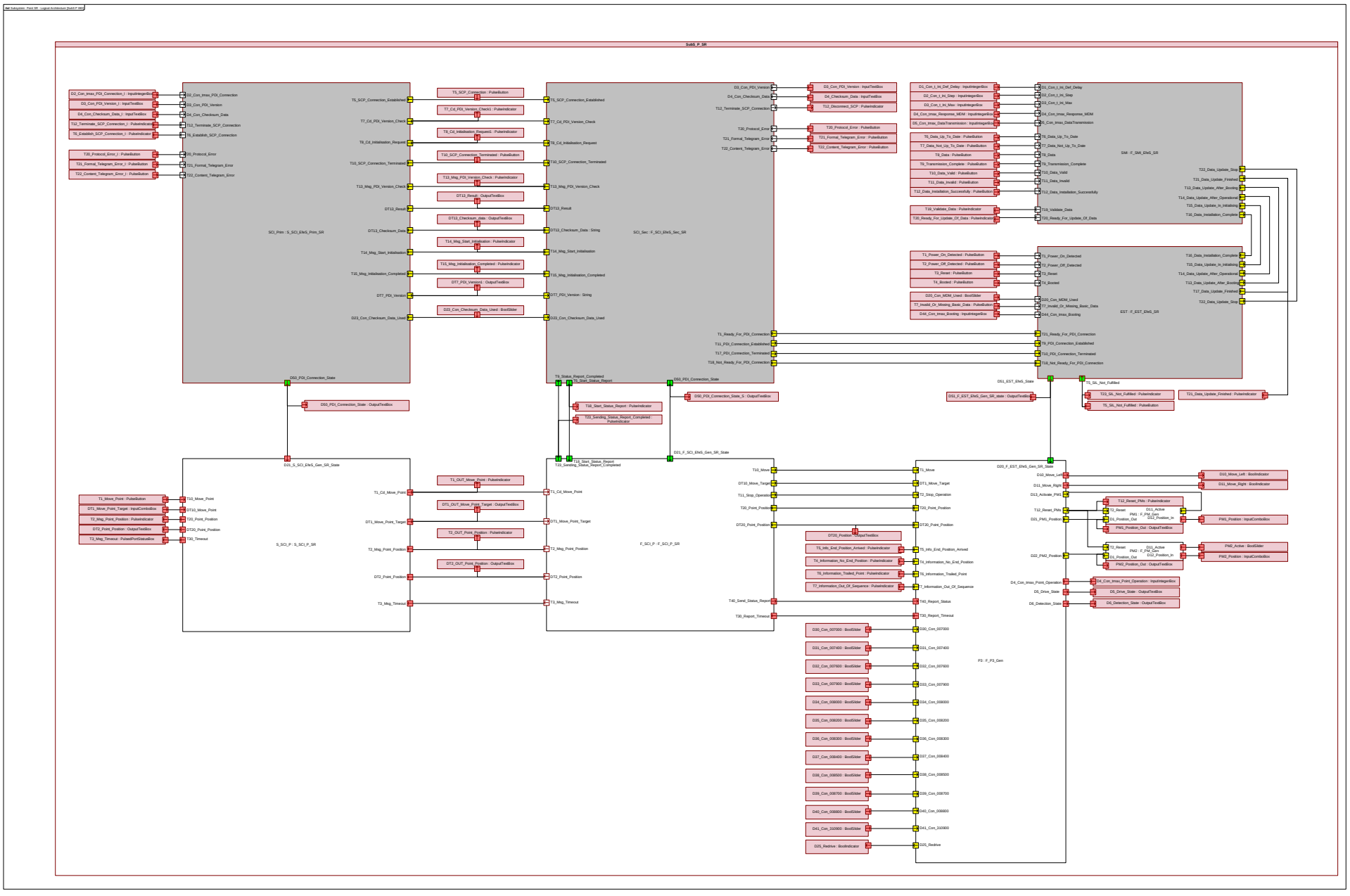
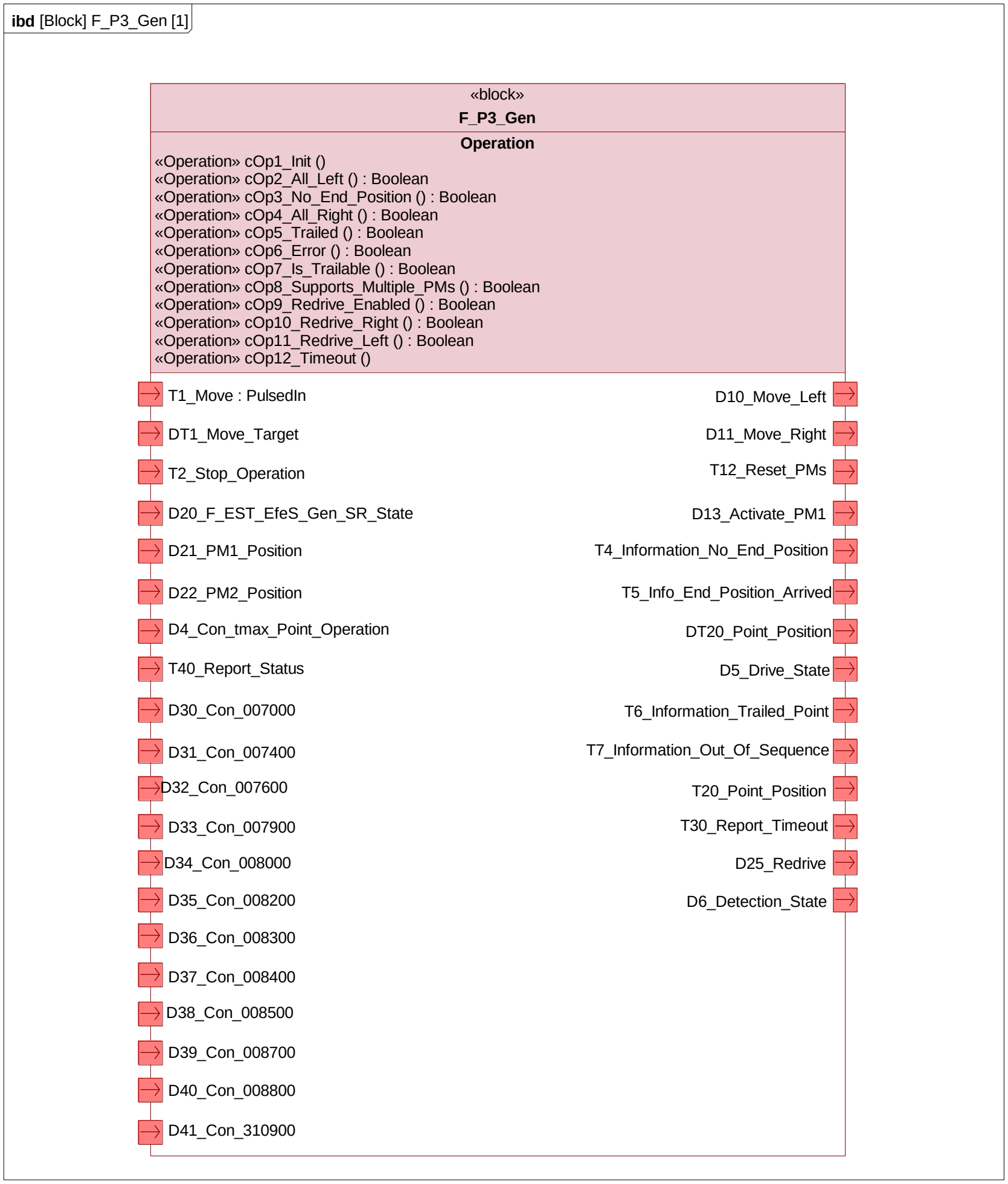
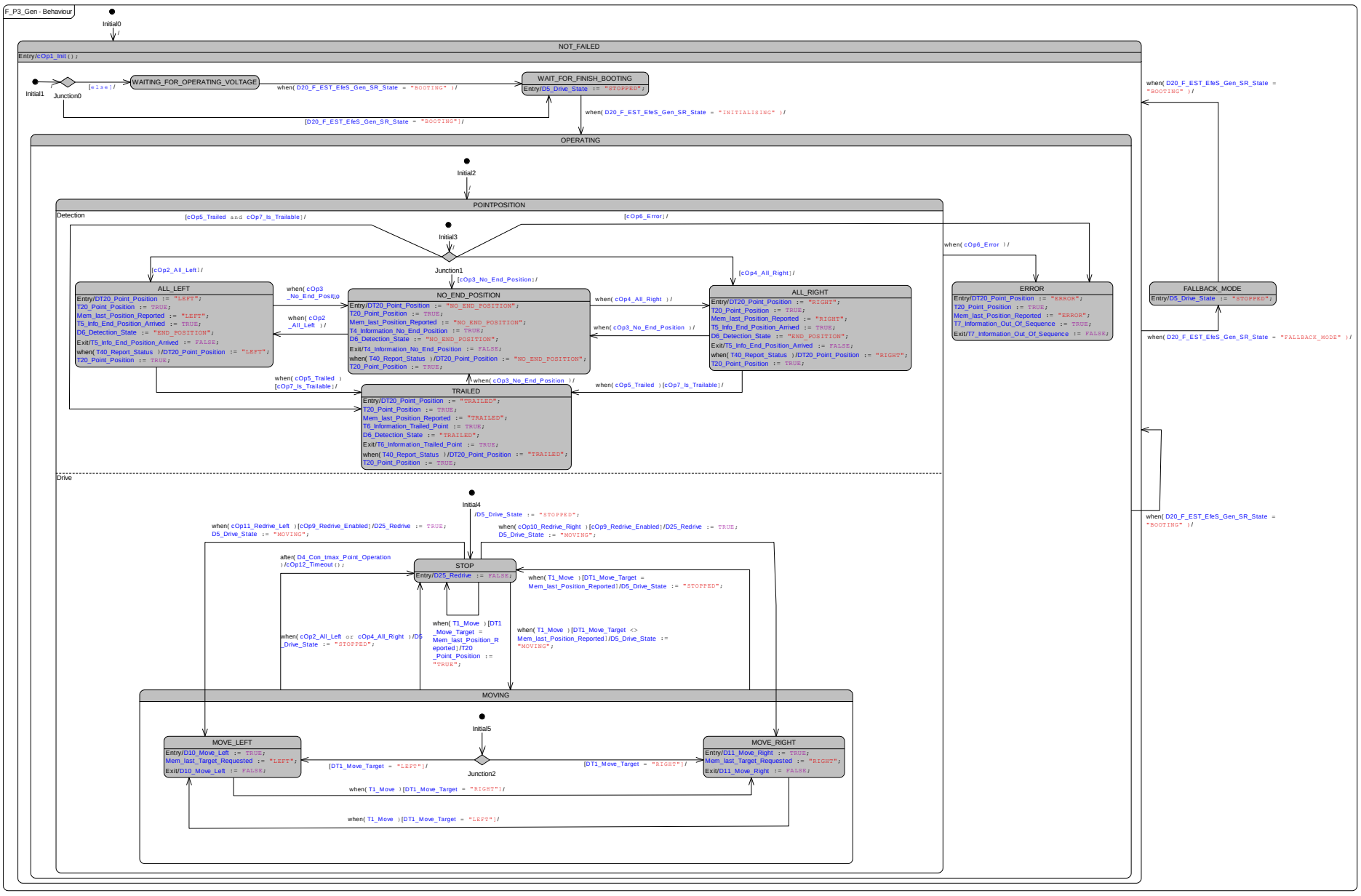
SUP_P_SR Logical Architecture

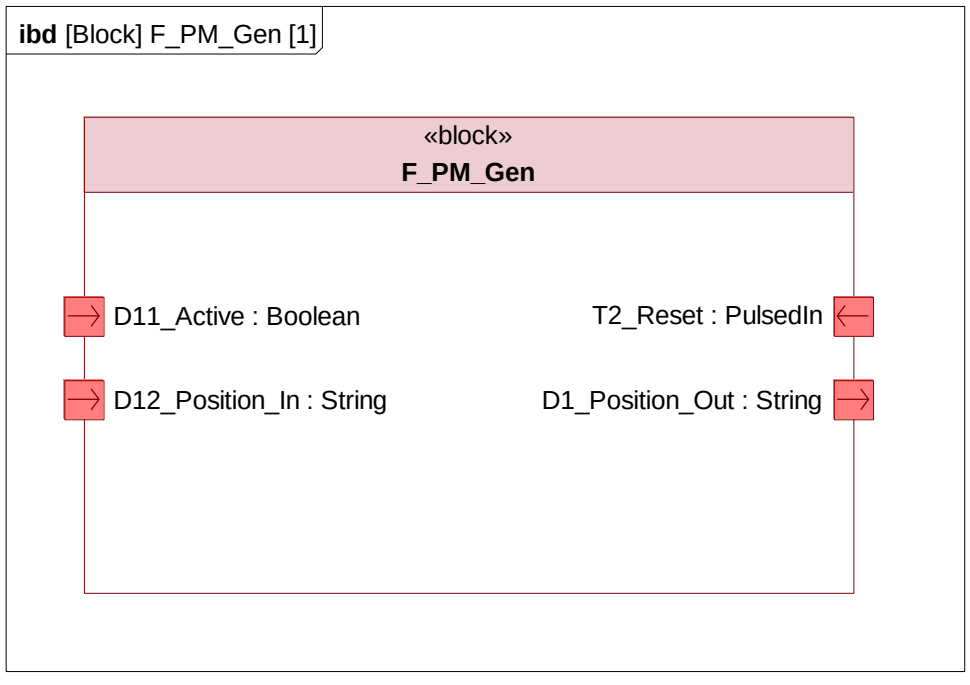


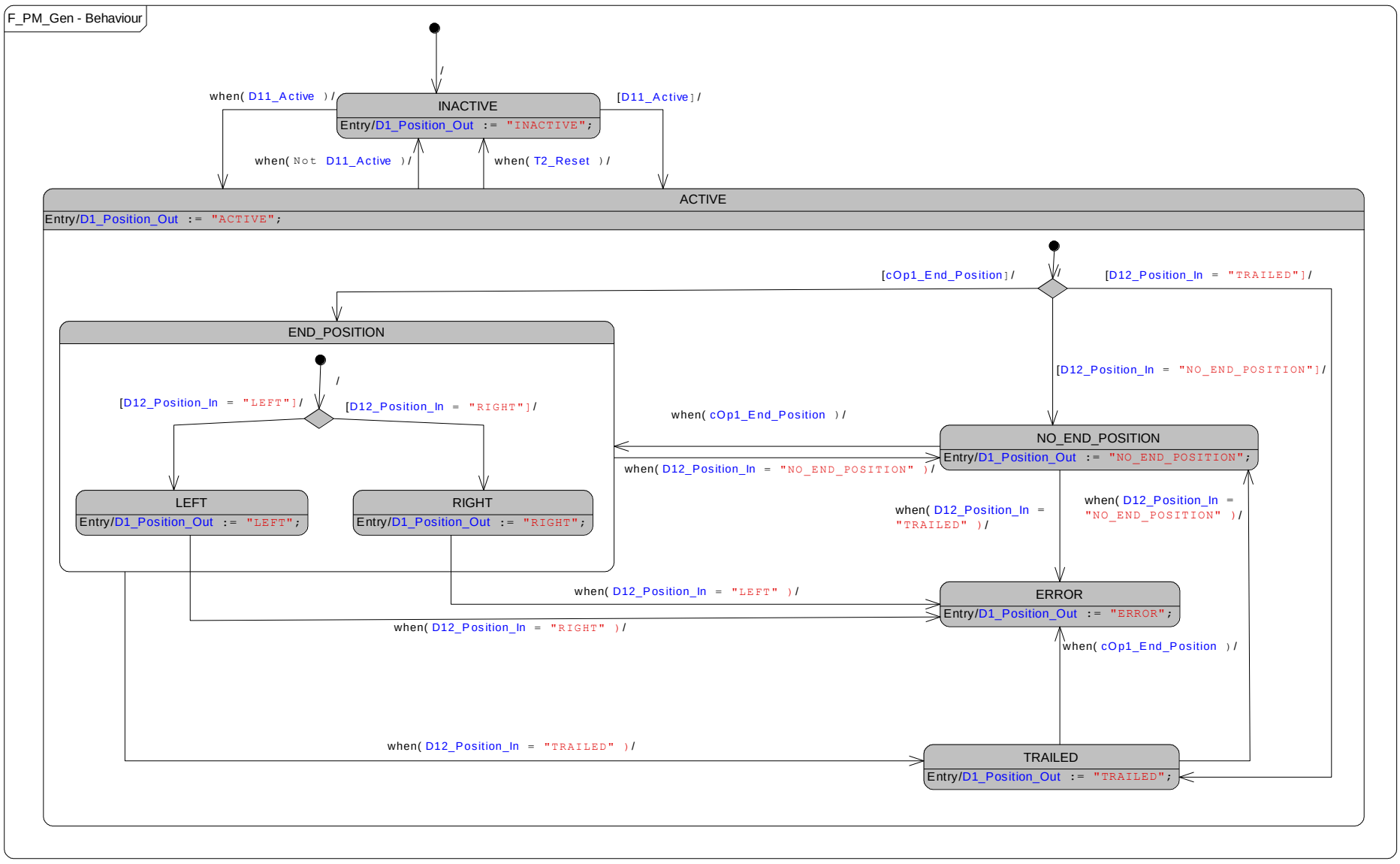
F_P3_Gen


ibd [Block] F_P3_Gen [1]
«block»
F_P3_Gen
Operation
«Operation» cOp1_Init ()
«Operation» cOp2_All_Left () : Boolean
«Operation» cOp3_No_End_Position () : Boolean
«Operation» cOp4_All_Right () : Boolean
«Operation» cOp5_Trailed () : Boolean
«Operation» cOp6_Error () : Boolean
«Operation» cOp7_Is_Trailable () : Boolean
«Operation» cOp8_Supports_Multiple_PMs () : Boolean
«Operation» cOp9_Redrive_Enabled () : Boolean
«Operation» cOp10_Redrive_Right () : Boolean
«Operation» cOp11_Redrive_Left () : Boolean
«Operation» cOp12_Timeout ()
T1_Move : PulsedIn
DT1_Move_Target
T2_Stop_Operation
D20_F_EST_EfeS_Gen_SR_State
D21_PM1_Position
D22_PM2_Position
D4_Con_tmax_Point_Operation
T40_Report_Status
D30_Con_007000
D31_Con_007400
