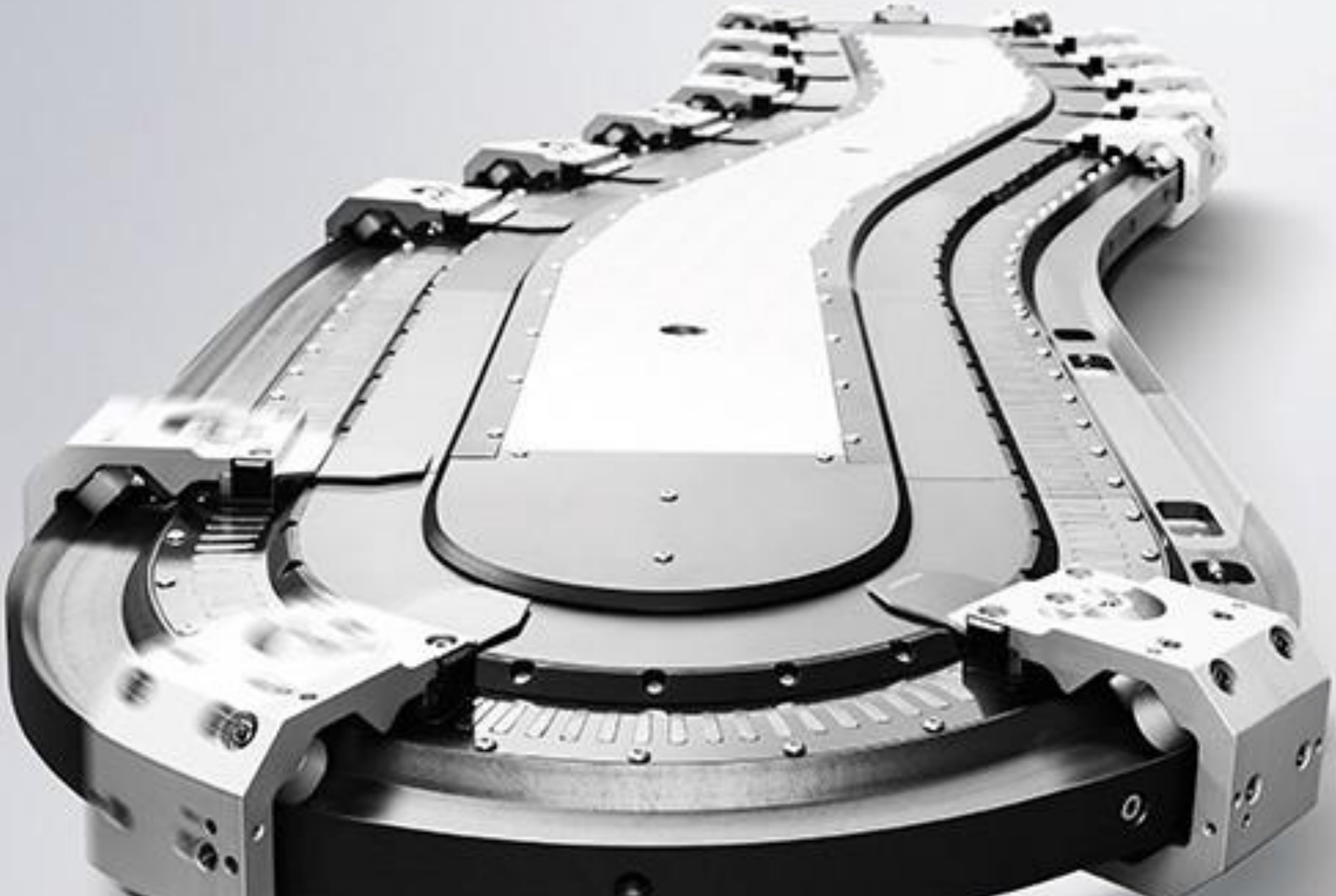
XTS TRANSPORT LAYER – Station Geometry



Motivation

- **Station based approach**
  - The station is the base building block of the XTS_TRANSPORT_LAYER
  - The station is described by its geometric parameters
  - The stations behaviour may be changed by changing its geometric parameters.
  - The station may be part of Processes.
  - A Process is controlling a range of stations via a single interface.

Motivation

- **Station based approach**
  - The station is tasked with the coordination of movement.
  - The station carries the geometric information on the track.
  - The stations’ list carries the mover information (The Ticket – ST_MOVER_DATA)
  - The geometric information has influence on the stations’ behaviour.
    - How many nests?
    - How long the distance until release?
  - The geometric information has influence on the movers’ behaviour within a station
    - Forward move to a nest
    - Backward move to a nest

Motivation

- **Station based approach**

  - The station is tasked with the coordination of mover targets
    - Halting in the states you answer to.
      - The station will wait here forever, isolated, doing nothing until commanded.
    - Sequential CASE execution states you only see in the logs.
      - The station must be able to execute without delay until the next roadblock (either a state you answer to or an error exit).
    - Handover is atomic and is either successful or leads to a fatal error.
      - The linked lists are not using dynamic memory allocation.
        - Static memory allocation [0.. MAX_LIST_NODES] is ensuring that no memory fault or heap fragmentation can occur.

