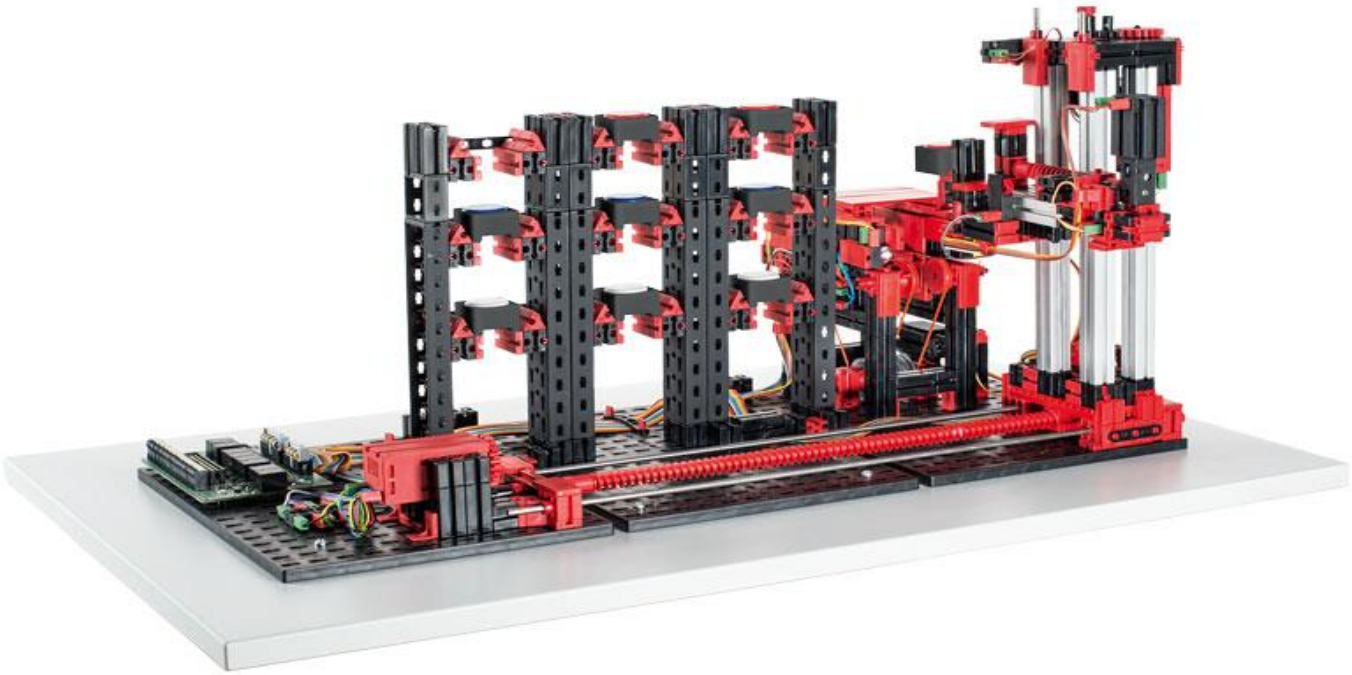
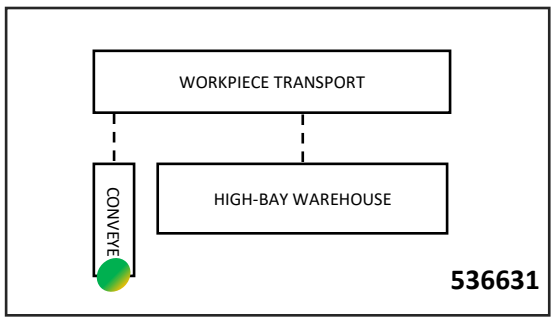
fischertechnik: Automated High-Bay warehouse

