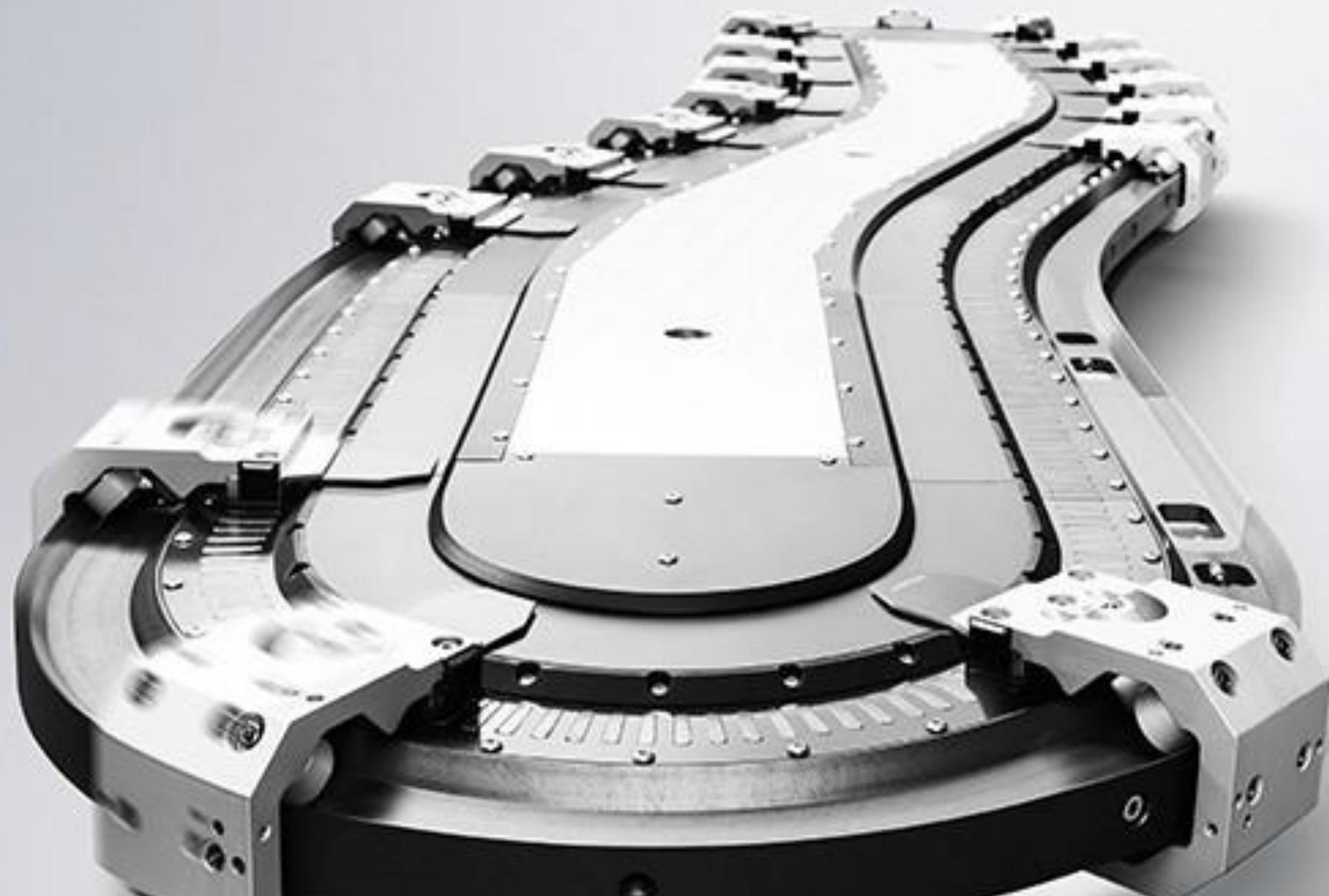


XTS TRANSPORT LAYER – Control and State structures for mapping

BECKHOFF

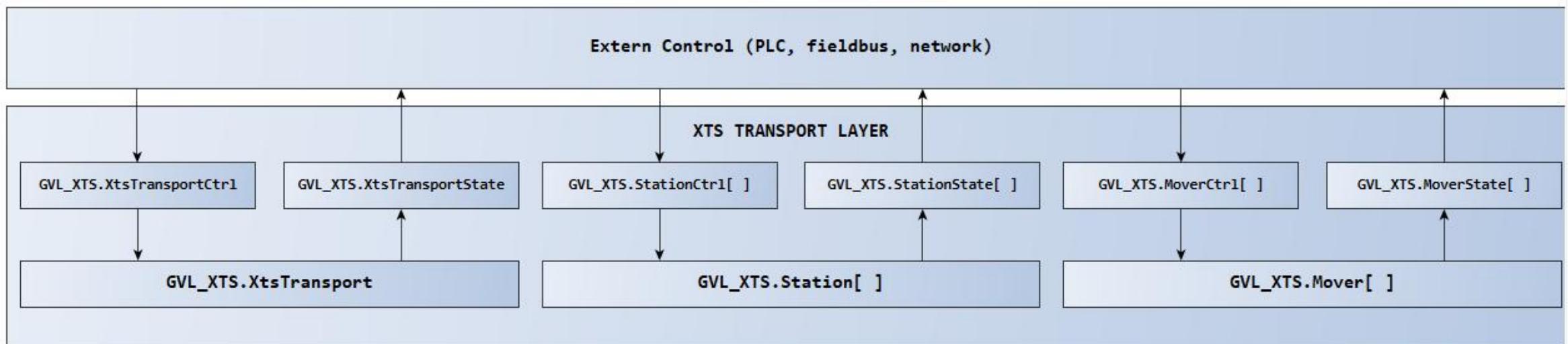


- 1. Introduction
- 2. Control / State Members
- 3. Implementation
 - 1. Profinet
 - 2. EAP



- XTS may work as subsystem in heterogen networks
 - Cyclic fieldbus networks
 - NonCyclic TCP/IP networks
- TwinCAT offers different protocols, the most common (used in this context) are:
 - EtherCAT
 - Profinet
 - EAP
- XTS_TRANSPORT_LAYER must be modified to match your use case
 - Size of mapping must be taken into consideration
 - Cycle time of mapping may be different to PLC cycle time

- Top Level structures:



- **XtsTransport structures:**

- ST_XTS_TRANSPORT_CTRL
used for getting XTS_TRANSPORT_LAYER
to a defined state.

```
E_XTS_TRANSPORT_CTRL # X
1 {attribute 'strict'}
2 {attribute 'to_string'}
3 TYPE E_XTS_TRANSPORT_CTRL :
4 (
5     CMD_NULL,
6     CMD_INIT          := 10,
7     CMD_IDLE,
8
9     CMD_MOVER_ENABLE   := 20,
10    CMD_MOVER_DISABLE,
11    CMD_MOVER_HALT_CA,
12    CMD_MOVER_STOP,
13
14    CMD_GROUP_CLEAR      := 30,
15    CMD_GROUP_BUILD,
16    CMD_GROUP_ENABLE,
17    CMD_GROUP_STOP,
18
19    CMD_TRANSPORT_START    := 40,
20    CMD_TRANSPORT_RESTART
21
22 )UINT;
23 END_TYPE
```

```
ST_XTS_TRANSPORT_CTRL # X
1 {attribute 'pack_mode' := '2'}
2 TYPE ST_XTS_TRANSPORT_CTRL :
3 STRUCT
4     Cmd      : E_XTS_TRANSPORT_CTRL;
5 END_STRUCT
6 END_TYPE
7
```