Motivation

- **Station based approach**

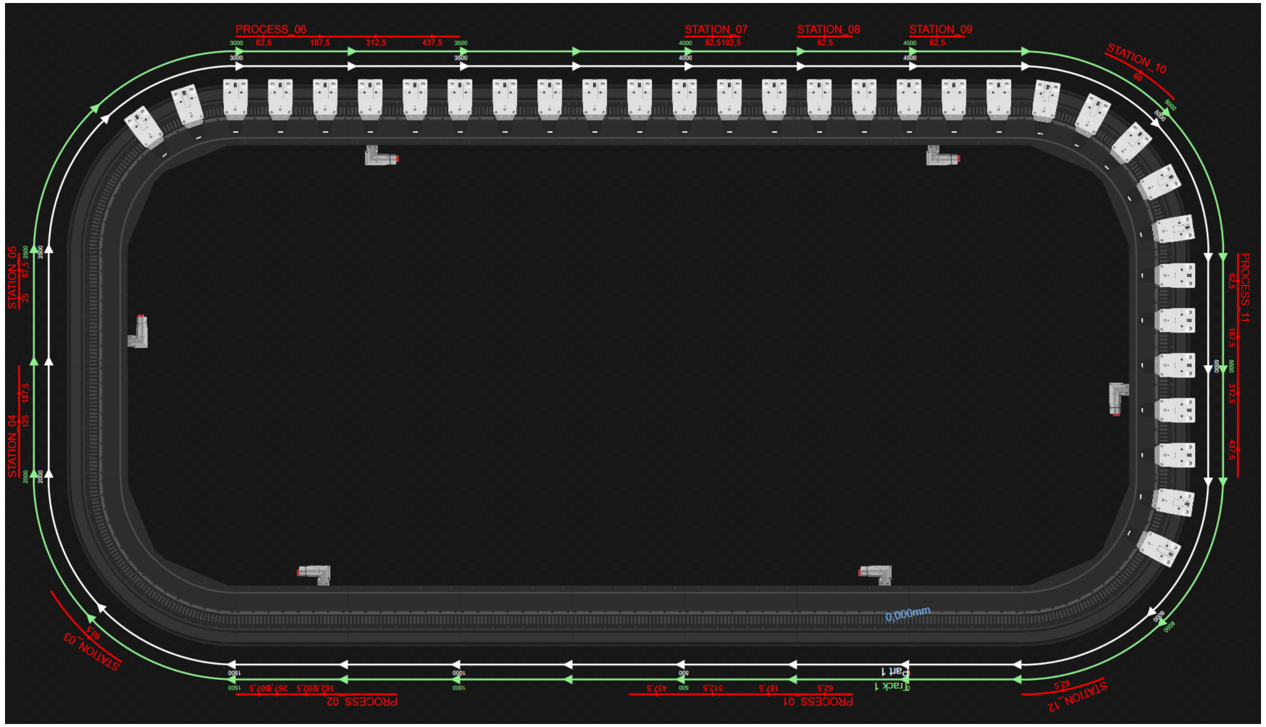
  - Closed loop trade offs
    - No passing at all times.
      - This constraint is the main reason why lists may stay sorted simply by mapping the topology onto the logic.

    - High throughput and high speed
      - While a track-switch enables highest flexibility regarding the mover targets
        - A track-switch introduces a bottleneck which must be accounted for.

Localisation

- **Station based approach**

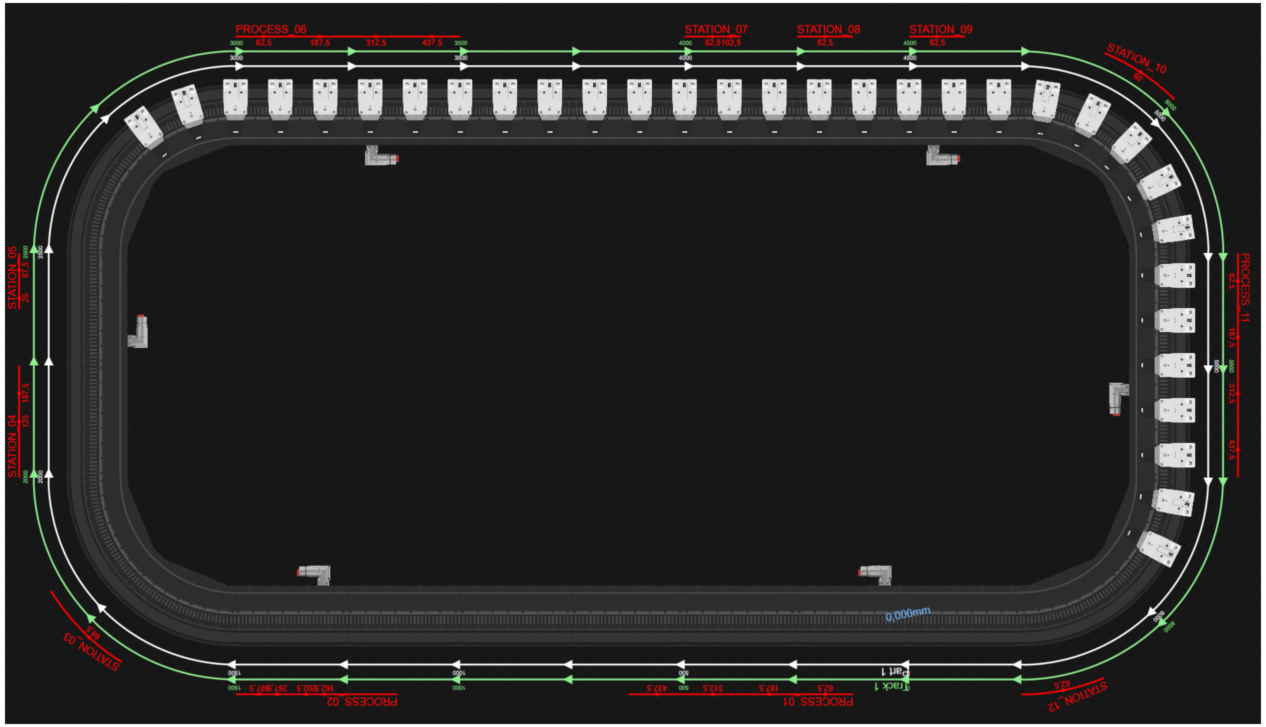
- **Example of a closed loop XTS:**
  - **PROCESS** may have multiple stations
    - **STATION** may have multiple nests
      - **NEST** is a stop position for a mover



Localisation

- **Station based approach**

- **Base building block:**
  - **STATION**
    - Since the station logic is fundamental for understanding processes, we start by describing the stations' geometry first.