Station no. 536631

- Program design

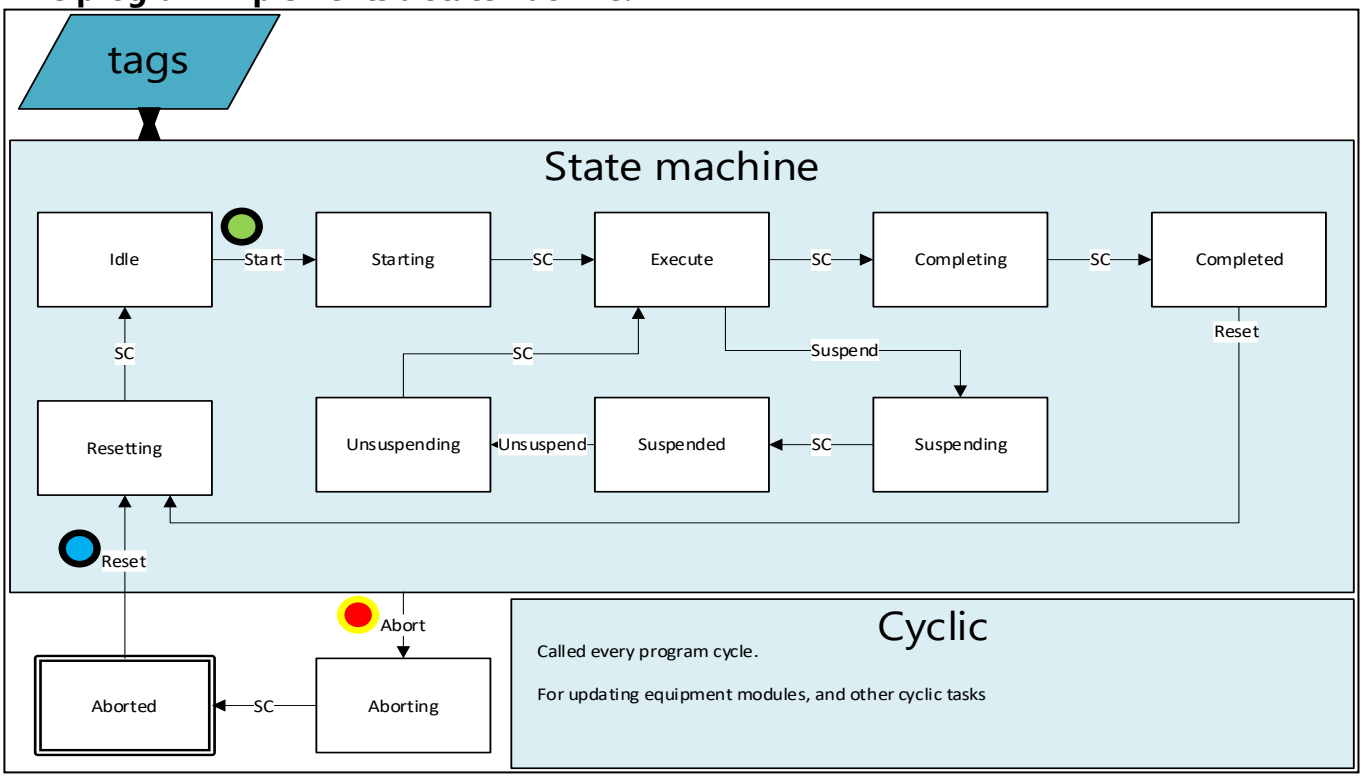




Pieces are placed and removed at this position

Program design

The program implements a statemachine.



Aborting	The init/startup state of the SFC program. No parts should be moving when the machine/unit is in this state.
Aborted	The resetting state should reset all the equipment modules. Any active errors are also reset in this state. All equipments are (if applicable) also initialized, when the initialization is done, the machine moves to next state.
Idle	The idle state indicates to the process cell that this unit can be started. The next state is only activated by a **start** command is given.
Starting	In the starting state things can be setup before moving to the execute state. For example starting a conveyer e.g.
Execute	The main exution takes place in this SFC state. In this state the operation on the piece(s) is performed. When the operation is done the next state is activated
Completing	The last things are wrapped up before moving to the execute state, for example stopping the belt e.g.
Completed	The last state of the SFC program, this state indicates to the process cell that the operations on the piece(s) are done. The next state (resetting) is activated by a **Reset** command.
Suspending	The suspend state can only be activated when the current state is Execute and a **Suspend** command is made. The suspend state prepares the program for the suspended state. E.g. slowing/stopping equipments
Suspended	In the suspended state the unit processes no pieces until the unsuspend command is given. Unlike the aborting/stopped state, the program can easily start again.
Unsuspending	In the unsuspending state preparations are made to move to execute state. E.g. starting/ramping up equipments. The unsuspeding state is activated by a **Unsuspend** command.