- **XtsTransport structures:**

- ST_XTS_TRANSPORT_STATE
- State delivers information about current Command, XPU and CAGroup.
- State feedback is a combined value of the active command and E_PROGRESS
- Check feedback for internal checks
- XpuState feedback is a combined value of the current xpu command and E_PROGRESS
- See “fb_TransportUnit.pdf” in doc folder of this project.

```
ST_XTS_TRANSPORT_STATE {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
```

- XtsTransport structures:

- E_XTS_TRANSPORT_STATE:
 - Enum feedback for ctrl
 - Combined with E_PROGRESS

```
E_PROGRESS + x
1 {attribute 'qualified_only'}
2 //{@attribute 'strict'}
3 {attribute 'to_string'}
4 TYPE E_PROGRESS :
5 (
6     // progress is used in project to
7     // mirror state of requested command/function
8     PROGRESS_INVALID,
9     PROGRESS_NOT_EXIST      := 100,
10    PROGRESS_INIT           := 1000,
11    PROGRESS_BUSY            := 2000,
12    PROGRESS_PREPARE         := 3000,
13    PROGRESS_STARTUP         := 4000,
14    PROGRESS_CHECK           := 5000,
15    PROGRESS_OCCUPIED        := 6000,
16    PROGRESS_WORKING         := 7000,
17    PROGRESS_STILL_WORKING   := 8000,
18    PROGRESS_ERROR           := 9000,
19    PROGRESS_DONE             := 10000
20 )UINT;
21 END_TYPE
```

```
E_XTS_TRANSPORT_STATE + x
1 {attribute 'to_string'}
2 TYPE E_XTS_TRANSPORT_STATE :
3 (
4     TRANSPORT_NULL,
5     TRANSPORT_INVALID,
6     TRANSPORT_INIT          := 10,
7     TRANSPORT_IDLE,
8
9     TRANSPORT_MOVER_ENABLE   := 20,
10    TRANSPORT_MOVER_DISABLE,
11    TRANSPORT_MOVER_HALT_CA,
12    TRANSPORT_MOVER_STOP,
13
14    TRANSPORT_GROUP_CLEAR    := 30,
15    TRANSPORT_GROUP_BUILD,
16    TRANSPORT_GROUP_ENABLE,
17    TRANSPORT_GROUP_STOP,
18
19    TRANSPORT_START          := 40,
20    TRANSPORT_RESTART
21 )UINT;
22 END_TYPE
23
24
```

- **XtsTransport structures:**

- E_XTS_TRANSPORT_CHECK:
 - Enum feedback for pointer checks
 - If no pointer error is detected, Check is set to DONE_CHECK.

```
E_XTS_TRANSPORT_CHECK # X
1 {attribute 'qualified_only'}
2 {attribute 'to_string'}
3 TYPE E_XTS_TRANSPORT_CHECK :
4 (
5     // info for cyclic checks
6     INIT_CHECK,
7     START_CHECK,    // check cycle start|  
XPU_INIT,        // Xpu: initialization
8
9
10    // errors in fb_TransportUnit.Check()
11    POINTER_CHECK           := 900,
12    POINTER_NULL_CTRL,
13
14    POINTER_NULL_STATE,
15    POINTER_NULL_XPU_CTRL,
16
17    POINTER_NULL_XPU_STATE,
18    POINTER_NULL_XPU_INFO,
19
20    POINTER_NULL_GROUP_ITF,
21    POINTER_NULL_GROUP_INFO,
22
23    POINTER_NULL_MOVER,
24    POINTER_NULL_MOVER_ITF,
25
26    POINTER_NULL_MOVER_INFO,
27    POINTER_NULL_MOVER_LAST_POS,
28    POINTER_NULL_MOVER_LAST_GAP,
29
30    POINTER_NULL_STATION_START,
31    POINTER_NULL_STATION_ITF,
32    POINTER_NULL_STATION_CONTROL,
33    POINTER_NULL_STATION_STATE,
34    POINTER_NULL_STATION_CTRL_ITF,
35
36    DONE_CHECK           := 10000
37 )UINT;
38 END_TYPE
39
40
```

- XtsTransport structures:

- ST_XPU_STATE:
 - Initialization state, (should be checked on startup)
 - State feedback combined with E_PROGRESS
 - Check feedback for cyclic plausibility checks
 - DcLink 48V feedback from **all** motor modules.

```
ST_XPU_STATE {attribute 'pack_mode' := '2'}  
TYPE ST_XPU_STATE :  
STRUCT  
    Init          : E_XPU_INIT;  
    State         : E_XPU_STATE;  
    Check         : E_XPU_CHECK;  
    DcLink        : BIT;  
END_STRUCT  
END_TYPE
```

- **XtsTransport structures:**

- **E_XPU_INIT:**

- Initialization state, (should be checked on startup)

```
E_XPU_INIT += x
1 {attribute 'to_string'}
2 TYPE E_XPU_INIT :
3 (
4     INIT_START,
5     INIT_ENVIRONMENT_OID           := 10,
6     INIT_INFO_SERVER_ITF,
7     INIT_INFO_SERVER_STATION_COUNT,
8     INIT_INFO_SERVER_STATION_OID,
9     INIT_CA_GROUP_OID              := 20,
10    INIT_CA_GROUP_GET_OID,
11    INIT_CA_GROUP_CHECK_OID,
12    INIT_CA_GROUP_COM,
13    INIT_PARAMETERSET_COUNT,
14    INIT_PARAMETERSET_OID,
15    INIT_PARAMETERSET_COM,
16    INIT_PARAMETERSET_COM_NEXT,
17    INIT_PROCESSING_UNIT_COM       := 30,
18    INIT_PROCESSING_UNIT_PART_COUNT,
19    INIT_PROCESSING_UNIT_PART_OID,
20    INIT_PROCESSING_UNIT_TRACK_COUNT,
21    INIT_PROCESSING_UNIT_TRACK_OID,
22    INIT_PROCESSING_UNIT_MOVER_COUNT,
23    INIT_PROCESSING_UNIT_MOVER_OID,
24    INIT_PROCESSING_UNIT_TASK_COUNT,
25    INIT_PROCESSING_UNIT_TASK_OID,
26    INIT_TRACK_COM                := 50,
27    INIT_PART_COM                 := 60,
28    INIT_PART_MODULE_COUNT,
29    INIT_PART_MODULE_OID,
30    INIT_PART_MODULE_COM,
31    INIT_PART_MODULE_COM_NEXT,
32    INIT_MOVER_COM                := 70,
33    INIT_MOVER_COM_NEXT,
34    INIT_NCT_CONTROLLER_OID        := 80,
35    INIT_NCT_CONTROLLER_COM,
36    INIT_NCT_BASE_UNIT_COUNT,
37    INIT_NCT_BASE_UNIT_ITF,
38    INIT_NCT_BASE_UNIT_NEXT,
39    INIT_DATA_GET                  := 90,
40    INIT_DATA_CHECK,
41    INIT_DONE                      := 100
42 )UINT;
43 END_TYPE
```

- XtsTransport structures:

- E_XPU_STATE:
 - State feedback combined with E_PROGRESS

```
E_PROGRESS + X
1 {attribute 'qualified_only'}
2 //{attribute 'strict'}
3 {attribute 'to_string'}
4 TYPE E_PROGRESS :
5 (
6     // progress is used in project to
7     // mirror state of requested command/function
8     PROGRESS_INVALID,
9     PROGRESS_NOT_EXIST      := 100,
10    PROGRESS_INIT           := 1000,
11    PROGRESS_BUSY            := 2000,
12    PROGRESS_PREPARE         := 3000,
13    PROGRESS_STARTUP         := 4000,
14    PROGRESS_CHECK           := 5000,
15    PROGRESS_OCCUPIED        := 6000,
16    PROGRESS_WORKING         := 7000,
17    PROGRESS_STILL_WORKING   := 8000,
18    PROGRESS_ERROR           := 9000,
19    PROGRESS_DONE             := 10000
20 )UINT;
21 END_TYPE
```

```
E_XPU_STATE + X
1 {attribute 'to_string'}
2 TYPE E_XPU_STATE :
3 (
4     XTS_NULL,
5     XTS_INVALID,
6     XTS_INIT           := 10,
7     XTS_STATION_INFO_READ := 20,
8     XTS_IDLE            := 30
9 )UINT;
10 END_TYPE
11
12
```

