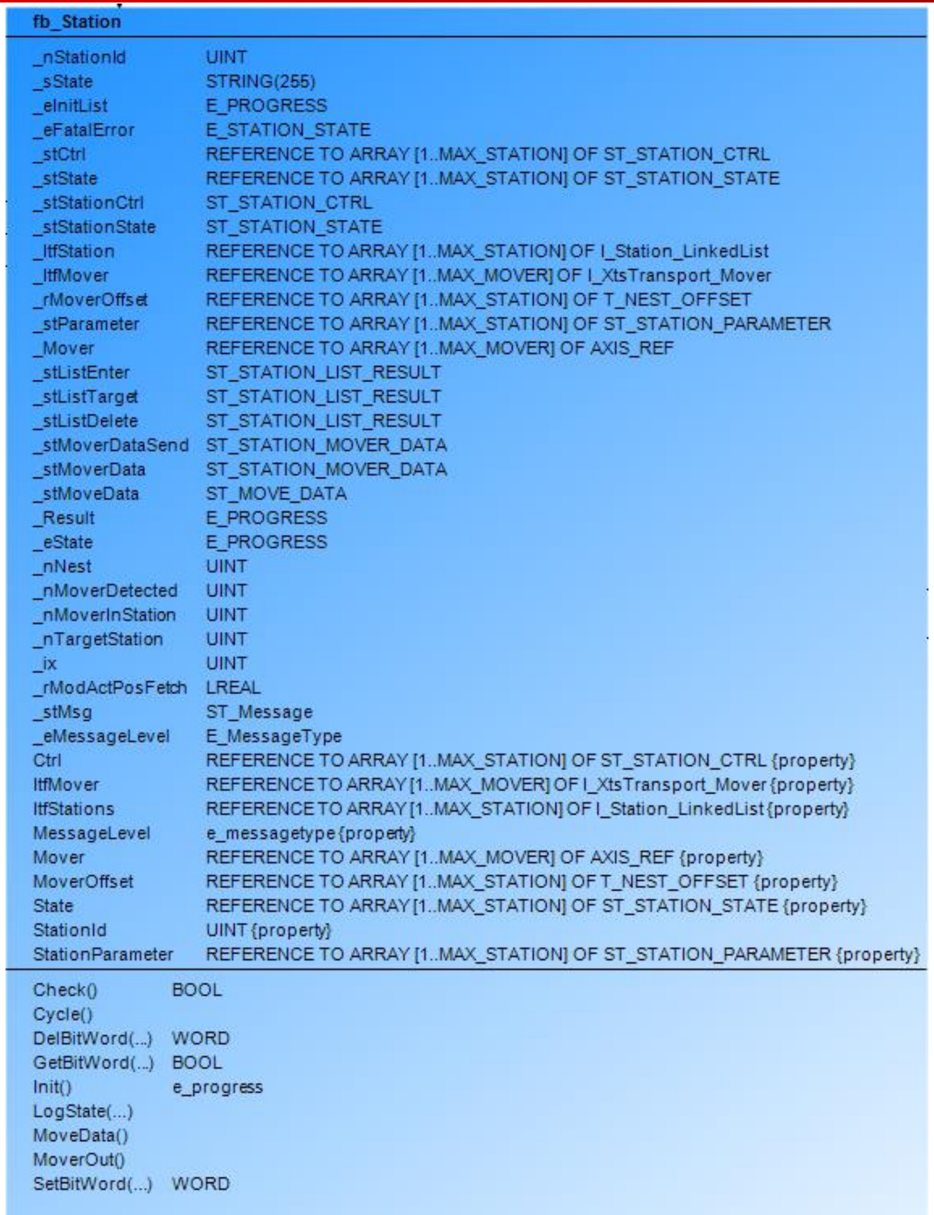


<<global>>
GVL_XTS

StationStart	ST_STATION_PARAMETER
Station	ARRAY [1..MAX_STATION] OF fb_Station
StationList	ARRAY [1..MAX_STATION] OF fb_Station_LinkedListCtrl
StationQueue	ARRAY [1..MAX_STATION] OF ARRAY [1..MAX_LIST_NODES] OF ST_STATION_MOVER_DATA
StationListItf	ARRAY [1..MAX_STATION] OF I_Station_LinkedList
StationCtrlItf	ARRAY [1..MAX_STATION] OF I_XtsTransport_Station
StationCtrl	ARRAY [1..MAX_STATION] OF ST_STATION_CTRL
StationState	ARRAY [1..MAX_STATION] OF ST_STATION_STATE
StationParameter	ARRAY [1..MAX_STATION] OF ST_STATION_PARAMETER
PositionOffset	ARRAY [1..MAX_STATION] OF T_NEST_OFFSET
XtsTransport	fb_TransportUnit
XtsTransportCtrl	ST_XTS_TRANSPORT_CTRL
XtsTransportState	ST_XTS_TRANSPORT_STATE
Xpu	fb_XpuCtrl
XpuCtrl	ST_XPU_CTRL
XpuState	ST_XPU_STATE
XpuInfo	ST_XPU_INFO
XpuModules	ARRAY [1..MAX_MODULE] OF Tc3_XTS_Utility.ST_InfoDataView
CaGroup	FB_CaGroup
CaGroupItf	I_XtsTransport_CaGroup
CaGroupRef	Tc3_McCoordinatedMotion.AXES_GROUP_REF
CaGroupInfo	ST_GROUP_INFO
Mover	ARRAY [1..MAX_MOVER] OF fb_MoverCtrl
MoverCtrl	ARRAY [1..MAX_MOVER] OF ST_MOVER_CTRL
MoverState	ARRAY [1..MAX_MOVER] OF ST_MOVER_STATE
MoverItf	ARRAY [1..MAX_MOVER] OF I_XtsTransport_Mover
LastPosition	ARRAY [1..MAX_MOVER] OF LREAL
LastGap	ARRAY [1..MAX_MOVER] OF LREAL
MoverInfo	ARRAY [1..MAX_MOVER] OF ST_MOVER_INFO
MoveData	ARRAY [1..MAX_MOVER] OF ST_MOVE_DATA
GearData	ARRAY [1..MAX_MOVER] OF ST_GEAR_DATA
AxisRefMover	ARRAY [1..MAX_MOVER] OF Tc2_MC2.AXIS_REF

3. Design

- GVL_XTS.Station
  - fb_Station[].Cycle
    - State machine for handshaking with extern control
      - Init (clears everything in station)
      - Enable
      - Mover Enter
      - Stop Position(s)
      - Mover Out
      - Empty
    - Control writes ticket for mover
      - MoverId
      - TargetStation
      - Mask
      - Offset