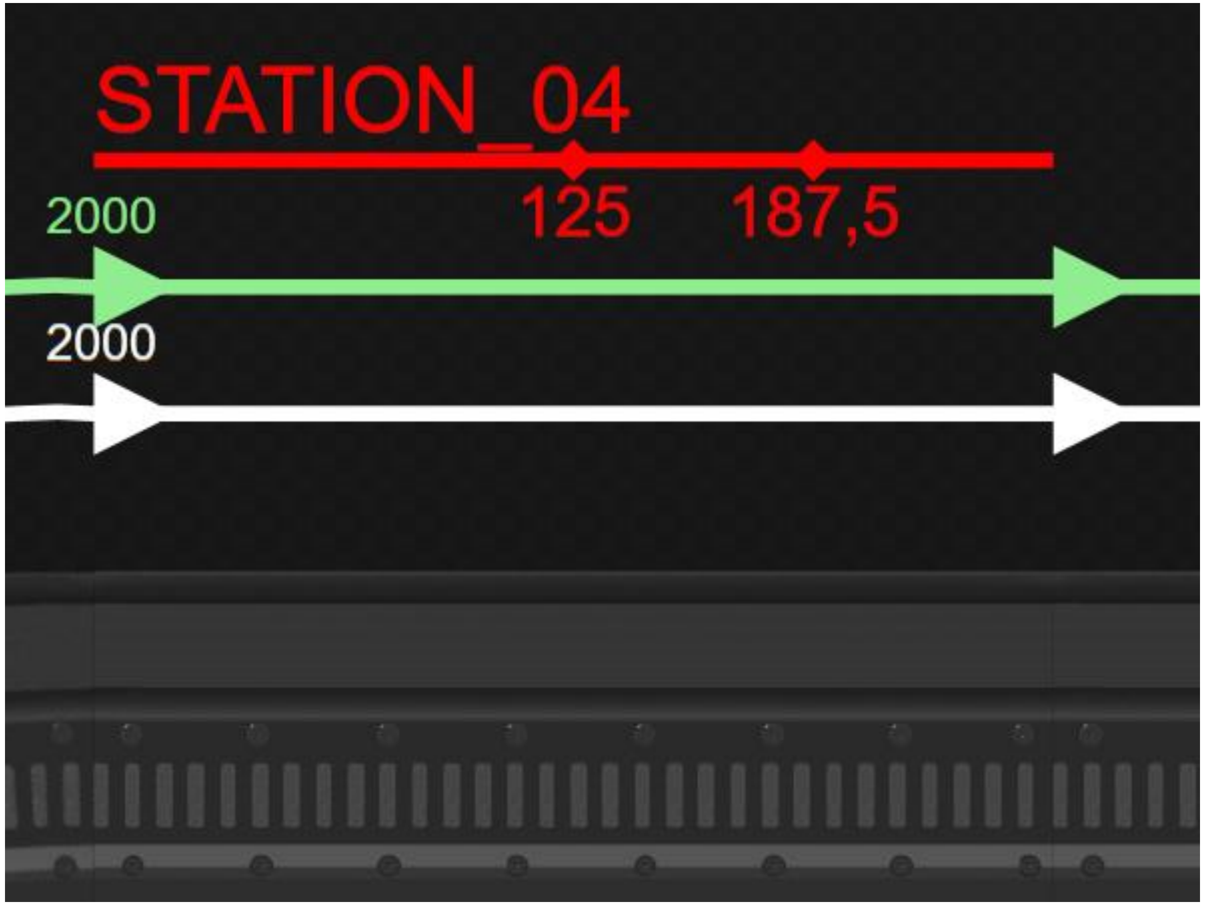


Localisation

BECKHOFF

- **Station based approach**

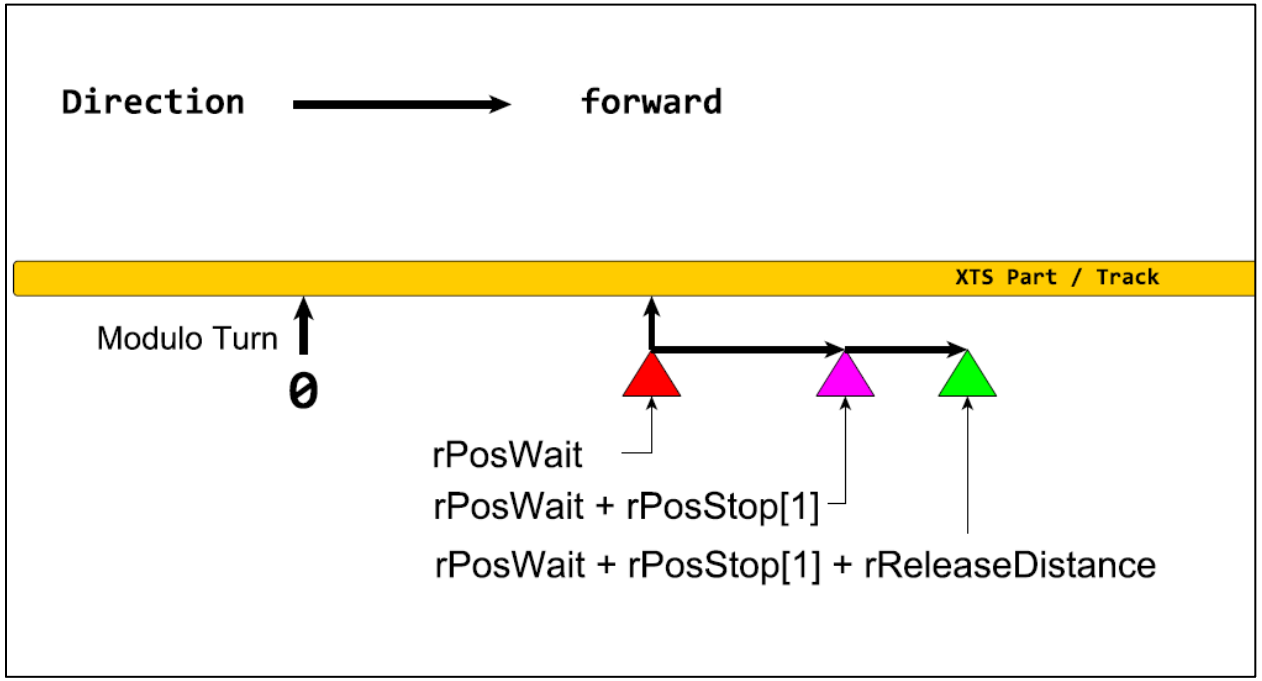
- **STATION geometry:**
  - **WaitPos**: a position on the track where the station starts, and any other station may send a mover to.
  - **StopPos**: 1 to 8 possible **relative** positions a mover may stop at.
  - **ReleaseDistance**: distance a mover has to travel to logically leave a station.