Program design

Tags.

The program implements tags in order to communicate with the ProcesCell (coordinates the different stations when ik factory config). These tags can also be changed during runtime by the user in order to control the station in a stand-alone configuration.

Status tags *used to get information from this station*

Name	***Comment***	***Datatype***
StateCurrent	The name of the current active state	STRING
ProdProcessedCount	No. Of products that have passed	INT
ErrorActive	TRUE when error is active in Unit	BOOL
WhitePieceCnt	no. of white pieces in the warehouse	INT
RedPieceCnt	no. of red pieces in the warehouse	INT
BluePieceCnt	no. of Blue pieces in the warehouse	INT

Command tags *used to set information in this station (to control the station)*

CntrlCmd		Control of the machine (start, reset, suspend e.g.)	ENUM ControlCommand
eProduct		The product we want to store or receive	ENUM.Product
GetPiece	**TRUE =**	Get a piece from the warehouse of eProduct type	BOOL
	FALSE =	Store a piece, eProduct value is saved	BOOL

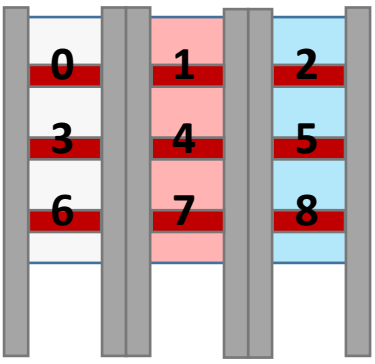
The station cannot be moved to execute when:

xGetpiece = TRUE & eProduct type not present in magazine

xGetpiece = FALSE & no more free spaces in high-bay warehouse

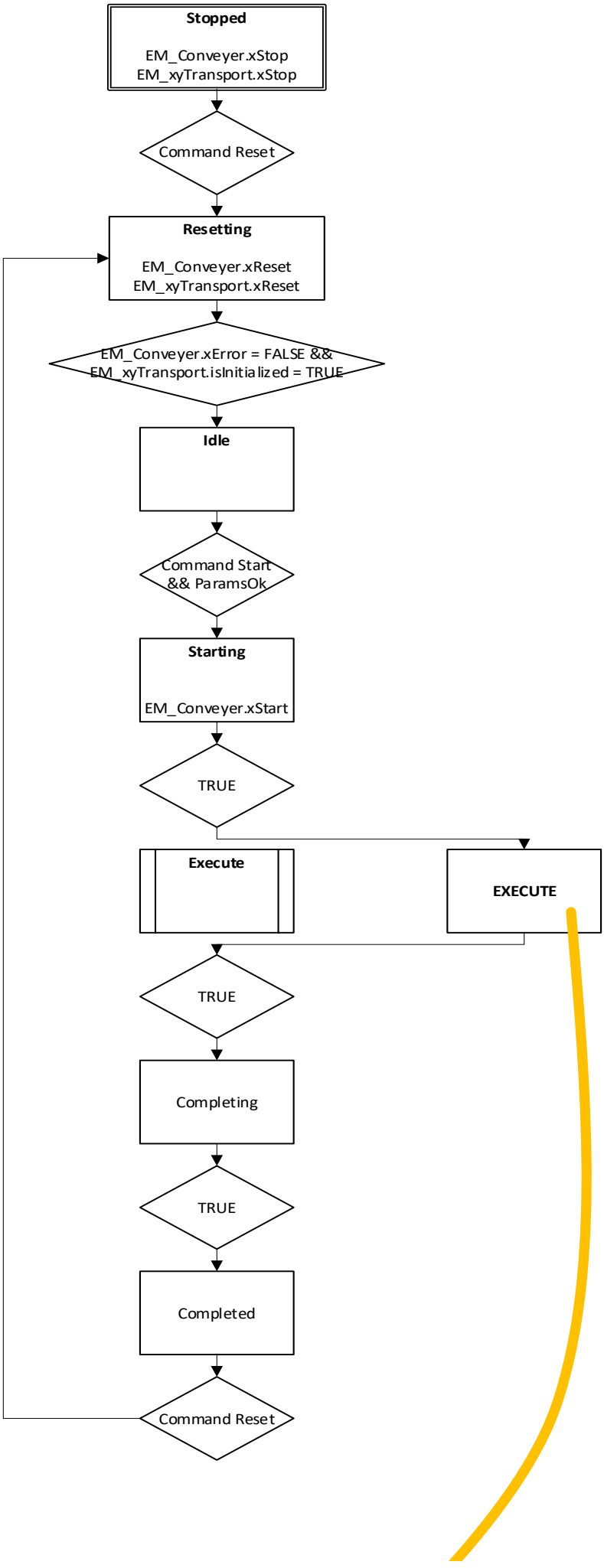