fb_Station

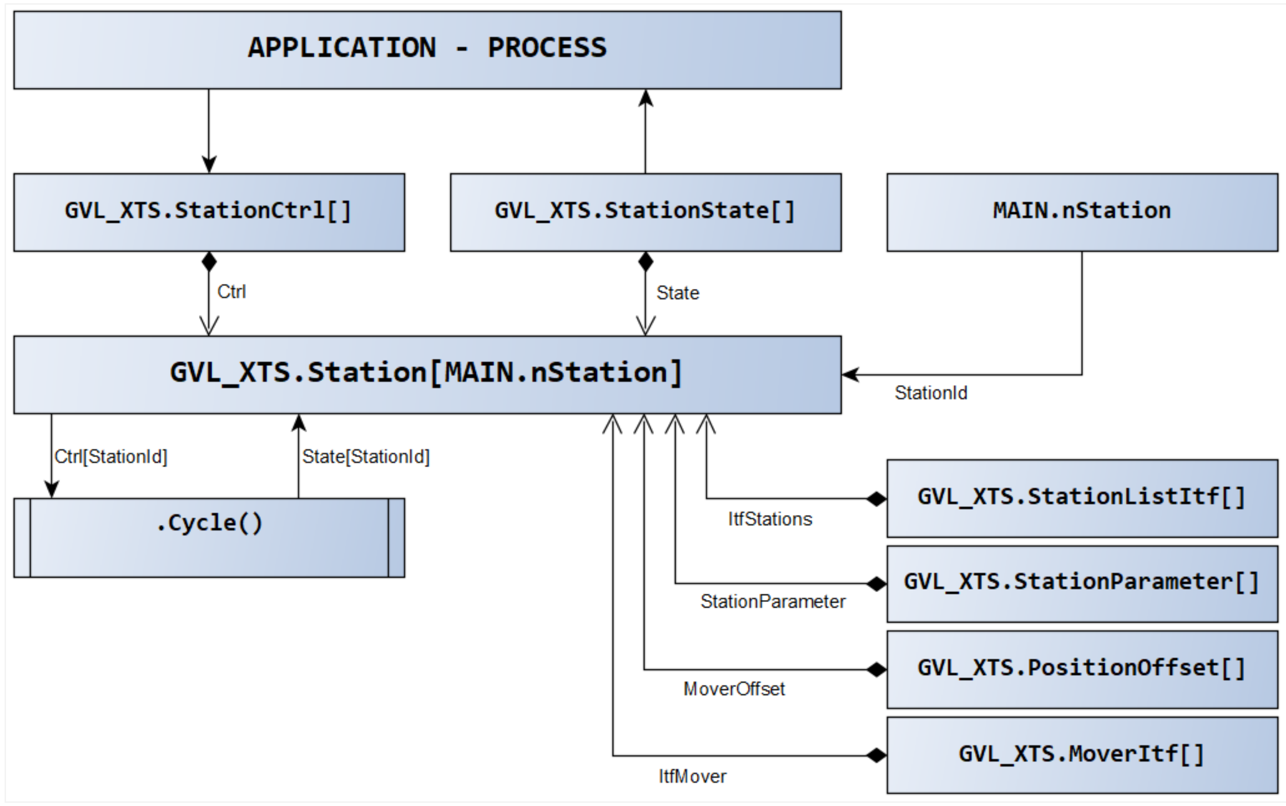
_nStationId	UINT
_sState	STRING(255)
_eInitList	E_PROGRESS
_eFatalError	E_STATION_STATE
_stCtrl	REFERENCE TO ARRAY [1..MAX_STATION] OF ST_STATION_CTRL
_stState	REFERENCE TO ARRAY [1..MAX_STATION] OF ST_STATION_STATE
_stStationCtrl	ST_STATION_CTRL
_stStationState	ST_STATION_STATE
_ItfStation	REFERENCE TO ARRAY [1..MAX_STATION] OF I_Station_LinkedList
_ItfMover	REFERENCE TO ARRAY [1..MAX_MOVER] OF I_XtsTransport_Mover
_rMoverOffset	REFERENCE TO ARRAY [1..MAX_STATION] OF T_NEST_OFFSET
_stParameter	REFERENCE TO ARRAY [1..MAX_STATION] OF ST_STATION_PARAMETER
_Mover	REFERENCE TO ARRAY [1..MAX_MOVER] OF AXIS_REF
_stListEnter	ST_STATION_LIST_RESULT
_stListTarget	ST_STATION_LIST_RESULT
_stListDelete	ST_STATION_LIST_RESULT
_stMoverDataSend	ST_STATION_MOVER_DATA
_stMoverData	ST_STATION_MOVER_DATA
_stMoveData	ST_MOVE_DATA
_Result	E_PROGRESS
_eState	E_PROGRESS
_nNest	UINT
_nMoverDetected	UINT
_nMoverInStation	UINT
_nTargetStation	UINT
_ix	UINT
_rModActPosFetch	LREAL
_stMsg	ST_Message
_eMessageLevel	E_MessageType
Ctrl	REFERENCE TO ARRAY [1..MAX_STATION] OF ST_STATION_CTRL {property}
ItfMover	REFERENCE TO ARRAY [1..MAX_MOVER] OF I_XtsTransport_Mover {property}
ItfStations	REFERENCE TO ARRAY [1..MAX_STATION] OF I_Station_LinkedList {property}
MessageLevel	e_messagetype {property}
Mover	REFERENCE TO ARRAY [1..MAX_MOVER] OF AXIS_REF {property}
MoverOffset	REFERENCE TO ARRAY [1..MAX_STATION] OF T_NEST_OFFSET {property}
State	REFERENCE TO ARRAY [1..MAX_STATION] OF ST_STATION_STATE {property}
StationId	UINT {property}
StationParameter	REFERENCE TO ARRAY [1..MAX_STATION] OF ST_STATION_PARAMETER {property}

Check()	BOOL
Cycle()	
DelBitWord(...)	WORD
GetBitWord(...)	BOOL
Init()	e_progress
LogState(...)	
MoveData()	
MoverOut()	
SetBitWord(...)	WORD

3. Design

BECKHOFF

- GVL_XTS.Station
  - nStation index is passed as value from caller
  - Global datafields are passed as references (REF=) into fb_Station properties
    - Ctrl / State: handshakes
    - ItfStations: interface pointer to linked list methods for getting and setting of mover data
    - StationParameter: Coordinates and dynamic constraint of XtsStation
    - MoverOffset: correction values for every mover in every station with every nest (StopPos[])
    - ItfMover: interface pointer to CA movements



3. Design

- GVL_XTS.Station
  - Ctrl[nStation] : ST_STATION_CTRL
    - eCmd :
      - enumeration for handshakes with State[nStation].eState
    - nMask :
      - bit mask to be used with multiple stop positions within a XtsStation.
        This mask tells the target station which StopPos[] (nest) has to be worked.
    - nTargetStation :
      - target to send mover to GVL_XTS.Station[nTargetStation].WaitPos
    - rOffset :
      - Optional offset for mover, used in target station in addition to static offset

3. Design

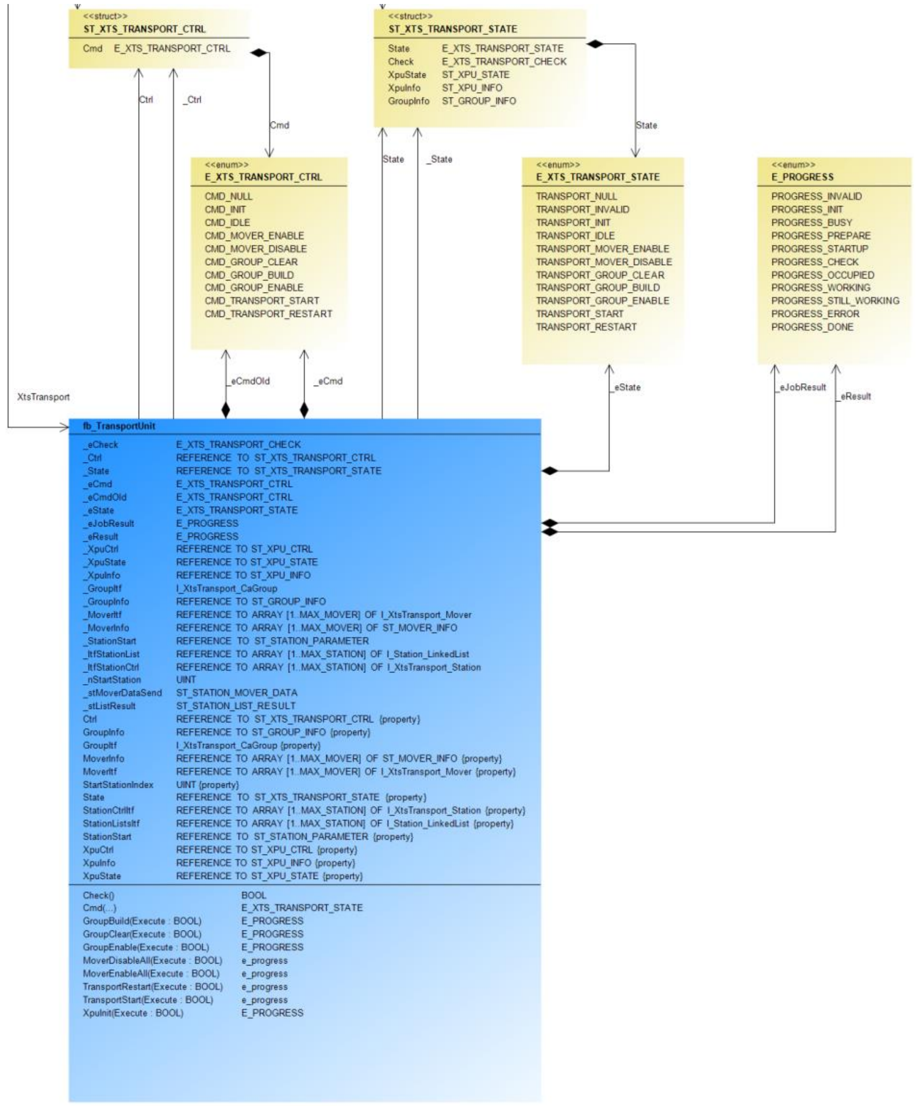
- GVL_XTS.Station
  - State[nStation] : ST_STATION_STATE
    - eState :
      - Enumeration for active station state, Ctrl has to react to
    - nMask :
      - Bitmask for active PosStop[] (nest)
    - nMoverId :
      - Active mover index in station
    - rMoverModPos :
      - Modulo position of active mover
    - nQueue :
      - Count of movers, which were sent to XtsStation

