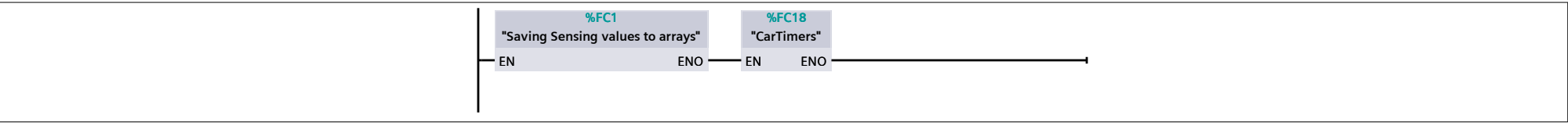


Program blocks

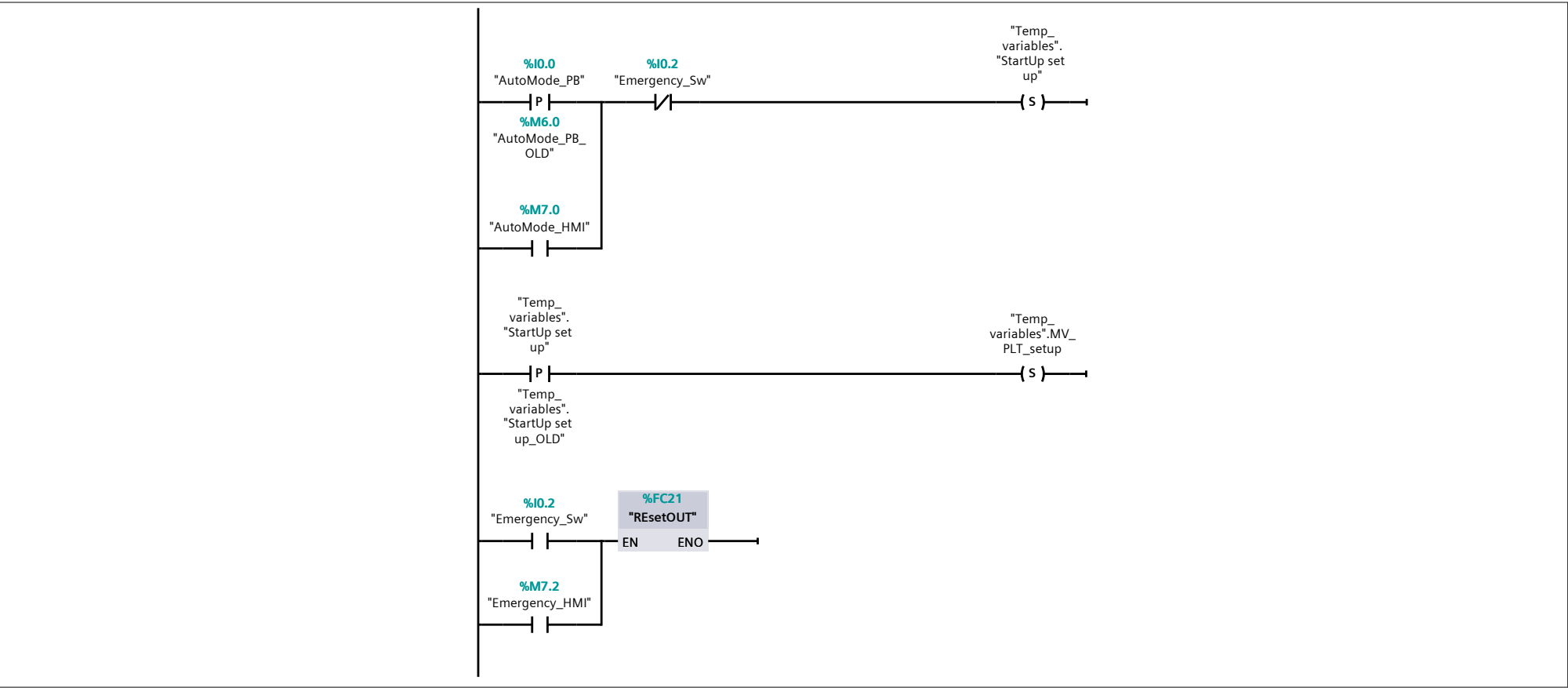
Main [OB1]

Main Properties							
General							
Name	Main	Number	1	Type	OB	Language	LAD
Numbering	Automatic						
Information							
Title	"Main Program Sweep (Cycle)"	Author		Comment		Family	
Version	0.1	User-defined ID					
Name			Data type		Default value		
▼ Input							
Initial_Call			Bool				
Remanence			Bool				
▼ Temp							
Calculating ID			Int				
car status			Bool				
Constant							

Network 1: Functions initiallization



Network 2: Sart and stop the system



Network 3: Start up set up (Moving platform)

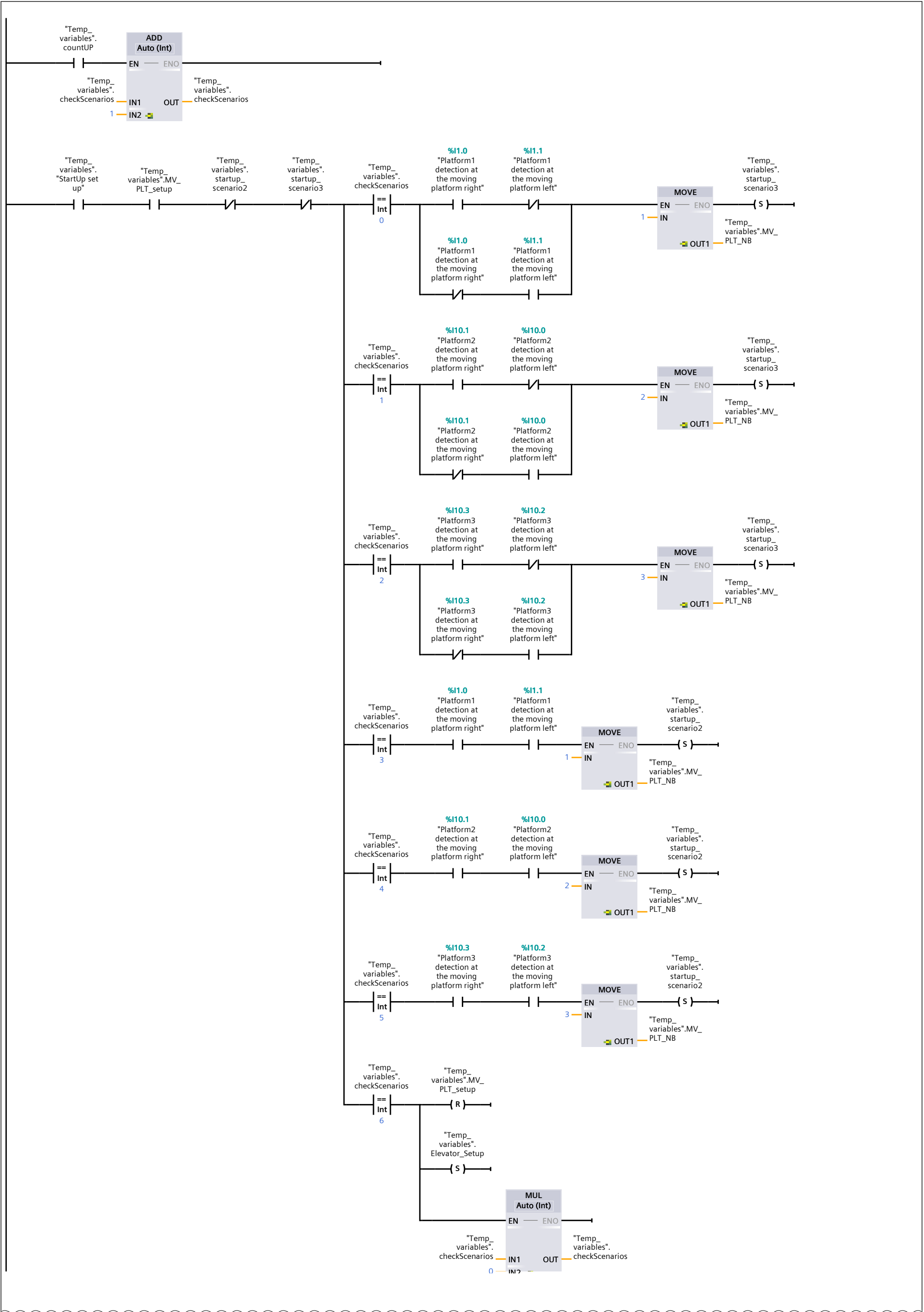
we want first to make sure that everything in the system is in the right position .

the aim of this function is to check the different scenario that the moving platform can be , this check will be applied for all the moving platform , but the operation will be sequential (only one will move per time )

scenario 2 the moving platform already contains a platform

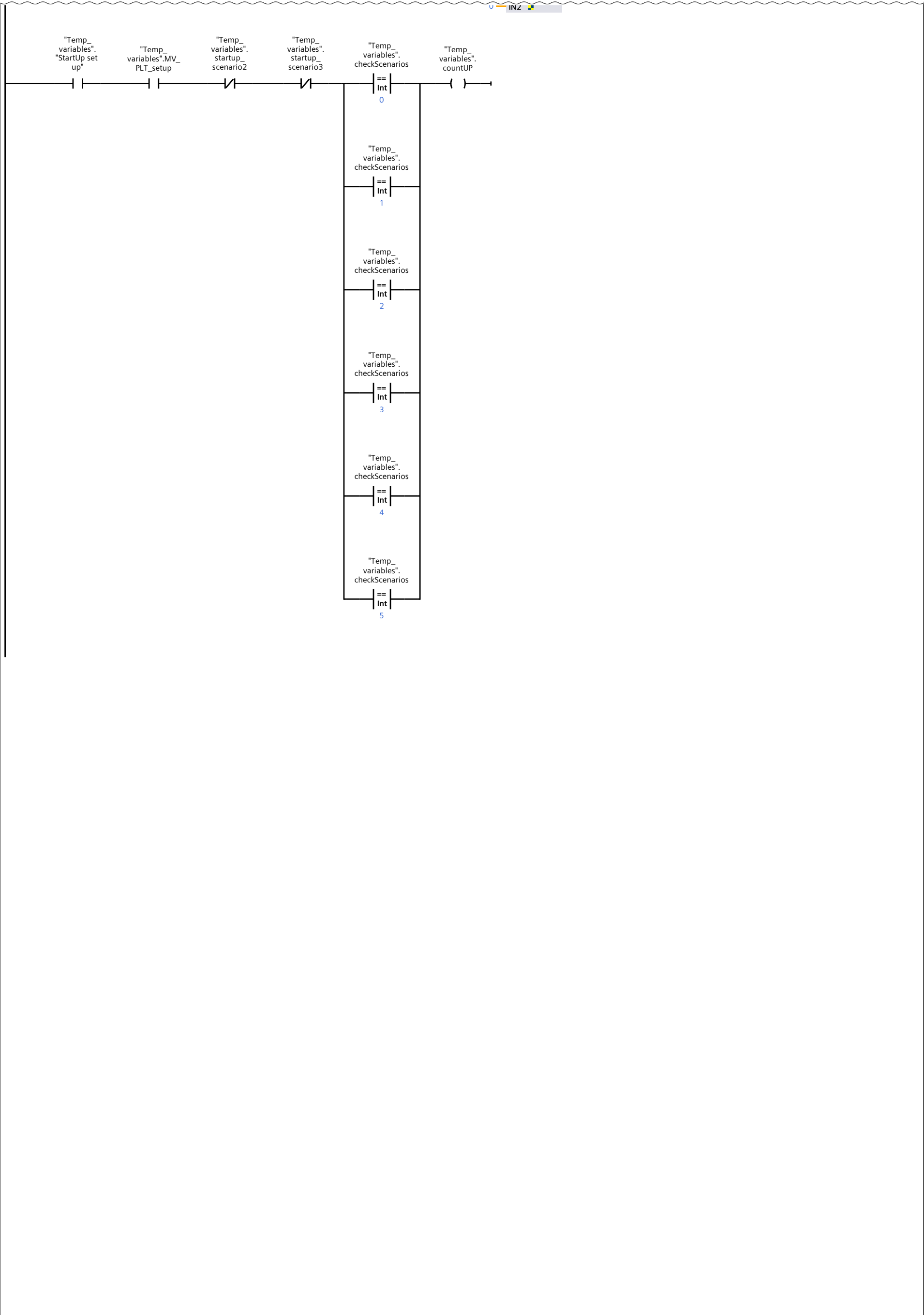
scenario 3 the car platform is not fully attached at the moving platform

Network 3: Start up set up (Moving platform) (1.1 / 2.1)



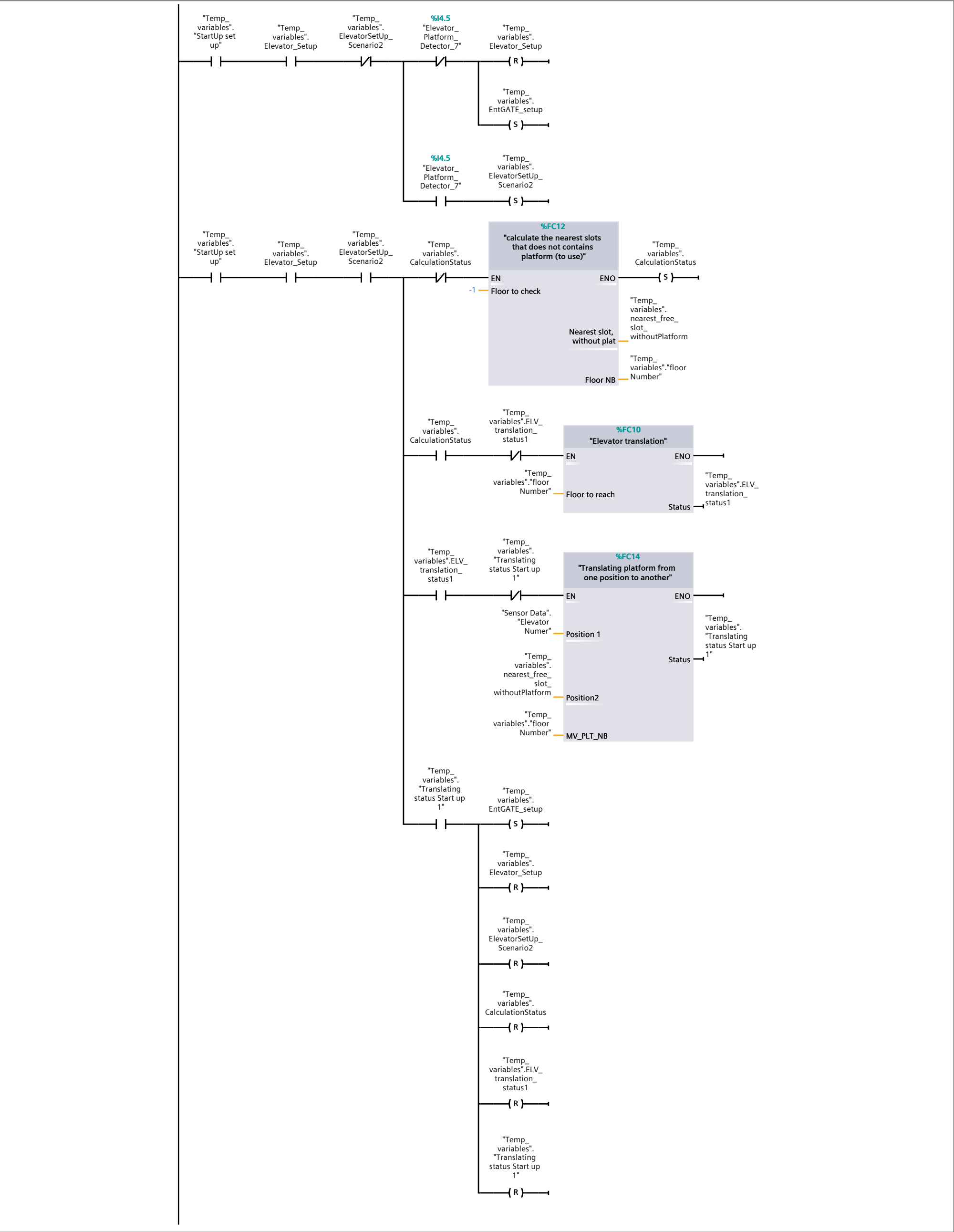
Network 3: Start up set up (Moving platform) (2.1 / 2.1)

1.1 ( Page1 - 2)



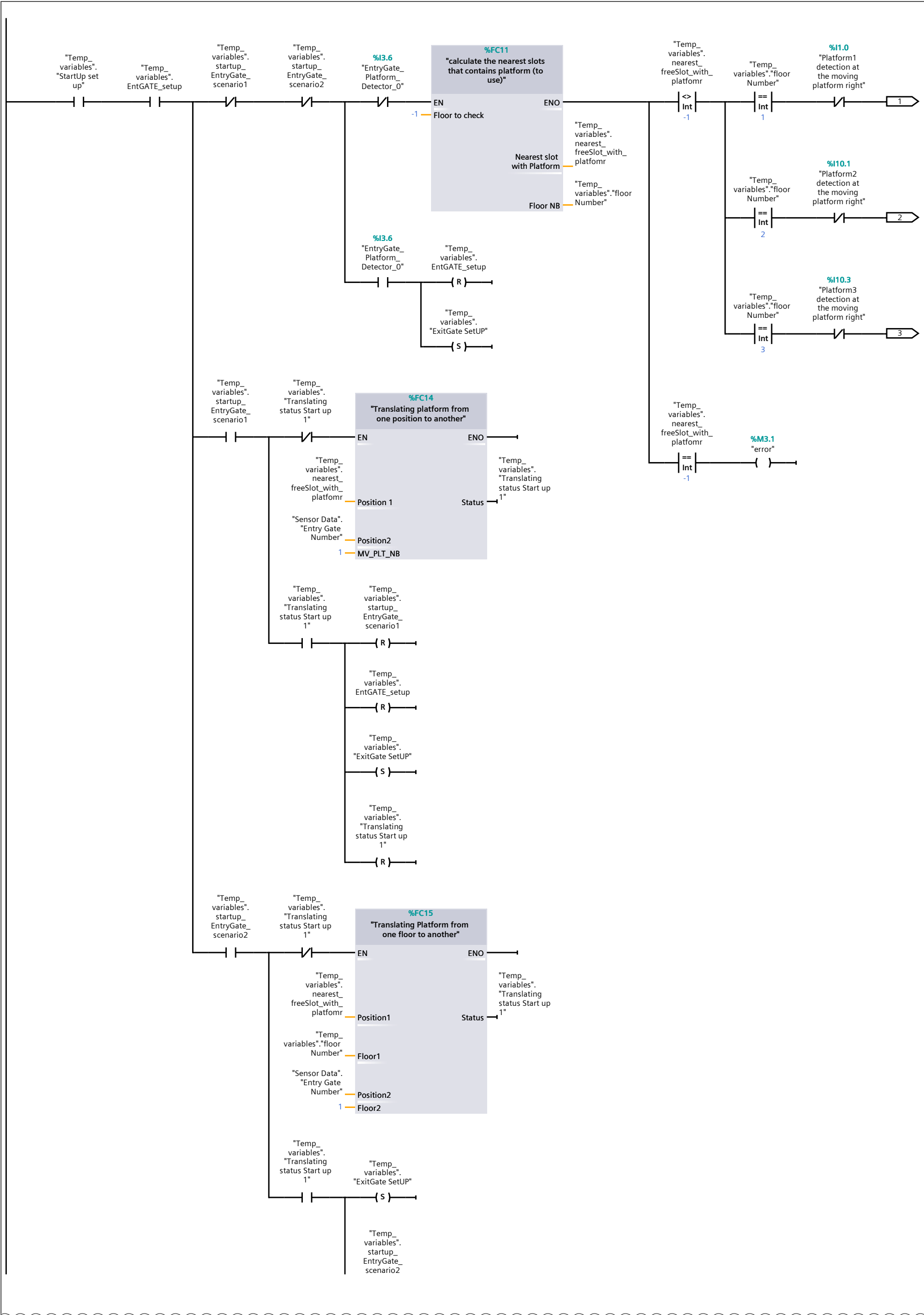
Network 4: Start up set up (Elevator)

check if there is a platform in the elevator,  
if there is one the system will search for the nearest slot that does not cointains a platform  
the elevator will go to that floor  
the moving platform will take this platform from the elevator to the free slot

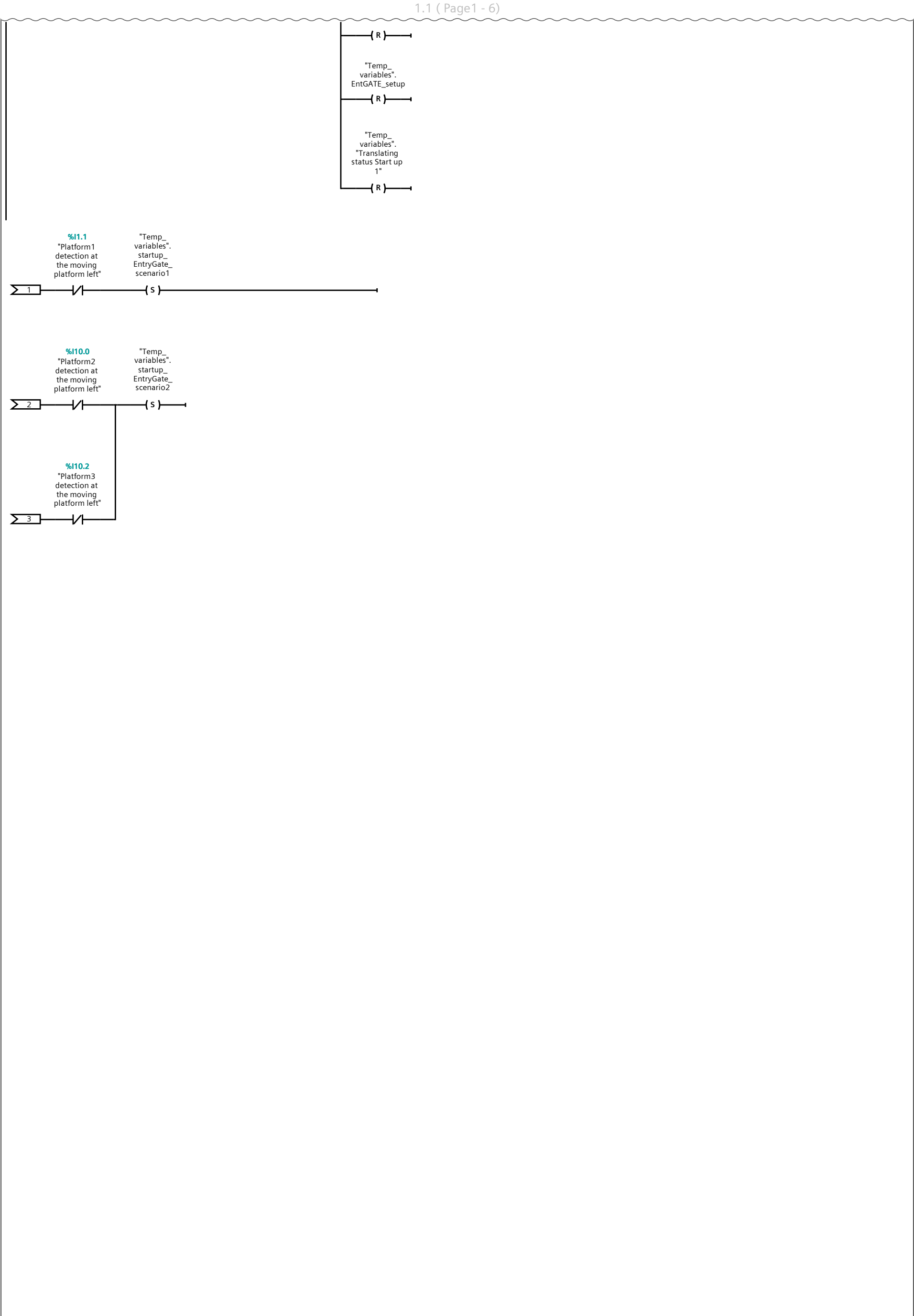


Totally Integrated Automation Portal		
<p><b>Network 5: Start up set up (Entry gate)</b></p> <p>we want the car platform to be Available at the entry gate , to do so we are going to make sure that the moving platform don't have a car platform we are going to check first the nearest free spot that contains a car platform , if we didn't find any we will get a error</p> <p>After finishing from this step we want to make sure that there is no platfomr at the exit gate and if there is one we are going to let it back to it's position</p>		

Network 5: Start up set up (Entry gate) (1.1 / 2.1)