D32_Con_007600
D33_Con_007900
D34_Con_008000
D35_Con_008200
D36_Con_008300
D37_Con_008400
D38_Con_008500
D39_Con_008700
D40_Con_008800
D41_Con_310900
D10_Move_Left
D11_Move_Right
T12_Reset_PMs
D13_Activate_PM1
T4_Information_No_End_Position
T5_Info_End_Position_Arrived
DT20_Point_Position
D5_Drive_State
T6_Information_Trailed_Point
T7_Information_Out_Of_Sequence
T20_Point_Position
T30_Report_Timeout
D25_Redrive
D6_Detection_State


F_P3_Gen - Behaviour
Initial0
/
NOT_FAILED
Entry/cOp1_Init();
Initial1
Junction0
[else]/
WAITING_FOR_OPERATING_VOLTAGE
when(D20_F_EST_EfeS_Gen_SR_State = "BOOTING")/
WAIT_FOR_FINISH_BOOTING
Entry/D5_Drive_State := "STOPPED";
[D20_F_EST_EfeS_Gen_SR_State = "BOOTING"]/
when(D20_F_EST_EfeS_Gen_SR_State = "INITIALISING")/
when(D20_F_EST_EfeS_Gen_SR_State = "BOOTING")/
OPERATING
Initial2
/
POINTPOSITION
Detection
[cOp5_Trailed and cOp7_Is_Trailable]/
[cOp6_Error]/
Initial3
/
when(cOp6_Error)/
[cOp2_All_Left]/
Junction1