Localisation

- **Station based approach**

- **STATION geometry:**
  - **WaitPos**: a position on the track where the station starts, and any other station may send a mover to.
  - **StopPos**: 1 to 8 possible **relative** positions a mover may stop at.
  - **ReleaseDistance**: distance a mover has to travel to logically leave a station.

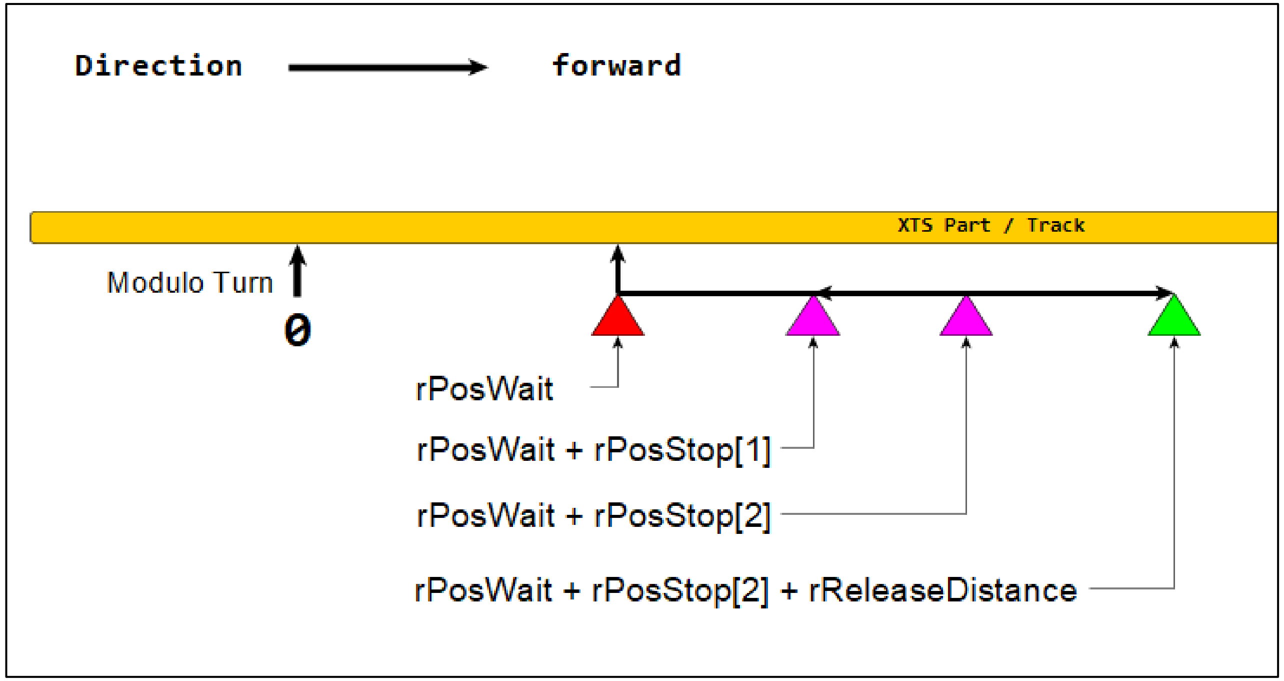


Localisation

BECKHOFF

- **Station based approach**

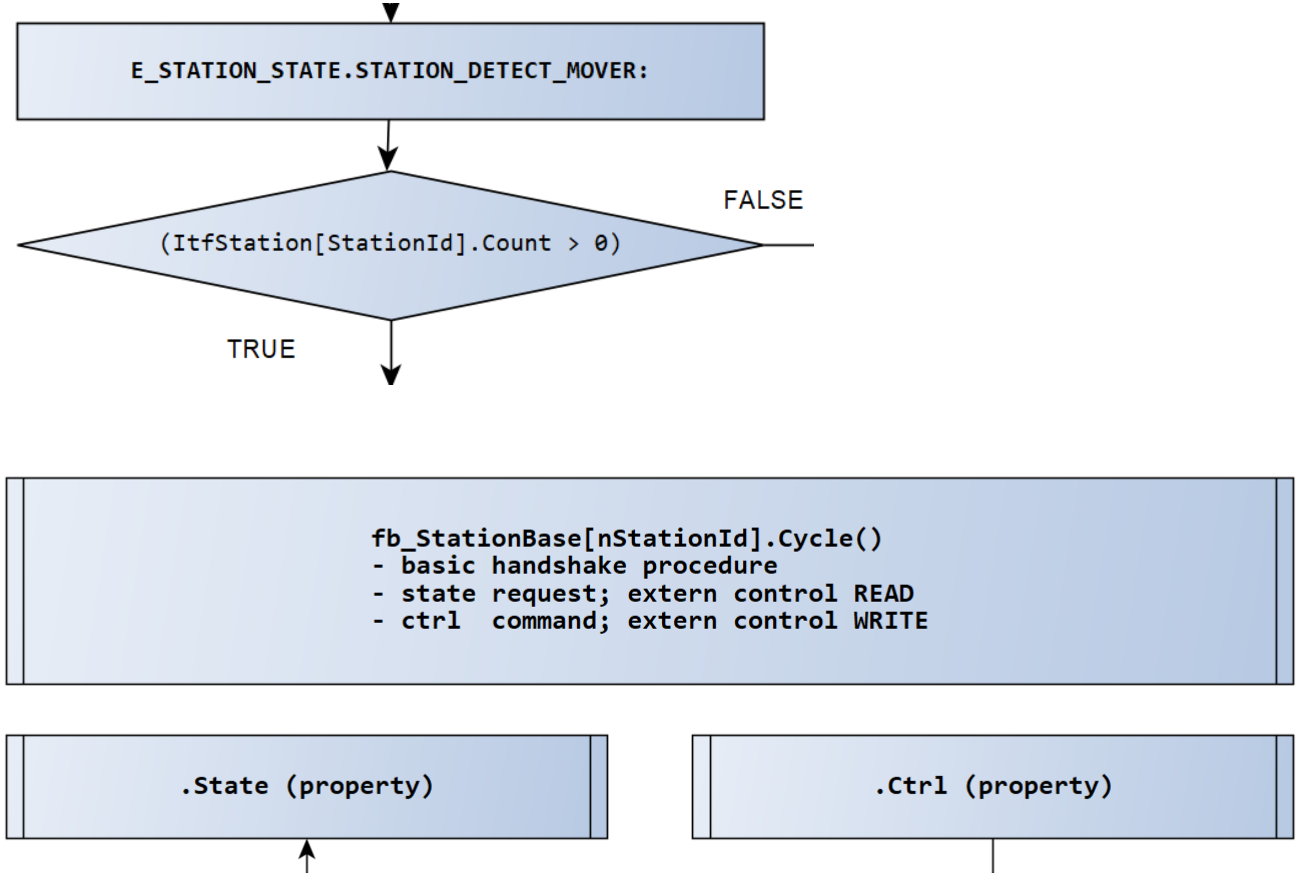
- **STATION geometry:**
  - **WaitPos**: a position on the track where the station starts, and any other station may send a mover to.
  - **StopPos**: 1 to 8 possible **relative** positions a mover may stop at.