3. Design

- GVL_XTS.StationParameter
  - sText :
    - Description only
  - rPosWait :
    - start of station, a sending station is using this value to send mover to
  - rReleaseDistance :
    - distance mover has to travel (from ActPos) in order for station to go back to mover detection
  - rGap :
    - Active gap on infeed and outfeed of station
  - rVelo :
    - Active velocity on infeed and outfeed of station
  - rAccDec :
    - Active dyn constraint
  - rJerk :
    - Active dyn constraint
  - nConfiguredStopCount :
    - Count of PosStop (nests) a mover has to stop at in XtsStation
  - rPosStop[] :
    - Relative to rPosWait

3. Design

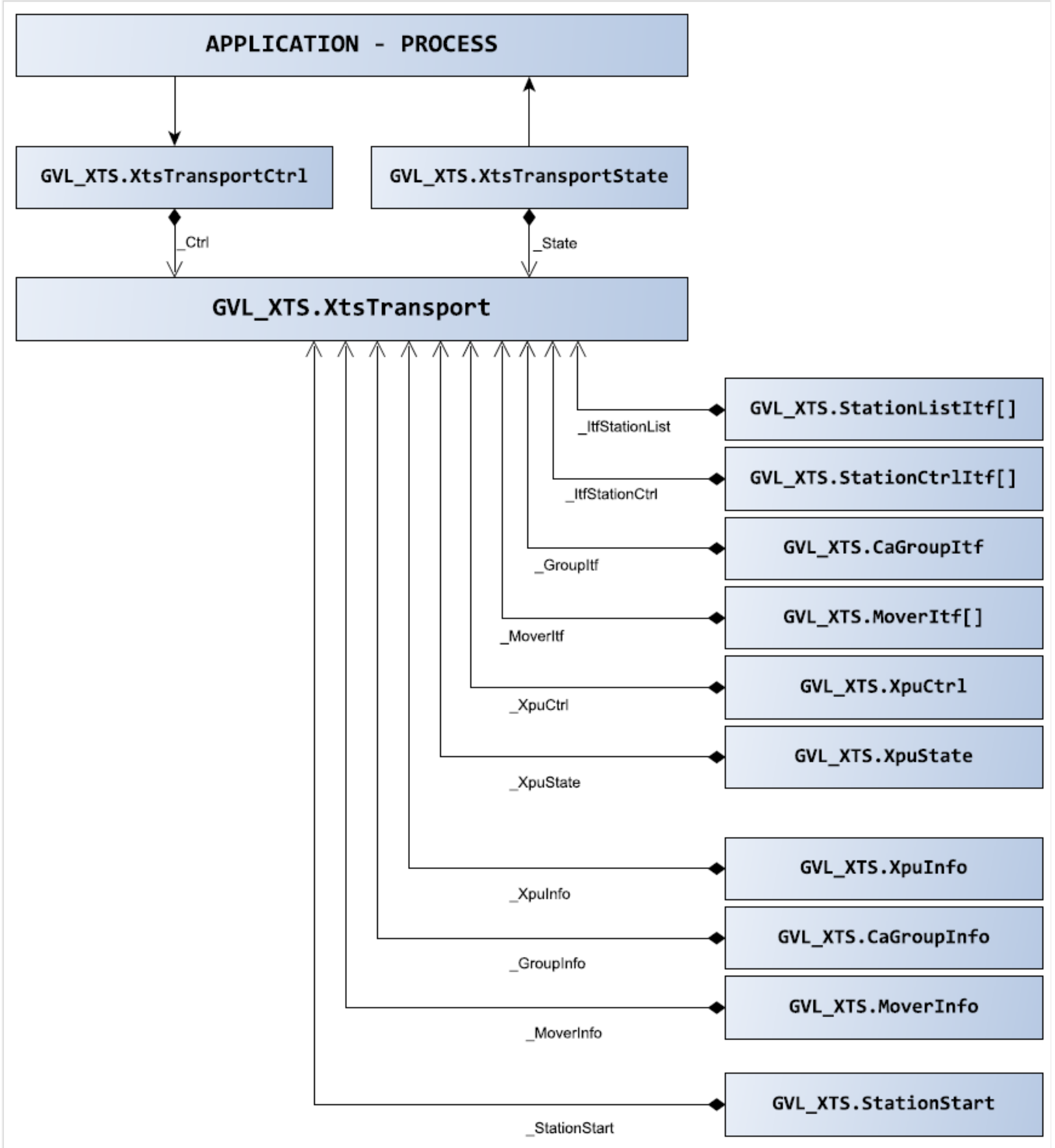
- TransportUnit
  - Fb_TransportUnit():
    - Top level control of XtsTransport
    - Cycle check for change of command:
      - E_XTS_TRANSPORT_CTRL.
        - CMD_INIT
        - CMD_IDLE
        - CMD_MOVER_ENABLE
        - CMD_MOVER_DISABLE
        - CMD_GROUP_CLEAR
        - CMD_GROUP_BUILD
        - CMD_GROUP_ENABLE
        - CMD_TRANSPORT_START



<<struct>>
ST_XTS_TRANSPORT_CTRL
Cmd E_XTS_TRANSPORT_CTRL
<<struct>>
ST_XTS_TRANSPORT_STATE
State E_XTS_TRANSPORT_STATE
Check E_XTS_TRANSPORT_CHECK
XpuState ST_XPU_STATE
XpuInfo ST_XPU_INFO
GroupInfo ST_GROUP_INFO
Ctrl
_Ctrl
Cmd
State
State
_State
<<enum>>
E_XTS_TRANSPORT_CTRL
CMD_NULL
CMD_INIT
CMD_IDLE
CMD_MOVER_ENABLE
CMD_MOVER_DISABLE
CMD_GROUP_CLEAR
CMD_GROUP_BUILD
CMD_GROUP_ENABLE
CMD_TRANSPORT_START
CMD_TRANSPORT_RESTART
<<enum>>
E_XTS_TRANSPORT_STATE
TRANSPORT_NULL
TRANSPORT_INVALID
TRANSPORT_INIT
TRANSPORT_IDLE
TRANSPORT_MOVER_ENABLE
TRANSPORT_MOVER_DISABLE
TRANSPORT_GROUP_CLEAR
TRANSPORT_GROUP_BUILD
TRANSPORT_GROUP_ENABLE
TRANSPORT_START
TRANSPORT_RESTART
<<enum>>
E_PROGRESS
PROGRESS_INVALID
PROGRESS_INIT
PROGRESS_BUSY
PROGRESS_PREPARE
PROGRESS_STARTUP
PROGRESS_CHECK
PROGRESS_OCCUPIED
PROGRESS_WORKING
PROGRESS_STILL_WORKING
PROGRESS_ERROR
PROGRESS_DONE
_eCmdOld
_eCmd
_eState
_eJobResult
_eResult
XtsTransport
fb_TransportUnit
_eCheck E_XTS_TRANSPORT_CHECK
_Ctrl REFERENCE TO ST_XTS_TRANSPORT_CTRL
_State REFERENCE TO ST_XTS_TRANSPORT_STATE
_eCmd E_XTS_TRANSPORT_CTRL
_eCmdOld E_XTS_TRANSPORT_CTRL
_eState E_XTS_TRANSPORT_STATE
_eJobResult E_PROGRESS
_eResult E_PROGRESS
_XpuCtrl REFERENCE TO ST_XPU_CTRL
_XpuState REFERENCE TO ST_XPU_STATE
_XpuInfo REFERENCE TO ST_XPU_INFO
_GroupItf I_XtsTransport_CaGroup
_GroupInfo REFERENCE TO ST_GROUP_INFO
_MoverItf REFERENCE TO ARRAY [1..MAX_MOVER] OF I_XtsTransport_Mover
_MoverInfo REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_MOVER_INFO
_StationStart REFERENCE TO ST_STATION_PARAMETER
_ItfStationList REFERENCE TO ARRAY [1..MAX_STATION] OF I_Station_LinkedList
_ItfStationCtrl REFERENCE TO ARRAY [1..MAX_STATION] OF I_XtsTransport_Station
_nStartStation UINT
_stMoverDataSend ST_STATION_MOVER_DATA
_stListResult ST_STATION_LIST_RESULT
Ctrl REFERENCE TO ST_XTS_TRANSPORT_CTRL {property}
GroupInfo REFERENCE TO ST_GROUP_INFO {property}
GroupItf I_XtsTransport_CaGroup {property}
MoverInfo REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_MOVER_INFO {property}
MoverItf REFERENCE TO ARRAY [1..MAX_MOVER] OF I_XtsTransport_Mover {property}
StartStationIndex UINT {property}
State REFERENCE TO ST_XTS_TRANSPORT_STATE {property}
StationCtrlItf REFERENCE TO ARRAY [1..MAX_STATION] OF I_XtsTransport_Station {property}
StationListsItf REFERENCE TO ARRAY [1..MAX_STATION] OF I_Station_LinkedList {property}
StationStart REFERENCE TO ST_STATION_PARAMETER {property}
XpuCtrl REFERENCE TO ST_XPU_CTRL {property}
XpuInfo REFERENCE TO ST_XPU_INFO {property}
XpuState REFERENCE TO ST_XPU_STATE {property}
Check() BOOL
Cmd(...) E_XTS_TRANSPORT_STATE
GroupBuild(Execute : BOOL) E_PROGRESS
GroupClear(Execute : BOOL) E_PROGRESS
GroupEnable(Execute : BOOL) E_PROGRESS
MoverDisableAll(Execute : BOOL) e_progress
MoverEnableAll(Execute : BOOL) e_progress
TransportRestart(Execute : BOOL) e_progress
TransportStart(Execute : BOOL) e_progress
XpuInit(Execute : BOOL) E_PROGRESS