- **XtsTransport structures:**

- **E_XPU_CHECK:**

- feedback for cyclic plausibility checks to the XTS Processing Unit

```
E_XPU_CHECK # x
1  [attribute 'qualified_only']
2  [attribute 'to_string']
3  TYPE E_XPU_CHECK :
4  (
5      INIT_CHECK,
6      INIT_CHECK_ERROR,
7      START_CHECK,
8      XPU_INSTANCE_NULL,
9      XPU_RAIL_LENGTH,
10     GROUP_RAIL_LENGTH,
11     RAIL_LENGTH_COMPARE,
12     GROUP_NOT_CONFIGURED,
13     GROUP_OID_MISMATCH,
14
15     MOVER_COUNT           := 10,
16     MOVER_COUNT_ZERO,
17     MOVER_COUNT_NOT_EQUAL,
18
19     PROCESSING_UNIT_POSITIONS_VALID      := 20,
20     PROCESSING_UNIT_GET_DATA_EXCEED,
21
22     MOVER_ID_STANDARD           := 30,
23     MOVER_ID_SIMULATION,
24     MOVER_ID_DETECTION_MODE,
25     MOVER_ID_DETECTION_VALID,
26     MOVER_ID_DETECTION_DC_LINK,
27     MOVER_ID_DETECTION_BUSY,
28     MOVER_ID_DETECTION_IDLE,
29     MOVER_ID_DETECTION_STATE,
30     MOVER_ID_MULTIPLE_NOT_SUPPORTED,
31
32     POINTER_CHECK           := 90,
33     POINTER_NULL_CTRL,
34     POINTER_NULL_STATE
35 )UINT;
36 END_TYPE
37
```

- **XtsTransport structures:**

- ST_XPU_INFO:
 - Data of cyclic checks to XTS ProcessingUnit

```
ST_XPU_INFO + x
1 {attribute 'pack_mode' := '2'}
2 TYPE ST_XPU_INFO :
3 STRUCT
4     AllPositionsValid      : BIT;
5     IdDetectionError      : BIT;
6     IdDetectionValid      : BIT;
7     IdDetectionActive     : BIT;
8
9     OperationMode         : UINT;
10
11    IdDetectionMode       : UINT;
12    MoverPositionAssignment : UINT;
13
14    nDetectedAxisCount   : UINT;
15    nExpectedAxisCount   : UINT;
16
17    nParameterSetCount   : UINT;
18
19 END_STRUCT
20 END_TYPE
21
```

- **XtsTransport structures:**

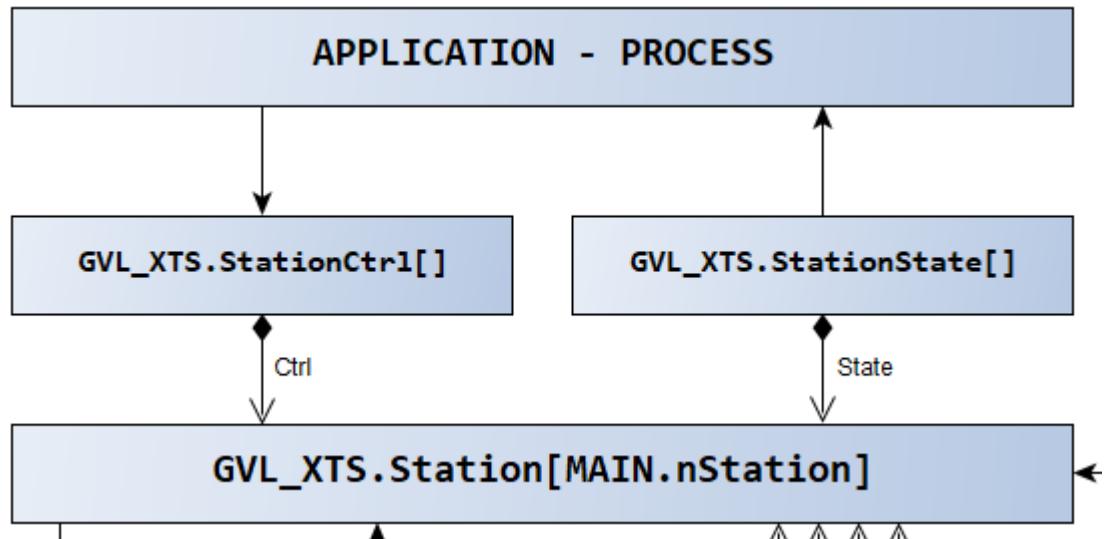
- ST_GROUP_INFO:

- Data of cyclic checks to Collision Avoidance Group (CAGroup)

```
ST_GROUP_INFO ▾ X
1  TYPE ST_GROUP_INFO : 
2  STRUCT
3      GroupStatusValid,
4      GroupStatusBusy,
5      GroupMoving,
6      GroupHoming,
7      GroupErrorStop,
8      GroupNotReady,
9      GroupStandby,
10     GroupStopping,
11
12     GroupDisabled,
13     AllAxesStanding,
14     ConstantVelocity,
15     Accelerating,
16     Decelerating,
17     InPosition,
18     GroupError          : BIT;
19
20     GroupErrorId        : UDINT;
21
22     AxisCount           : UDINT;
23     AxisCountEnabled    : UDINT;
24
25     {attribute 'displaymode' := 'hex'}
26     CaGroupOID          : OTCID;
27
28     CaGroupState         : E_CA_GROUP_STATE;
29
30 END_STRUCT
31 END_TYPE
```

- **XtsStation structures:**

- ST_STATION_CTRL:
 - Command for handshaking mover transport
 - Command parameter for sending mover to a target station.
- ST_STATION_STATE:
 - State for handshaking mover transport
 - State details about current mover



- **XtsStation structures:**

- ST_STATION_CTRL:
 - E_STATION_CTRL:
 - Command for handshaking mover transport in station
 - Command parameter for sending mover to a target station.
 - nMask sets index of PosStop[] in target
 - nTargetStation sets index of target station
 - rOffset sets optional mover offset in target station

```
ST_STATION_CTRL  + X
1 {attribute 'pack_mode' := '2'}
2 TYPE ST_STATION_CTRL :
3 STRUCT
4   eCmd          : E_STATION_CTRL;
5   {attribute 'displaymode' := 'bin'}
6   nMask         : BYTE;
7   nTargetStation : USINT;
8   rOffset        : REAL;
9
10 END_STRUCT
11 END_TYPE
12
```