  - **ReleaseDistance**: distance a mover has to travel to logically leave a station.



Localisation

- **Station based approach**

- **STATION operation:**
  - **StationList**: a list in which a sending station writes the mover ticket for this station. This station checks its own list cyclically and reports the detection of a new mover via the StationState.
  - **StationCtrl**: control struct to command station.
  - **StationState**: state information you have to react to.



E_STATION_STATE.STATION_DETECT_MOVER:

(ItfStation[StationId].Count > 0)

FALSE

TRUE

```
fb_StationBase[nStationId].Cycle()
- basic handshake procedure
- state request; extern control READ
- ctrl  command; extern control WRITE
```

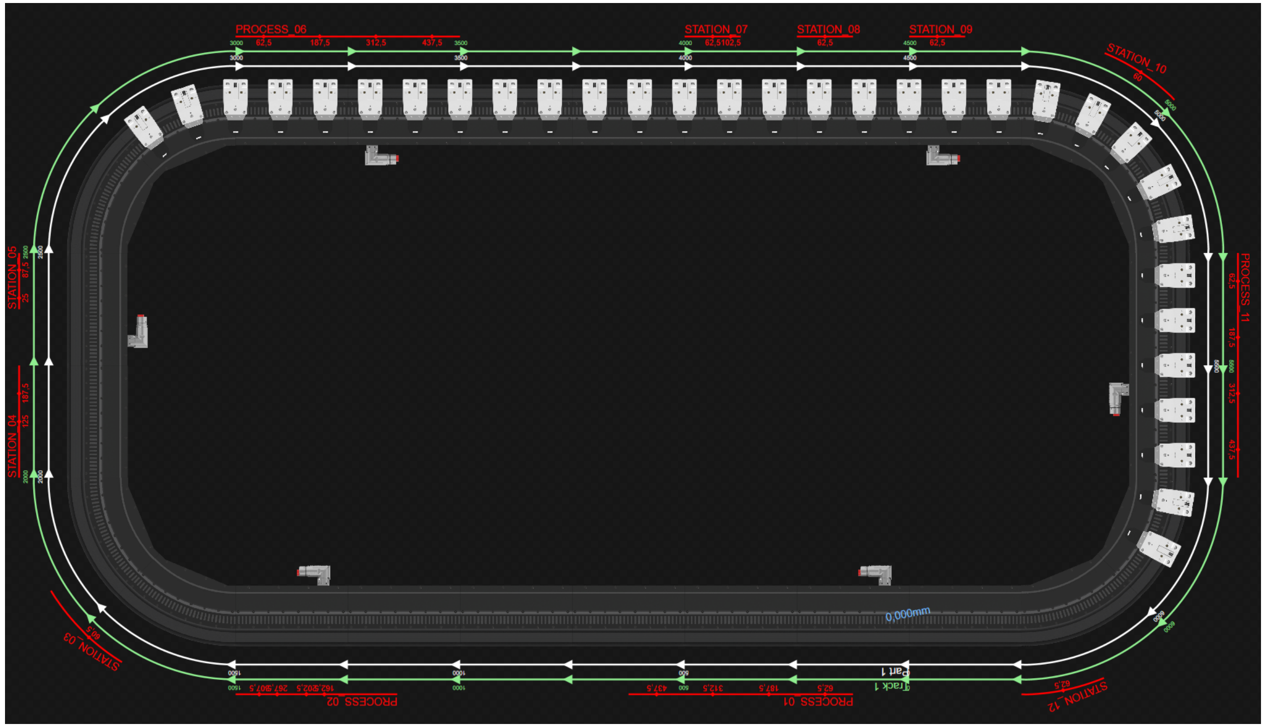
.State (property)