3. Design

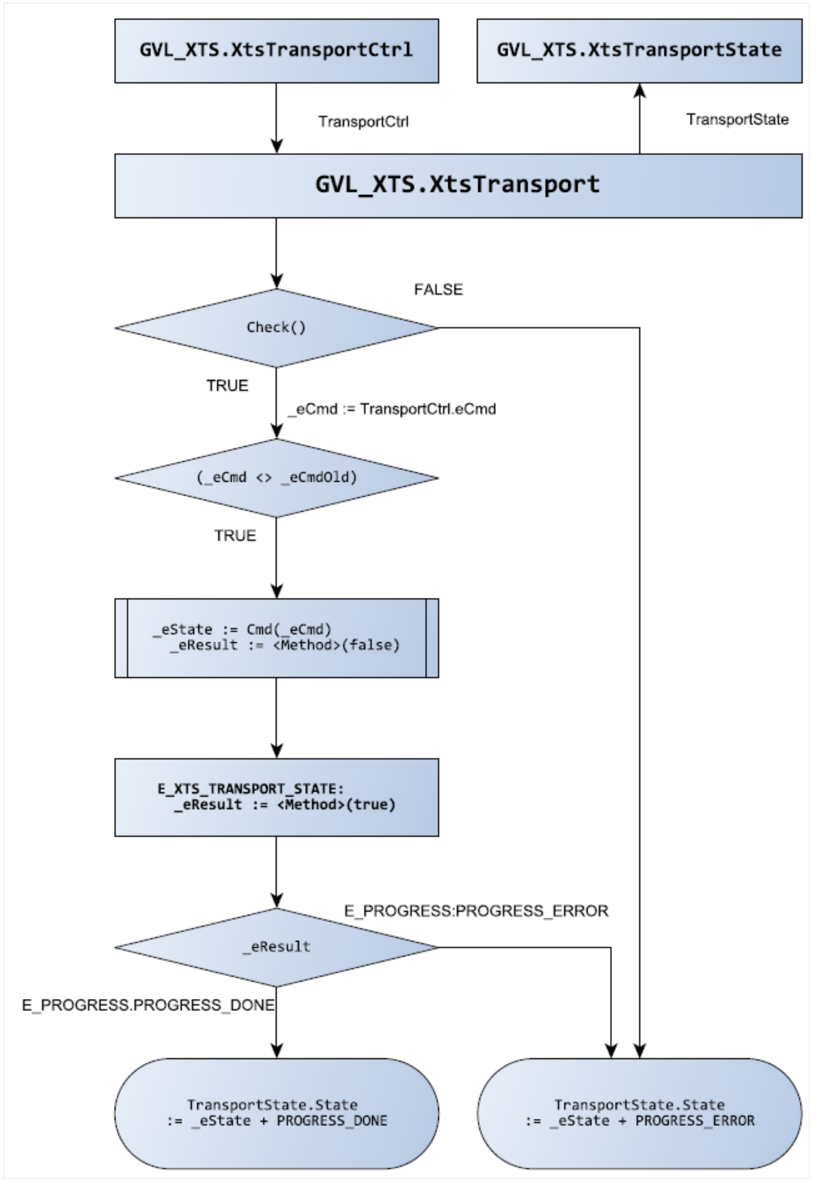
- TransportUnit
  - Fb_TransportUnit():
    - Members:



3. Design

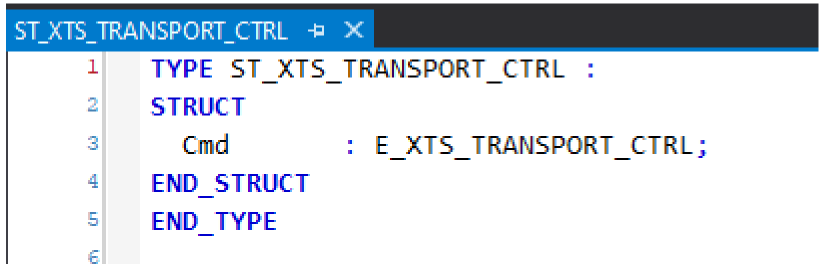
BECKHOFF

- TransportUnit
  - Fb_TransportUnit():
    - Change of command triggers execution
    - Execution result is added to state
    - Extern control needs to react to BUSY, DONE or ERROR

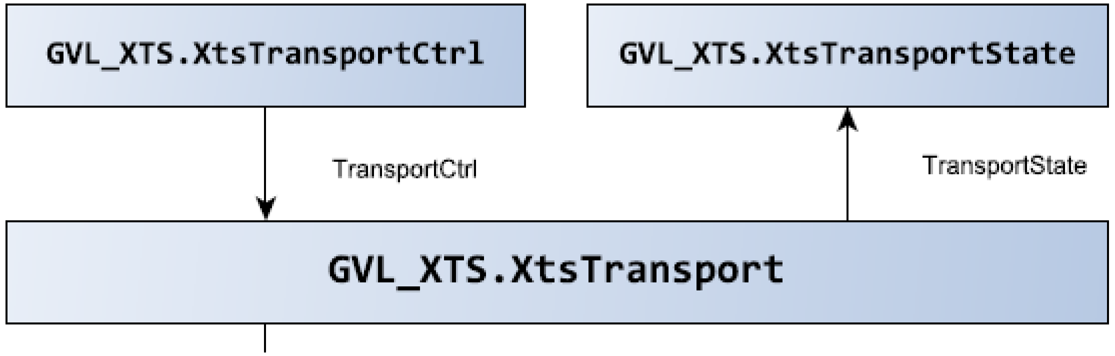

GVL_XTS.XtsTransportCtrl
GVL_XTS.XtsTransportState
TransportCtrl
TransportState
GVL_XTS.XtsTransport
FALSE
Check()
TRUE
_eCmd := TransportCtrl.eCmd
(_eCmd <> _eCmdOld)
TRUE
_eState := Cmd(_eCmd)
_eResult := <Method>(false)
E_XTS_TRANSPORT_STATE:
_eResult := <Method>(true)
E_PROGRESS:PROGRESS_ERROR
_eResult
E_PROGRESS.PROGRESS_DONE
TransportState.State := _eState + PROGRESS_DONE
TransportState.State := _eState + PROGRESS_ERROR