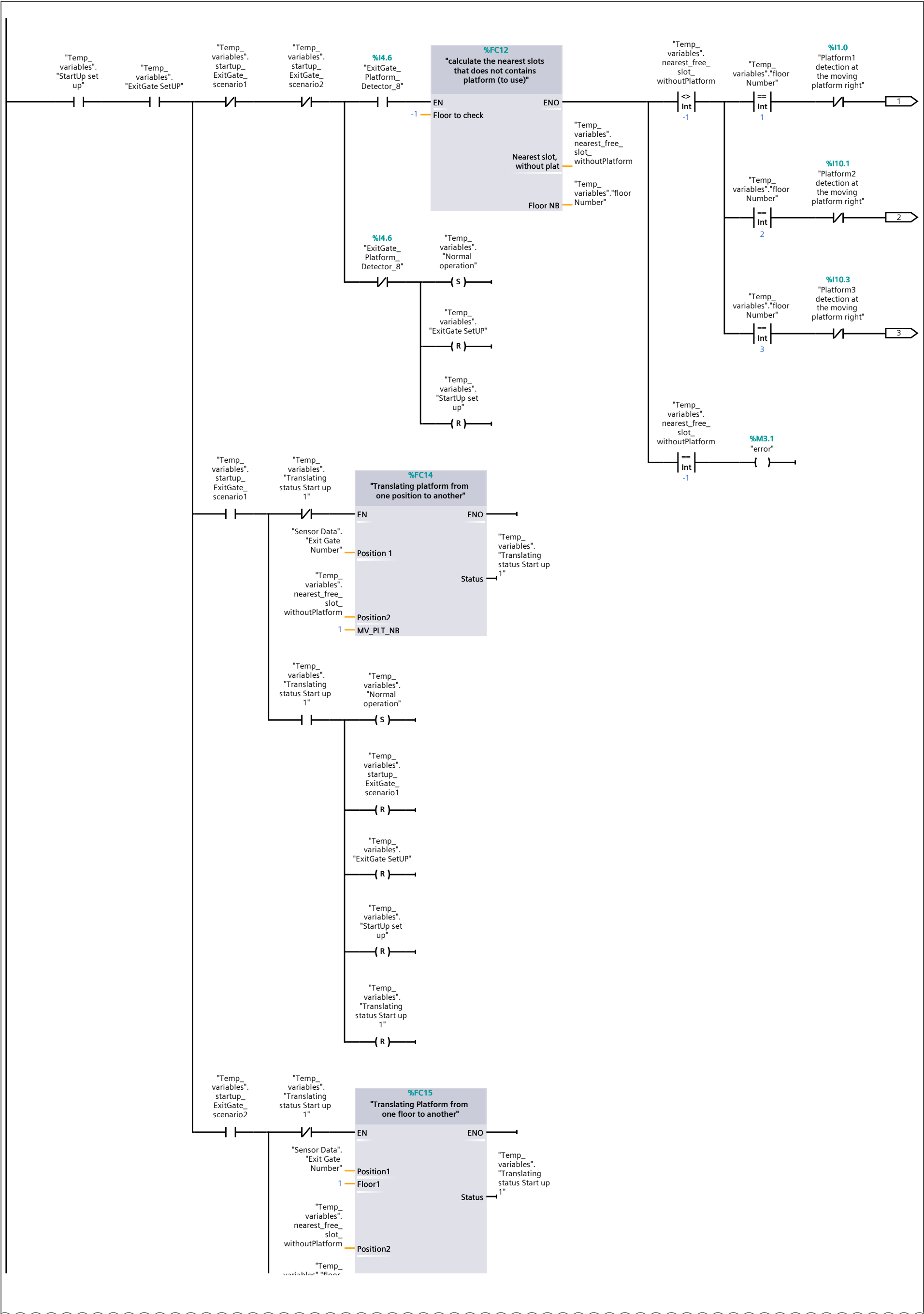
Network 5: Start up set up (Entry gate) (2.1 / 2.1)



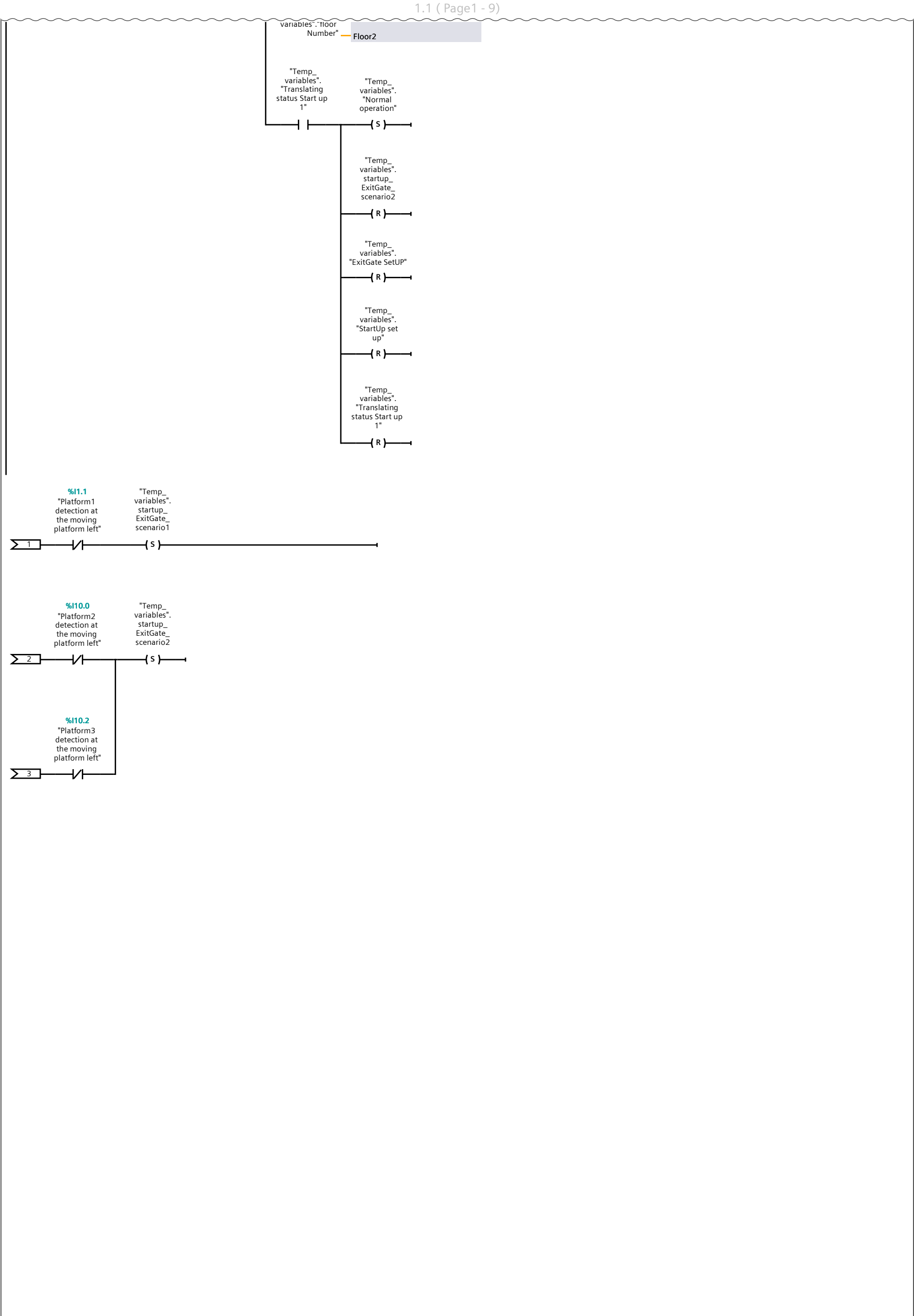
Totally Integrated Automation Portal		
<div>Network 6: Start up set up ( Exit gate)</div> <div>the system will check if there is a platform at the exit gate : if no : start normal operation if yes : check the scenarios , and take this platform and put it in the right place then start the normal operation Scenario 1 : the free slot is in the floor 1 scenario 2 : the free slot is on the floor 2 or 3 we need to use the elevator</div>		



Network 6: Start up set up ( Exit gate) (1.1 / 2.1)

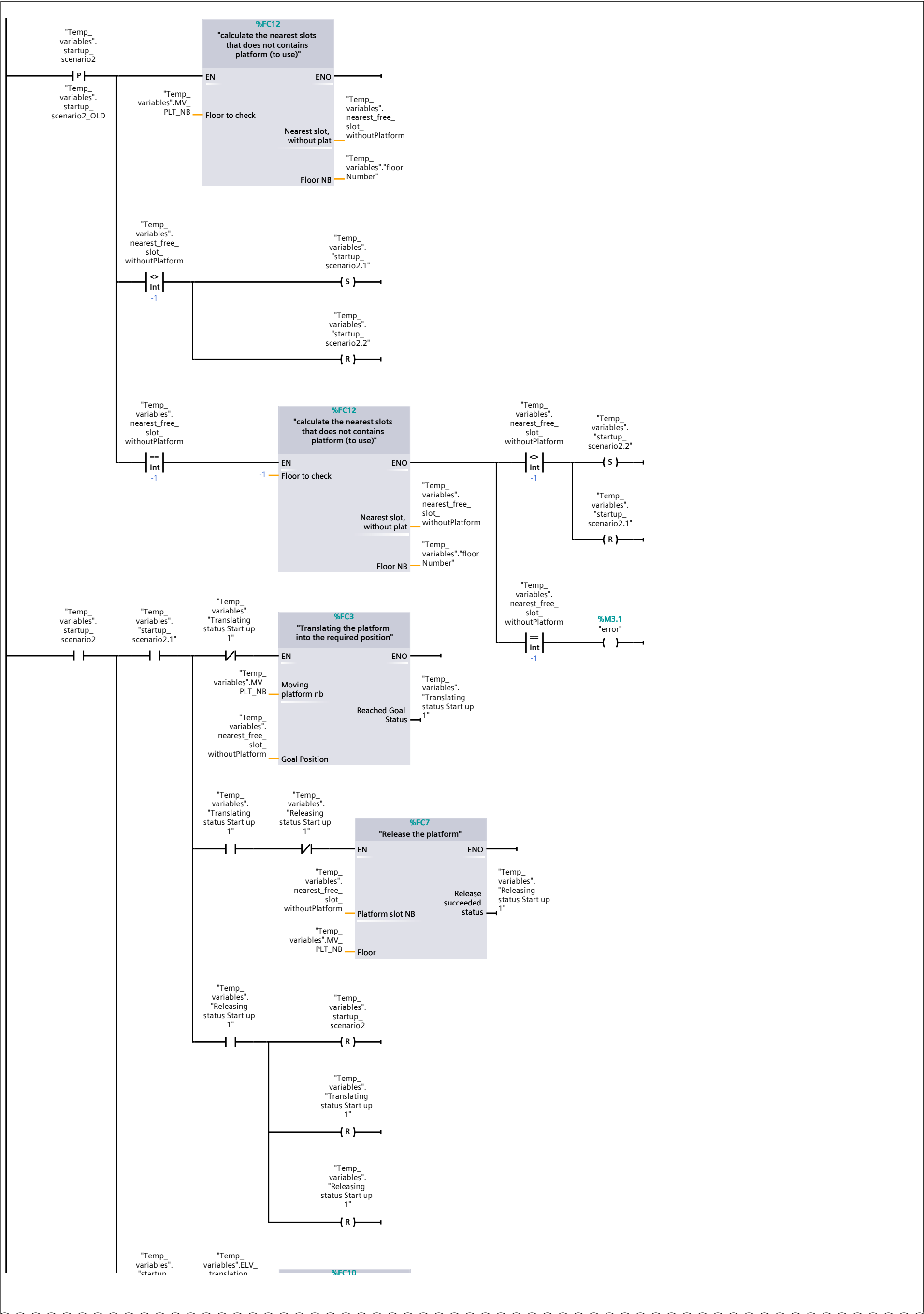


Network 6: Start up set up ( Exit gate) (2.1 / 2.1)



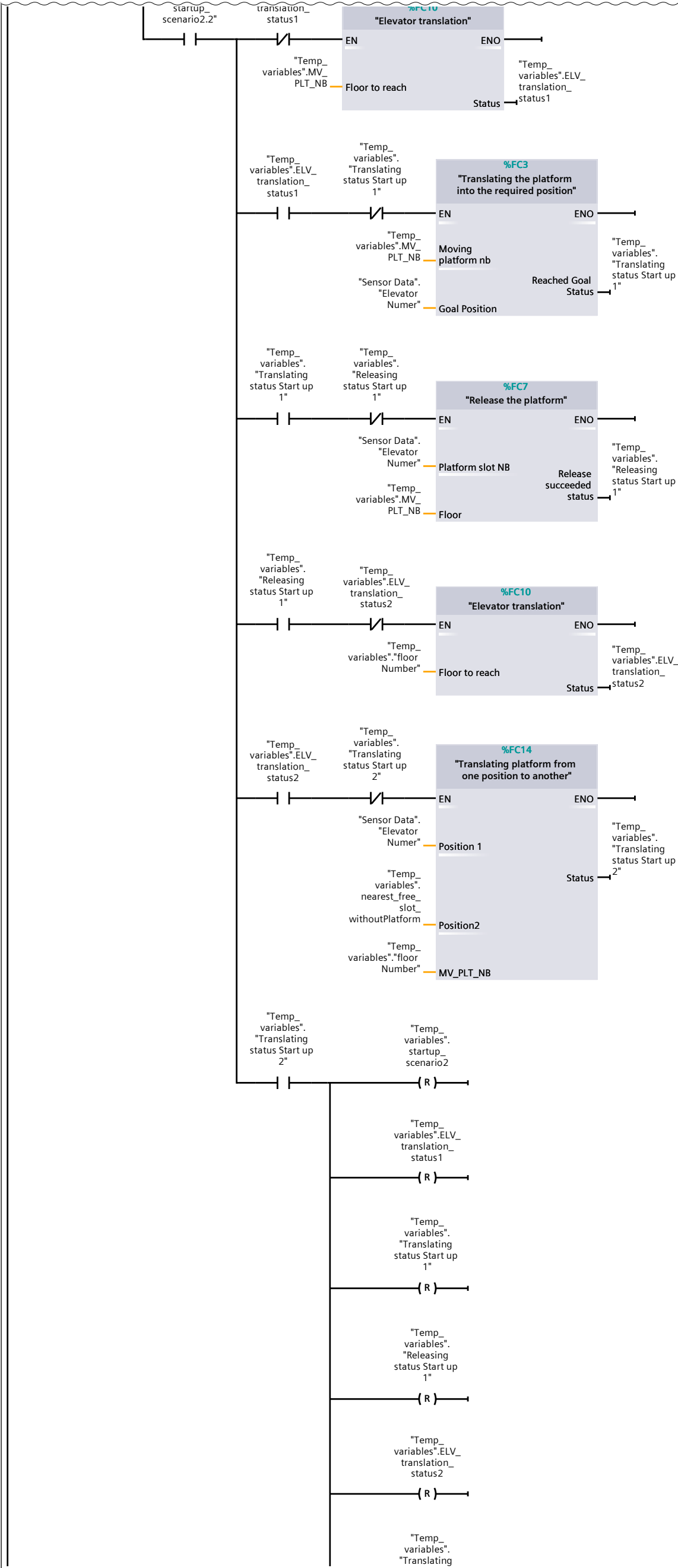
Totally Integrated Automation Portal		
<div>Network 7: Start up scenario 2</div> <div>the moving platform already have a car platform and it's fully attached , we might have three scenarios : 1-2- there is no car at the plataform 2- there is a car at this platform but the moving platform will take the same action in the two scenarios , it will search for the nearest (nearest to the entry gate) slot tht does not contain a platform and release the platform inside it. 3-if the nearest slot is in differnet floor (-1) the scenario 2 will be activated : the moving platform will put the platform in the elevator the elevator will go to the required floor the moving platform of this floor will take the platform and put it in the free spot</div>		

Network 7: Start up scenario 2 (1.1 / 3.1)



Network 7: Start up scenario 2 (2.1 / 3.1)

1.1 ( Page1 - 12)

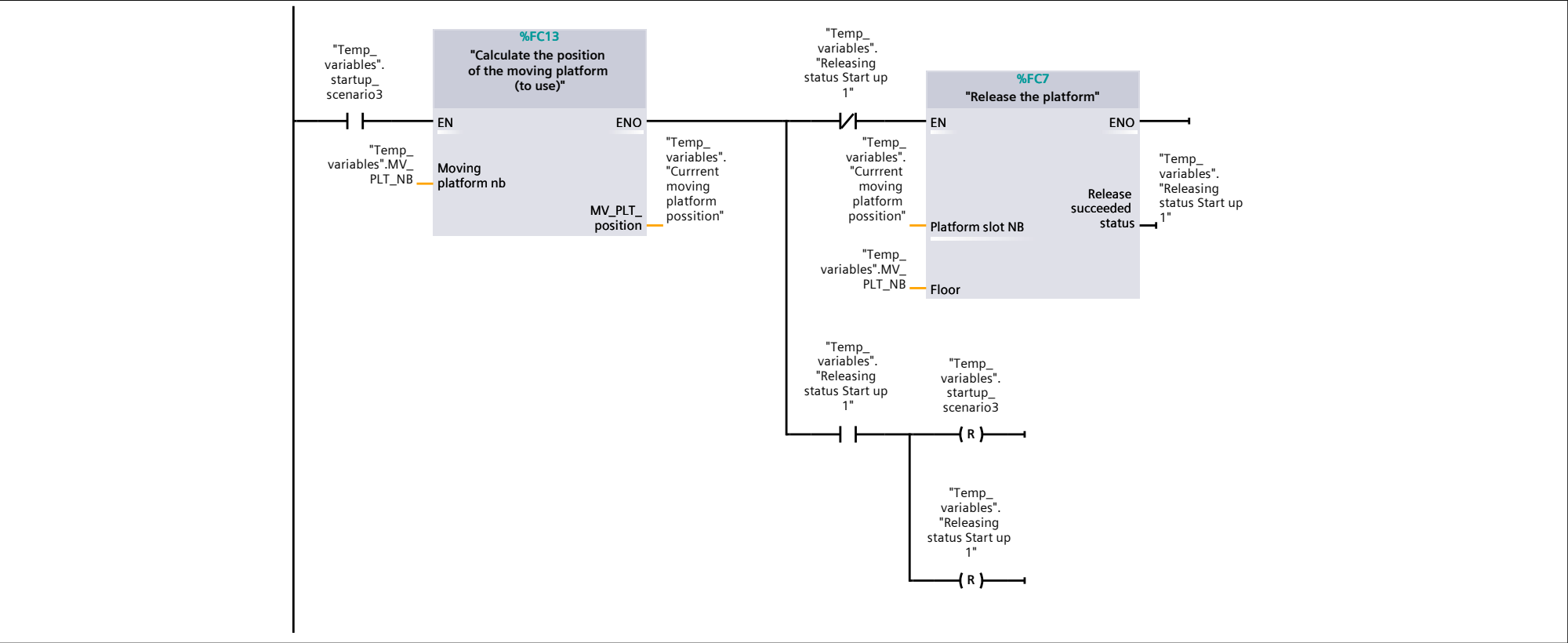


3.1 ( Page1 - 14)

Totally Integrated Automation Portal		
<div>Network 7: Start up scenario 2 (3.1 / 3.1)</div> <div>2.1 ( Page1 - 13)</div> <div><div><div></div><div>status Start up 2"</div><div>( R )</div></div></div>		

Network 8: Start up scenario 3

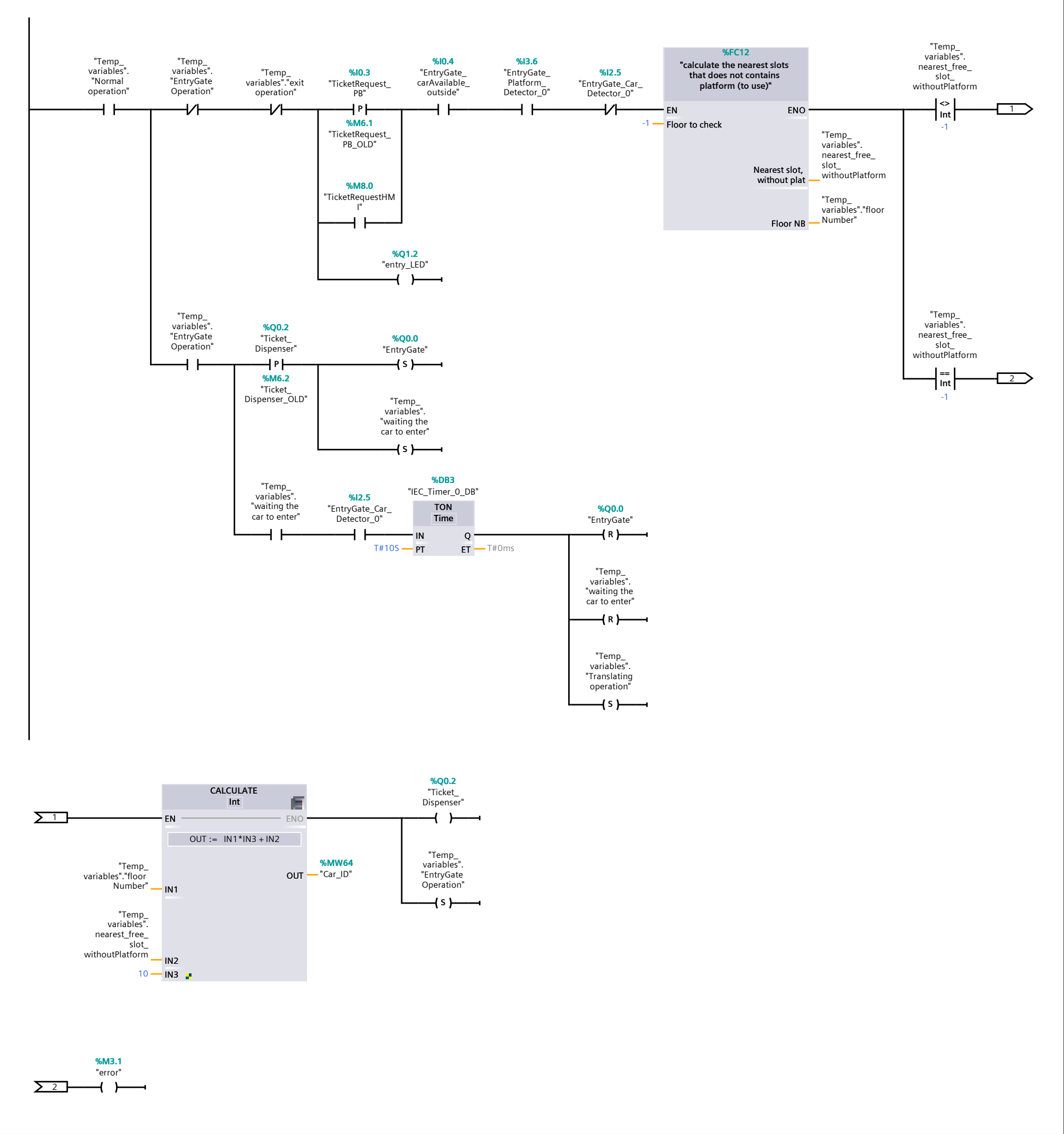
the moving platform already have a car platform and it's not fully attached :  
the moving platform will simply release this platform and reset the scenario 3 but not the start up.



Network 9: entering to the parking phase 1 : Request a Ticket and opening the gate

this phase represent when the user want to enter to the parking so  
first the system will check if the user pushed the requrest button and there is a car detected at the entry gate  
  
second it will search for the nearest free park that does not contains a platform and give a ID based on this spot.  
The ID will be formed of 2 digits the first represents the floor number and the secons represents the car slot (eg: 14 : floor 1 slot 4)  
  
third we will open the gate and the gate will close after 10 second after he put his car in the right position  
Note: when this operation start the next user shall wait until this process and the translating process finish .  
A LED will be turned on to tell the user when he can push the ticked dispenser button .