[cOp3_No_End_Position]/
[cOp4_All_Right]/
ALL_LEFT
Entry/DT20_Point_Position := "LEFT";
T20_Point_Position := TRUE;
Mem_last_Position_Reported := "LEFT";
T5_Info_End_Position_Arrived := TRUE;
D6_Detection_State := "END_POSITION";
Exit/T5_Info_End_Position_Arrived := FALSE;
when(T40_Report_Status)/DT20_Point_Position := "LEFT";
T20_Point_Position := TRUE;
when(cOp3_No_End_Positio
when(cOp2_All_Left)/
NO_END_POSITION
Entry/DT20_Point_Position := "NO_END_POSITION";
T20_Point_Position := TRUE;
Mem_last_Position_Reported := "NO_END_POSITION";
T4_Information_No_End_Position := TRUE;
D6_Detection_State := "NO_END_POSITION";
Exit/T4_Information_No_End_Position := FALSE;
when(T40_Report_Status)/DT20_Point_Position := "NO_END_POSITION";
T20_Point_Position := TRUE;
when(cOp4_All_Right)/
when(cOp3_No_End_Position)/
ALL_RIGHT
Entry/DT20_Point_Position := "RIGHT";
T20_Point_Position := TRUE;
Mem_last_Position_Reported := "RIGHT";
T5_Info_End_Position_Arrived := TRUE;