3. Design

- TransportUnit
  - GVL_XTS.XtsTransportCtrl: ST_TRANSPORT_UNIT_CTRL
    - Struct for commanding FB_TransportUnit
    - eCmd : E_XTS_TRANSPORT_CTRL

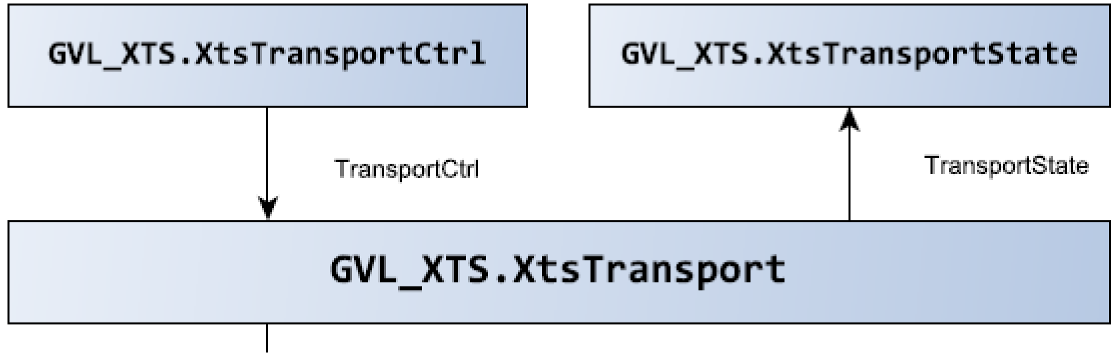

ST_XTS_TRANSPORT_CTRL

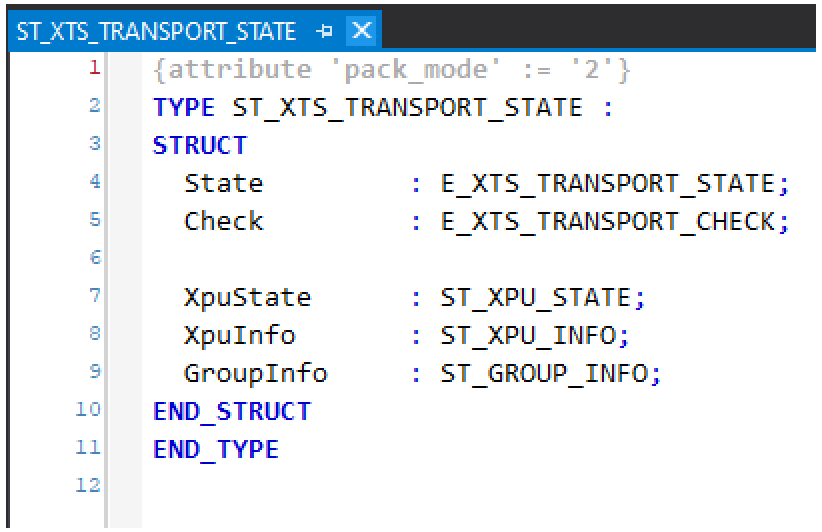
```
TYPE ST_XTS_TRANSPORT_CTRL :
STRUCT
  Cmd        : E_XTS_TRANSPORT_CTRL;
END_STRUCT
END_TYPE
```



3. Design

- TransportUnit
  - GVL_XTS.XtsTransportState: ST_TRANSPORT_UNIT_STATE
  - State: combines active command and result
  - Check: cyclic pointer checks
  - XpuState: state from fb_Xpu
  - XpuInfo: cyclic plausibility checks to TcCOM Objects
  - GroupInfo: cyclic information from FB_CaGroup



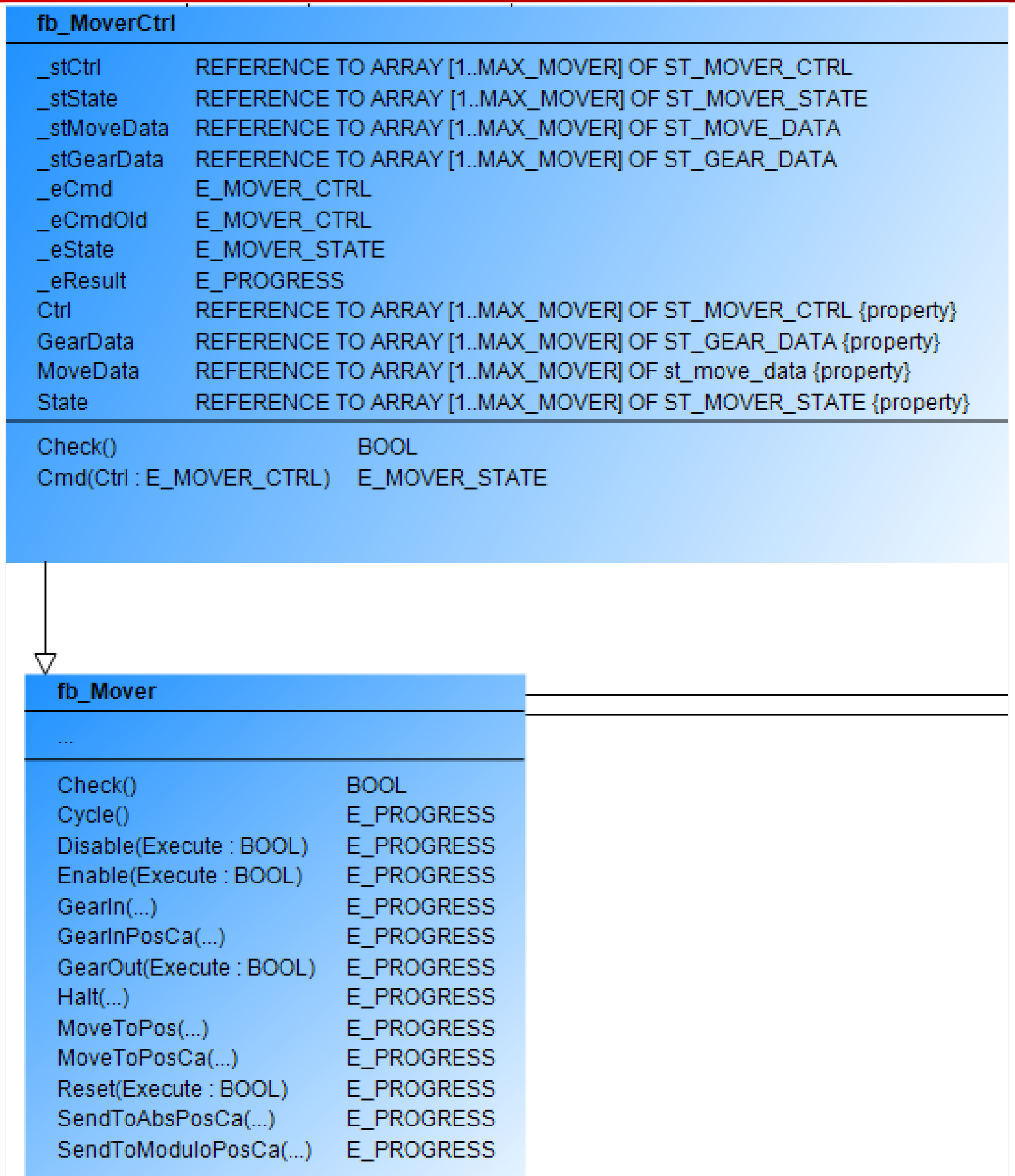


```
{attribute 'pack_mode' := '2'}
TYPE ST_XTS_TRANSPORT_STATE :
STRUCT
  State          : E_XTS_TRANSPORT_STATE;
  Check          : E_XTS_TRANSPORT_CHECK;

  XpuState       : ST_XPU_STATE;
  XpuInfo        : ST_XPU_INFO;
  GroupInfo      : ST_GROUP_INFO;
END_STRUCT
END_TYPE