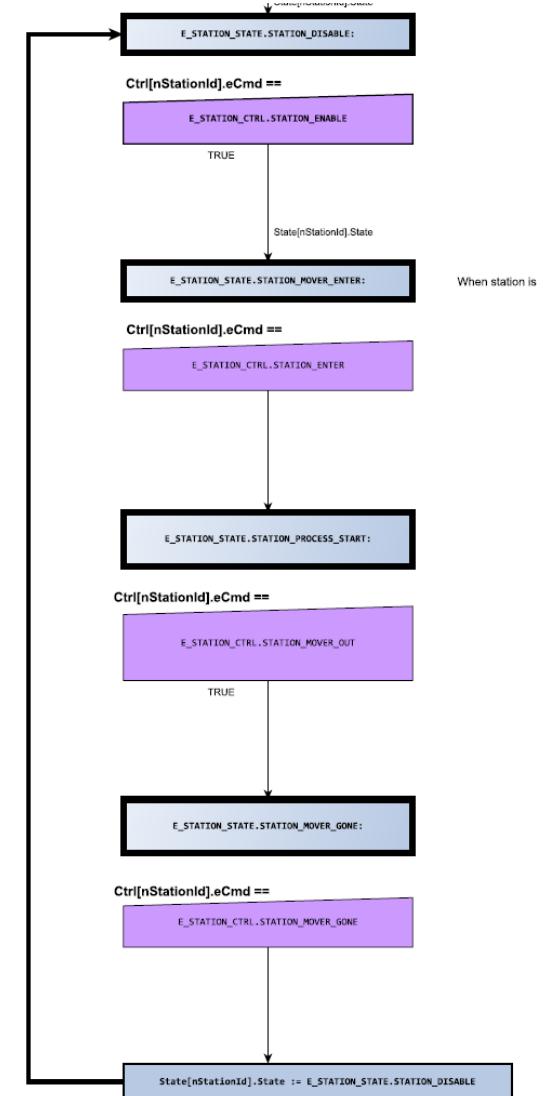
- **XtsStation structures:**

- ST_STATION_STATE:
 - E_STATION_STATE:
 - State for handshaking mover transport in station
 - nMask:
 - In case of ConfiguredStopCount > 1 nMask shows current index of PosStop[] in station
 - nMoverId:
 - Active mover in station
 - rMoverModPos:
 - Modulo position of mover in station
 - nQueue
 - Count of movers which were sent to station

```
ST_STATION_STATE • X
1 {attribute 'pack_mode' := '2'}
2 TYPE ST_STATION_STATE :
3 STRUCT
4     eState          : E_STATION_STATE;
5     {attribute 'displaymode' := 'bin'}
6     nMask           : BYTE;
7     nMoverId        : USINT;
8     rMoverModPos    : LREAL;
9     nQueue          : USINT;
10 END_STRUCT
11 END_TYPE
12
```

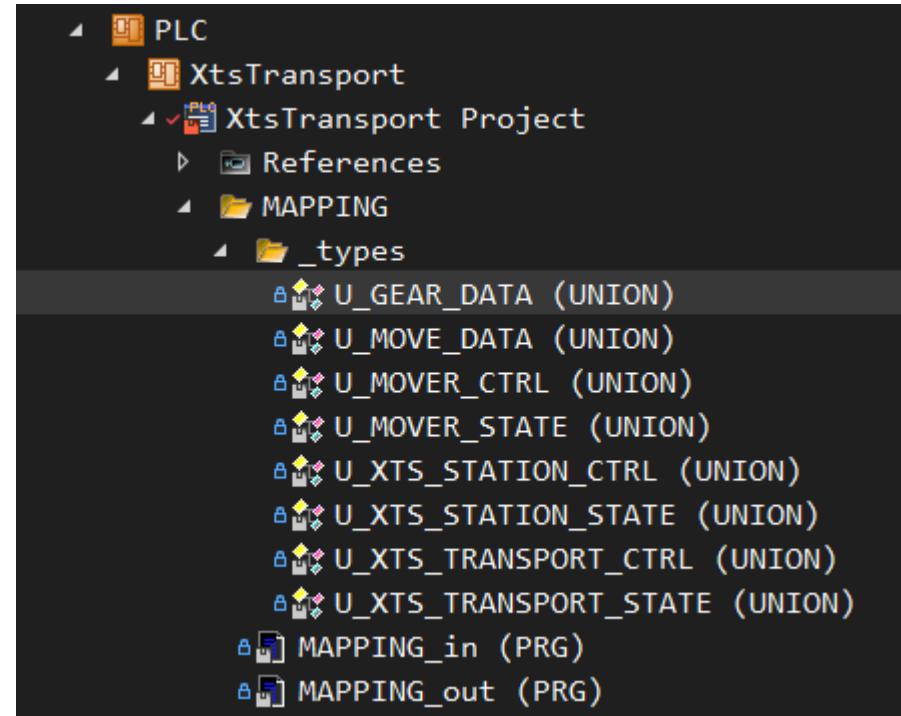
- **XtsStation structures:**

- E_STATION_CTRL / E_STATION_STATE
 - Handshake procedures you'll find in example pdfs in the doc folder of this project.



- **Mapping structures:**

- YOU can decide which structures you need for your application
- Use of UNIONs in TwinCAT
- Dedicated PRGs for mapping
 - Mapping_In
 - Mapping_Out
- All interfacing structures → pack mode := 2
- All fieldbus mappings as array of byte



■ Profinet:

- **EL6631-0010 Profinet Decive terminal**
 - See Beckhoff Infosys for documentation.
 - <https://infosys.beckhoff.com/index.php>

Technical data	Supplement
PROFINET Version	RT Class 1 ConformanceClassB
Number of device interfaces) ¹	8
Topology	variable
Quantity of user data	per device, maximum one Ethernet frame length) ² 1500 bytes of user data, inc. IOPS and IOCS
Cycle time	≥ 1 ms

)¹ see the chapter on virtual PROFINET devices

)² Depending on the cycle time, the PROFINET cycle time, and the CPU being used

Conditions required for operation

The following points must be observed when using the PROFINET supplement:



- Only Ethernet cards with Intel chipset allowed.
- RealTime Ethernet driver must be installed.
- No other RealTime protocols must be connected through this interface.
- The real-time capability can only be guaranteed in the transmit direction; in the receive direction, the possibility of incorrect use means that it cannot be guaranteed. This might, for example, be the copying of large quantities of data through this interface.
It is recommended that the PROFINET network is separated from other networks.

- **Profinet:**

- **EL6633-0010 Profinet Decive terminal**
 - See Beckhoff Infosys for documentation.
 - <https://infosys.beckhoff.com/index.php>

- **EAP:**

- The EtherCAT Automation Protocol (EAP) device enables the cyclic, highly deterministic exchange of any desired variables between PCs that are connected by Ethernet. Communication between EAP devices takes place according to the Publisher/Subscriber principle and is specified by the EtherCAT Technology Group (ETG) (ETG 1005 – see webpage www.ethercat.org).
 - The real-time Ethernet driver for TwinCAT must be installed for the TwinCAT EAP device in order for highly deterministic communication to take place
 - <https://infosys.beckhoff.com/index.php>

▪ Mapping structures:

<pre> U_XTS_TRANSPORT_CTRL + X U_XTS_TRANSPORT_STATE 1 {attribute 'pack_mode' := '2'} 2 TYPE U_XTS_TRANSPORT_CTRL : 3 UNION 4 stCtrl : ST_XTS_TRANSPORT_CTRL; 5 byCtrl AT %I* : ARRAY[1..SIZEOF(ST_XTS_TRANSPORT_CTRL)] OF BYTE; 6 END_UNION 7 END_TYPE </pre>	<p>TC_XTS_BASE</p> <table border="1"> <thead> <tr> <th>Variable</th> <th>Flags</th> <th>Online</th> <th>Online List</th> </tr> </thead> <tbody> <tr> <td>Name:</td> <td colspan="3">MAPPING_in.TransportControl.byCtrl</td> </tr> <tr> <td>Type:</td> <td colspan="3">ARRAY [1..2] OF BYTE</td> </tr> <tr> <td>Group:</td> <td>PlcTask Inputs</td> <td>Size:</td> <td>2.0</td> </tr> <tr> <td>Address:</td> <td>441010 (0x6BAB2)</td> <td>User ID:</td> <td>0</td> </tr> <tr> <td>Linked to...</td> <td colspan="3">MAPPING_out.TransportControl.byCtrl . PLC_1 Outputs . ExtemControl Instance . ExtemControl</td> </tr> <tr> <td>Comment:</td> <td colspan="3"></td> </tr> </tbody> </table>	Variable	Flags	Online	Online List	Name:	MAPPING_in.TransportControl.byCtrl			Type:	ARRAY [1..2] OF BYTE			Group:	PlcTask Inputs	Size:	2.0	Address:	441010 (0x6BAB2)	User ID:	0	Linked to...	MAPPING_out.TransportControl.byCtrl . PLC_1 Outputs . ExtemControl Instance . ExtemControl			Comment:			
Variable	Flags	Online	Online List																										
Name:	MAPPING_in.TransportControl.byCtrl																												
Type:	ARRAY [1..2] OF BYTE																												
Group:	PlcTask Inputs	Size:	2.0																										
Address:	441010 (0x6BAB2)	User ID:	0																										
Linked to...	MAPPING_out.TransportControl.byCtrl . PLC_1 Outputs . ExtemControl Instance . ExtemControl																												
Comment:																													
<pre> U_XTS_TRANSPORT_CTRL U_XTS_TRANSPORT_STATE + X 1 {attribute 'pack_mode' := '2'} 2 TYPE U_XTS_TRANSPORT_STATE : 3 UNION 4 stState : ST_XTS_TRANSPORT_STATE; 5 byState AT %I* : ARRAY[1..SIZEOF(ST_XTS_TRANSPORT_STATE)] OF BYTE; 6 END_UNION 7 END_TYPE </pre>	<p>TC_XTS_BASE</p> <table border="1"> <thead> <tr> <th>Variable</th> <th>Flags</th> <th>Online</th> <th>Online List</th> </tr> </thead> <tbody> <tr> <td>Name:</td> <td colspan="3">MAPPING_out.TransportState.byIdState</td> </tr> <tr> <td>Type:</td> <td colspan="3">ARRAY [1..50] OF BYTE</td> </tr> <tr> <td>Group:</td> <td>PlcTask Outputs</td> <td>Size:</td> <td>50.0</td> </tr> <tr> <td>Address:</td> <td>442892 (0x6C20C)</td> <td>User ID:</td> <td>0</td> </tr> <tr> <td>Linked to...</td> <td colspan="3">MAPPING_in.TransportState.byIdState . PLC_1 Inputs . ExtemControl Instance . ExtemControl</td> </tr> <tr> <td>Comment:</td> <td colspan="3"></td> </tr> </tbody> </table>	Variable	Flags	Online	Online List	Name:	MAPPING_out.TransportState.byIdState			Type:	ARRAY [1..50] OF BYTE			Group:	PlcTask Outputs	Size:	50.0	Address:	442892 (0x6C20C)	User ID:	0	Linked to...	MAPPING_in.TransportState.byIdState . PLC_1 Inputs . ExtemControl Instance . ExtemControl			Comment:			
Variable	Flags	Online	Online List																										
Name:	MAPPING_out.TransportState.byIdState																												
Type:	ARRAY [1..50] OF BYTE																												
Group:	PlcTask Outputs	Size:	50.0																										
Address:	442892 (0x6C20C)	User ID:	0																										
Linked to...	MAPPING_in.TransportState.byIdState . PLC_1 Inputs . ExtemControl Instance . ExtemControl																												
Comment:																													

- **Mapping structures:**

U_XTS_STATION_CTRL ➔ X **U_XTS_STATION_STATE**

```

1  TYPE U_XTS_STATION_CTRL :
2
3    UNION
4      stCtrl          : ARRAY[1..MAX_STATION] OF ST_STATION_CTRL;
5      | byCtrl        AT %I*   : ARRAY[1..MAX_STATION] OF
6                                ARRAY[1..SIZEOF(ST_STATION_CTRL)] OF BYTE;
7
8    END_UNION
9
10   END_TYPE

```

TC_XTS_BASE ➔ X

Variable	Flags	Online	Online List
Name:	MAPPING_in.StationControl.byCtrl[1]		
Type:	ARRAY [1..8] OF BYTE		
Group:	PlcTask Inputs	Size:	8.0
Address:	440970 (0x6BA8A)	User ID:	0
Linked to...			
Comment:			

U_XTS_STATION_CTRL ➔ X **U_XTS_STATION_STATE** ➔ X

```

1  TYPE U_XTS_STATION_STATE :
2
3    UNION
4      stState         : ARRAY[1..MAX_STATION] OF ST_STATION_STATE;
5      | byState        AT %I*   : ARRAY[1..MAX_STATION] OF
6                                ARRAY[1..SIZEOF(ST_STATION_STATE)] OF BYTE;
7
8    END_UNION
9
10   END_TYPE

```

TC_XTS_BASE ➔ X

Variable	Flags	Online	Online List
Name:	MAPPING_out.StationState.byIdState[1]		
Type:	ARRAY [1..14] OF BYTE		
Group:	PlcTask Outputs	Size:	14.0
Address:	442822 (0x6C1C6)	User ID:	0
Linked to...			
Comment:			

Implementation

BECKHOFF

▪ Mapping structures:

```
U_MOVER_CTRL  U_MOVER_STATE  U_MOVE_DATA  U_GEAR_DATA
1  TYPE U_MOVER_CTRL :
2  UNION
3      stCtrl          : ARRAY[1..MAX_MOVER] OF ST_MOVER_CTRL;
4
5      byCtrl    AT %I*   : ARRAY[1..MAX_MOVER] OF
6                      ARRAY[1..SIZEOF(ST_MOVER_CTRL)] OF BYTE;
7
8  END_UNION
9  END_TYPE
```

```
U_MOVER_CTRL  U_MOVER_STATE  U_MOVE_DATA  U_GEAR_DATA
1  TYPE U_MOVER_STATE :
2  UNION
3      stState          : ARRAY[1..MAX_MOVER] OF ST_MOVER_STATE;
4
5      byState   AT %Q*   : ARRAY[1..MAX_MOVER] OF
6                      ARRAY[1..SIZEOF(ST_MOVER_STATE)] OF BYTE;
7
8  END_UNION
9  END_TYPE
```

TC_XTS_BASE

Variable	Flags	Online	Online List
Name:	MAPPING_in.MoverCtrl.byCtrl[1]		
Type:	ARRAY [1..2] OF BYTE		
Group:	PlcTask Inputs	Size:	2.0
Address:	439970 (0x6B6A2)	User ID:	0
Linked to...			
Comment:			

TC_XTS_BASE

Variable	Flags	Online	Online List
Name:	MAPPING_out.MoverState.byId[1]		
Type:	ARRAY [1..4] OF BYTE		
Group:	PlcTask Outputs	Size:	4.0
Address:	442562 (0x6C0C2)	User ID:	0
Linked to...			
Comment:			

- **Mapping structures:**

U_MOVER_CTRL	U_MOVER_STATE	U_MOVE_DATA			U_GEAR_DATA
--------------	---------------	--------------------	--	--	-------------

```

1  TYPE U_MOVE_DATA :
2  UNION
3      stData          : ARRAY[1..MAX_MOVER] OF ST_MOVE_DATA;
4
5      byData    AT %I*   : ARRAY[1..MAX_MOVER] OF
6                      ARRAY[1..SIZEOF(ST_MOVE_DATA)] OF BYTE;
7
8  END_UNION
9  END_TYPE

```

TC_XTS_BASE					
Variable Flags Online Online List					
Name:	MAPPING_in.MoveData.byData[1]				
Type:	ARRAY [1..64] OF BYTE				
Group:	PlcTask Inputs	Size:	64.0		
Address:	439990 (0x6B6B6)		User ID:	0	
Linked to...					
Comment:					