D6_Detection_State := "END_POSITION";
Exit/T5_Info_End_Position_Arrived := FALSE;
when(T40_Report_Status)/DT20_Point_Position := "RIGHT";
T20_Point_Position := TRUE;
ERROR
Entry/DT20_Point_Position := "ERROR";
T20_Point_Position := TRUE;
Mem_last_Position_Reported := "ERROR";
T7_Information_Out_Of_Sequence := TRUE;
Exit/T7_Information_Out_Of_Sequence := FALSE;
FALLBACK_MODE
Entry/D5_Drive_State := "STOPPED";
when(D20_F_EST_EfeS_Gen_SR_State = "FALLBACK_MODE")/
when(cOp5_Trailed)
[cOp7_Is_Trailable]/
when(cOp3_No_End_Position)/
when(cOp5_Trailed)[cOp7_Is_Trailable]/
TRAILED
Entry/DT20_Point_Position := "TRAILED";
T20_Point_Position := TRUE;
Mem_last_Position_Reported := "TRAILED";
T6_Information_Trailed_Point := TRUE;
D6_Detection_State := "TRAILED";
Exit/T6_Information_Trailed_Point := TRUE;
when(T40_Report_Status)/DT20_Point_Position := "TRAILED";
T20_Point_Position := TRUE;
Drive
Initial4
/D5_Drive_State := "STOPPED";
when(cOp11_Redrive_Left)[cOp9_Redrive_Enabled]/D25_Redrive := TRUE;