```

3. Design

BECKHOFF

- fb_MoverCtrl:
  - Inherits fb_Mover
    - Access to MC function blocks in library
    - Implements Interface for use in other classes
  - Contains cyclic interface
    - Ctrl datafield for setting commands
    - State data field for checking responses
    - Parameter datafields for using motion functions



fb_MoverCtrl

_stCtrl	REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_MOVER_CTRL
_stState	REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_MOVER_STATE
_stMoveData	REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_MOVE_DATA
_stGearData	REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_GEAR_DATA
_eCmd	E_MOVER_CTRL
_eCmdOld	E_MOVER_CTRL
_eState	E_MOVER_STATE
_eResult	E_PROGRESS
Ctrl	REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_MOVER_CTRL {property}
GearData	REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_GEAR_DATA {property}
MoveData	REFERENCE TO ARRAY [1..MAX_MOVER] OF st_move_data {property}
State	REFERENCE TO ARRAY [1..MAX_MOVER] OF ST_MOVER_STATE {property}

Check()	BOOL
Cmd(Ctrl : E_MOVER_CTRL)	E_MOVER_STATE

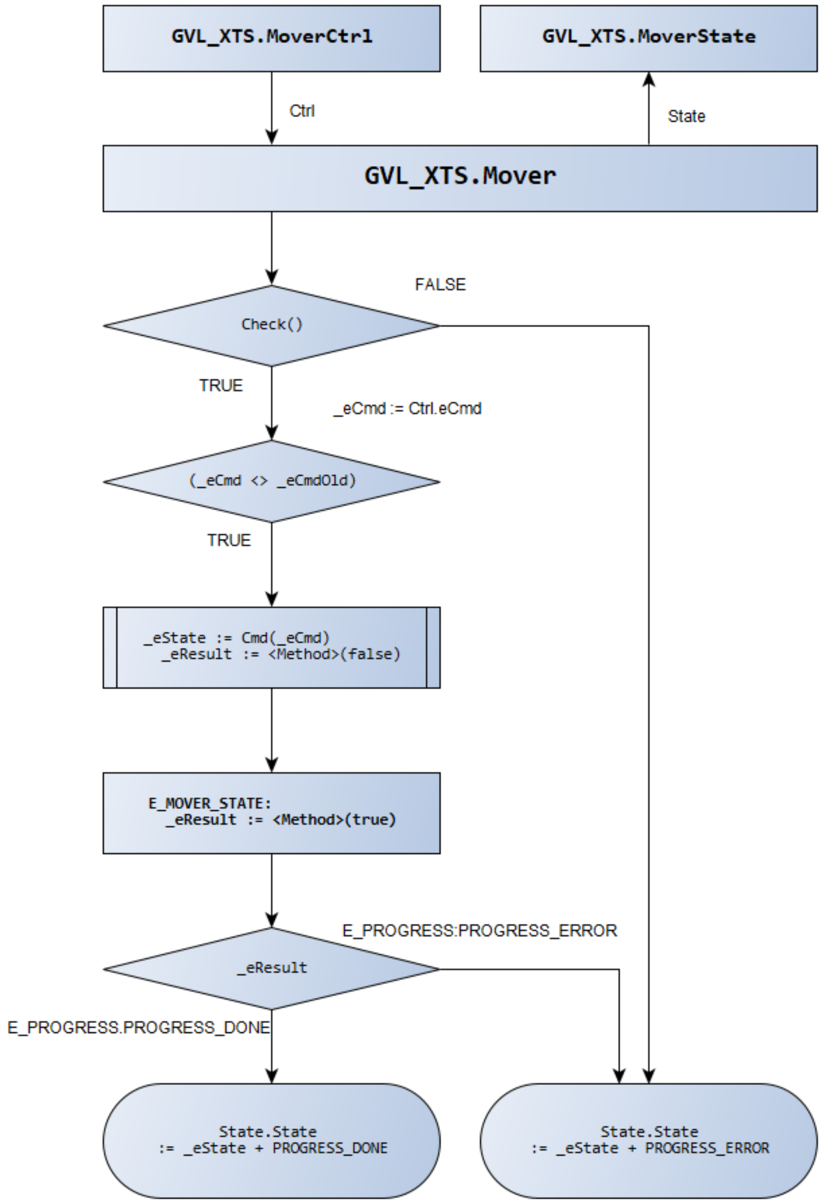
fb_Mover

...

Check()	BOOL
Cycle()	E_PROGRESS
Disable(Execute : BOOL)	E_PROGRESS
Enable(Execute : BOOL)	E_PROGRESS
GearIn(...)	E_PROGRESS
GearInPosCa(...)	E_PROGRESS
GearOut(Execute : BOOL)	E_PROGRESS
Halt(...)	E_PROGRESS
MoveToPos(...)	E_PROGRESS
MoveToPosCa(...)	E_PROGRESS
Reset(Execute : BOOL)	E_PROGRESS
SendToAbsPosCa(...)	E_PROGRESS
SendToModuloPosCa(...)	E_PROGRESS

3. Design

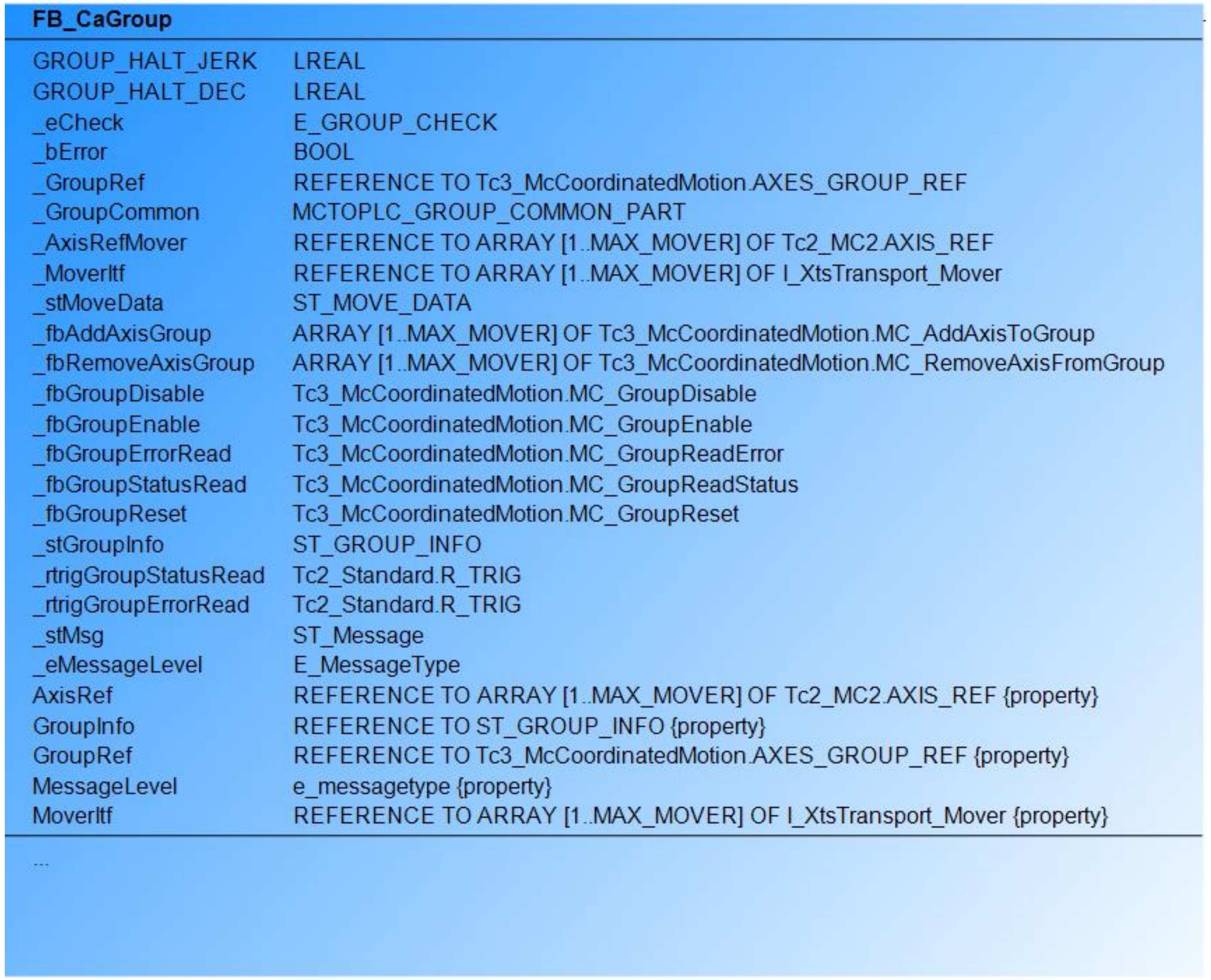
- fb_MoverCtrl:
  - Mover index is passed as value from caller
  - Global datafields are passed as references (REF=) into fb_MoverCtrl properties
    - Ctrl / State: handshakes
    - standard return value for method
    - Log LastPosition on CA/MC function execute
    - Log LastGap on CA function execute