Network 9: entering to the parking phase 1 : Request a Ticket and opening the gate

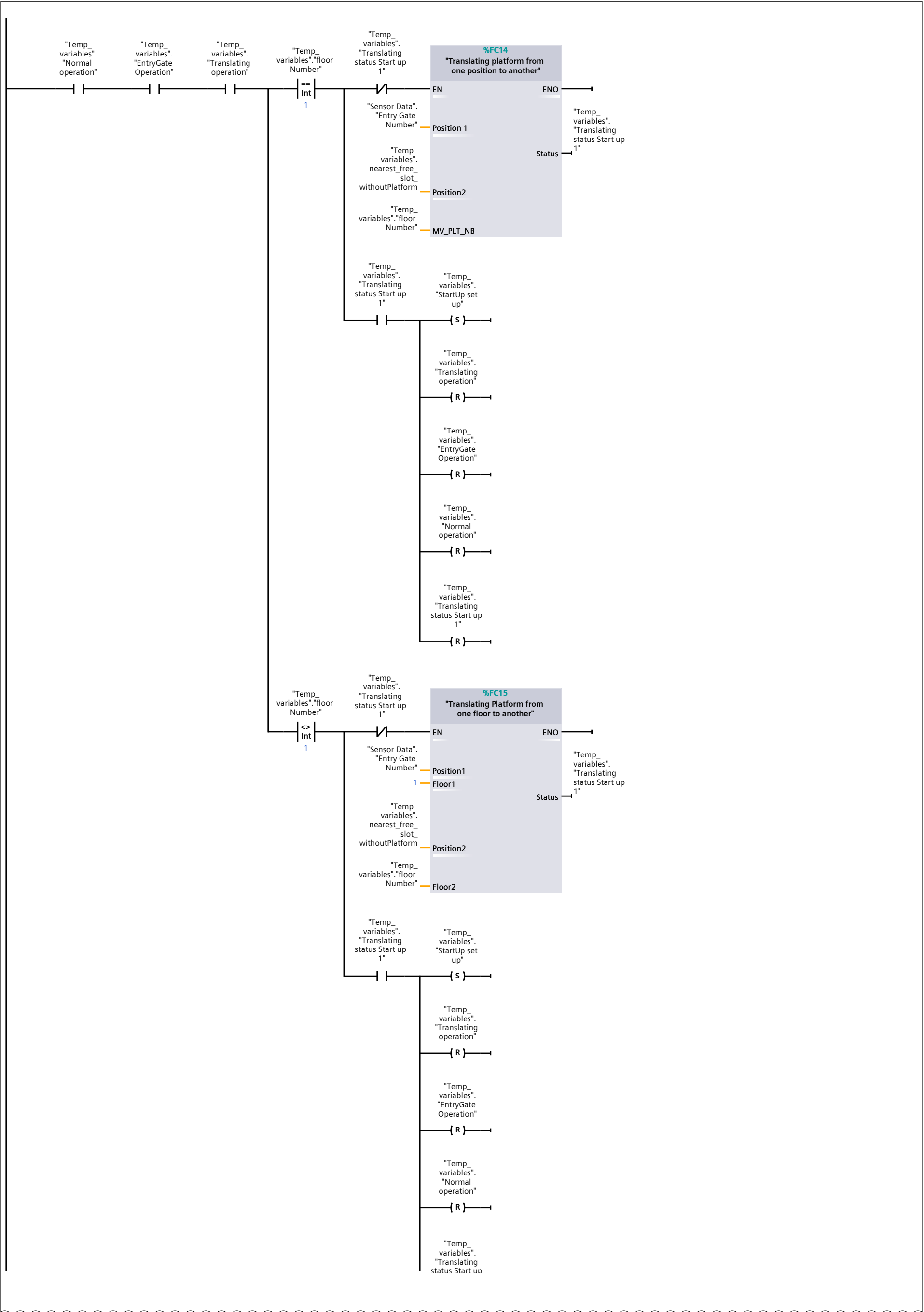


Network 10: entering to the parking phase 2 : translating operation plus bringing the platform at the entry gate

- 1- moving platform will go to the entry gate
- 2- attach the platform
- 3-translate it to the free slot that does not contains a platform (ID)
- 4- Release the platform
- 5- the start up set up will be set again to let the moving platform go and bring a platform and attach it to the entry gate
- 5.1- while doing this the normal operation will be stoped (even if another user is requesting to get a ID he won't get anything if this process is working)



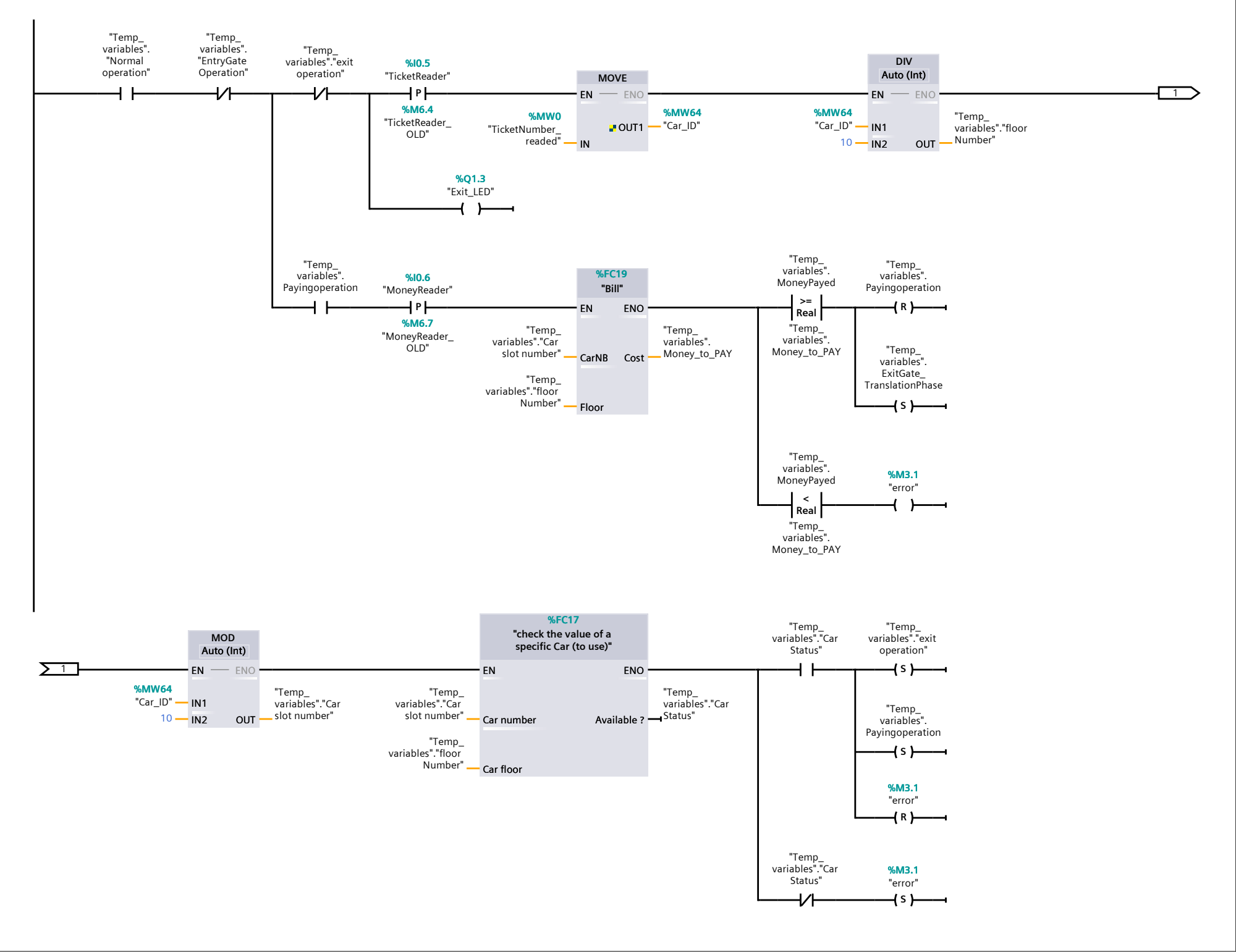
Network 10: entering to the parking phase 2 : translating operation plus bringing the platform at the entry gate (1.1 / 2.1)



Totally Integrated Automation Portal		
Network 10: entering to the parking phase 2 : translating operation plus bringing the platform at the entry gate (2.1 / 2.1)		
<div>1.1 ( Page1 - 17)</div> <div><div>status start up 1"</div><div>( R )</div></div>		

Network 11: exiting from the parking phase 1 : Car Request and paying

Network 11: exiting from the parking phase 1 : Car Request and paying

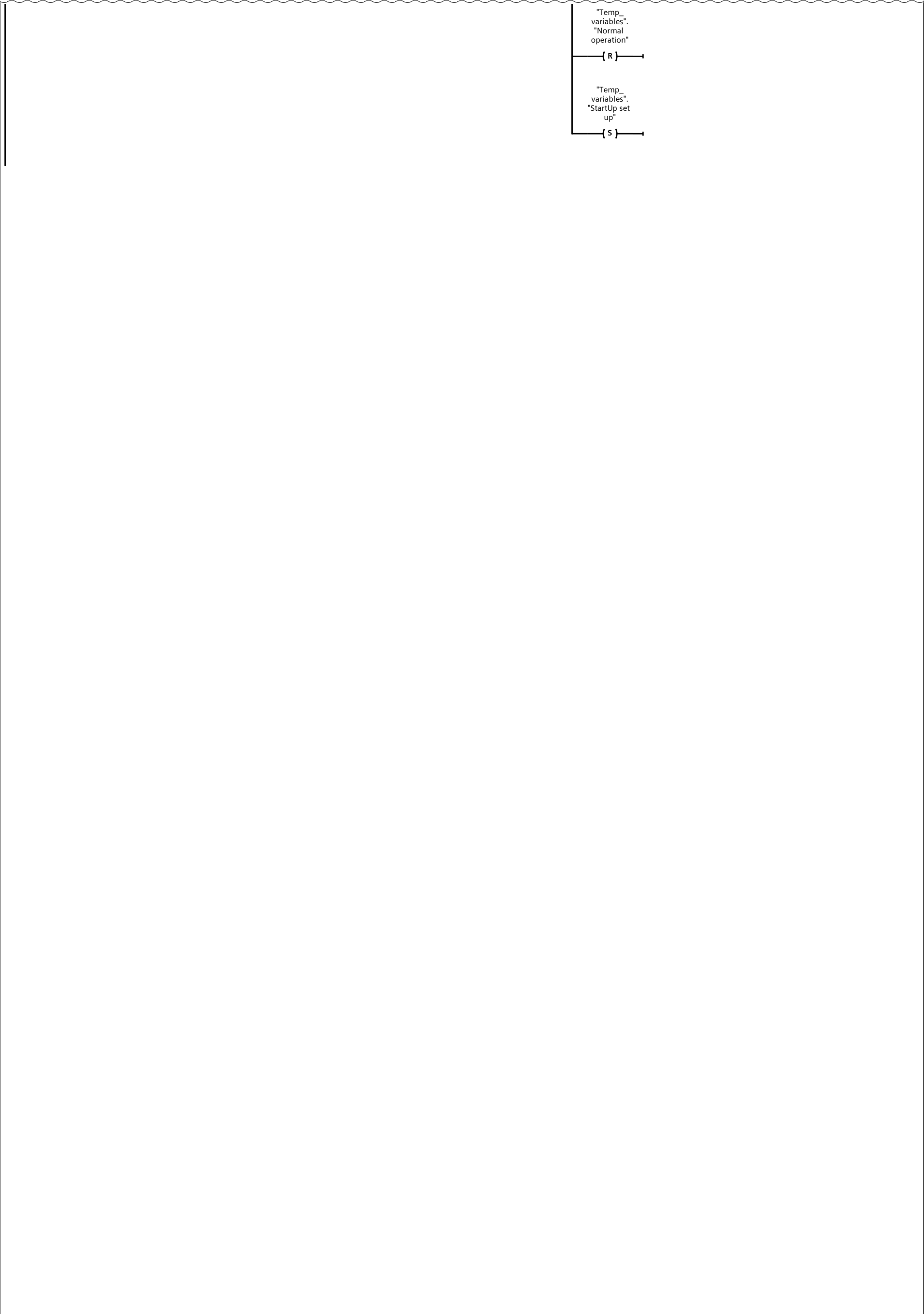


Network 12: exiting from the parking phase 2 : Car Translation and Opening the exit gate



Network 12: exiting from the parking phase 2 : Car Translation and Opening the exit gate (2.1 / 2.1)

1.1 ( Page1 - 20)



Totally Integrated Automation Portal

### Program blocks

#### Sensor Data [DB1]

Sensor Data Properties

General

Name	Sensor Data	Number	1	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Start value	Retain
▼ Static			
ParkingSlot_Car_Detector 1	Array[0..8] of Bool		False
ParkingSlot_Car_Detector 2	Array[0..8] of Bool		False
ParkingSlot_Car_Detector 3	Array[0..8] of Bool		False
ParkingSlot_Car_Detector_OLD1	Array[0..8] of Bool		False
ParkingSlot_Car_Detector_OLD2	Array[0..8] of Bool		False
ParkingSlot_Car_Detector_OLD3	Array[0..8] of Bool		False
ParkingSlop_Platform_Detector 1	Array[0..8] of Bool		False
ParkingSlop_Platform_Detector 2	Array[0..8] of Bool		False
ParkingSlop_Platform_Detector 3	Array[0..8] of Bool		False
Platform 1 LevelSensor	Array[0..8] of Bool		False
Platform 2 LevelSensor	Array[0..8] of Bool		False
Platform 3 LevelSensor	Array[0..8] of Bool		False
Entry Gate Number	Int	0	False
Elevator Numer	Int	7	False
Exit Gate Number	Int	8	False

## Program blocks

## Saving Sensing values to arrays [FC1]

Saving Sensing values to arrays Properties							
General							
Name	Saving Sensing values to arrays	Number	1	Type	FC	Language	LAD
Numbering Information	Automatic						
Title		Author		Comment		Family	
Version	0.1	User-defined ID					
Name			Data type		Default value		
Input							
Output							
InOut							
Temp							
Constant							
▼ Return							
Saving Sensing values to arrays			Void				

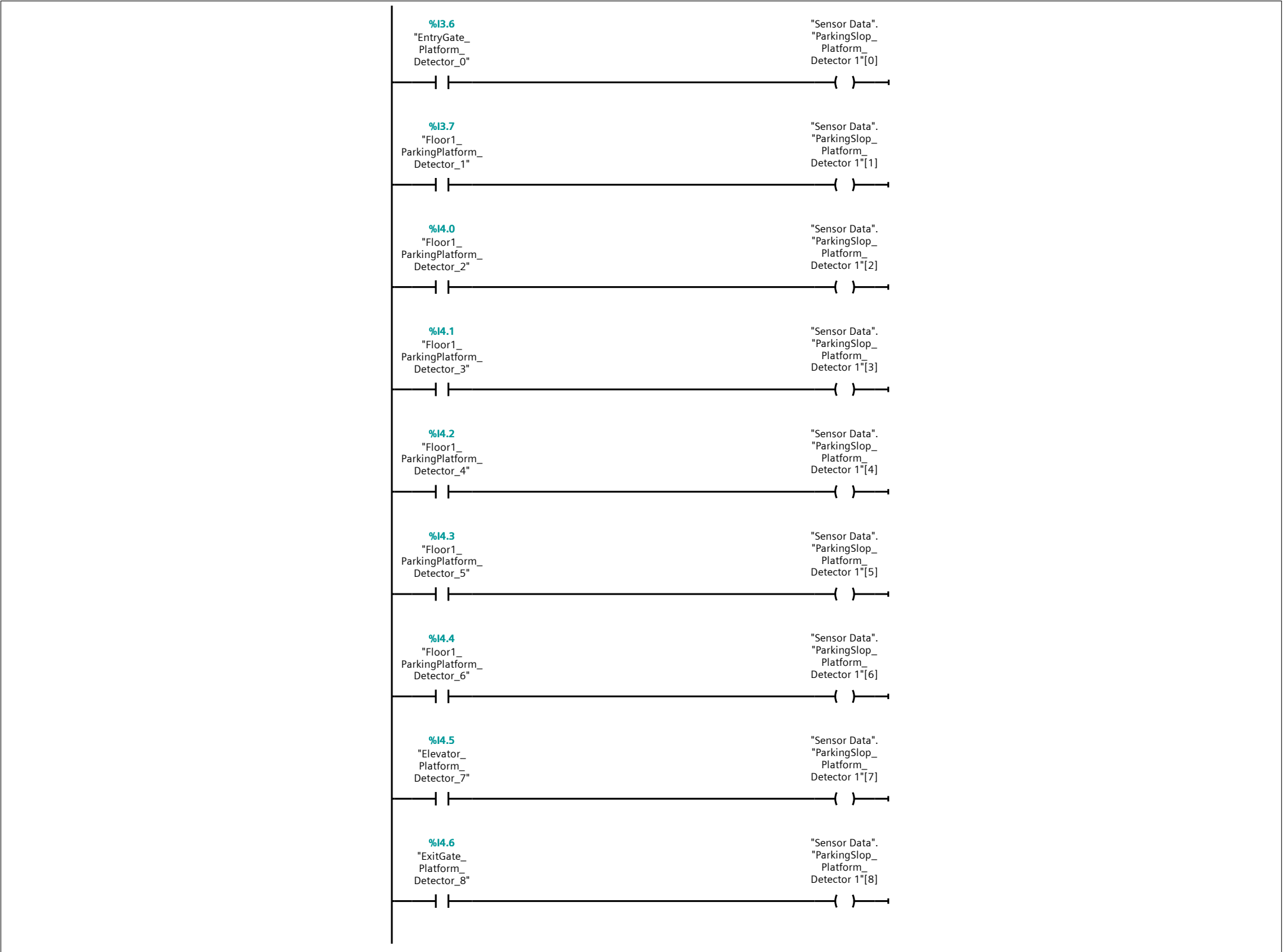
## Network 1: Saving values from the Car detection sensors floor1 into arrays



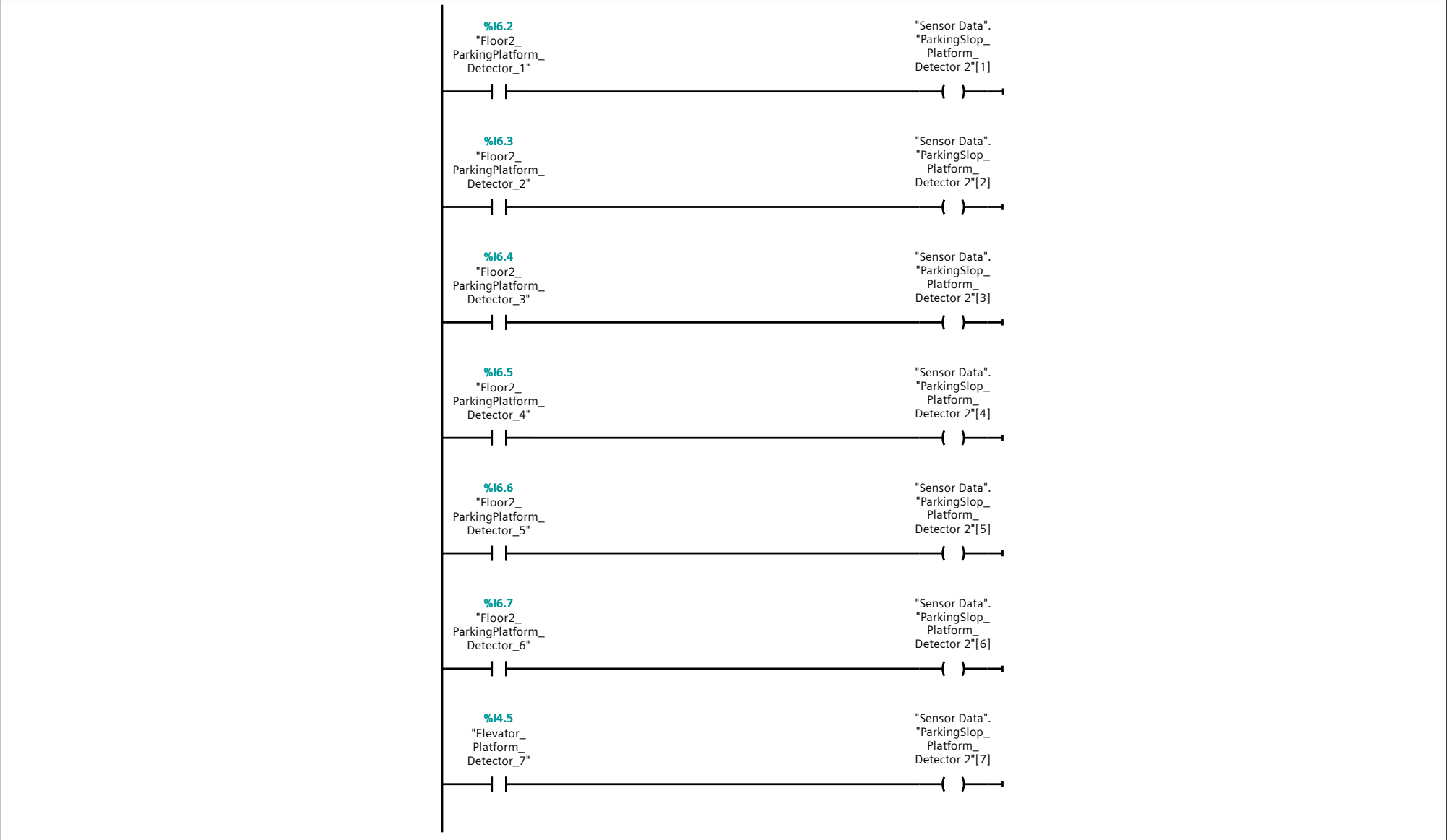
## Network 2: Saving values from the Car detection sensors floor1 into arrays







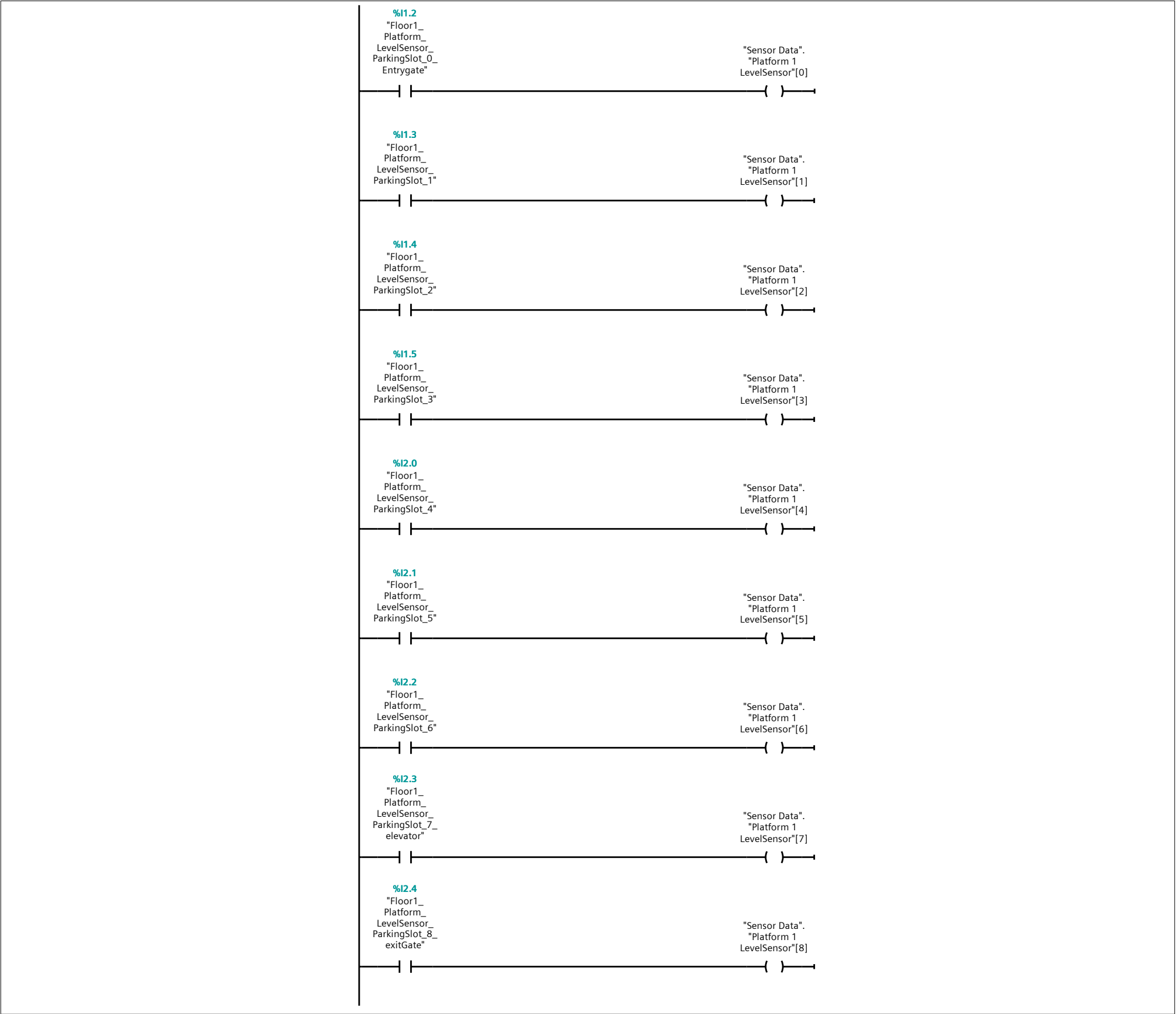
Network 5: Saving values from the Platform detection sensors floor 2 into arrays



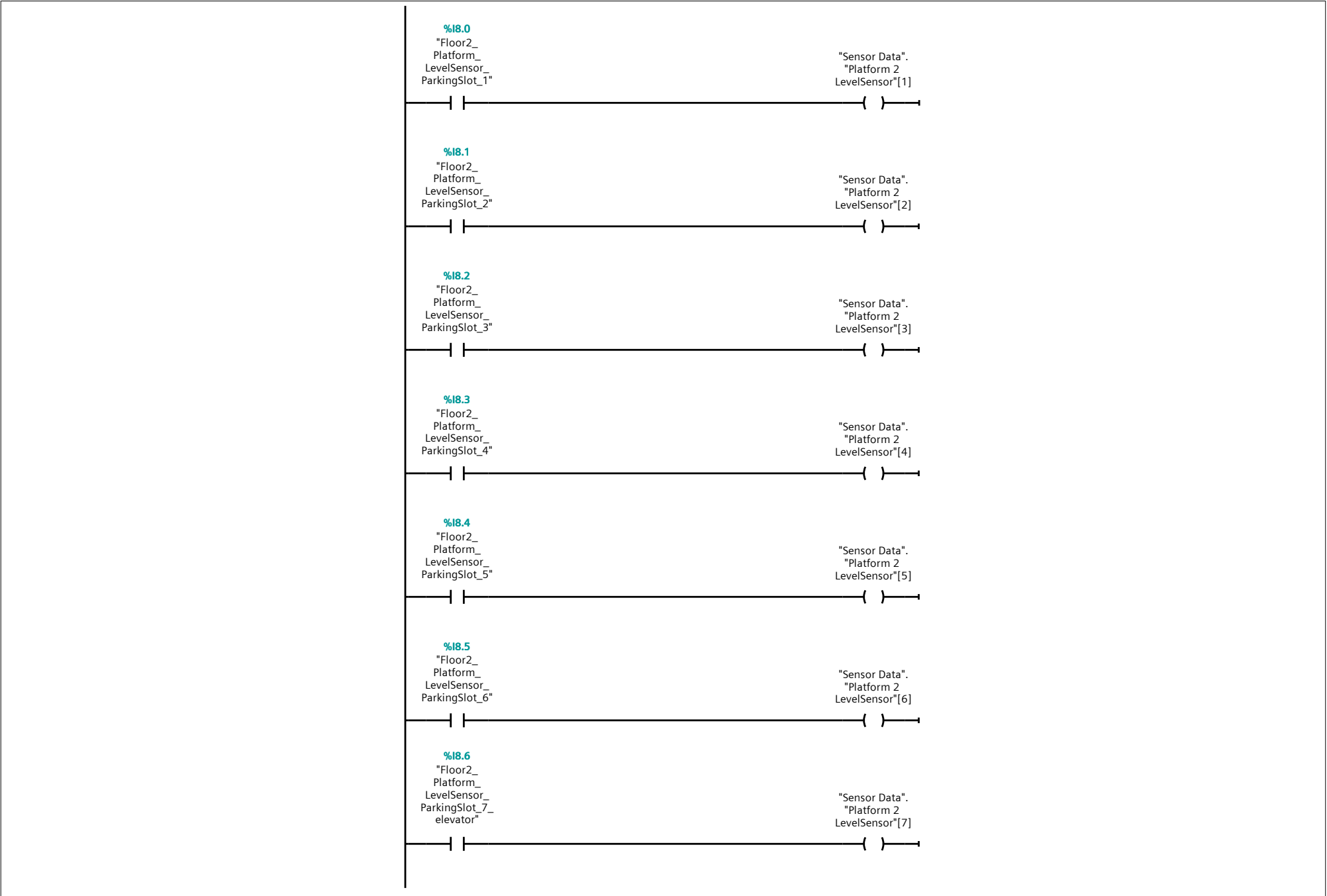
Network 6: Saving values from the Platform detection sensors floor 3 into arrays