D5_Drive_State := "MOVING";
when(cOp10_Redrive_Right)[cOp9_Redrive_Enabled]/D25_Redrive := TRUE;
D5_Drive_State := "MOVING";
after(D4_Con_tmax_Point_Operation)/cOp12_Timeout();
STOP
Entry/D25_Redrive := FALSE;
when(T1_Move)[DT1_Move_Target = Mem_last_Position_Reported]/D5_Drive_State := "STOPPED";
when(cOp2_All_Left or cOp4_All_Right)/D5_Drive_State := "STOPPED";
when(T1_Move)[DT1_Move_Target = Mem_last_Position_Reported]/T20_Point_Position := "TRUE";
when(T1_Move)[DT1_Move_Target <> Mem_last_Position_Reported]/D5_Drive_State := "MOVING";
MOVING
Initial5
MOVE_LEFT
Entry/D10_Move_Left := TRUE;
Mem_last_Target_Requested := "LEFT";
Exit/D10_Move_Left := FALSE;
[DT1_Move_Target = "LEFT"]/
Junction2
[DT1_Move_Target = "RIGHT"]/
MOVE_RIGHT
Entry/D11_Move_Right := TRUE;
Mem_last_Target_Requested := "RIGHT";
Exit/D11_Move_Right := FALSE;
when(T1_Move)[DT1_Move_Target = "RIGHT"]/
when(T1_Move)[DT1_Move_Target = "LEFT"]/

F_PM_Gen