U_MOVER_CTRL	U_MOVER_STATE	U_MOVE_DATA			U_GEAR_DATA
--------------	---------------	-------------	--	--	-------------

```

1  TYPE U_GEAR_DATA :
2  UNION
3      stData          : ARRAY[1..MAX_MOVER] OF ST_GEAR_DATA;
4
5      byData    AT %I*   : ARRAY[1..MAX_MOVER] OF
6                      ARRAY[1..SIZEOF(ST_GEAR_DATA)] OF BYTE;
7
8  END_UNION
9  END_TYPE

```

TC_XTS_BASE					
Variable Flags Online Online List					
Name:	MAPPING_in.GearData.byData[1]				
Type:	ARRAY [1..34] OF BYTE				
Group:	PlcTask Inputs	Size:	34.0		
Address:	440630 (0x6B936)		User ID:	0	
Linked to...					
Comment:					

- Mapping structures:

```

U_MOVER_INFO  TC_XTS_BASE
1  TYPE U_MOVER_INFO :
2  UNION
3      stInfo          : ARRAY[1..MAX_MOVER] OF ST_MOVER_INFO;
4
5      byInfo    AT %Q* : ARRAY[1..MAX_MOVER] OF
6                      ARRAY[1..SIZEOF(ST_MOVER_INFO)] OF BYTE;
7
8  END_UNION
9  END_TYPE
  
```

Variable	Flags	Online	Online List
Name:	MAPPING_out.MoverInfo.byInfo[1]		
Type:	ARRAY [1..22] OF BYTE		
Group:	PlcTask Outputs	Size:	22.0
Address:	442602 (0x6C0EA)	User ID:	0
Linked to...			
Comment:			
ADS Info:	Port: 353, IGrp: 0x8502000, IOffs: 0x8106C0EA, Len: 22		
Symbol Info:	Port: 851, 'MAPPING_out.MoverInfo.byInfo[1]'		
Full Name:	TIPC^XtsTransport^XtsTransport Instance^PlcTask Outputs^MAPPING_out.MoverInfo.byInfo[1]		

- **Mapping example:**

- **5 Stations**
- **10 Mover**

```
▷ MAIN (PRG)
  ▷ PlcTask (PLC_1)
    ▷ MAPPING_in
    ▷ MAIN
    ▷ MAPPING_out
  ▷ README.md
  ▷ XtsTransport.tmc
  ▷ XtsTransport Instance
    ▷ PlcTask Inputs
      ▷ MAPPING_in.MoverCtrl.byCtrl
      ▷ MAPPING_in.MoveData.byIdata
      ▷ MAPPING_in.GearData.byIdata
      ▷ MAPPING_in.StationControl.byCtrl
      ▷ MAPPING_in.TransportControl.byCtrl
      ▷ GVL_XTS.CaGroupRef.NcToPlc
      ▷ GVL_XTS.AxisRefMover[1].NcToPlc
      ▷ GVL_XTS.AxisRefMover[2].NcToPlc
      ▷ GVL_XTS.AxisRefMover[3].NcToPlc
      ▷ GVL_XTS.AxisRefMover[4].NcToPlc
      ▷ GVL_XTS.AxisRefMover[5].NcToPlc
      ▷ GVL_XTS.AxisRefMover[6].NcToPlc
      ▷ GVL_XTS.AxisRefMover[7].NcToPlc
      ▷ GVL_XTS.AxisRefMover[8].NcToPlc
      ▷ GVL_XTS.AxisRefMover[9].NcToPlc
      ▷ GVL_XTS.AxisRefMover[10].NcToPlc
```

```
▷ MAIN (PRG)
  ▷ PlcTask (PLC_1)
    ▷ MAPPING_in
    ▷ MAIN
    ▷ MAPPING_out
  ▷ README.md
  ▷ XtsTransport.tmc
  ▷ XtsTransport Instance
    ▷ PlcTask Inputs
    ▷ PlcTask Outputs
      ▷ MAPPING_out.StationState.byIdState
      ▷ MAPPING_out.TransportState.byIdState
      ▷ GVL_XTS.CaGroupRef.PlcToNc
      ▷ GVL_XTS.AxisRefMover[1].PlcToNc
      ▷ GVL_XTS.AxisRefMover[2].PlcToNc
      ▷ GVL_XTS.AxisRefMover[3].PlcToNc
      ▷ GVL_XTS.AxisRefMover[4].PlcToNc
      ▷ GVL_XTS.AxisRefMover[5].PlcToNc
      ▷ GVL_XTS.AxisRefMover[6].PlcToNc
      ▷ GVL_XTS.AxisRefMover[7].PlcToNc
      ▷ GVL_XTS.AxisRefMover[8].PlcToNc
      ▷ GVL_XTS.AxisRefMover[9].PlcToNc
      ▷ GVL_XTS.AxisRefMover[10].PlcToNc
```

- **Mapping example:**

- **5 Stations**
- **10 Mover**

```
└─ MAPPING_in.MoverCtrl.byCtrl
    └─ MAPPING_in.MoverCtrl.byCtrl[1]
        └─ MAPPING_in.MoverCtrl.byCtrl[1][1]
        └─ MAPPING_in.MoverCtrl.byCtrl[1][2]
    └─ MAPPING_in.MoverCtrl.byCtrl[2]
    └─ MAPPING_in.MoverCtrl.byCtrl[3]
    └─ MAPPING_in.MoverCtrl.byCtrl[4]
    └─ MAPPING_in.MoverCtrl.byCtrl[5]
    └─ MAPPING_in.MoverCtrl.byCtrl[6]
    └─ MAPPING_in.MoverCtrl.byCtrl[7]
    └─ MAPPING_in.MoverCtrl.byCtrl[8]
    └─ MAPPING_in.MoverCtrl.byCtrl[9]
    └─ MAPPING_in.MoverCtrl.byCtrl[10]
```

```
└─ PlcTask Outputs
    └─ MAPPING_out.MoverState.byIdState
        └─ MAPPING_out.MoverState.byIdState[1]
            └─ MAPPING_out.MoverState.byIdState[1][1]
            └─ MAPPING_out.MoverState.byIdState[1][2]
            └─ MAPPING_out.MoverState.byIdState[1][3]
            └─ MAPPING_out.MoverState.byIdState[1][4]
        └─ MAPPING_out.MoverState.byIdState[2]
        └─ MAPPING_out.MoverState.byIdState[3]
        └─ MAPPING_out.MoverState.byIdState[4]
        └─ MAPPING_out.MoverState.byIdState[5]
        └─ MAPPING_out.MoverState.byIdState[6]
        └─ MAPPING_out.MoverState.byIdState[7]
        └─ MAPPING_out.MoverState.byIdState[8]
        └─ MAPPING_out.MoverState.byIdState[9]
        └─ MAPPING_out.MoverState.byIdState[10]
    └─ MAPPING_out.MoverInfo.byIdInfo
    └─ MAPPING_out.StationState.byIdState
    └─ MAPPING_out.TransportState.byIdState
```