3. Design

BECKHOFF

- fb_CaGroup:
  - Collision Avoidance class wrapper
  - Implements _I_Transport_CaGroup
  - Cyclic information from AXES_GROUP_REF
  - Mover commands throug interface I_XtsTransport_Mover



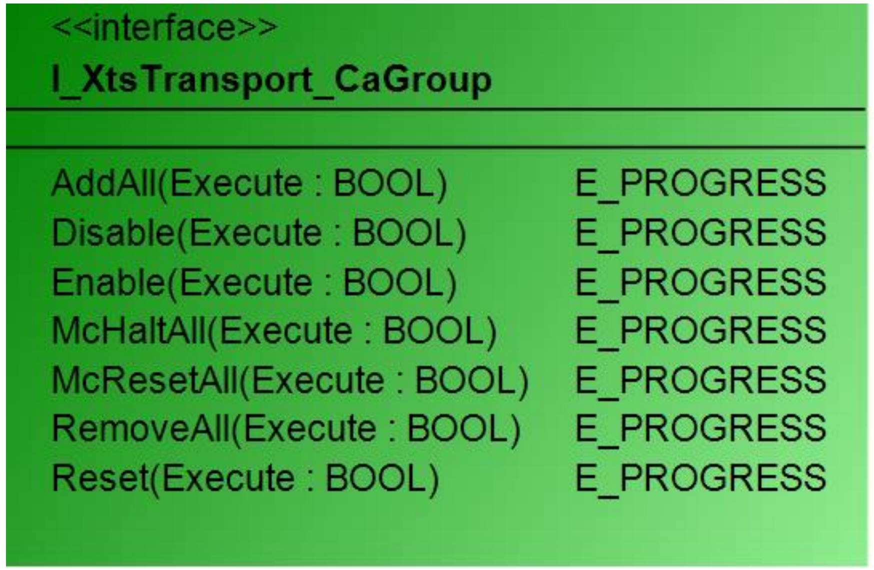
FB_CaGroup

GROUP_HALT_JERK	LREAL
GROUP_HALT_DEC	LREAL
_eCheck	E_GROUP_CHECK
_bError	BOOL
_GroupRef	REFERENCE TO Tc3_McCoordinatedMotion.AXES_GROUP_REF
_GroupCommon	MCTOPLC_GROUP_COMMON_PART
_AxisRefMover	REFERENCE TO ARRAY [1..MAX_MOVER] OF Tc2_MC2.AXIS_REF
_MoverItf	REFERENCE TO ARRAY [1..MAX_MOVER] OF I_XtsTransport_Mover
_stMoveData	ST_MOVE_DATA
_fbAddAxisGroup	ARRAY [1..MAX_MOVER] OF Tc3_McCoordinatedMotion.MC_AddAxisToGroup
_fbRemoveAxisGroup	ARRAY [1..MAX_MOVER] OF Tc3_McCoordinatedMotion.MC_RemoveAxisFromGroup
_fbGroupDisable	Tc3_McCoordinatedMotion.MC_GroupDisable
_fbGroupEnable	Tc3_McCoordinatedMotion.MC_GroupEnable
_fbGroupErrorRead	Tc3_McCoordinatedMotion.MC_GroupReadError
_fbGroupStatusRead	Tc3_McCoordinatedMotion.MC_GroupReadStatus
_fbGroupReset	Tc3_McCoordinatedMotion.MC_GroupReset
_stGroupInfo	ST_GROUP_INFO
_rtrigGroupStatusRead	Tc2_Standard.R_TRIG
_rtrigGroupErrorRead	Tc2_Standard.R_TRIG
_stMsg	ST_Message
_eMessageLevel	E_MessageType
AxisRef	REFERENCE TO ARRAY [1..MAX_MOVER] OF Tc2_MC2.AXIS_REF {property}
GroupInfo	REFERENCE TO ST_GROUP_INFO {property}
GroupRef	REFERENCE TO Tc3_McCoordinatedMotion.AXES_GROUP_REF {property}
MessageLevel	e_messagetype {property}
MoverItf	REFERENCE TO ARRAY [1..MAX_MOVER] OF I_XtsTransport_Mover {property}

...

3. Design

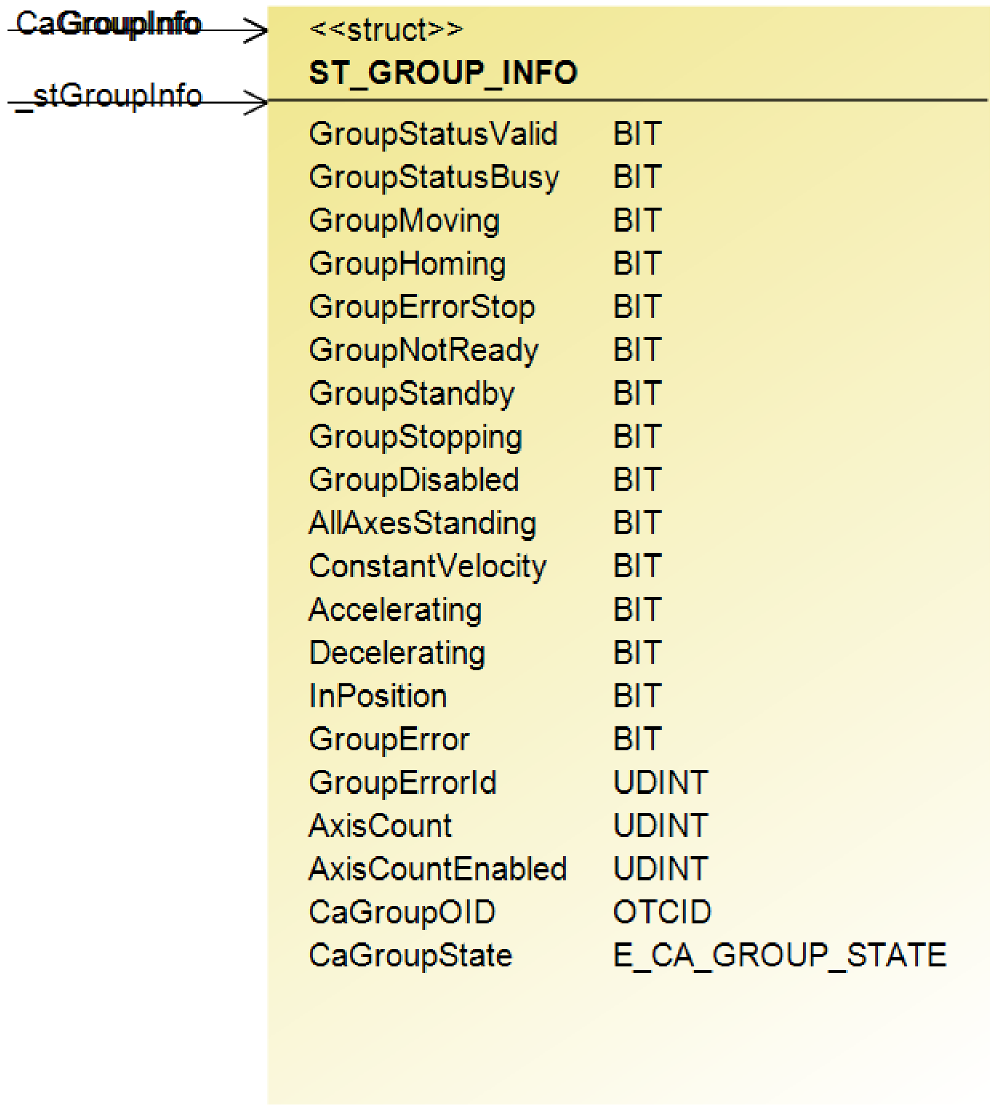
- fb_CaGroup:
  - Implements _I_Transport_CaGroup