---

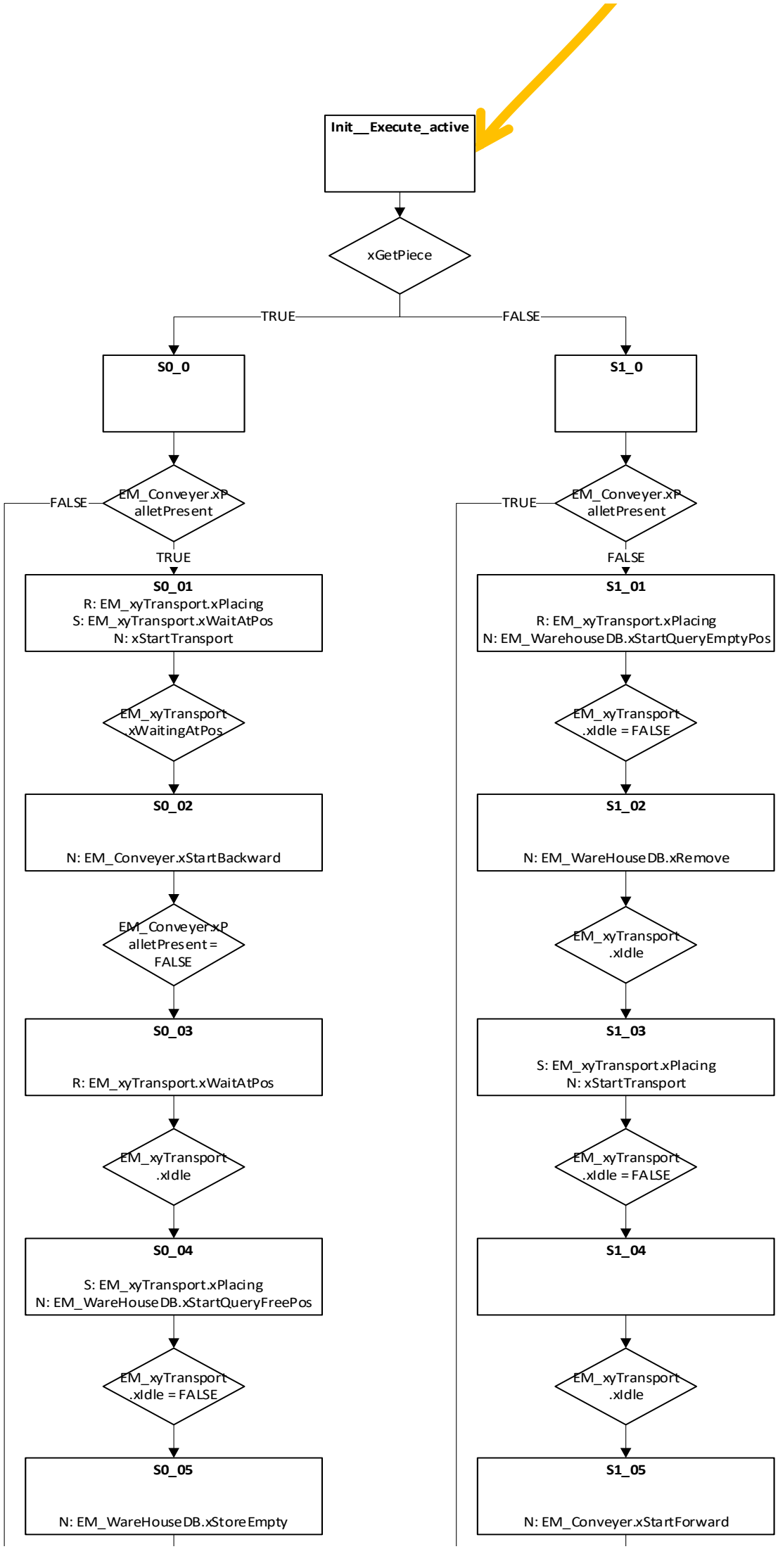
The high-Bay warehouse location are initialized at startup with the following values