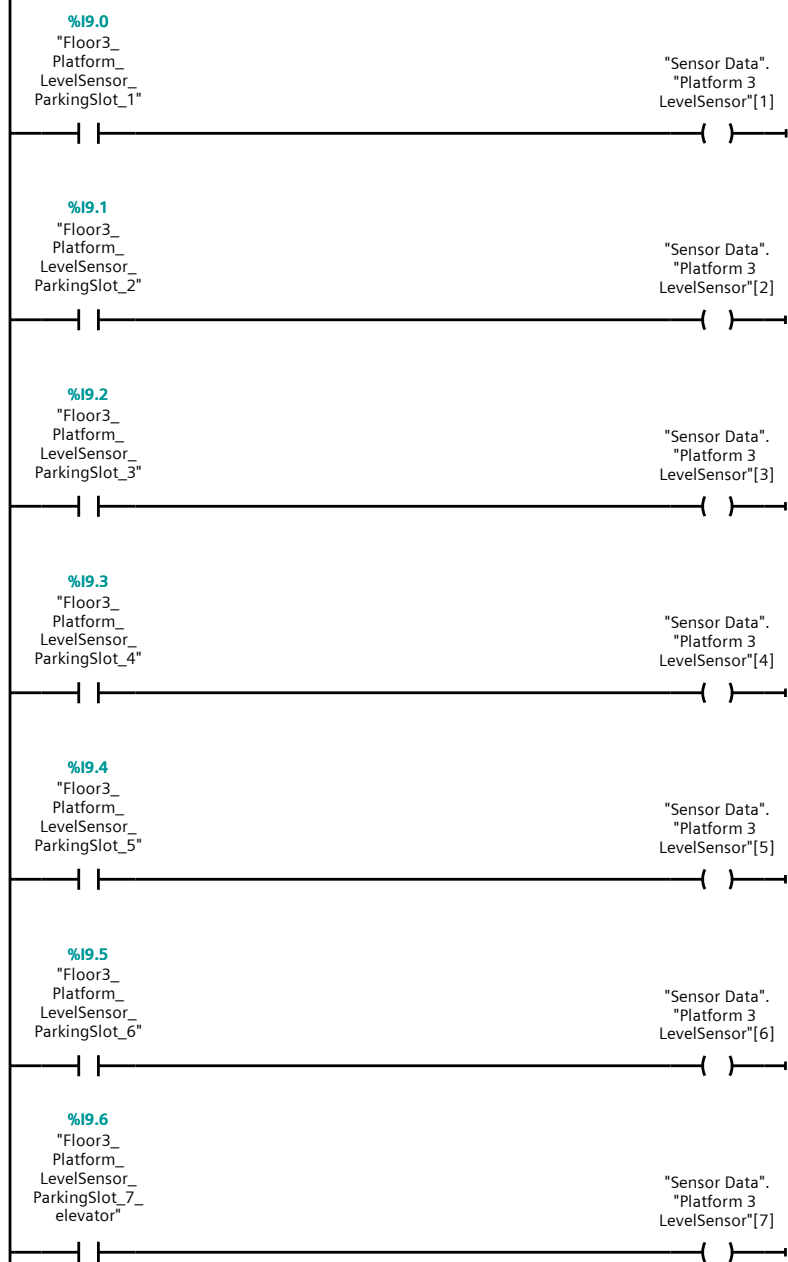
Network 7: Saving values from the Platform level sensors in floor 1 into arrays



## Network 8: Saving values from the Platform level sensors in floor 2 into arrays



## Network 9: Saving values from the Platform level sensors in floor 3 into arrays



Totally Integrated Automation Portal

### Program blocks

calculate the nearest slots that contains platform [FC2]

calculate the nearest slots that contains platform Properties							
General							
Name	calculate the nearest slots that contains platform	Number	2	Type	FC	Language	SCL
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Search_Specific_Floor	Int	
parking_car_detector floor1	Array[0..8] of Bool	
ParkingSlop_Platform_Detector floor1	Array[0..8] of Bool	
parking_car_detector floor2	Array[0..8] of Bool	
ParkingSlop_Platform_Detector floor2	Array[0..8] of Bool	
parking_car_detector floor3	Array[0..8] of Bool	
ParkingSlop_Platform_Detector floor3	Array[0..8] of Bool	
▼ Output		
nearest free slot with platform	Int	
Floor NB	Int	
InOut		
▼ Temp		
col	Int	
found	Bool	
Constant		
▼ Return		
calculate the nearest slots that contains platform	Void	

```
0001 // Initialize counters
0002 #nearest free slot with platform := -1;
0003 #Floor NB := -1;
0004
0005 // Flag to prevent overwriting
0006 #found := FALSE;
0007
0008 CASE #Search_Specific_Floor OF
0009   1:
0010     FOR #col := 1 TO 6 DO
0011       IF #parking_car_detector floor1[#col] = FALSE AND #ParkingSlop_Platform_Detector floor1[#col] = TRUE THEN
0012         #nearest free slot with platform := #col;
0013         #Floor NB := 1;
0014         #found := TRUE; // Mark that we found a slot
0015         EXIT;
0016       END_IF;
0017     END_FOR;
0018   ;
0019   2:
0020     FOR #col := 1 TO 6 DO
0021       IF #parking_car_detector floor2[#col] = FALSE AND #ParkingSlop_Platform_Detector floor2[#col] = TRUE THEN
0022         #nearest free slot with platform := #col;
0023         #Floor NB := 2;
0024         #found := TRUE;
0025         EXIT;
0026       END_IF;
0027     END_FOR;
0028   ;
0029   3:
0030     FOR #col := 1 TO 6 DO
0031       IF #parking_car_detector floor3[#col] = FALSE AND #ParkingSlop_Platform_Detector floor3[#col] = TRUE THEN
0032         #nearest free slot with platform := #col;
0033         #Floor NB := 3;
0034         #found := TRUE;
0035         EXIT;
0036       END_IF;
0037     END_FOR;
0038   ;
0039   ELSE
0040     FOR #col := 1 TO 6 DO
0041       IF #parking_car_detector floor1[#col] = FALSE AND #ParkingSlop_Platform_Detector floor1[#col] = TRUE THEN
0042         #nearest free slot with platform := #col;
0043         #Floor NB := 1;
0044         #found := TRUE; // Mark that we found a slot
0045         EXIT;
0046       END_IF;
0047     END_FOR;
0048
0049     // Loop through Floor 2 (Only if no slot was found before)
0050     IF NOT #found THEN
0051       FOR #col := 1 TO 6 DO
```



Program blocks

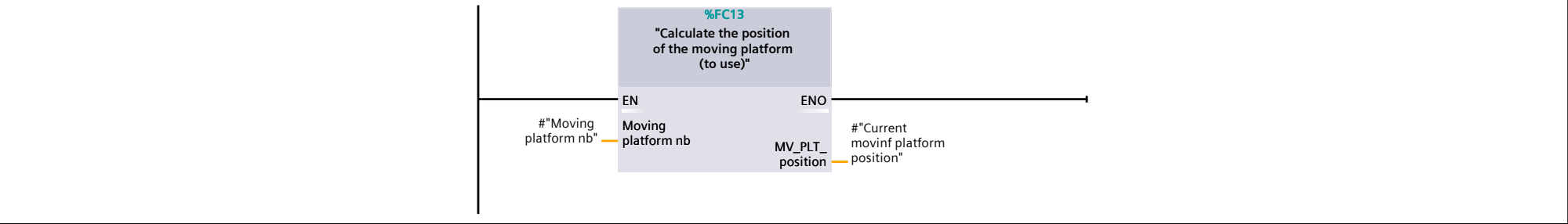
Translating the platform into the required position [FC3]

Translating the platform into the required position Properties							
General							
Name	Translating the platform into the required position	Number	3	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Moving platform nb	Int	
Goal Position	Int	
▼ Output		
Reached Goal Status	Bool	
InOut		
▼ Temp		
Current movinf platform position	Int	
Up or Down Direction	Int	
Constant		
▼ Return		
Translating the platform into the required position	Void	

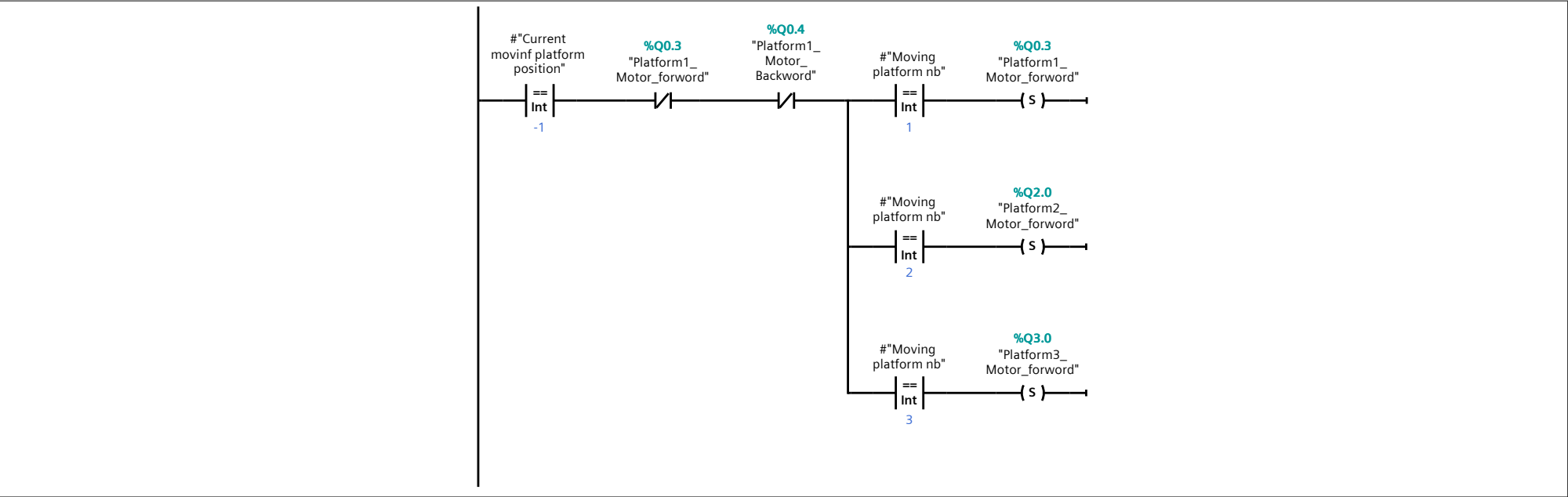
Network 1: Set which moving platform we want to use

Note: the floor 2 and 3 contain only 7 (1-7) position sensors  
but the floor 1 contains 9 (0-8)  
But the same function will be used for both. the level sensor nb 0-8 (entery exit gate)in the floor 1-2 will always be 0



Network 2: Checking the moving platform position

if we can't read the position of the moving platform and the platform is not moving we are going simply to make the moving platform go forward until one of the Level sensors detect the platform.

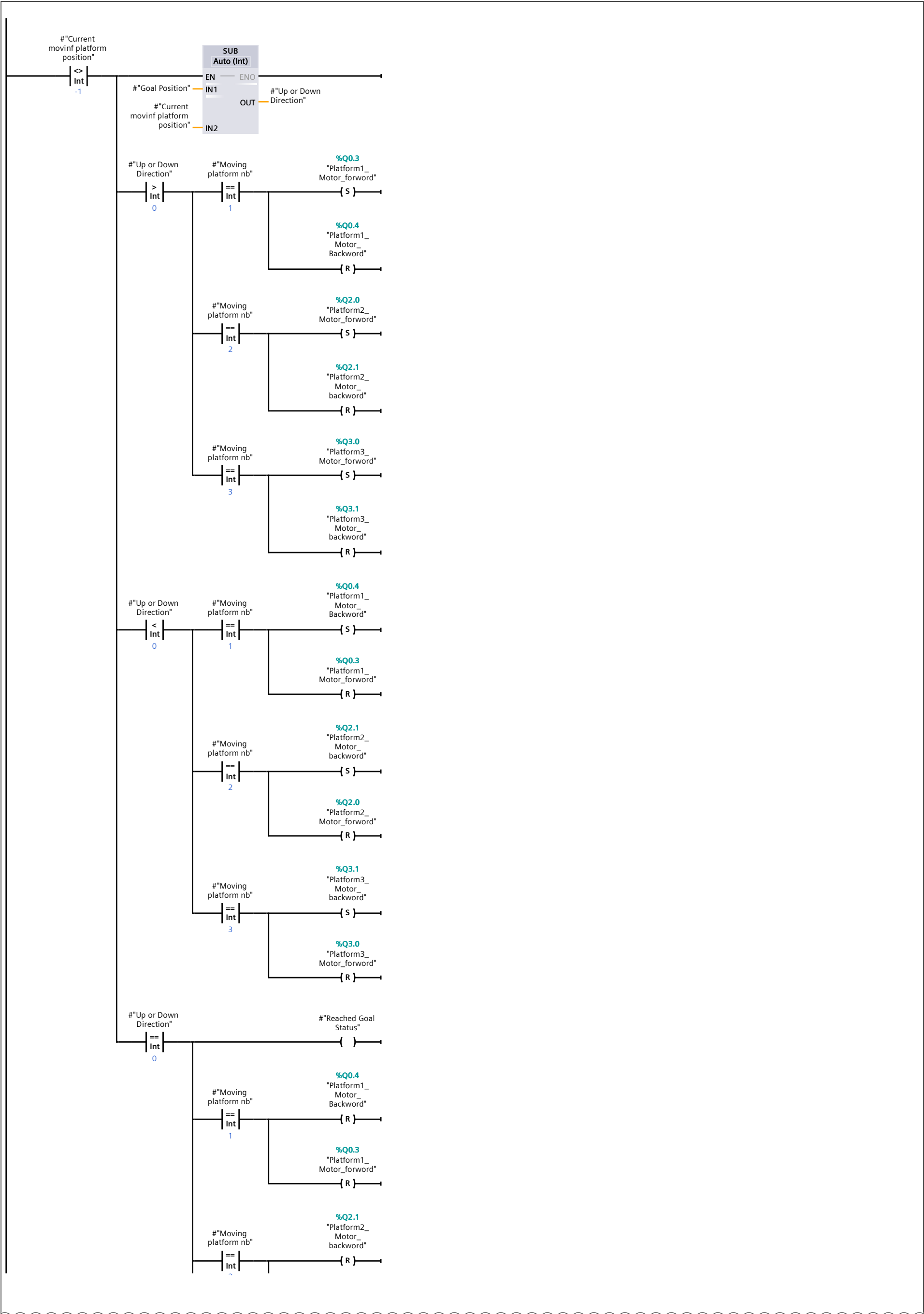


Network 3: Moving Platform direction detection

we will substract the goal position from the current position if the value is + that's mean i shall move upword and if - downword.  
if the "#Current movinf platform position" equal to -1 this netwoek won't run to avoid erros.

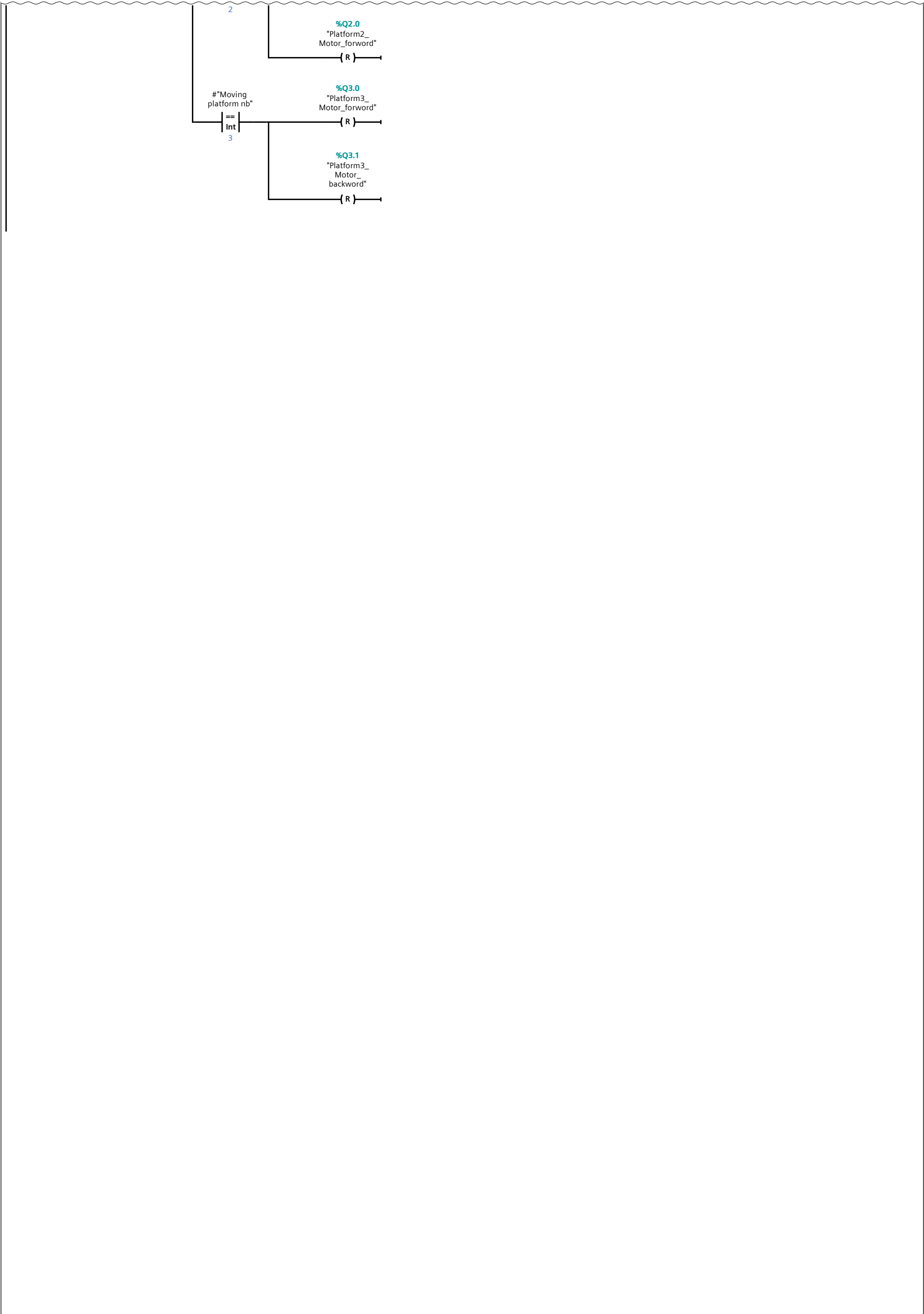


Network 3: Moving Platform direction detection (1.1 / 2.1)



Network 3: Moving Platform direction detection (2.1 / 2.1)

1.1 ( Page5 - 2)



Program blocks

Calculate the position of the Moving platform [FC5]

Calculate the position of the Moving platform Properties							
General							
Name	Calculate the position of the Moving platform	Number	5	Type	FC	Language	SCL
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Platform level	Array[0..8] of Bool	
▼ Output		
Current_Position	Int	
InOut		
▼ Temp		
col	Int	
Constant		
▼ Return		
Calculate the position of the Moving platform	Void	

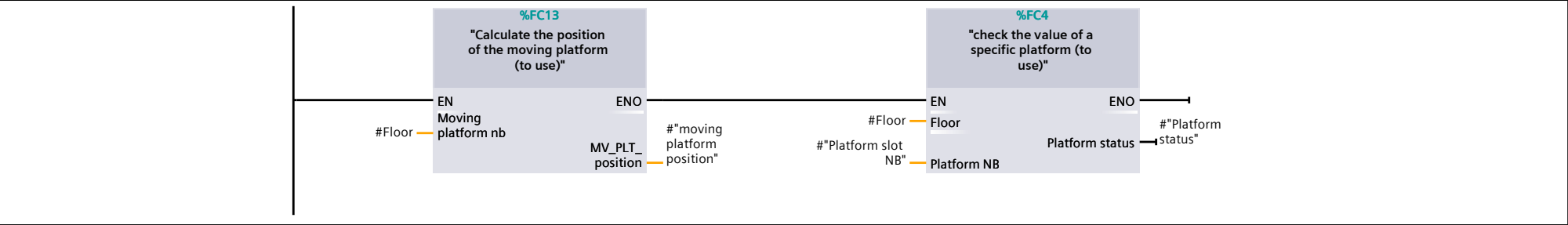
```
0001
0002 # "Current_Position" := -1;
0003
0004 // Loop through the 1D array
0005 FOR #col := 0 TO 8 DO
0006     IF # "Platform level" [#col] = TRUE THEN
0007         # "Current_Position" := #col; // Store the first free slot (column index)
0008         EXIT;
0009     END_IF;
0010 END_FOR;
0011
0012
```

Program blocks

Attach the platform [FC6]

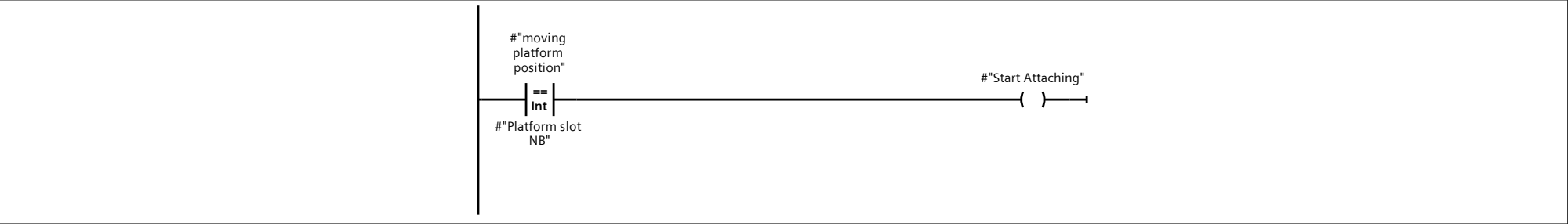
Attach the platform Properties							
General							
Name	Attach the platform	Number	6	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					
Name				Data type		Default value	
▼ Input							
Platform slot NB				Int			
Floor				Int			
▼ Output							
Attach succeeded status				Bool			
InOut							
▼ Temp							
moving platform position				Int			
Start Attaching				Bool			
Attach succeeded				Bool			
Platform status				Bool			
Constant							
▼ Return							
Attach the platform				Void			

Network 1: check the position of the moving platform and the status of the platform



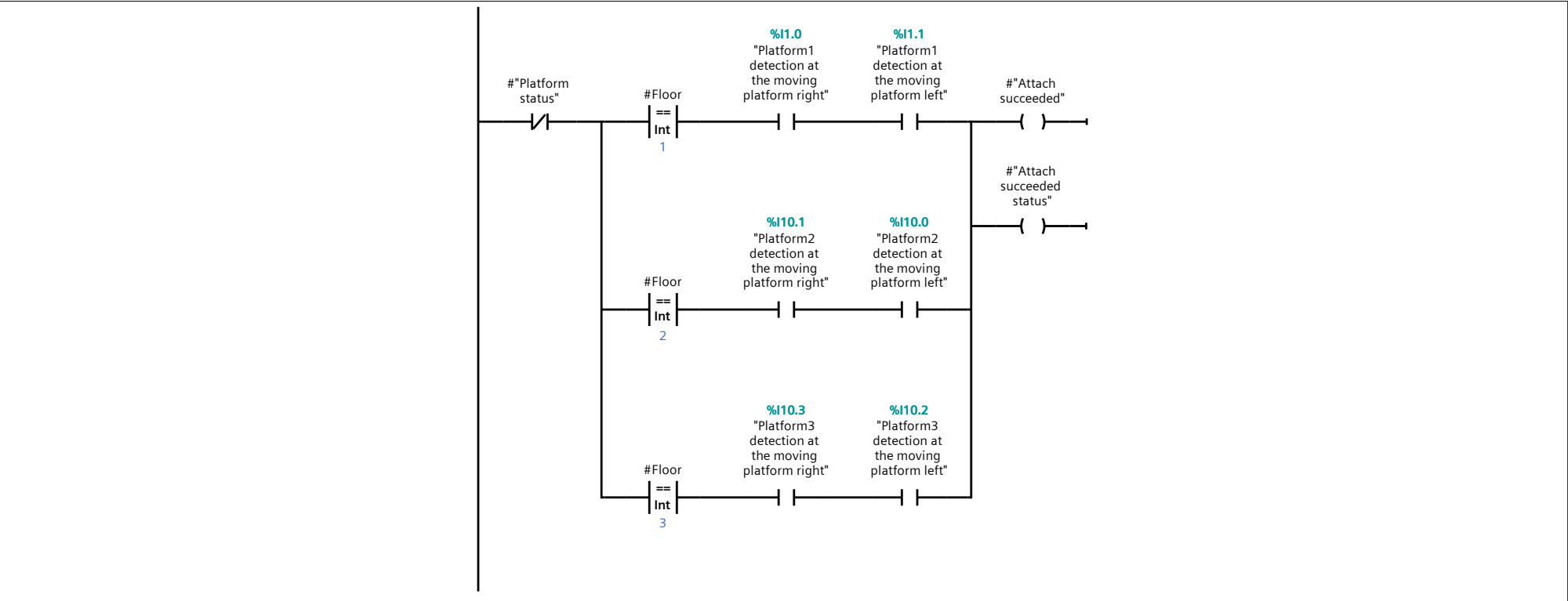
Network 2: Moving platform is in the right position