- **Mapping example:**

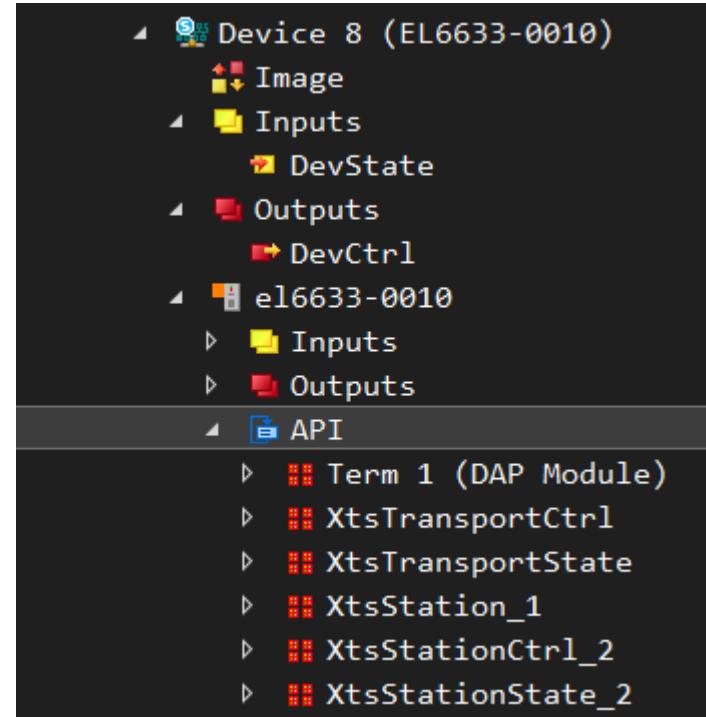
- **5 Stations**
- **10 Mover**

```
    ▶ ↗ MAPPING_in.StationControl.byCtrl
      ▲ ↗ MAPPING_in.StationControl.byCtrl[1]
        ↗ MAPPING_in.StationControl.byCtrl[1][1]
        ↗ MAPPING_in.StationControl.byCtrl[1][2]
        ↗ MAPPING_in.StationControl.byCtrl[1][3]
        ↗ MAPPING_in.StationControl.byCtrl[1][4]
        ↗ MAPPING_in.StationControl.byCtrl[1][5]
        ↗ MAPPING_in.StationControl.byCtrl[1][6]
        ↗ MAPPING_in.StationControl.byCtrl[1][7]
        ↗ MAPPING_in.StationControl.byCtrl[1][8]
      ▷ ↗ MAPPING_in.StationControl.byCtrl[2]
      ▷ ↗ MAPPING_in.StationControl.byCtrl[3]
      ▷ ↗ MAPPING_in.StationControl.byCtrl[4]
      ▷ ↗ MAPPING_in.StationControl.byCtrl[5]

    ▶ ↗ MAPPING_out.StationState.byId
      ▲ ↗ MAPPING_out.StationState.byId[1]
        ↗ MAPPING_out.StationState.byId[1][1]
        ↗ MAPPING_out.StationState.byId[1][2]
        ↗ MAPPING_out.StationState.byId[1][3]
        ↗ MAPPING_out.StationState.byId[1][4]
        ↗ MAPPING_out.StationState.byId[1][5]
        ↗ MAPPING_out.StationState.byId[1][6]
        ↗ MAPPING_out.StationState.byId[1][7]
        ↗ MAPPING_out.StationState.byId[1][8]
        ↗ MAPPING_out.StationState.byId[1][9]
        ↗ MAPPING_out.StationState.byId[1][10]
        ↗ MAPPING_out.StationState.byId[1][11]
        ↗ MAPPING_out.StationState.byId[1][12]
        ↗ MAPPING_out.StationState.byId[1][13]
        ↗ MAPPING_out.StationState.byId[1][14]
      ▷ ↗ MAPPING_out.StationState.byId[2]
      ▷ ↗ MAPPING_out.StationState.byId[3]
      ▷ ↗ MAPPING_out.StationState.byId[4]
      ▷ ↗ MAPPING_out.StationState.byId[5]
```

- PN - EL6633 example:

- TransportCtrl
- TransportState
- 2 Stations



- PN - EL6633 example:
 - TransportCtrl (2 Byte Out)

The screenshot shows the project structure and a table of variables for the el6633-0010 module.

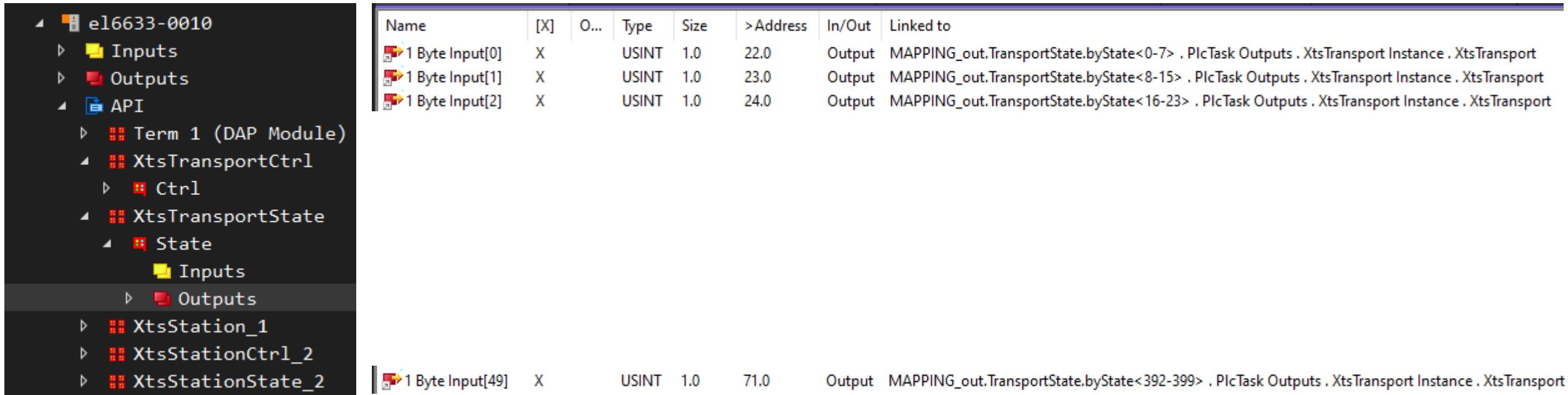
Project Structure:

- el6633-0010
 - Inputs
 - Outputs
 - API
 - Term 1 (DAP Module)
 - XtsTransportCtrl
 - Ctrl
 - Inputs
 - Outputs
 - XtsTransportState
 - XtsStation_1
 - XtsStationCtrl_2
 - XtsStationState_2

Table of Variables:

Name	[X]	Online	Type	Size	>Address	In/Out	Linked to
1 Byte Output[0]	X		USINT	1.0	2.0	Input	MAPPING_in.TransportControl.byCtrl<0-7> . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[1]	X		USINT	1.0	3.0	Input	MAPPING_in.TransportControl.byCtrl<8-15> . PlcTask Inputs . XtsTransport Instance . XtsTransport

- PN - EL6633 example:
 - TransportState (64 Byte In)



The screenshot shows a software interface for configuring a Beckhoff EL6633 module. On the left, a tree view displays the module structure:

- el6633-0010
 - Inputs
 - Outputs
 - API
 - Term 1 (DAP Module)
 - XtsTransportCtrl
 - Ctrl
 - XtsTransportState
 - State
 - Inputs
 - Outputs
 - XtsStation_1
 - XtsStationCtrl_2
 - XtsStationState_2

On the right, a table lists the input parameters:

Name	[X]	O...	Type	Size	>Address	In/Out	Linked to
1 Byte Input[0]	X		USINT	1.0	22.0	Output	MAPPING_out.TransportState.byIdState<0-7>.PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[1]	X		USINT	1.0	23.0	Output	MAPPING_out.TransportState.byIdState<8-15>.PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[2]	X		USINT	1.0	24.0	Output	MAPPING_out.TransportState.byIdState<16-23>.PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[49]	X		USINT	1.0	71.0	Output	MAPPING_out.TransportState.byIdState<392-399>.PlcTask Outputs . XtsTransport Instance . XtsTransport