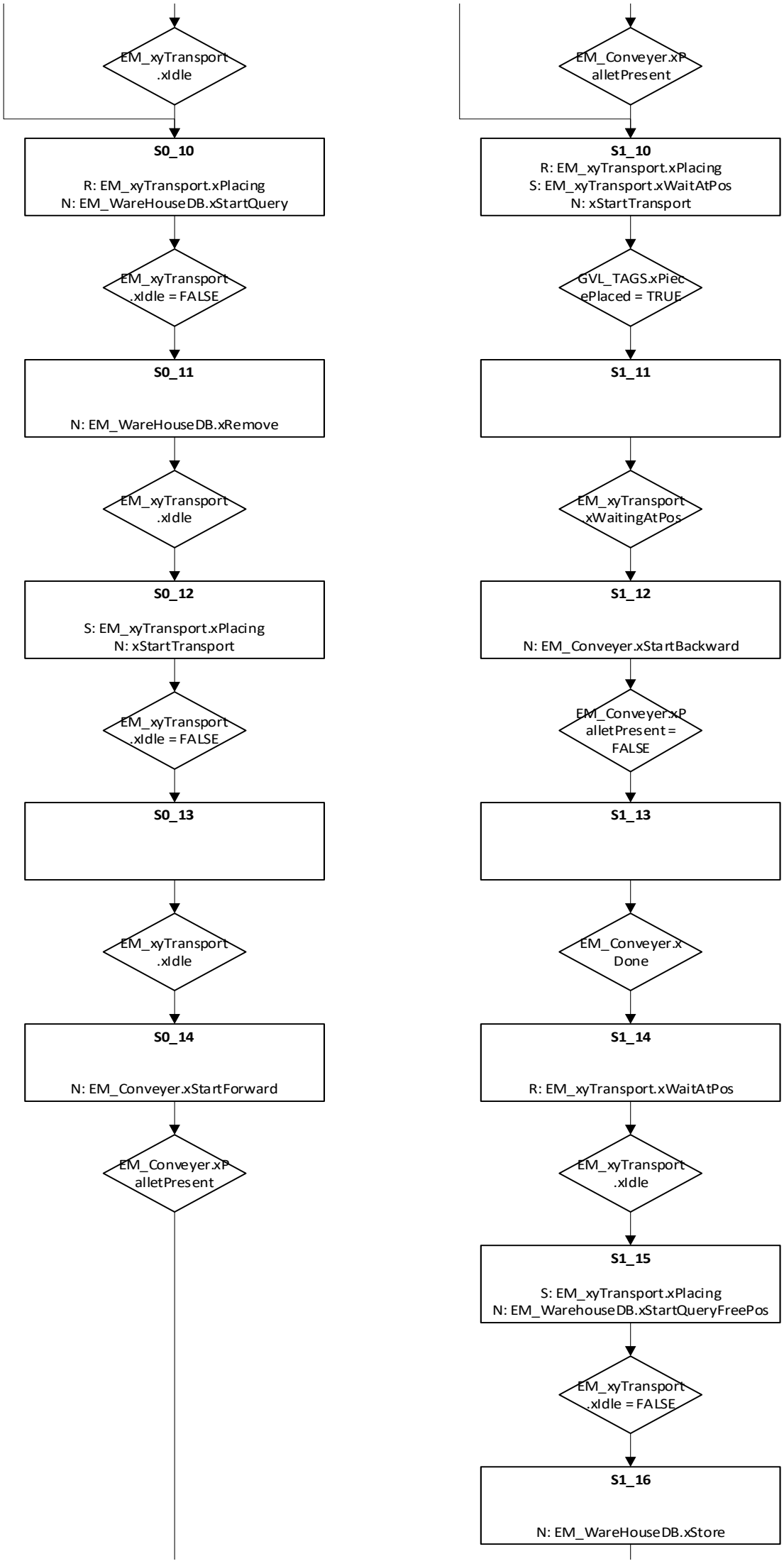


0, 3, 6	White pieces
1, 4, 7	Red pieces
2, 5, 8	Blue pieces

Program design | SFC program




Init__Execute_active
xGetPiece
TRUE
FALSE
S0_0
S1_0
EM_Conveyer.xPalletPresent
FALSE
TRUE
TRUE
FALSE
S0_01
R: EM_xyTransport.xPlacing
S: EM_xyTransport.xWaitAtPos
N: xStartTransport
S1_01
R: EM_xyTransport.xPlacing
N: EM_WarehouseDB.xStartQueryEmptyPos
EM_xyTransport.xWaitingAtPos
EM_xyTransport.xIdle = FALSE
S0_02
N: EM_Conveyer.xStartBackward
S1_02
N: EM_WareHouseDB.xRemove
EM_Conveyer.xPalletPresent = FALSE
EM_xyTransport.xIdle
S0_03
R: EM_xyTransport.xWaitAtPos
S1_03
S: EM_xyTransport.xPlacing
N: xStartTransport
EM_xyTransport.xIdle
EM_xyTransport.xIdle = FALSE
S0_04
S: EM_xyTransport.xPlacing
N: EM_WareHouseDB.xStartQueryFreePos
S1_04
EM_xyTransport.xIdle = FALSE
EM_xyTransport.xIdle
S0_05
N: EM_WareHouseDB.xStoreEmpty
S1_05
N: EM_Conveyer.xStartForward


EM_xyTransport .xIdle
S0_10
R: EM_xyTransport.xPlacing
N: EM_WareHouseDB.xStartQuery
EM_xyTransport .xIdle = FALSE
S0_11
N: EM_WareHouseDB.xRemove
EM_xyTransport .xIdle
S0_12
S: EM_xyTransport.xPlacing
N: xStartTransport
EM_xyTransport .xIdle = FALSE
S0_13
EM_xyTransport .xIdle
S0_14
N: EM_Conveyer.xStartForward
EM_Conveyer.xP alletPresent
EM_Conveyer.xP alletPresent
S1_10
R: EM_xyTransport.xPlacing
S: EM_xyTransport.xWaitAtPos
N: xStartTransport
GVL_TAGS.xPiec ePlaced = TRUE
S1_11
EM_xyTransport .xWaitingAtPos
S1_12
N: EM_Conveyer.xStartBackward
EM_Conveyer.xP alletPresent = FALSE
S1_13
EM_Conveyer.x Done
S1_14
R: EM_xyTransport.xWaitAtPos
EM_xyTransport .xIdle
S1_15
S: EM_xyTransport.xPlacing
N: EM_WarehouseDB.xStartQueryFreePos
EM_xyTransport .xIdle = FALSE
S1_16
N: EM_WareHouseDB.xStore


EM_xyTransport
.xIdle
EXECUTE_DONE