Making sure thatthe moving platform is in the right position



Network 3: Stop Attach proces

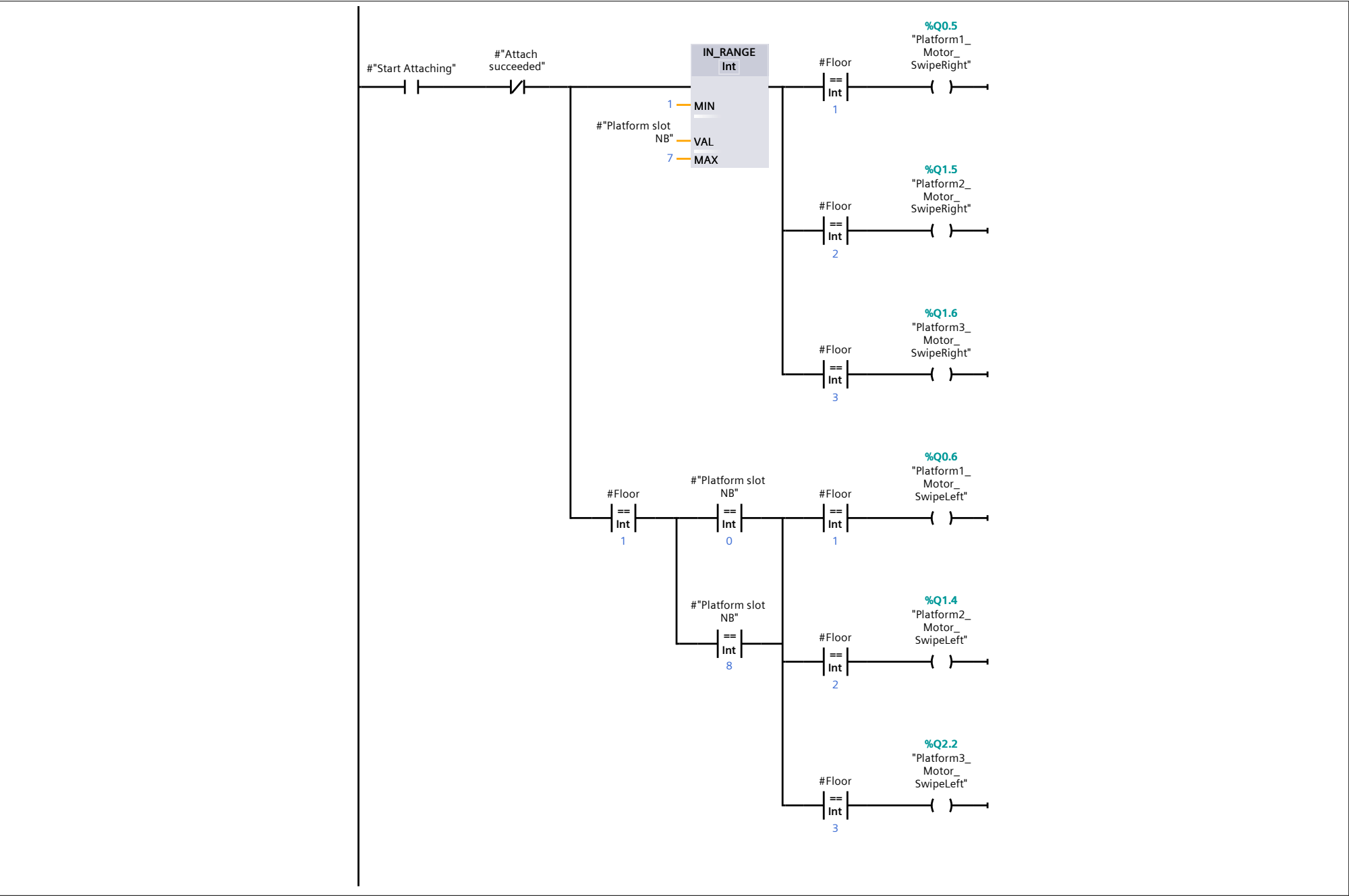
if the two sensors at the moving platform are high (the fix platform is available at the movable platform) and the Platform status is low we will stop the attaching process and give a result the the process has finished



Network 4: choosing the direction of swiping

The sweeping direction is based on the geometry of the parking area and changes when moving from the right side to the left side. On the right side, sweeping is done using the right motor (slots 1–3). On the left side, the same motor sweeps in the opposite direction (slots 4–7).

Slot 0 represents the entry gate, and slot 8 is the exit gate. These two slots operate in the opposite direction compared to slots 1–7, so we use the other motor, called "sweep left," for them.



Program blocks

Release the platform [FC7]

Release the platform Properties

General

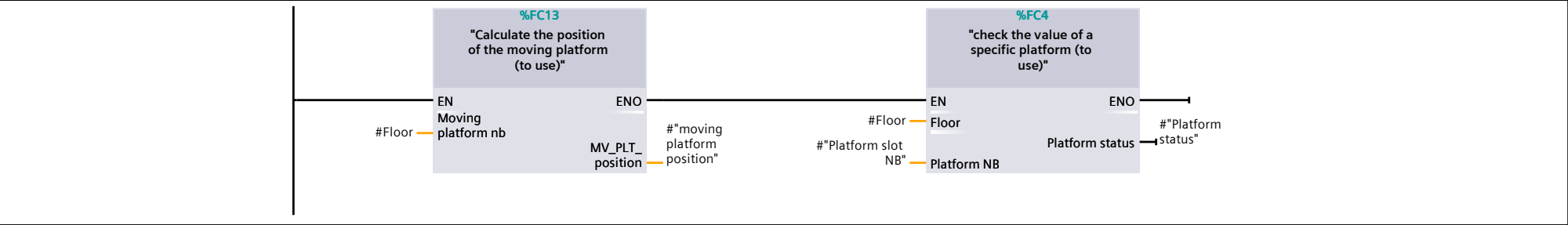
Name	Release the platform	Number	7	Type	FC	Language	LAD
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

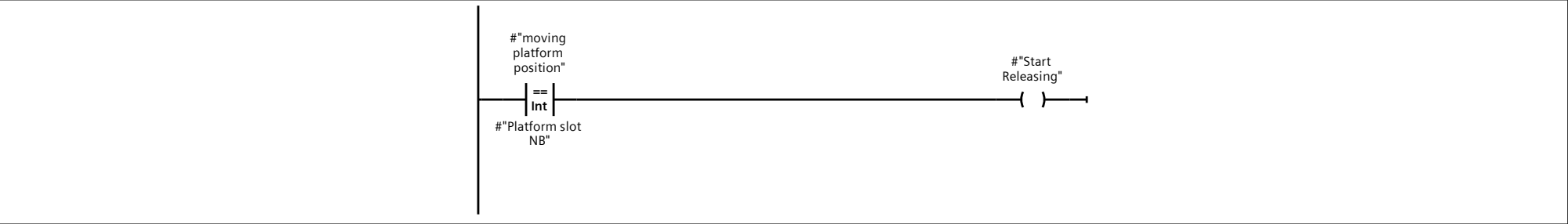
Name	Data type	Default value
▼ Input		
Platform slot NB	Int	
Floor	Int	
▼ Output		
Release succeeded status	Bool	
InOut		
▼ Temp		
moving platform position	Int	
Start Releasing	Bool	
Releasing succeeded	Bool	
Platform status	Bool	
Constant		
▼ Return		
Release the platform	Void	

Network 1: check the position of the moving platform and the status of the platform



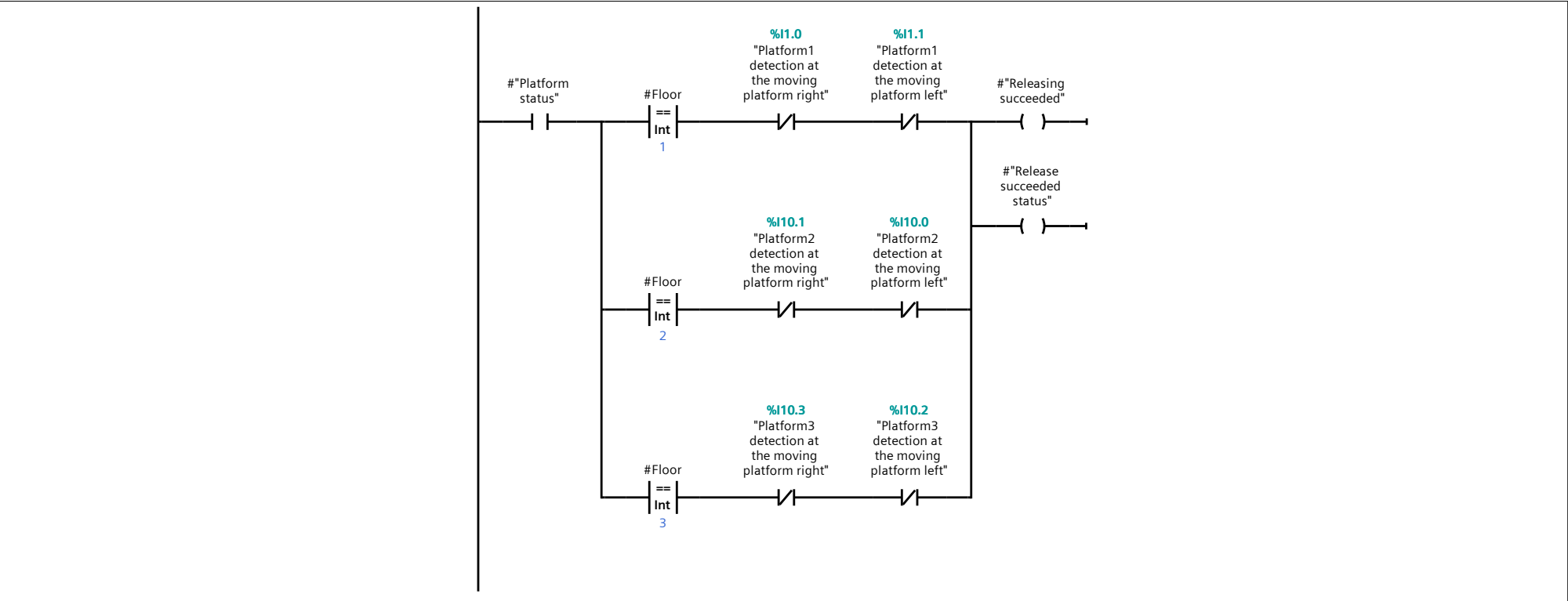
Network 2: Moving platform is in the right position

Making sure thatthe moving platform is in the right position



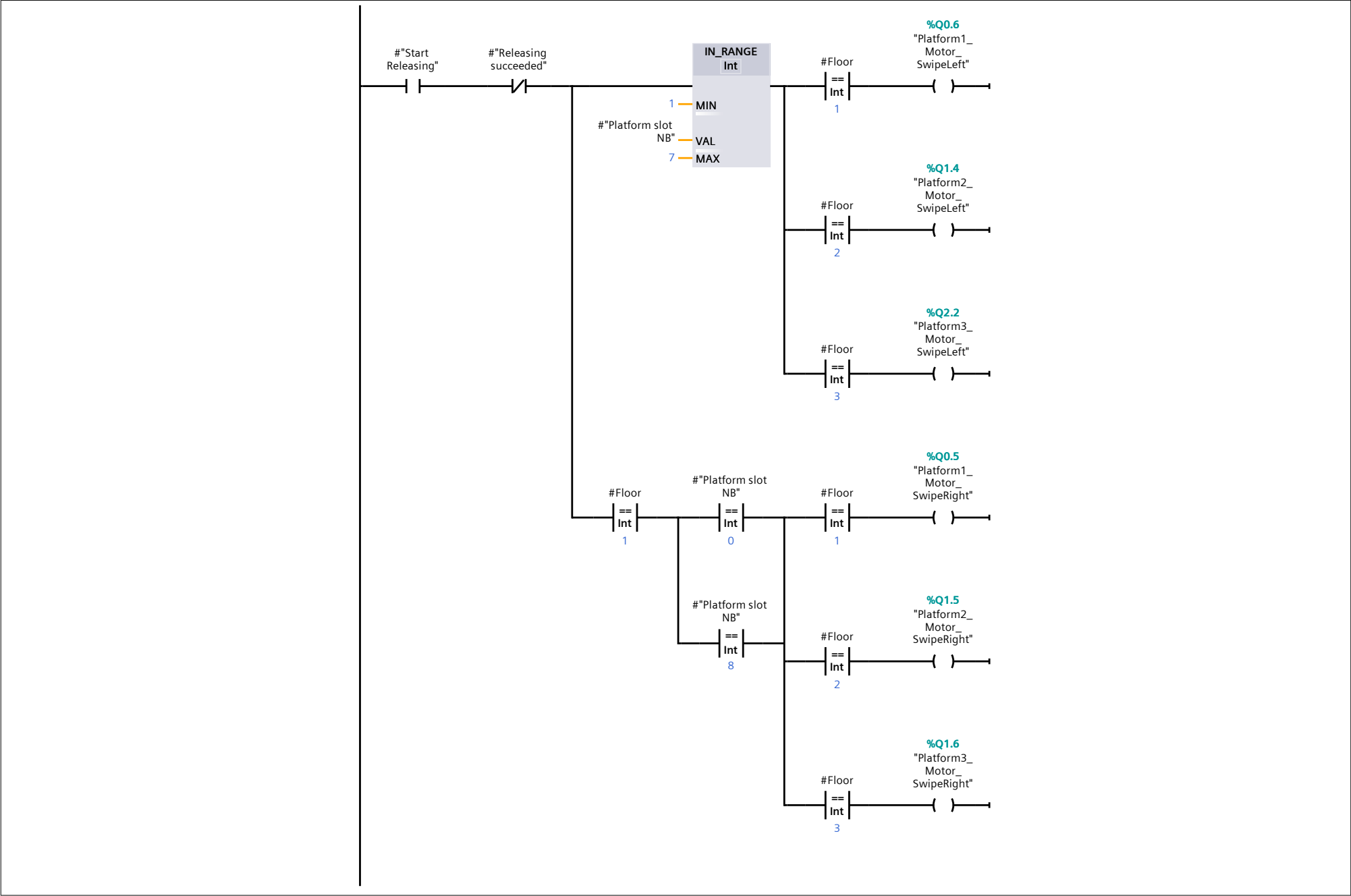
Network 3: Stop releasing process

if the two sensors at the moving platform are Low (the fix platform is not available at the movable platform) and the Platform status is higj we will stop the releasing process and give a result the the process has finished



Network 4: choosing the direction of swiping

The releasing algorithm is the same as the attach but the only differnt that we have cahnged the motor swipe right ---> motor swipe left and vise versa



Totally Integrated Automation Portal

Program blocks

Temp\_variables [DB2]

Temp\_variables Properties

General

Name	Temp_variables	Number	2	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Start value	Retain
▼ Static			
nearest_freeSlot_with_platfomr	Int	-1	False
nearest_free_slot_withoutPlatform	Int	-1	False
Currrent moving platform possition	Int	-1	False
floor Number	Int	0	False
MV_PLT_NB	Int	0	False
Car slot number	Int	0	False
Money_to_PAY	Real	0.0	False
MoneyPaid	Real	0.0	False
Car Status	Bool	false	False
StartUp set up	Bool	false	False
StartUp set up_OLD	Bool	false	False
Normal operation	Bool	false	False
exit operation	Bool	false	False
startup_EntryGate_scenario1	Bool	false	False
startup_scenario2	Bool	false	False
startup_scenario2_OLD	Bool	false	False
startup_scenario3	Bool	false	False
waiting the car to enter	Bool	false	False
waiting the car to enter OD	Bool	false	False
Translating operation	Bool	false	False
Opening Exit gate	Bool	false	False
waiting the car to exit	Bool	false	False
Opening Exit gate OLD	Bool	false	False
EntryGate Operation	Bool	false	False
ExitGate SetUP	Bool	false	False
startup_ExitGate_scenario1	Bool	false	False
checkScenarios	Int	0	False
countUP	Bool	false	False
RESETcount	Bool	false	False
MV_PLT_setup	Bool	false	False
Elevator_Setup	Bool	false	False
EntGATE_setup	Bool	false	False
startup_scenario2.1	Bool	false	False
startup_scenario2.2	Bool	false	False
StartUpSetValues	Bool	false	False
CalculationStatus	Bool	false	False
startup_EntryGate_scenario2	Bool	false	False
startup_ExitGate_scenario2	Bool	false	False
countUPOLD	Bool	false	False
StartupValues2	Bool	false	False
ElevatorSetUp_Scenario2	Bool	false	False
Payingoperation	Bool	false	False
ExitGate_TranslationPhase	Bool	false	False
Translating status Start up 1	Bool	false	False
Attaching status Start up 1	Bool	false	False
Translating status Start up 2	Bool	false	False
Releasing status Start up 1	Bool	false	False
ELV_translation_status1	Bool	false	False
ELV_translation_status2	Bool	false	False
TR_1	Bool	false	False
TR_2	Bool	false	False
TR_3	Bool	false	False
RL_1	Bool	false	False
RL_2	Bool	false	False
AT_1	Bool	false	False
AT_2	Bool	false	False
ELVTR_1	Bool	false	False
ELVTR_2	Bool	false	False
TRR1	Bool	false	False
TRR2	Bool	false	False
ELVTRR1	Bool	false	False
ELVTRR2	Bool	false	False



Totally Integrated Automation Portal

### Program blocks

calculate the nearest slots that does not contains platform [FC8]

calculate the nearest slots that does not contains platform Properties							
General							
Name	calculate the nearest slots that does not contains plat-form	Number	8	Type	FC	Language	SCL
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Search_Specific_Floor	Int	
parking_car_detector floor1	Array[0..8] of Bool	
ParkingSlop_Platform_Detector floor1	Array[0..8] of Bool	
parking_car_detector floor2	Array[0..8] of Bool	
ParkingSlop_Platform_Detector floor2	Array[0..8] of Bool	
parking_car_detector floor3	Array[0..8] of Bool	
ParkingSlop_Platform_Detector floor3	Array[0..8] of Bool	
▼ Output		
nearest free slot with platform	Int	
Floor NB	Int	
InOut		
▼ Temp		
col	Int	
found	Bool	
Constant		
▼ Return		
calculate the nearest slots that does not contains platform	Void	

```
0001 // Initialize counters
0002 #nearest free slot with platform := -1;
0003 #Floor NB := -1;
0004
0005 // Flag to prevent overwriting
0006 #found := FALSE;
0007
0008 CASE #Search_Specific_Floor OF
0009   1:
0010     FOR #col := 1 TO 6 DO
0011       IF #parking_car_detector floor1[#col] = FALSE AND #ParkingSlop_Platform_Detector floor1[#col] = FALSE THEN
0012         #nearest free slot with platform := #col;
0013         #Floor NB := 1;
0014         #found := TRUE; // Mark that we found a slot
0015         EXIT;
0016       END_IF;
0017     END_FOR;
0018   ;
0019   2:
0020     FOR #col := 1 TO 6 DO
0021       IF #parking_car_detector floor2[#col] = FALSE AND #ParkingSlop_Platform_Detector floor2[#col] = FALSE THEN
0022         #nearest free slot with platform := #col;
0023         #Floor NB := 2;
0024         #found := TRUE;
0025         EXIT;
0026       END_IF;
0027     END_FOR;
0028   ;
0029   3:
0030     FOR #col := 1 TO 6 DO
0031       IF #parking_car_detector floor3[#col] = FALSE AND #ParkingSlop_Platform_Detector floor3[#col] = FALSE THEN
0032         #nearest free slot with platform := #col;
0033         #Floor NB := 3;
0034         #found := TRUE;
0035         EXIT;
0036       END_IF;
0037     END_FOR;
0038   ;
0039 ELSE
0040     FOR #col := 1 TO 6 DO
0041       IF #parking_car_detector floor1[#col] = FALSE AND #ParkingSlop_Platform_Detector floor1[#col] = FALSE THEN
0042         #nearest free slot with platform := #col;
0043         #Floor NB := 1;
0044         #found := TRUE; // Mark that we found a slot
0045         EXIT;
0046       END_IF;
0047     END_FOR;
0048
0049 // Loop through Floor 2 (Only if no slot was found before)
0050 IF NOT #found THEN
```

Totally Integrated Automation Portal		
<div>0051       FOR #col := 1 TO 6 DO</div> <div>0052       IF #"parking_car_detector floor2"[#col] = FALSE AND #"ParkingSlop_Platform_Detector floor2"[#col] = FALSE THEN</div> <div>0053        #"nearest free slot with platform" := #col;</div> <div>0054        #"Floor NB" := 2;</div> <div>0055        #found := TRUE;</div> <div>0056        EXIT;</div> <div>0057        END_IF;</div> <div>0058       END_FOR;</div> <div>0059   END_IF;</div> <div>0060</div> <div>0061   // Loop through Floor 3 (Only if no slot was found before)</div> <div>0062   IF NOT #found THEN</div> <div>0063     FOR #col := 1 TO 6 DO</div> <div>0064       IF #"parking_car_detector floor3"[#col] = FALSE AND #"ParkingSlop_Platform_Detector floor3"[#col] = FALSE THEN</div> <div>0065        #"nearest free slot with platform" := #col;</div> <div>0066        #"Floor NB" := 3;</div> <div>0067        #found := TRUE;</div> <div>0068        EXIT;</div> <div>0069        END_IF;</div> <div>0070     END_FOR;</div> <div>0071   END_IF;</div> <div>0072   ;</div> <div>0073 END_CASE;</div> <div>0074</div> <div>0075</div>		

Totally Integrated Automation Portal

### Program blocks

check the value of a specific platform [FC9]

check the value of a specific platform Properties

General

Name	check the value of a specific platform	Number	9	Type	FC	Language	SCL
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
ParkingSlop_Platform_Detector	Array[0..8] of Bool	
Platform_number_to_check	Int	
▼ Output		
Platform status	Bool	
InOut		
Temp		
Constant		
▼ Return		
check the value of a specific platform	Void	

```
0001 // Initialize counters
0002 #"Platform status" := FALSE;
0003
0004 IF #ParkingSlop_Platform_Detector[#Platform_number_to_check] = TRUE THEN
0005     #"Platform status" := TRUE; // Store the first free slot (column index)
0006 END_IF;
0007
0008
0009
```

Program blocks

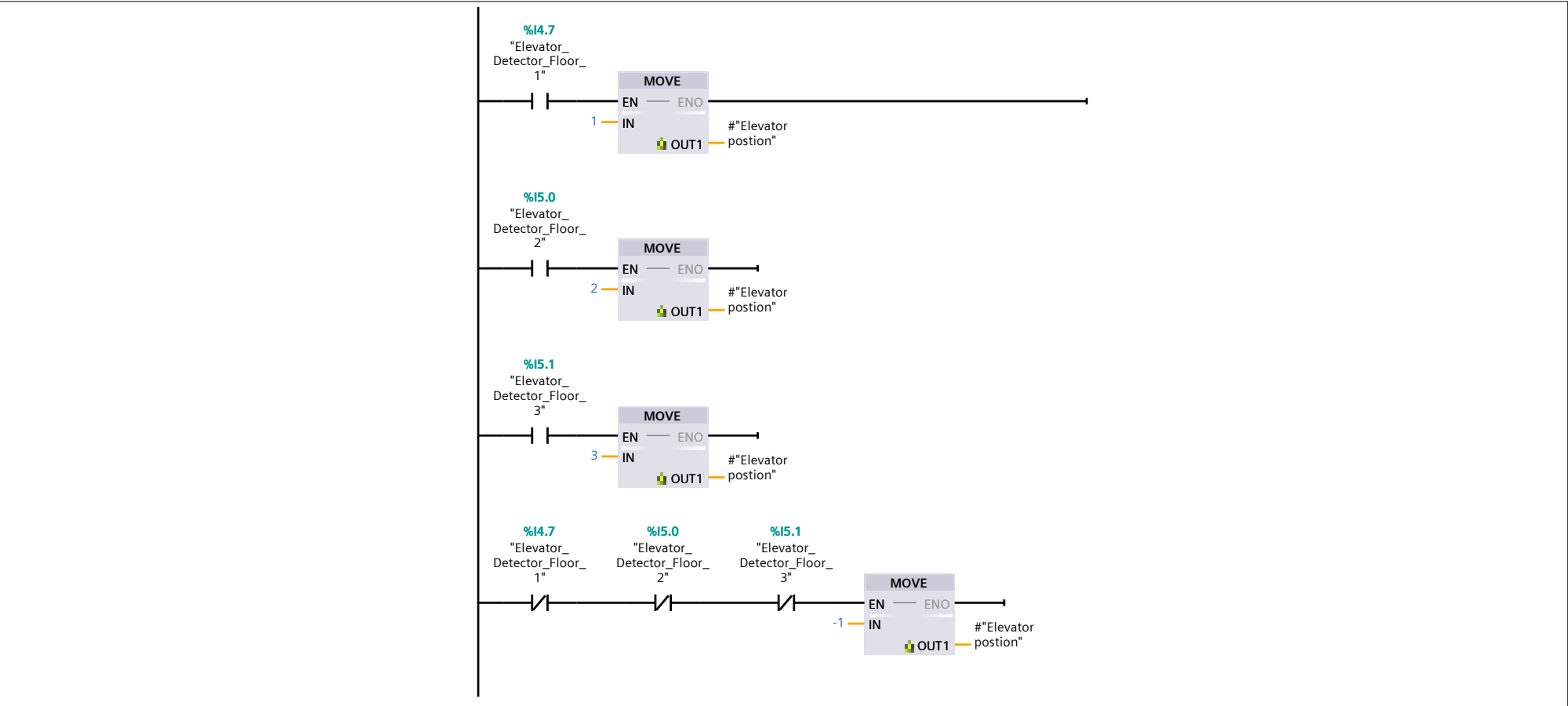
Elevator translation [FC10]