<<interface>>
I_XtsTransport_CaGroup

AddAll(Execute : BOOL)	E_PROGRESS
Disable(Execute : BOOL)	E_PROGRESS
Enable(Execute : BOOL)	E_PROGRESS
McHaltAll(Execute : BOOL)	E_PROGRESS
McResetAll(Execute : BOOL)	E_PROGRESS
RemoveAll(Execute : BOOL)	E_PROGRESS
Reset(Execute : BOOL)	E_PROGRESS

3. Design

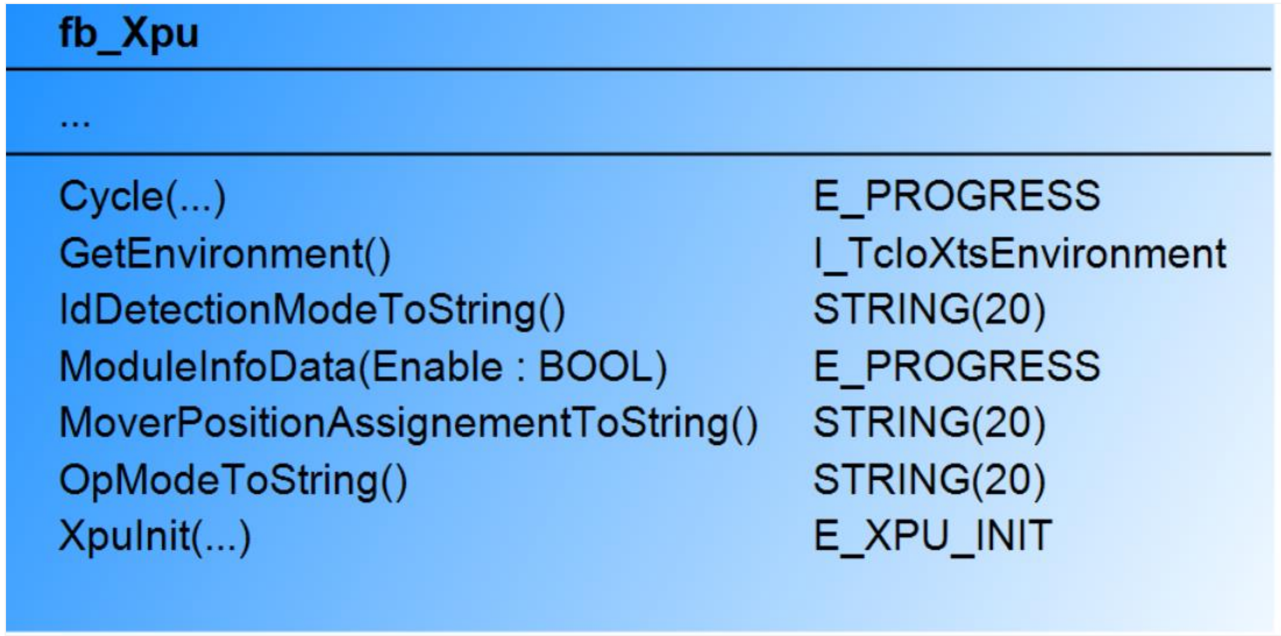
- fb_CaGroup:
  - Cyclic information to ST_GROUP_INFO



GroupStatusValid	BIT
GroupStatusBusy	BIT
GroupMoving	BIT
GroupHoming	BIT
GroupErrorStop	BIT
GroupNotReady	BIT
GroupStandby	BIT
GroupStopping	BIT
GroupDisabled	BIT
AllAxesStanding	BIT
ConstantVelocity	BIT
Accelerating	BIT
Decelerating	BIT
InPosition	BIT
GroupError	BIT
GroupErrorId	UDINT
AxisCount	UDINT
AxisCountEnabled	UDINT
CaGroupOID	OTCID
CaGroupState	E_CA_GROUP_STATE

3. Design

- fb_Xpu:
  - Class for interacting with XTS ProcessingUnit
  - XpuInit()
    - Connects to OTCIDs of XTS TcCOM Objects



fb_Xpu

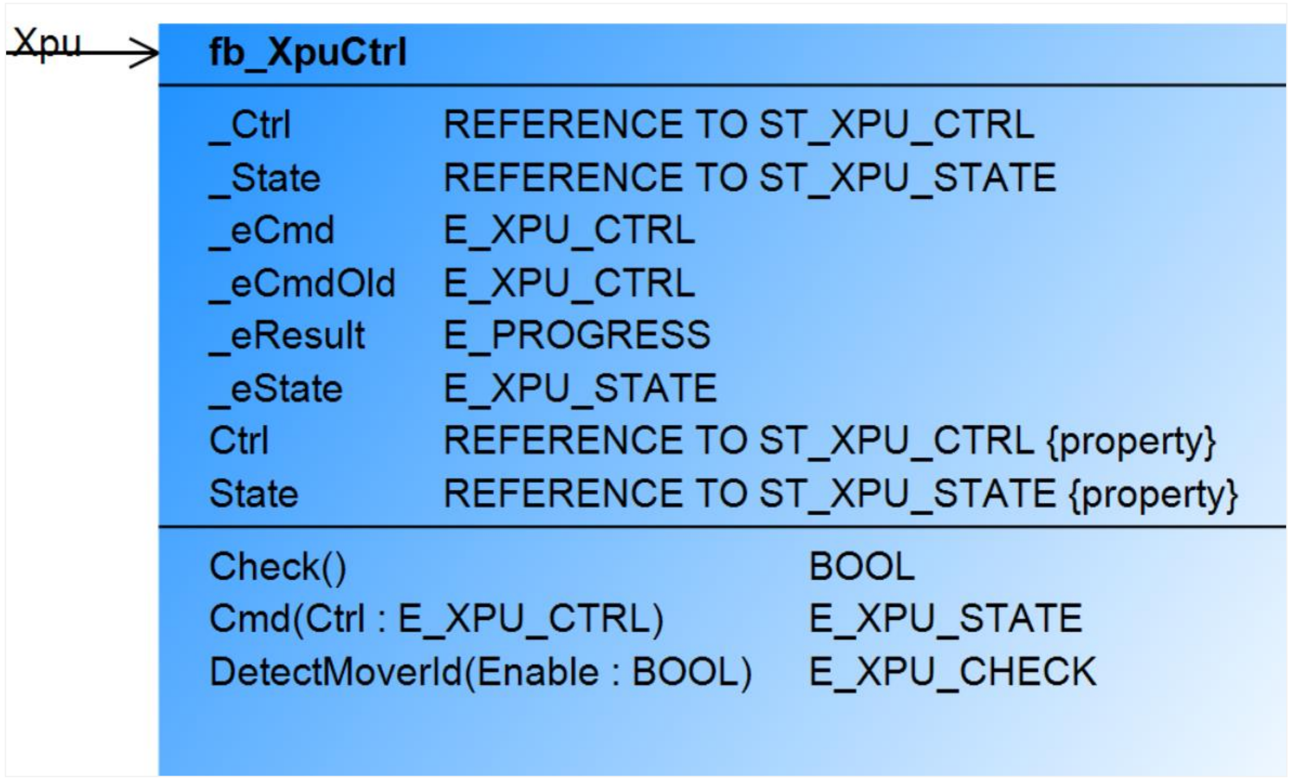
...

Cycle(...)	E_PROGRESS
GetEnvironment()	I_TcIoXtsEnvironment
IdDetectionModeToString()	STRING(20)
ModuleInfoData(Enable : BOOL)	E_PROGRESS
MoverPositionAssignementToString()	STRING(20)
OpModeToString()	STRING(20)
XpuInit(...)	E_XPU_INIT

3. Design

BECKHOFF

- fb_XpuCtrl:
  - Cyclic check for command change
  - Wraps cyclic execution of fb_Xpu



_Ctrl	REFERENCE TO ST_XPU_CTRL
_State	REFERENCE TO ST_XPU_STATE
_eCmd	E_XPU_CTRL
_eCmdOld	E_XPU_CTRL
_eResult	E_PROGRESS
_eState	E_XPU_STATE
Ctrl	REFERENCE TO ST_XPU_CTRL {property}
State	REFERENCE TO ST_XPU_STATE {property}
Check()	BOOL
Cmd(Ctrl : E_XPU_CTRL)	E_XPU_STATE
DetectMoverId(Enable : BOOL)	E_XPU_CHECK

4. LICENSE

BECKHOFF

XTS_TRANSPORT_LAYER project