ibd [Block] F_PM_Gen [1]
«block»
F_PM_Gen
D11_Active : Boolean
D12_Position_In : String
T2_Reset : PulsedIn
D1_Position_Out : String


F_PM_Gen - Behaviour
/
when(D11_Active)/
[D11_Active]/
INACTIVE
Entry/D1_Position_Out := "INACTIVE";
when(Not D11_Active)/
when(T2_Reset)/
ACTIVE
Entry/D1_Position_Out := "ACTIVE";
[cOp1_End_Position]/
/
[D12_Position_In = "TRAILED"]/
END_POSITION
/
[D12_Position_In = "LEFT"]/
[D12_Position_In = "RIGHT"]/
LEFT
Entry/D1_Position_Out := "LEFT";
RIGHT
Entry/D1_Position_Out := "RIGHT";
[D12_Position_In = "NO_END_POSITION"]/
when(cOp1_End_Position)/
when(D12_Position_In = "NO_END_POSITION")/
NO_END_POSITION
Entry/D1_Position_Out := "NO_END_POSITION";
when(D12_Position_In = "TRAILED")/
when(D12_Position_In = "NO_END_POSITION")/
when(D12_Position_In = "LEFT")/
when(D12_Position_In = "RIGHT")/
ERROR
Entry/D1_Position_Out := "ERROR";
when(cOp1_End_Position)/
when(D12_Position_In = "TRAILED")/
TRAILED
Entry/D1_Position_Out := "TRAILED";