Elevator translation Properties							
General							
Name	Elevator translation	Number	10	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Floor to reach	Int	
▼ Output		
Status	Bool	
InOut		
▼ Temp		
Elevator postion	Int	
Up/Down direction	Int	
Constant		
▼ Return		
Elevator translation	Void	

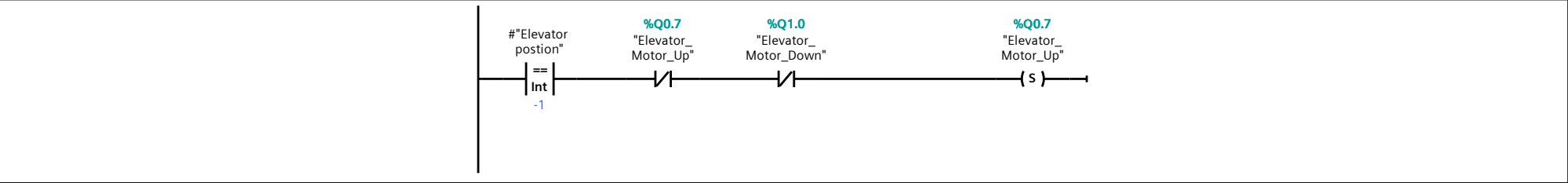
Network 1: elevator position

Set up the position of the elevator



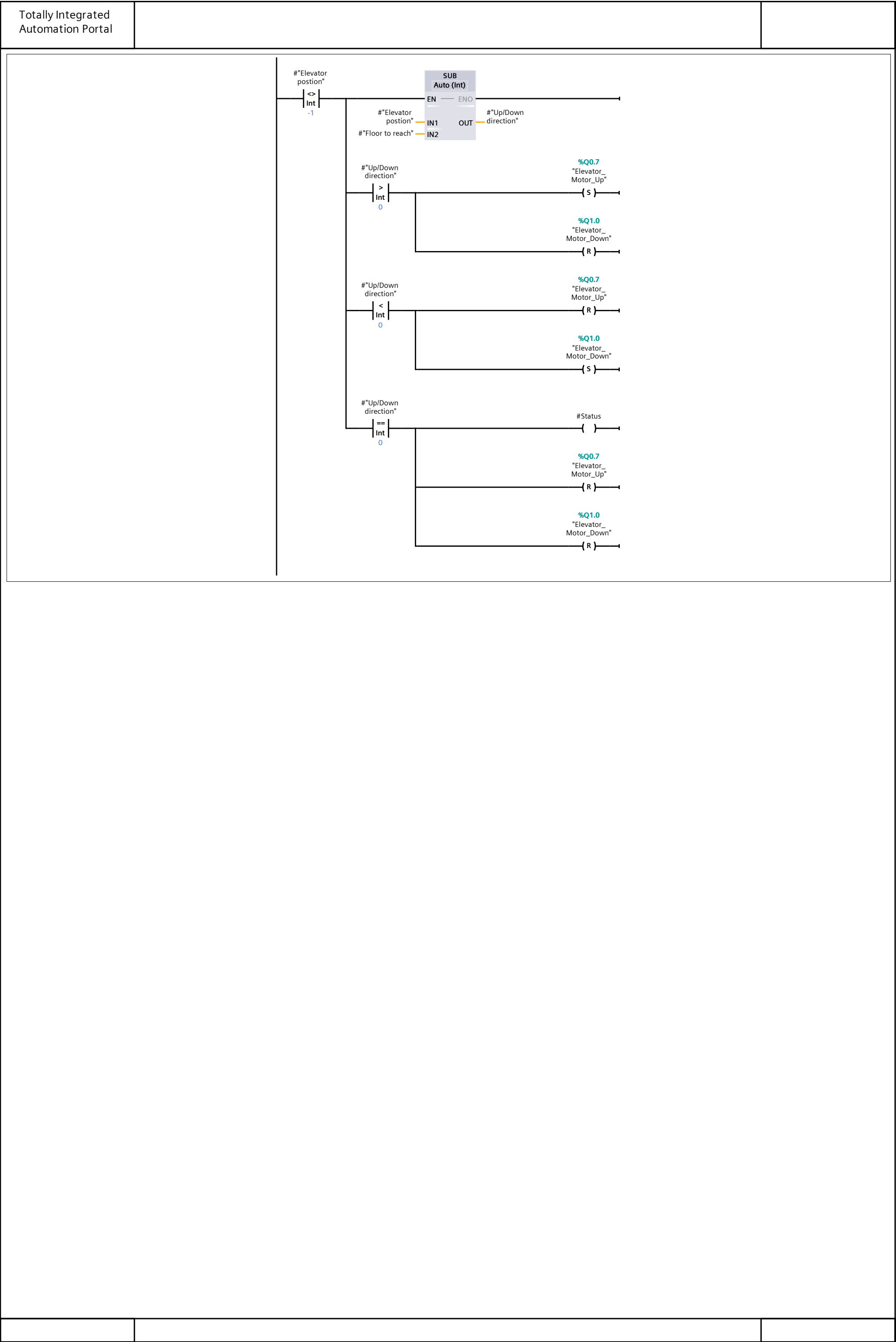
Network 2: Elevator set up

we will check first in which floor the elevator is .  
If we can't read the position of the elevator and it's not working we will let the elevator go up automatically



Network 3: movinf to the required floor

we are going to folow the same algorithm of the moving platform ,  
we are going to substract the elevator position from the floor to reach :  
if the value is + --> go up  
if the value is - --> go down  
if the value = 0 --> position has reached



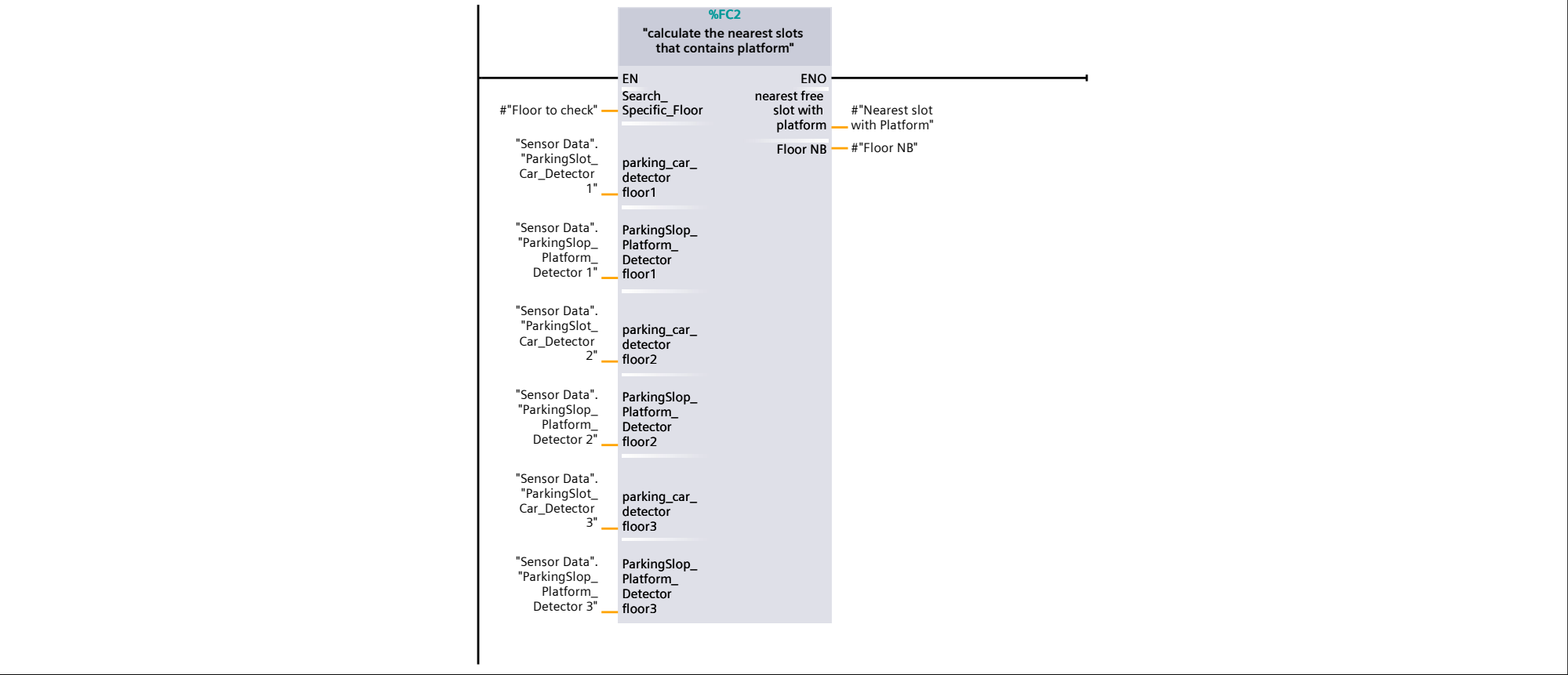
Program blocks

calculate the nearest slots that contains platform (to use) [FC11]

calculate the nearest slots that contains platform (to use) Properties							
General							
Name	calculate the nearest slots that contains platform (to use)	Number	11	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Floor to check	Int	
▼ Output		
Nearest slot with Platform	Int	
Floor NB	Int	
InOut		
Temp		
Constant		
▼ Return		
calculate the nearest slots that contains platform (to use)	Void	

Network 1:



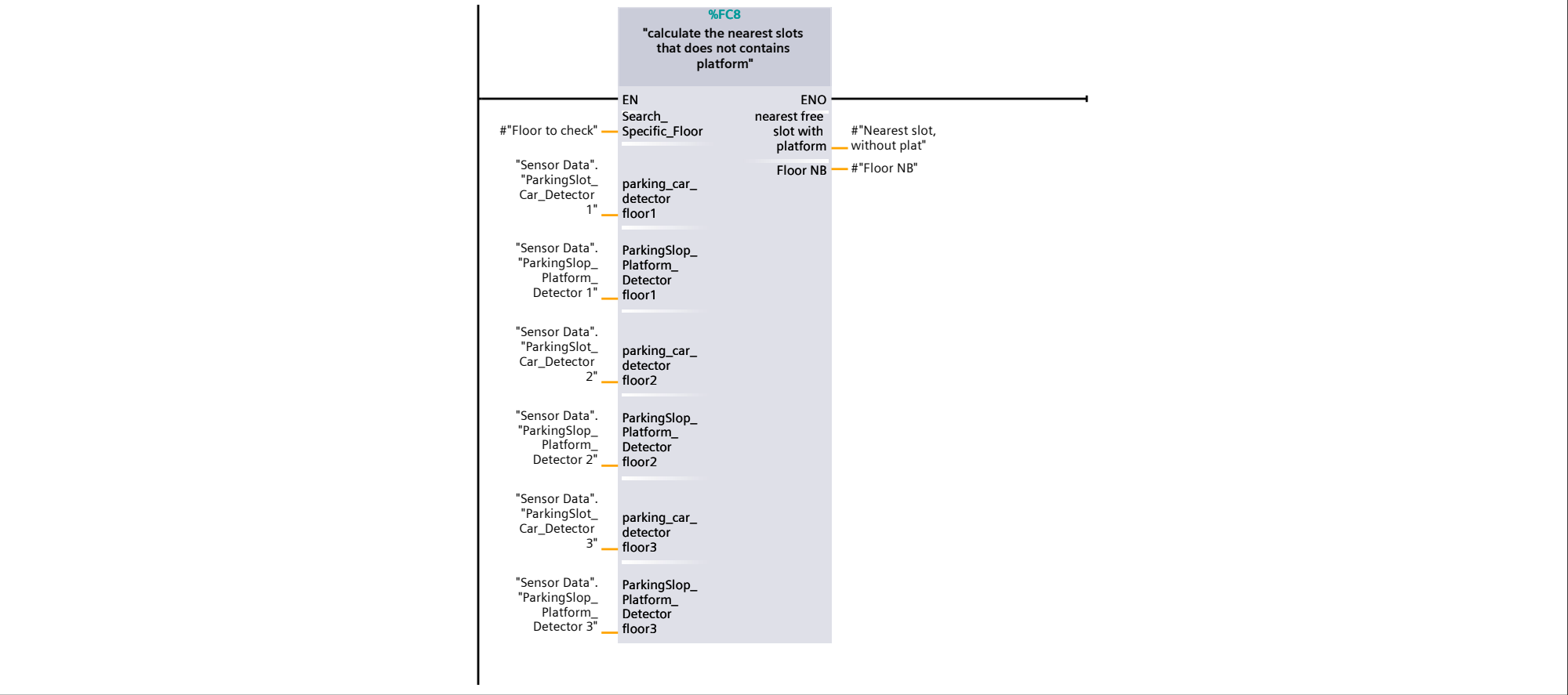
Program blocks

calculate the nearest slots that does not contains platform (to use) [FC12]

calculate the nearest slots that does not contains platform (to use) Properties							
General							
Name	calculate the nearest slots that does not contains platform (to use)	Number	12	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Floor to check	Int	
▼ Output		
Nearest slot, without plat	Int	
Floor NB	Int	
InOut		
Temp		
Constant		
▼ Return		
calculate the nearest slots that does not contains platform (to use)	Void	

Network 1:

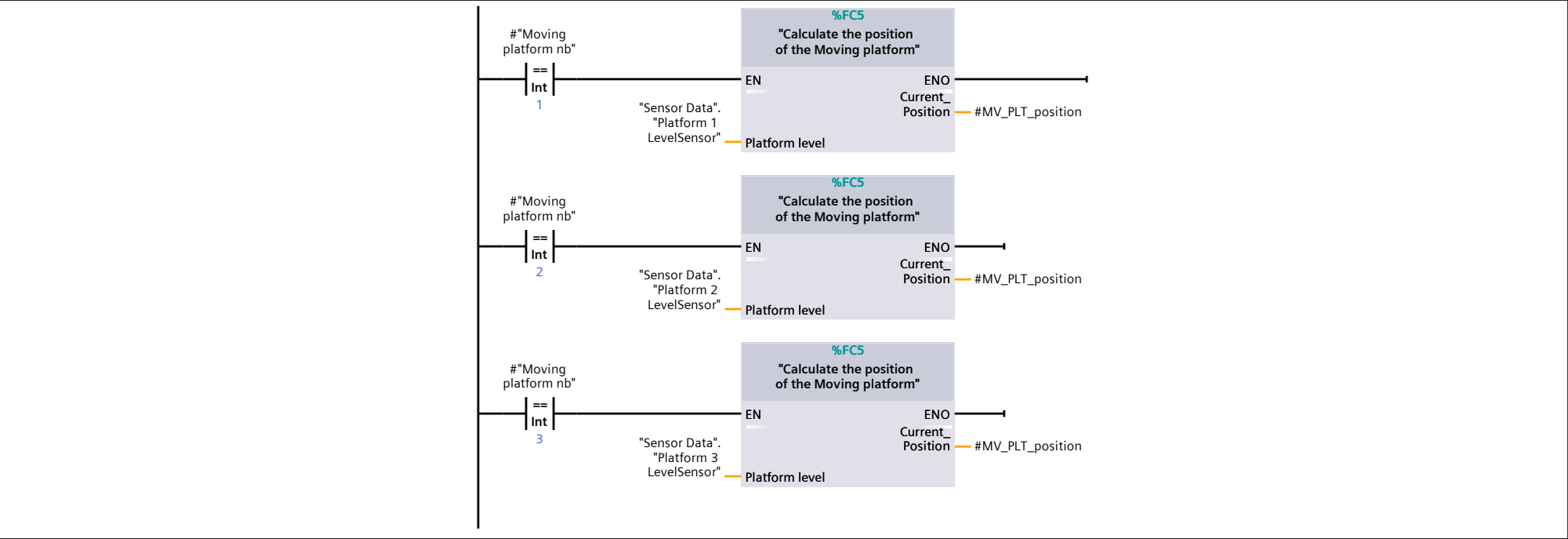


Program blocks

Calculate the position of the moving platform (to use) [FC13]

Calculate the position of the moving platform (to use) Properties							
General							
Name	Calculate the position of the moving platform (to use)	Number	13	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					
Name				Data type		Default value	
▼ Input							
Moving platform nb				Int			
▼ Output							
MV_PLT_position				Int			
InOut							
Temp							
Constant							
▼ Return							
Calculate the position of the moving platform (to use)				Void			

Network 1:





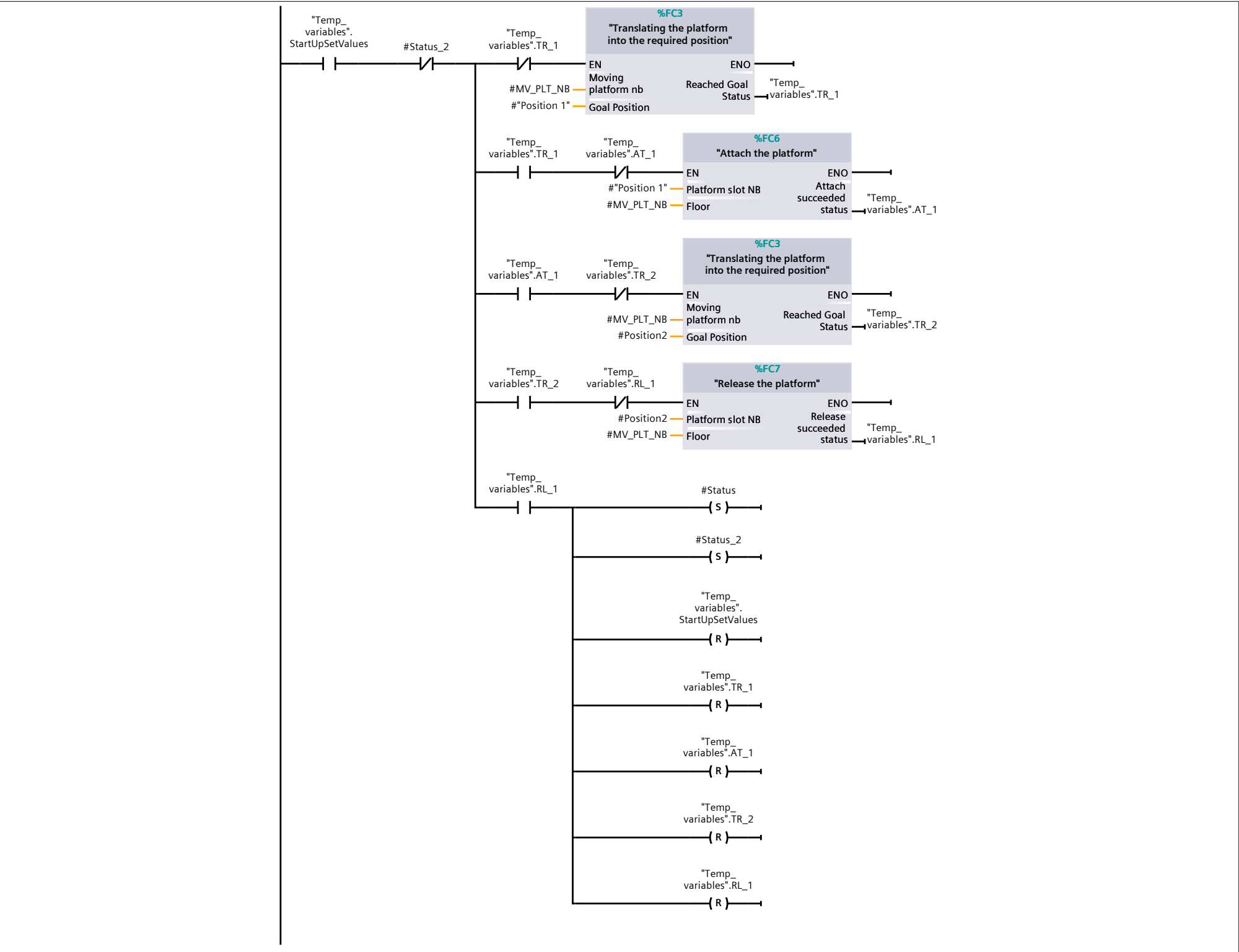
Program blocks

Translating platform from one position to another [FC14]

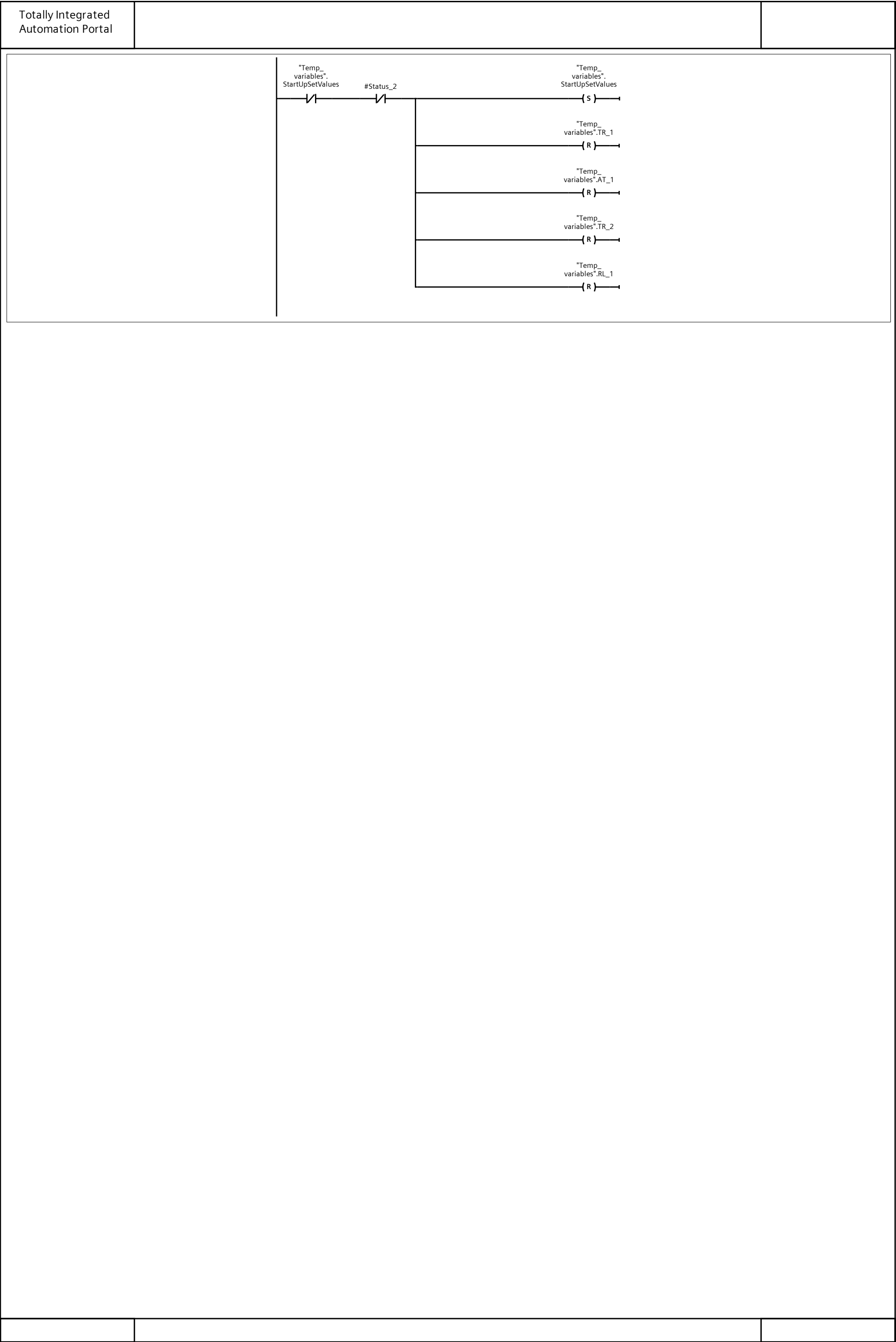
Translating platform from one position to another Properties							
General							
Name	Translating platform from one position to another	Number	14	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Position 1	Int	
Position2	Int	
MV_PLT_NB	Int	
▼ Output		
Status	Bool	
InOut		
▼ Temp		
Status_2	Bool	
Constant		
▼ Return		
Translating platform from one position to another	Void	

Network 1:



Network 2: startup



Program blocks

Translating Platform from one floor to another [FC15]