.Ctrl (property)

Localisation

- **Station based approach**

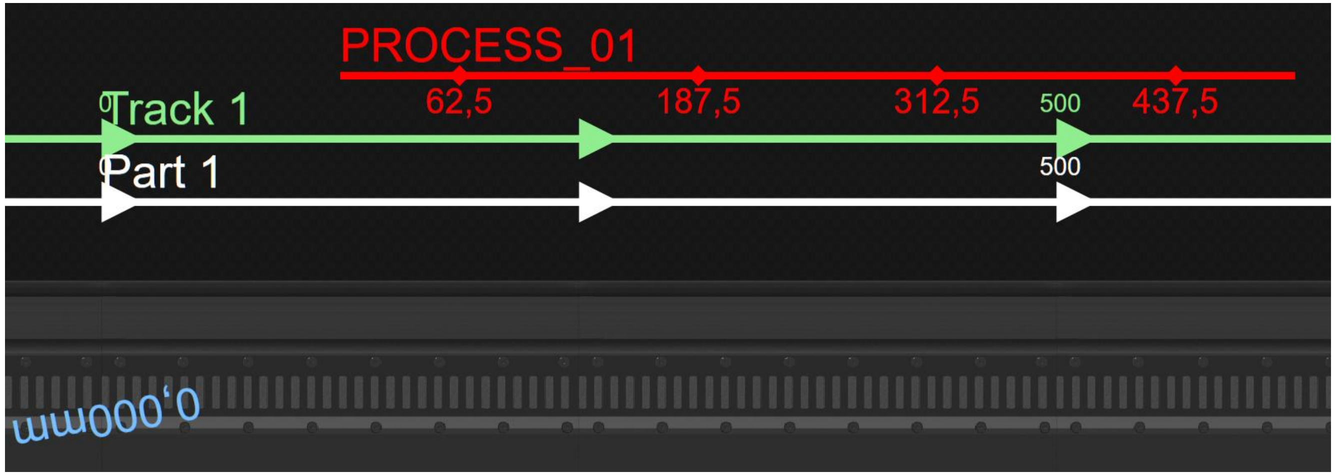
- **Grouping building block:**
  - **PROCESS**
    - The process abstraction is intended to give you control over a range of stations by using one dedicated interface.
    - The process commands and checks all its stations.
    - Stations my be muted during runtime, so event driven decision making is possible for every batch individually.



Localisation

- **Station based approach**

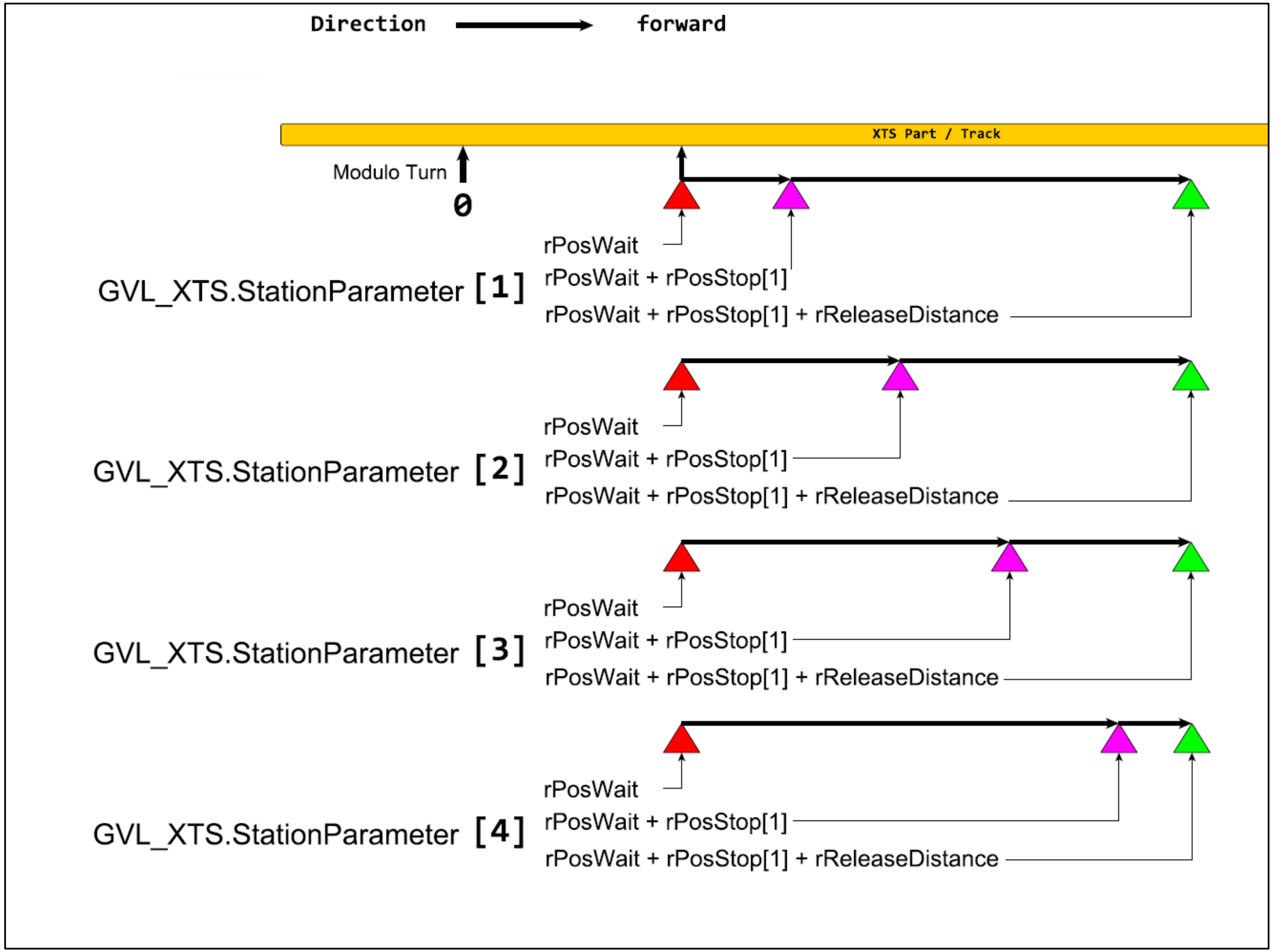
- **PROCESS:**
  - may have one or many stations
  - handshakes stations simultaneously
  - may mute stations
  - Stations in processes may have multiple nests