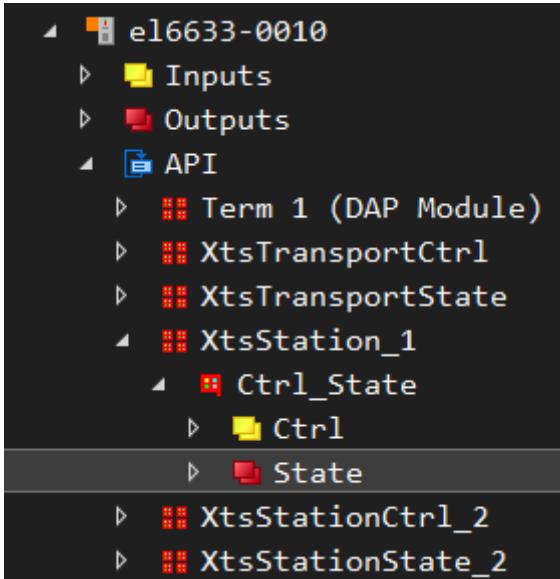
- PN - EL6633 example:

- XtsStation_1 (16 Byte In/Out)

Name	[X]	O.	Type	Size	>A...	In/Out	Linked to
1 Byte Output[0]	X		USINT	1.0	4.0	Input	MAPPING_in.StationControl.byCtrl[1]<0-7> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[1]	X		USINT	1.0	5.0	Input	MAPPING_in.StationControl.byCtrl[1]<8-15> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[2]	X		USINT	1.0	6.0	Input	MAPPING_in.StationControl.byCtrl[1]<16-23> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[3]	X		USINT	1.0	7.0	Input	MAPPING_in.StationControl.byCtrl[1]<24-31> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[4]	X		USINT	1.0	8.0	Input	MAPPING_in.StationControl.byCtrl[1]<32-39> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[5]	X		USINT	1.0	9.0	Input	MAPPING_in.StationControl.byCtrl[1]<40-47> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[6]	X		USINT	1.0	10.0	Input	MAPPING_in.StationControl.byCtrl[1]<48-55> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[7]	X		USINT	1.0	11.0	Input	MAPPING_in.StationControl.byCtrl[1]<56-63> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport

- PN - EL6633 example:

- XtsStation_1 (16 Byte In/Out)



Name	[X]	O.	Type	S...	>...	In/Out	Linked to
1 Byte Input[0]	X		USINT	1.0	6.0	Output	MAPPING_out.StationState.byId[1]<0-7> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[1]	X		USINT	1.0	7.0	Output	MAPPING_out.StationState.byId[1]<8-15> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[2]	X		USINT	1.0	8.0	Output	MAPPING_out.StationState.byId[1]<16-23> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[3]	X		USINT	1.0	9.0	Output	MAPPING_out.StationState.byId[1]<24-31> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[4]	X		USINT	1.0	10.0	Output	MAPPING_out.StationState.byId[1]<32-39> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[5]	X		USINT	1.0	11.0	Output	MAPPING_out.StationState.byId[1]<40-47> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[6]	X		USINT	1.0	12.0	Output	MAPPING_out.StationState.byId[1]<48-55> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[7]	X		USINT	1.0	13.0	Output	MAPPING_out.StationState.byId[1]<56-63> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[8]	X		USINT	1.0	14.0	Output	MAPPING_out.StationState.byId[1]<64-71> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[9]	X		USINT	1.0	15.0	Output	MAPPING_out.StationState.byId[1]<72-79> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[10]	X		USINT	1.0	16.0	Output	MAPPING_out.StationState.byId[1]<80-87> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[11]	X		USINT	1.0	17.0	Output	MAPPING_out.StationState.byId[1]<88-95> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[12]	X		USINT	1.0	18.0	Output	MAPPING_out.StationState.byId[1]<96-103> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[13]	X		USINT	1.0	19.0	Output	MAPPING_out.StationState.byId[1]<104-111> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport

- PN - EL6633 example:
 - XtsStationCtrl_2 (8 Byte Out)

The screenshot shows a software interface with a tree-based navigation on the left and a detailed table on the right.

Left Panel (Tree View):

- e16633-0010
 - Inputs
 - Outputs
 - API
 - Term 1 (DAP Module)
 - XtsTransportCtrl
 - XtsTransportState
 - XtsStation_1
 - XtsStationCtrl_2
 - Ctrl
 - Inputs
 - Outputs
 - XtsStationState_2

Right Panel (Table):

Name	[X]	O.	Type	S...	>A...	In/Out	Linked to
1 Byte Output[0]	X		USINT	1.0	20.0	Input	MAPPING_in.StationControl.byCtrl[2]<0-7> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[1]	X		USINT	1.0	21.0	Input	MAPPING_in.StationControl.byCtrl[2]<8-15> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[2]	X		USINT	1.0	22.0	Input	MAPPING_in.StationControl.byCtrl[2]<16-23> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[3]	X		USINT	1.0	23.0	Input	MAPPING_in.StationControl.byCtrl[2]<24-31> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[4]	X		USINT	1.0	24.0	Input	MAPPING_in.StationControl.byCtrl[2]<32-39> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[5]	X		USINT	1.0	25.0	Input	MAPPING_in.StationControl.byCtrl[2]<40-47> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[6]	X		USINT	1.0	26.0	Input	MAPPING_in.StationControl.byCtrl[2]<48-55> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport
1 Byte Output[7]	X		USINT	1.0	27.0	Input	MAPPING_in.StationControl.byCtrl[2]<56-63> . MAPPING_in.StationControl.byCtrl . PlcTask Inputs . XtsTransport Instance . XtsTransport

- PN - EL6633 example:

- XtsStationState_2 (16 Byte In)

The screenshot shows the configuration of the XtsStationState_2 module. On the left, the module structure is displayed with various components like Inputs, Outputs, and API. The API section contains several sub-components related to transport and station state. On the right, a detailed table lists the 16 inputs and their mappings to PLC outputs.

Name	[X]	O.	Type	S...	>A...	In/Out	Linked to
1 Byte Input[0]	X		USINT	1.0	86.0	Output	MAPPING_out.StationState.byId[2]<0-7> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[1]	X		USINT	1.0	87.0	Output	MAPPING_out.StationState.byId[2]<8-15> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[2]	X		USINT	1.0	88.0	Output	MAPPING_out.StationState.byId[2]<16-23> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[3]	X		USINT	1.0	89.0	Output	MAPPING_out.StationState.byId[2]<24-31> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[4]	X		USINT	1.0	90.0	Output	MAPPING_out.StationState.byId[2]<32-39> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[5]	X		USINT	1.0	91.0	Output	MAPPING_out.StationState.byId[2]<40-47> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[6]	X		USINT	1.0	92.0	Output	MAPPING_out.StationState.byId[2]<48-55> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[7]	X		USINT	1.0	93.0	Output	MAPPING_out.StationState.byId[2]<56-63> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[8]	X		USINT	1.0	94.0	Output	MAPPING_out.StationState.byId[2]<64-71> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[9]	X		USINT	1.0	95.0	Output	MAPPING_out.StationState.byId[2]<72-79> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[10]	X		USINT	1.0	96.0	Output	MAPPING_out.StationState.byId[2]<80-87> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[11]	X		USINT	1.0	97.0	Output	MAPPING_out.StationState.byId[2]<88-95> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[12]	X		USINT	1.0	98.0	Output	MAPPING_out.StationState.byId[2]<96-103> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport
1 Byte Input[13]	X		USINT	1.0	99.0	Output	MAPPING_out.StationState.byId[2]<104-111> . MAPPING_out.StationState.byId . PlcTask Outputs . XtsTransport Instance . XtsTransport