Translating Platform from one floor to another Properties

General

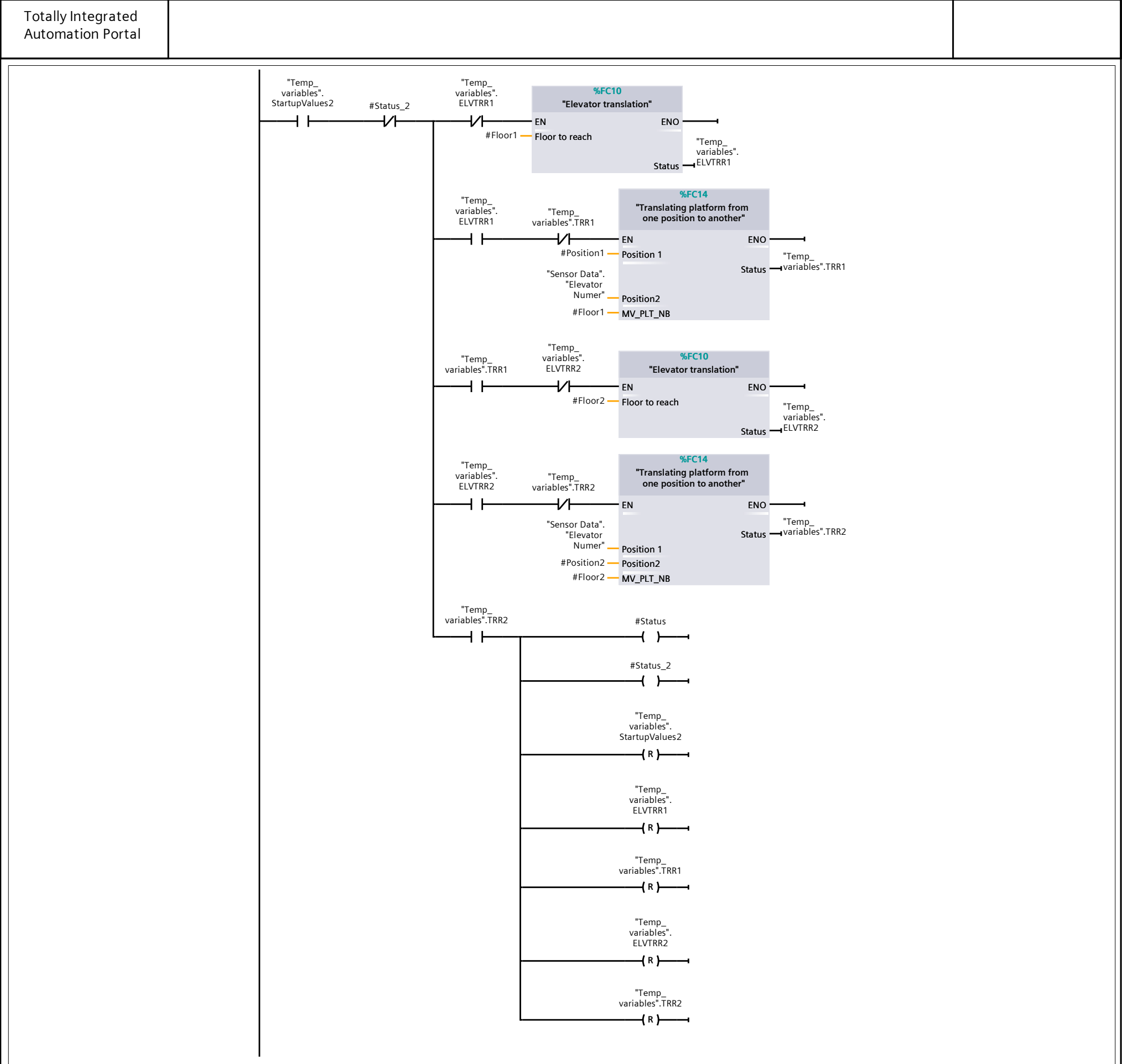
Name	Translating Platform from one floor to another	Number	15	Type	FC	Language	LAD
Numbering	Automatic						

Information

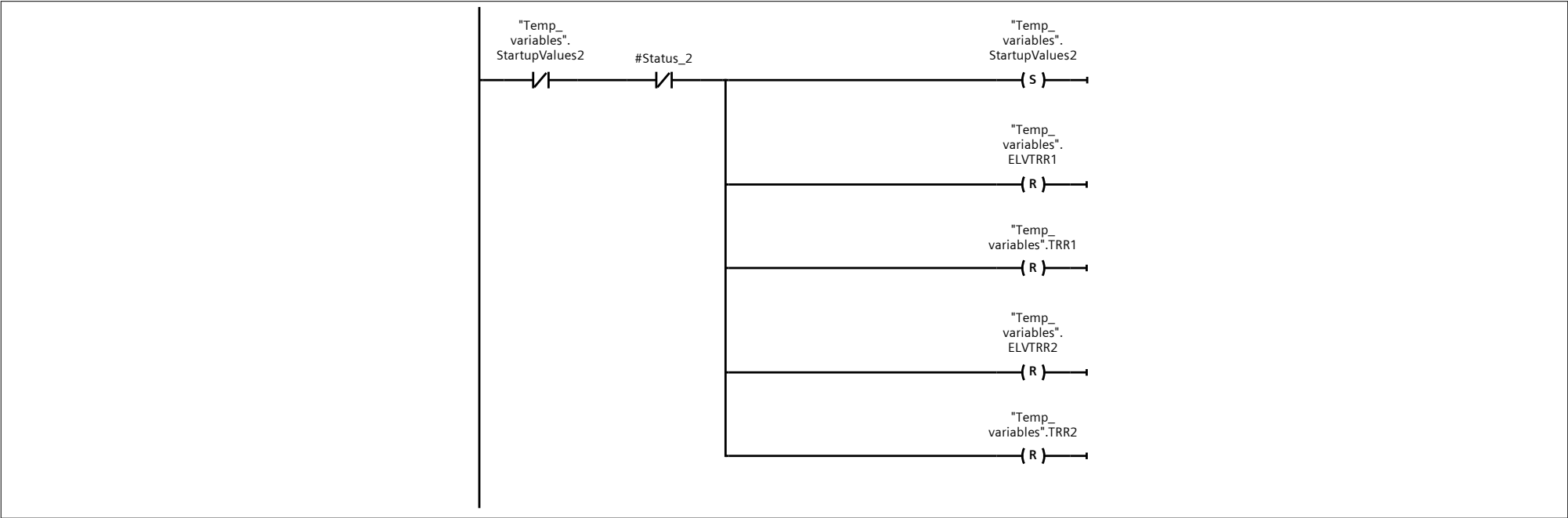
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Position1	Int	
Floor1	Int	
Position2	Int	
Floor2	Int	
▼ Output		
Status	Bool	
InOut		
▼ Temp		
Status_2	Bool	
Constant		
▼ Return		
Translating Platform from one floor to another	Void	

Network 1:



Network 2: startup



Program blocks

check the value of a specific Car [FC16]

check the value of a specific Car Properties							
General							
Name	check the value of a specific Car	Number	16	Type	FC	Language	SCL
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Car_number_to_check	Int	
Floor	Int	
ParkingSlot_Car_Detector_1	Array[0..8] of Bool	
ParkingSlot_Car_Detector_2	Array[0..8] of Bool	
ParkingSlot_Car_Detector_3	Array[0..8] of Bool	
▼ Output		
Car Status	Bool	
InOut		
Temp		
Constant		
▼ Return		
check the value of a specific Car	Void	

```
0001 // Initialize counters
0002 #"Car Status" := FALSE;
0003
0004 CASE #"Floor" OF
0005   1: // Statement section case 1
0006     IF #ParkingSlot_Car_Detector_1[#Car_number_to_check] = TRUE THEN
0007       #"Car Status" := TRUE; // Store the first free slot (column index)
0008     END_IF;
0009   ;
0010   2: // Statement section case 2
0011     IF #ParkingSlot_Car_Detector_2[#Car_number_to_check] = TRUE THEN
0012       #"Car Status" := TRUE; // Store the first free slot (column index)
0013     END_IF;
0014   ;
0015   3:
0016     IF #ParkingSlot_Car_Detector_3[#Car_number_to_check] = TRUE THEN
0017       #"Car Status" := TRUE; // Store the first free slot (column index)
0018     END_IF;
0019   ;
0020 ELSE // Statement section ELSE
0021   ;
0022 END_CASE;
```

Program blocks

check the value of a specific Car (to use) [FC17]

check the value of a specific Car (to use) Properties

General

Name	check the value of a specific Car (to use)	Number	17	Type	FC	Language	LAD
------	--	--------	----	------	----	----------	-----

Numbering

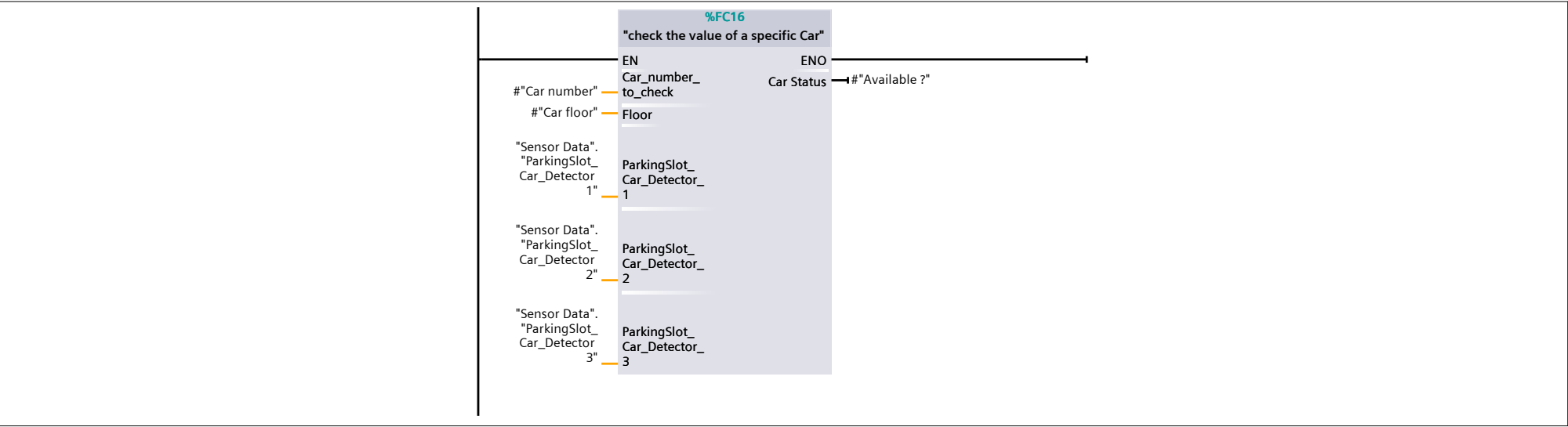
Automatic

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
Car number	Int	
Car floor	Int	
▼ Output		
Available ?	Bool	
InOut		
Temp		
Constant		
▼ Return		
check the value of a specific Car (to use)	Void	

Network 1:

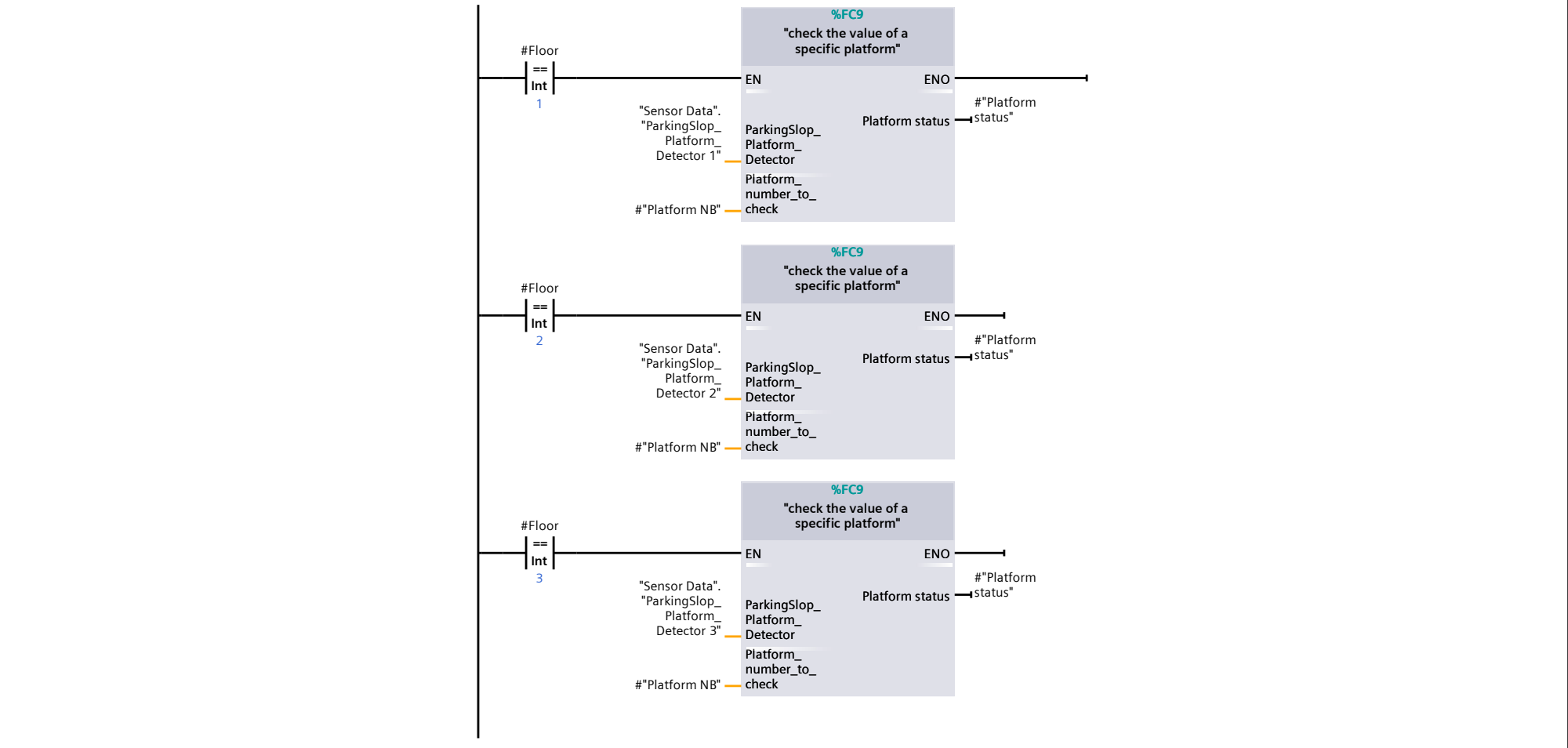


Program blocks

check the value of a specific platform (to use) [FC4]

check the value of a specific platform (to use) Properties							
General							
Name	check the value of a specific platform (to use)	Number	4	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					
Name				Data type		Default value	
▼ Input							
Floor				Int			
Platform NB				Int			
▼ Output							
Platform status				Bool			
InOut							
Temp							
Constant							
▼ Return							
check the value of a specific platform (to use)				Void			

Network 1:



Totally Integrated Automation Portal

Program blocks

CarTimers [FC18]

CarTimers Properties

General

Name	CarTimers	Number	18	Type	FC	Language	LAD
Numbering	Automatic						

Information

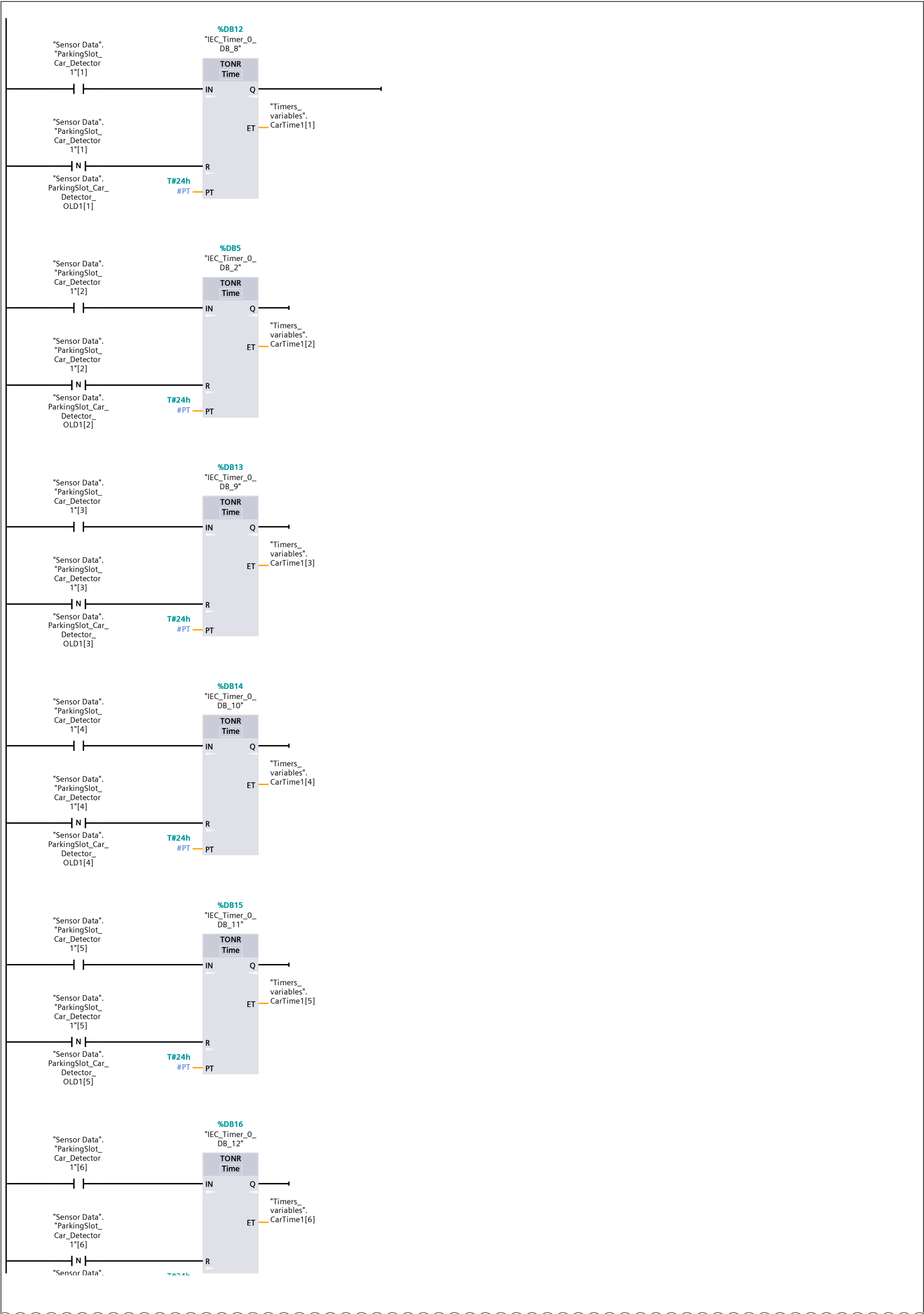
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
Input		
Output		
InOut		
Temp		
▼ Constant		
PT	Time	T#24h
▼ Return		
CarTimers	Void	

Network 1: floor1



Network 1: floor1 (1.1 / 2.1)



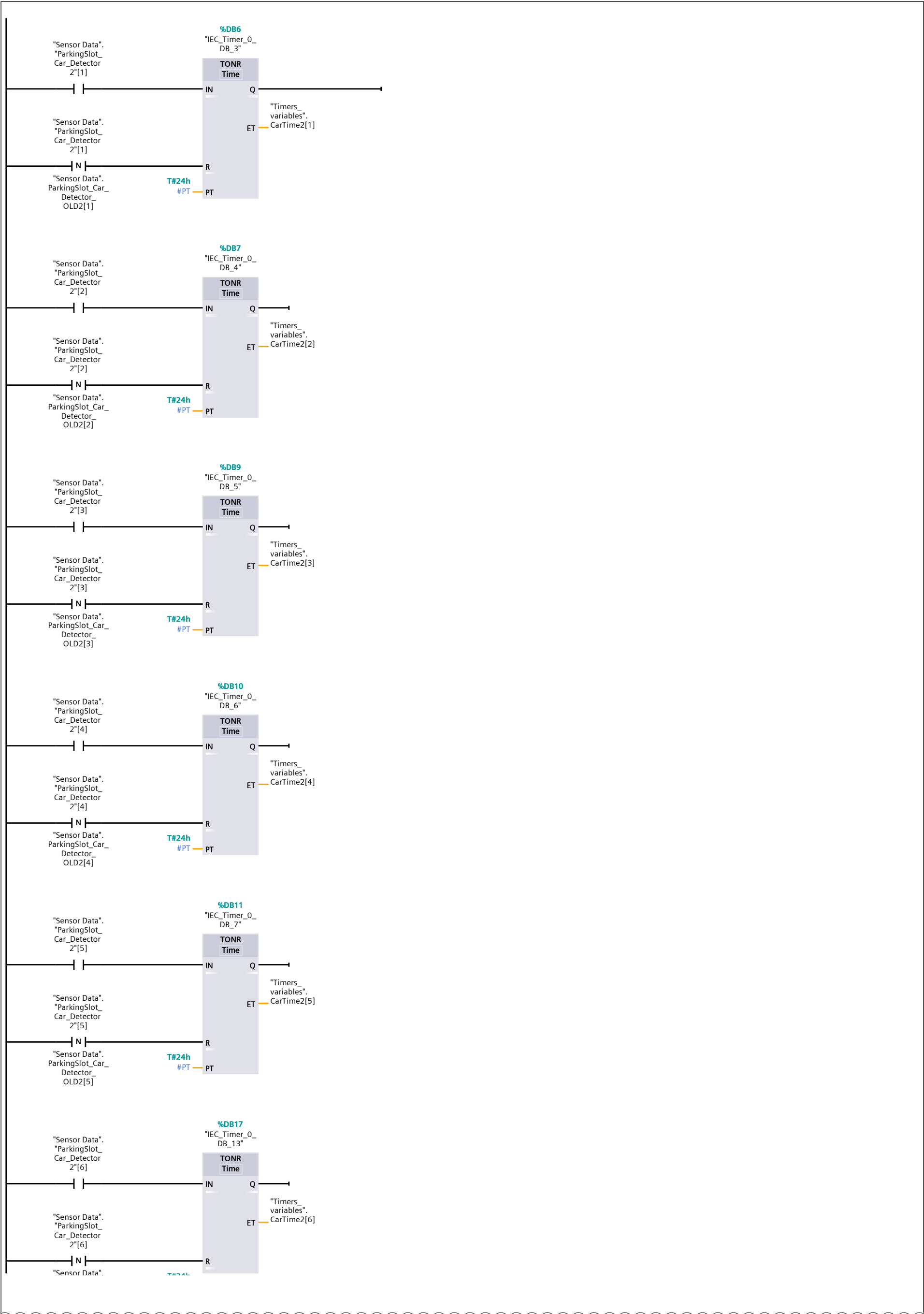
Sensor Data :  
ParkingSlot\_Car\_  
Detector\_  
OLD1[6]

T#24h  
#PT

PT

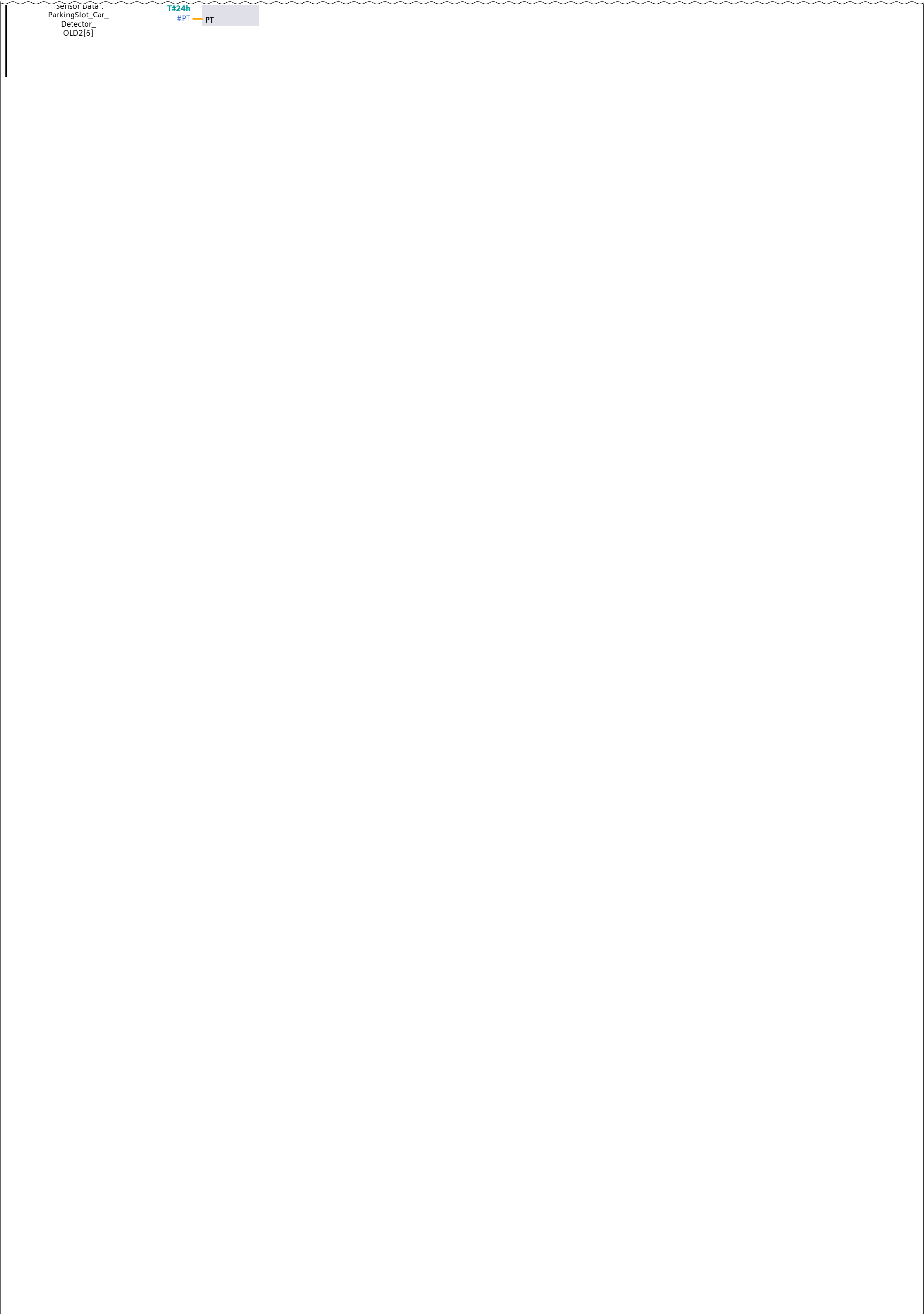
Totally Integrated Automation Portal		
Network 2: floor2		

Network 2: floor2 (1.1 / 2.1)