Localisation

- **Station based approach**

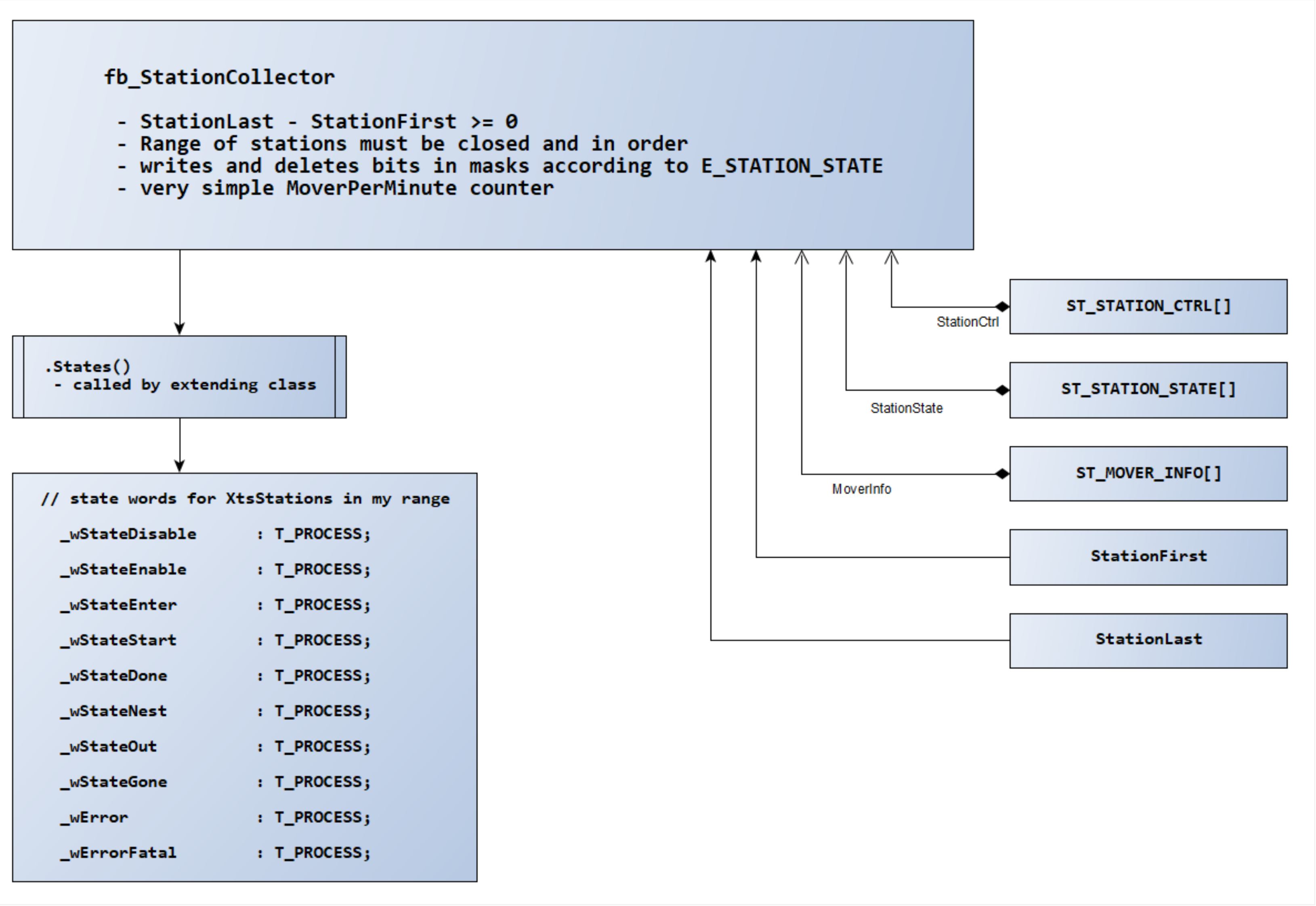
- **PROCESS:**
  - may have one or many stations
  - handshakes stations simultaneously
  - may mute stations
  - Stations in processes may have multiple nests


Direction forward
XTS Part / Track
Modulo Turn
0
GVL_XTS.StationParameter [1]
rPosWait
rPosWait + rPosStop[1]
rPosWait + rPosStop[1] + rReleaseDistance
GVL_XTS.StationParameter [2]
rPosWait
rPosWait + rPosStop[1]
rPosWait + rPosStop[1] + rReleaseDistance
GVL_XTS.StationParameter [3]
rPosWait
rPosWait + rPosStop[1]
rPosWait + rPosStop[1] + rReleaseDistance
GVL_XTS.StationParameter [4]
rPosWait
rPosWait + rPosStop[1]
rPosWait + rPosStop[1] + rReleaseDistance

Localisation

- **Station based approach**

- **PROCESS:**
  - **StationCollector**
  - collects information from the stations.



```
fb_StationCollector

 - StationLast - StationFirst >= 0
 - Range of stations must be closed and in order
 - writes and deletes bits in masks according to E_STATION_STATE
 - very simple MoverPerMinute counter
```

```
.States()
 - called by extending class
```

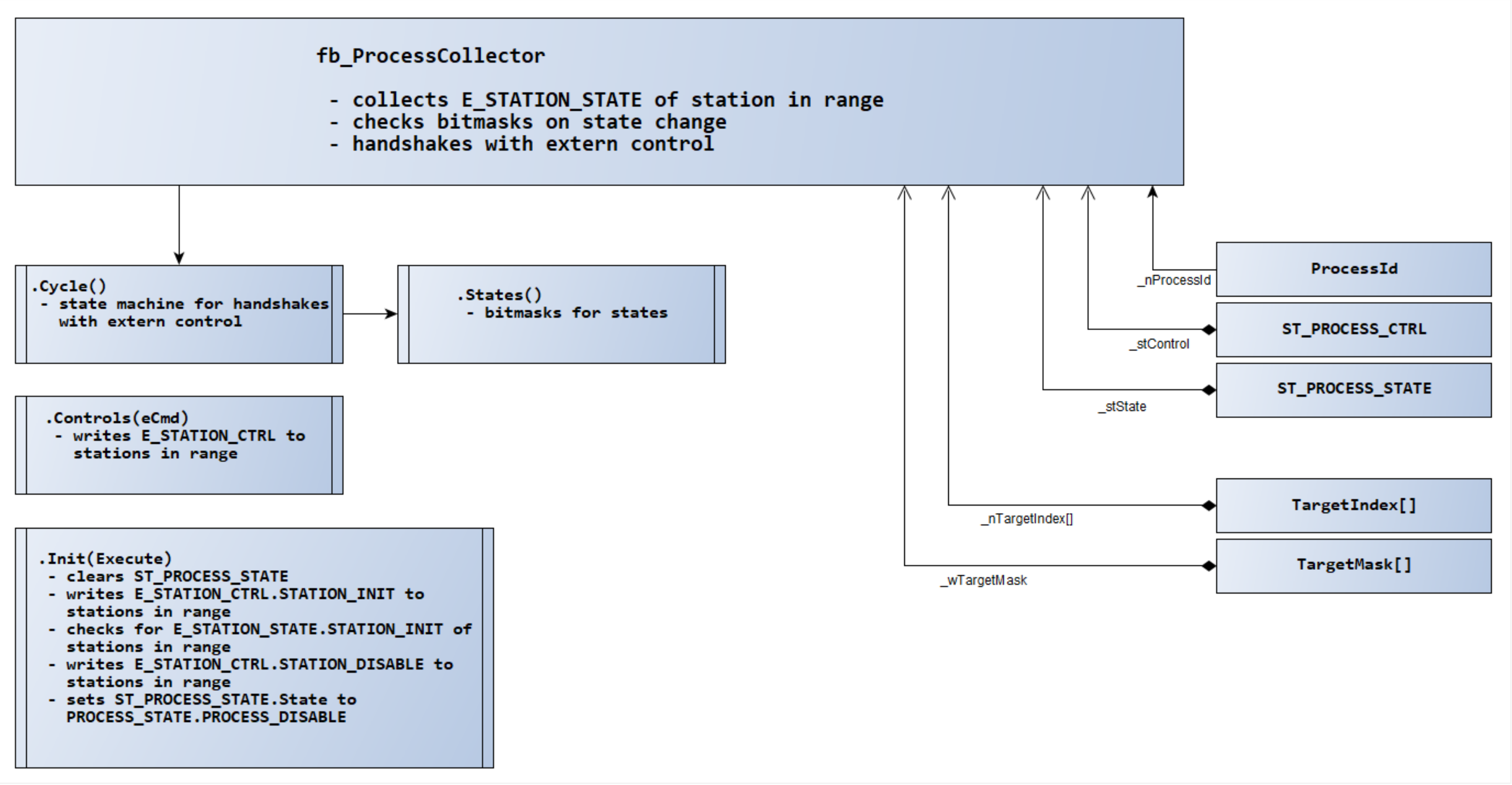
```
// state words for XtsStations in my range
  _wStateDisable      : T_PROCESS;
  _wStateEnable       : T_PROCESS;
  _wStateEnter        : T_PROCESS;
  _wStateStart        : T_PROCESS;
  _wStateDone         : T_PROCESS;
  _wStateNest         : T_PROCESS;
  _wStateOut          : T_PROCESS;
  _wStateGone         : T_PROCESS;
  _wError             : T_PROCESS;
  _wErrorFatal        : T_PROCESS;
```

Localisation

BECKHOFF

- **Station based approach**

- **PROCESS:**
  - **ProcessCollector:**
  - commands stations via dedicated structures
  - Checks range of activated stations if their states match the commanding bitmask.



```
fb_ProcessCollector

 - collects E_STATION_STATE of station in range
 - checks bitmasks on state change
 - handshakes with extern control
```

```
.Cycle()
 - state machine for handshakes
   with extern control
```

```
.States()
 - bitmasks for states
```

```
.Controls(eCmd)
 - writes E_STATION_CTRL to
   stations in range
```

```
.Init(Execute)
 - clears ST_PROCESS_STATE
 - writes E_STATION_CTRL.STATION_INIT to
   stations in range
 - checks for E_STATION_STATE.STATION_INIT of
   stations in range
 - writes E_STATION_CTRL.STATION_DISABLE to
   stations in range
 - sets ST_PROCESS_STATE.State to
   PROCESS_STATE.PROCESS_DISABLE
```

LICENSE

XTS_TRANSPORT_LAYER project