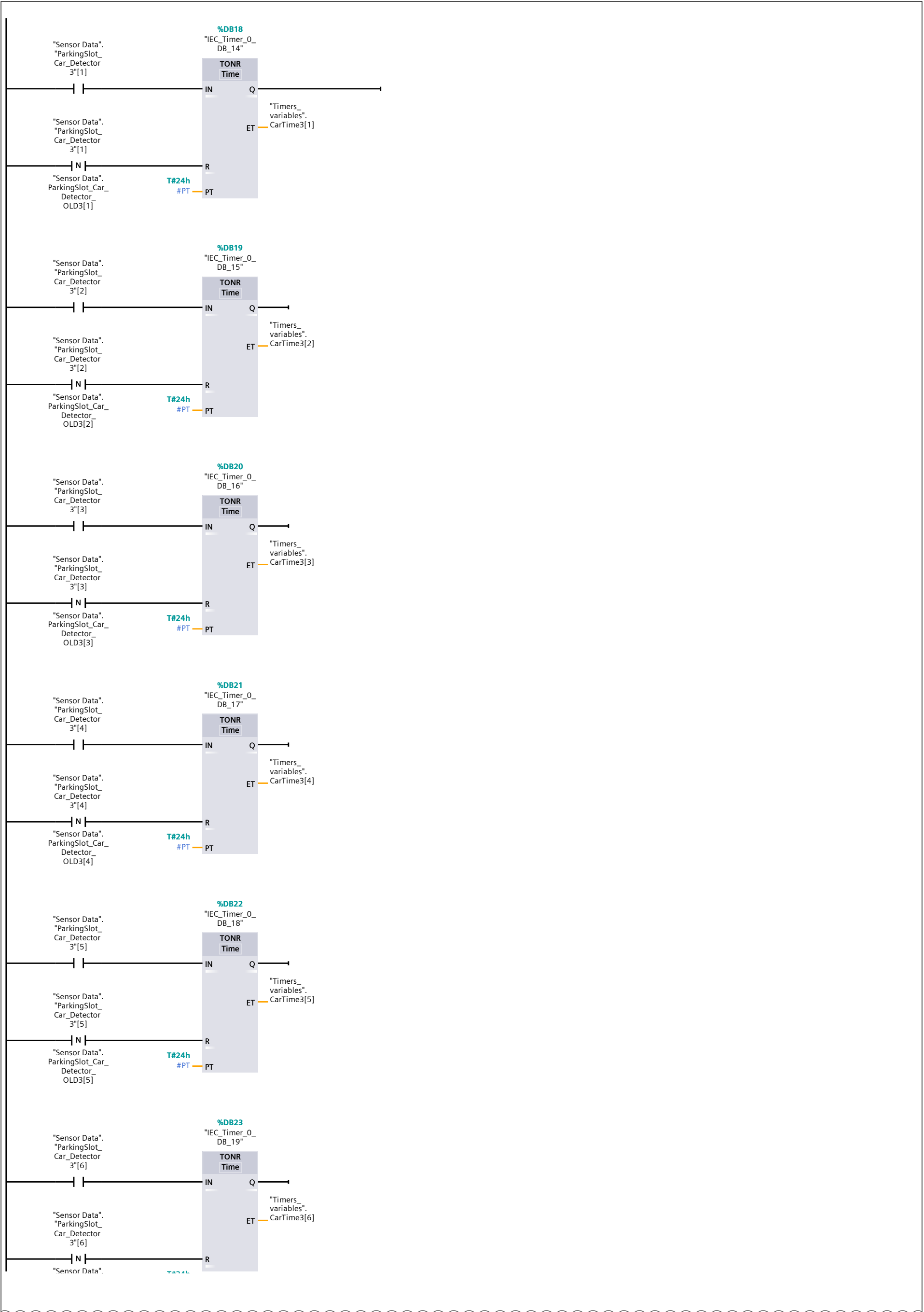
Network 2: floor2 (2.1 / 2.1)

1.1 ( Page21 - 5)



Totally Integrated Automation Portal		
Network 3: floor3		

Network 3: floor3 (1.1 / 2.1)



Sensor Data :  
ParkingSlot\_Car\_  
Detector\_  
OLD3[6]

T#24h  
#PT — PT

T#24h  
#PT — PT

PT



Totally Integrated Automation Portal

### Program blocks

#### Timers\_variables [DB8]

Timers\_variables Properties

General

Name	Timers_variables	Number	8	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Start value	Retain
▼ Static			
CarTime1	Array[1..6] of Time		False
CarTime2	Array[1..6] of Time		False
CarTime3	Array[1..6] of Time		False

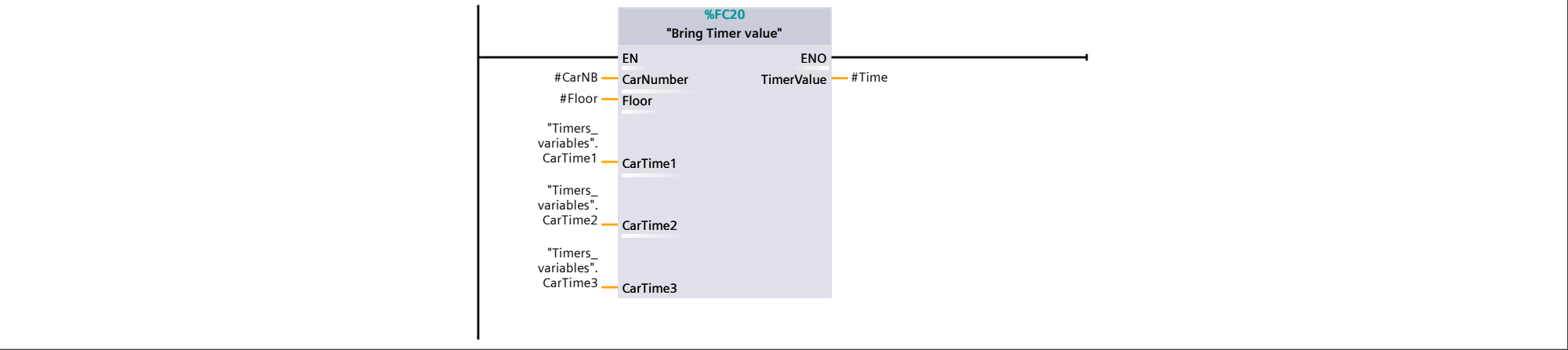
Program blocks

Bill [FC19]

Bill Properties							
General							
Name	Bill	Number	19	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

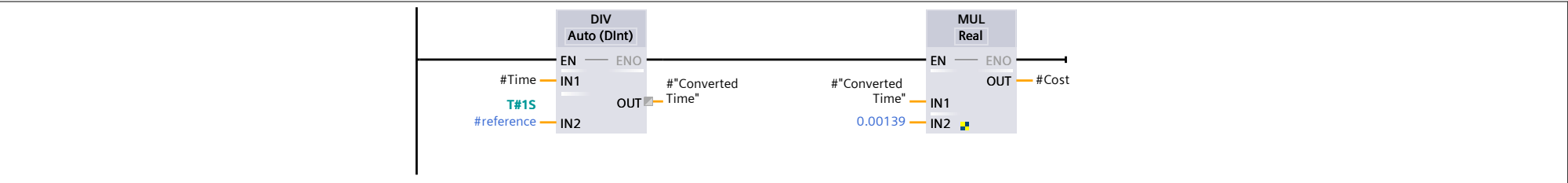
Name	Data type	Default value
▼ Input		
CarNB	Int	
Floor	Int	
▼ Output		
Cost	Real	
InOut		
▼ Temp		
Converted Time	Real	
Time	Time	
▼ Constant		
reference	Time	T#1S
▼ Return		
Bill	Void	

Network 1: Bring the timer



Network 2: cost calculation

The 1h cost 5\$ we will apply a linear calculation based on this information  
because of the integer value that we are going to get from division we are going to use the folowing formula :  
1s --->0.00139  
xS -----> ?  
we will divide the time with 1s to have a REAL number of seconds



Program blocks

Bring Timer value [FC20]

Bring Timer value Properties							
General							
Name	Bring Timer value	Number	20	Type	FC	Language	SCL
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
▼ Input		
CarNumber	Int	
Floor	Int	
CarTime1	Array[1..6] of Time	
CarTime2	Array[1..6] of Time	
CarTime3	Array[1..6] of Time	
▼ Output		
TimerValue	Time	
InOut		
Temp		
Constant		
▼ Return		
Bring Timer value	Void	

```
0001 #TimerValue := T#0S;
0002 CASE #"Floor" OF
0003   1:  // Statement section case 1
0004     #TimerValue := #CarTime1[#CarNumber]
0005   ;
0006   2:  // Statement section case 2
0007     #TimerValue := #CarTime2[#CarNumber]
0008   ;
0009   3:
0010     #TimerValue := #CarTime3[#CarNumber]
0011   ;
0012 END_CASE;
```

Totally Integrated Automation Portal

Program blocks

REsetOUT [FC21]

REsetOUT Properties

General

Name	REsetOUT	Number	21	Type	FC	Language	LAD
Numbering	Automatic						

Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
Input		
Output		
InOut		
Temp		
Constant		
▼ Return		
REsetOUT	Void	

Network 1: Output to reset

Totally Integrated Automation Portal		
	<div><div></div><div><div>%Q0.0 "EntryGate"</div><div>( R )</div></div><div><div>%Q0.1 "ExitGate"</div><div>( R )</div></div><div><div>%Q0.2 "Ticket_ Dispenser"</div><div>( R )</div></div><div><div>%Q0.3 "Platform1_ Motor_forward"</div><div>( R )</div></div><div><div>%Q0.4 "Platform1_ Motor_ Backword"</div><div>( R )</div></div><div><div>%Q0.5 "Platform1_ Motor_ SwipeRight"</div><div>( R )</div></div><div><div>%Q0.6 "Platform1_ Motor_ SwipeLeft"</div><div>( R )</div></div><div><div>%Q0.7 "Elevator_ Motor_Up"</div><div>( R )</div></div><div><div>%Q1.0 "Elevator_ Motor_Down"</div><div>( R )</div></div><div><div>%Q1.2 "entry_LED"</div><div>( R )</div></div><div><div>%Q1.3 "Exit_LED"</div><div>( R )</div></div><div><div>%Q2.0 "Platform2_ Motor_forward"</div><div>( R )</div></div><div><div>%Q2.1 "Platform2_ Motor_ backward"</div><div>( R )</div></div><div><div>%Q1.5 "Platform2_ Motor_ SwipeRight"</div><div>( R )</div></div><div><div>%Q1.4 "Platform2_ Motor_ SwipeLeft"</div><div>( R )</div></div><div><div>%Q3.1 "Platform3_ Motor_ backward"</div><div>( R )</div></div><div><div>%Q3.0 "Platform3_ Motor_forward"</div><div>( R )</div></div><div><div>%Q2.2 "Platform3_ Motor_ SwipeLeft"</div><div>( R )</div></div><div><div>%Q1.6 "Platform3_ Motor_ SwipeRight"</div><div>( R )</div></div></div>	

Program blocks

RSet [FC22]

RSet Properties

General

Name	RSet	Number	22	Type	FC	Language	LAD
Numbering	Automatic						

Information

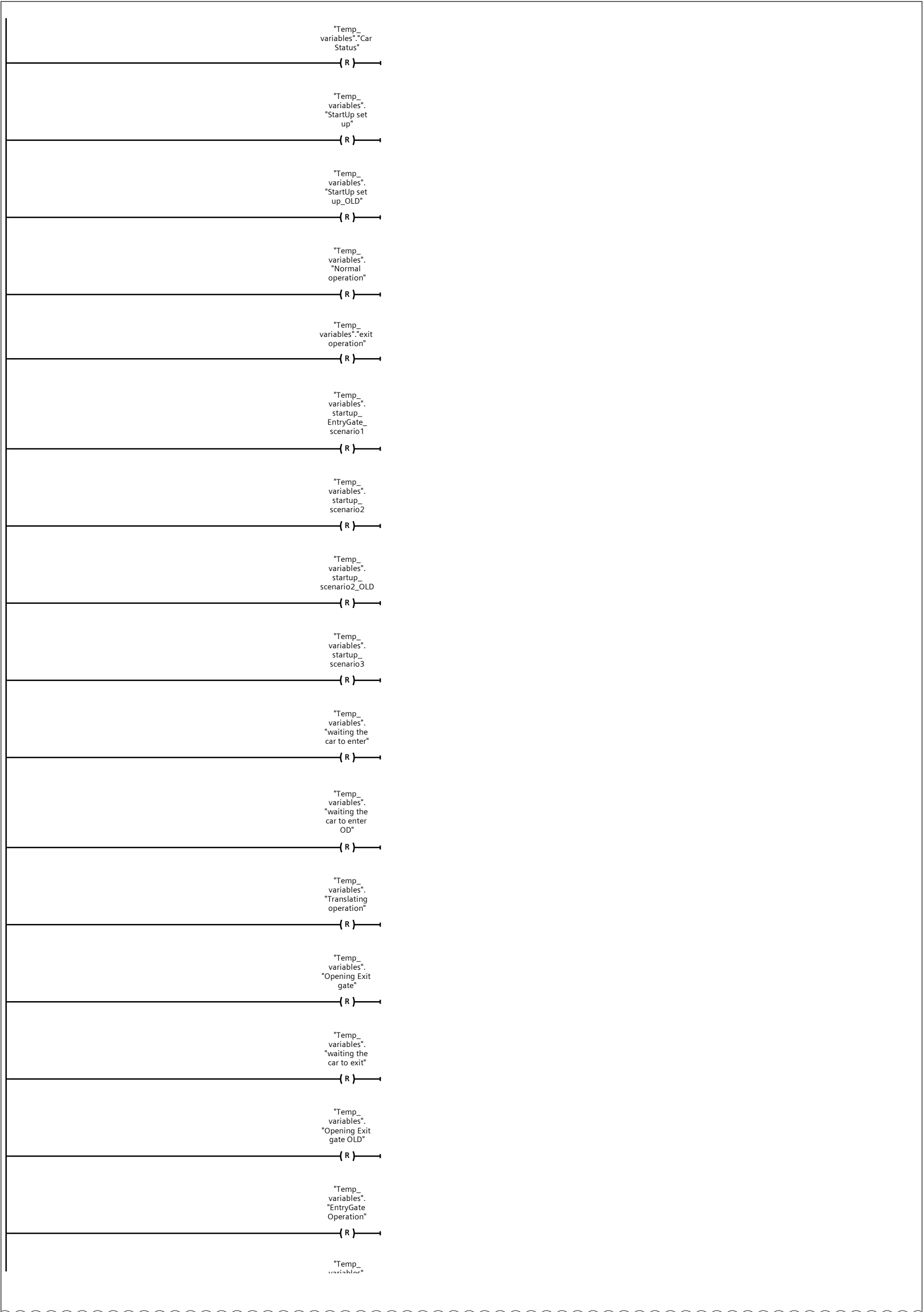
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value
Input		
Output		
InOut		
Temp		
Constant		
▼ Return		
RSet	Void	

Network 1: Output to reset



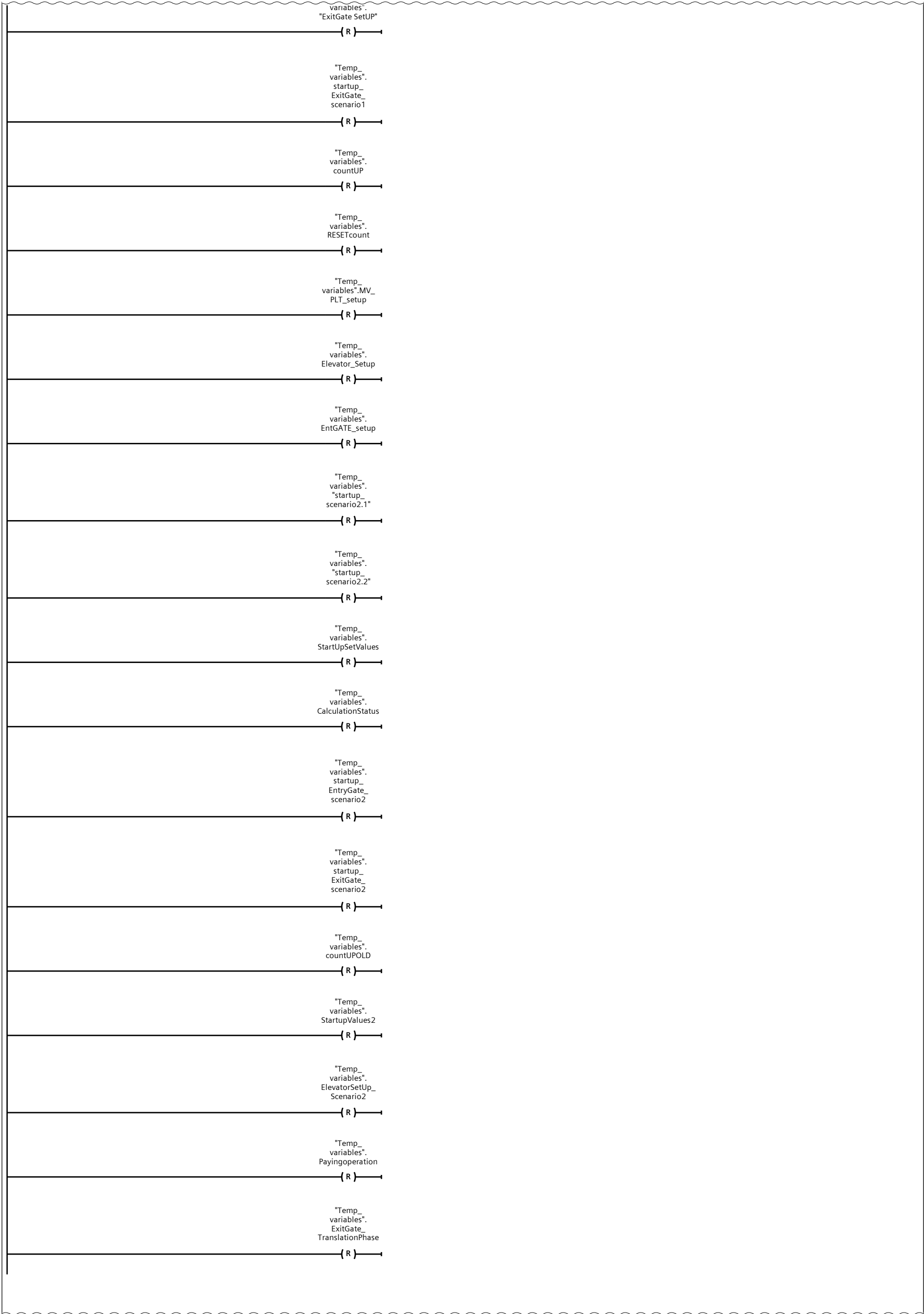
Network 2: TemporaryTags (1.1 / 3.1)





Network 2: TemporaryTags (2.1 / 3.1)

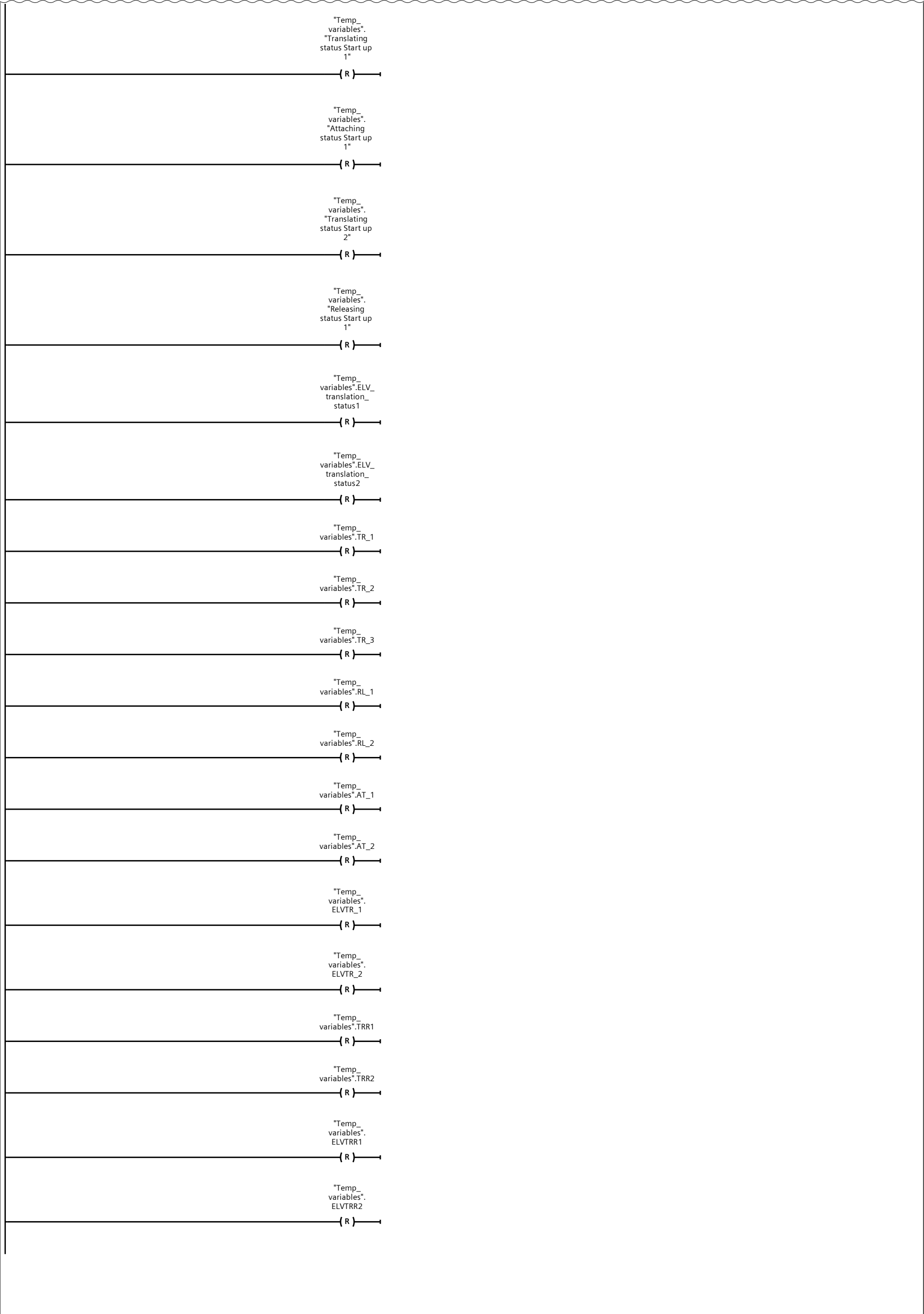
1.1 ( Page26 - 3)



3.1 ( Page26 - 5)

Network 2: TempororyTags (3.1 / 3.1)

2.1 ( Page26 - 4)



Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB [DB3]

IEC\_Timer\_0\_DB Properties

General

Name	IEC_Timer_0_DB	Number	3	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_1 [DB4]

IEC_Timer_0_DB_1 Properties							
General							
Name	IEC_Timer_0_DB_1	Number	4	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_8 [DB12]

IEC_Timer_0_DB_8 Properties							
General							
Name	IEC_Timer_0_DB_8	Number	12	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_2 [DB5]

IEC_Timer_0_DB_2 Properties							
General							
Name	IEC_Timer_0_DB_2	Number	5	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_9 [DB13]

IEC_Timer_0_DB_9 Properties							
General							
Name	IEC_Timer_0_DB_9	Number	13	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Totally Integrated Automation Portal

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_10 [DB14]

IEC\_Timer\_0\_DB\_10 Properties

General

Name	IEC_Timer_0_DB_10	Number	14	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False



Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_11 [DB15]

IEC\_Timer\_0\_DB\_11 Properties

General

Name	IEC_Timer_0_DB_11	Number	15	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_12 [DB16]

IEC_Timer_0_DB_12 Properties							
General							
Name	IEC_Timer_0_DB_12	Number	16	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_3 [DB6]

IEC\_Timer\_0\_DB\_3 Properties

General

Name	IEC_Timer_0_DB_3	Number	6	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_4 [DB7]

IEC\_Timer\_0\_DB\_4 Properties

General

Name	IEC_Timer_0_DB_4	Number	7	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_5 [DB9]

IEC_Timer_0_DB_5 Properties							
General							
Name	IEC_Timer_0_DB_5	Number	9	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_6 [DB10]

IEC\_Timer\_0\_DB\_6 Properties

General

Name	IEC_Timer_0_DB_6	Number	10	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_7 [DB11]

IEC_Timer_0_DB_7 Properties							
General							
Name	IEC_Timer_0_DB_7	Number	11	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_13 [DB17]

IEC\_Timer\_0\_DB\_13 Properties

General

Name	IEC_Timer_0_DB_13	Number	17	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False



Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_14 [DB18]

IEC\_Timer\_0\_DB\_14 Properties

General

Name	IEC_Timer_0_DB_14	Number	18	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_15 [DB19]

IEC_Timer_0_DB_15 Properties							
General							
Name	IEC_Timer_0_DB_15	Number	19	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type		Start value		Retain
▼ Static							
PT			Time		T#0ms		False
ET			Time		T#0ms		False
IN			Bool		false		False
Q			Bool		false		False

Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_16 [DB20]

IEC\_Timer\_0\_DB\_16 Properties

General

Name	IEC_Timer_0_DB_16	Number	20	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Program blocks / System blocks / Program resources

IEC\_Timer\_0\_DB\_17 [DB21]

IEC_Timer_0_DB_17 Properties							
General							
Name	IEC_Timer_0_DB_17	Number	21	Type	DB	Language	DB
Numbering	Automatic						
Information							
Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				
Name			Data type	Start value		Retain	
▼ Static							
PT			Time	T#0ms		False	
ET			Time	T#0ms		False	
IN			Bool	false		False	
Q			Bool	false		False	

Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_18 [DB22]

IEC\_Timer\_0\_DB\_18 Properties

General

Name	IEC_Timer_0_DB_18	Number	22	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False

Totally Integrated Automation Portal

### Program blocks / System blocks / Program resources

#### IEC\_Timer\_0\_DB\_19 [DB23]

IEC\_Timer\_0\_DB\_19 Properties

General

Name	IEC_Timer_0_DB_19	Number	23	Type	DB	Language	DB
Numbering	Automatic						

Information

Title		Author	Simatic	Comment		Family	IEC
Version	1.0	User-defined ID	IEC_TMR				

Name	Data type	Start value	Retain
▼ Static			
PT	Time	T#0ms	False
ET	Time	T#0ms	False
IN	Bool	false	False
Q	Bool	false	False