Last Codesys Mini Project: Develop the logic and HMI for an elevator used in a 5-storey building.

Description: The objective of this assignment is to develop a:

- An elevator model (simulator) that is used to simulate the position of an elevator in a building assumed to have five floors (ground + 4) labelled as floors 0 to 4.
- A controller that automates the operation of the elevator based on sensor and user inputs.

STEP 1: Develop a simplified simulation of the elevator.

This is the first step where you will develop a program (a function block) that calculates and displays the position and movements of the elevator. The inputs and outputs of this program (elevator model) are:

Inputs:

- Elevator UP (Boolean): The elevator moves up when this input is true.
- Elevator DN (Boolean): The elevator moves down when this input is true.
- Elevator Door (Boolean): Used to open / close elevator door.

  During the initial testing of the elevator model, the Elevator UP and Elevator DN inputs are manually generated using two push buttons. The elevator is stationary if both Elevator UP = Elevator DN = 0. Elevator UP and Elevator DN cannot be simultaneously true.

Outputs:

- **Elevator position** (UINT): A number that uniquely defines the position of the elevator. This typically varies from 0 to a suitable maximum value (depending on your monitor) which causes the position of the lift to change from the near bottom of the screen to the near top of the screen, as the elevator moves from the ground floor to the top floor. The elevator may be represented by a rectangle or a stored image. Its vertical position is animated (varied) by assigning the elevator position variable to Properties > Absolute movement > Y axis.

- **Floor Switches** (Boolean): Floor 0 to Floor 4.

  The position of an elevator is determined by sensors (which are either limit switches or proximity sensors) installed on each floor. When the elevator reaches a particular floor, the output of the sensor installed on that floor is activated indicating the presence of the lift on that floor. In this assignment, we will simulate these sensors.

  The output of all floor sensors is LOW if the lift is in between any two floors.

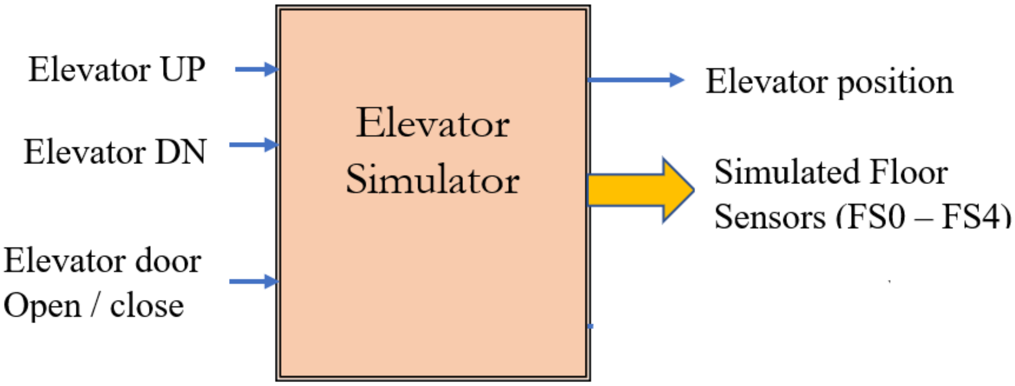
  The output of at most one-floor sensor can be HIGH at any point in time because the lift cannot simultaneously exist on multiple floors.

  The state of a simulated floor sensor is derived from the elevator position**.** The output of a floor sensor is true when the elevator position is exactly aligned with the floor.

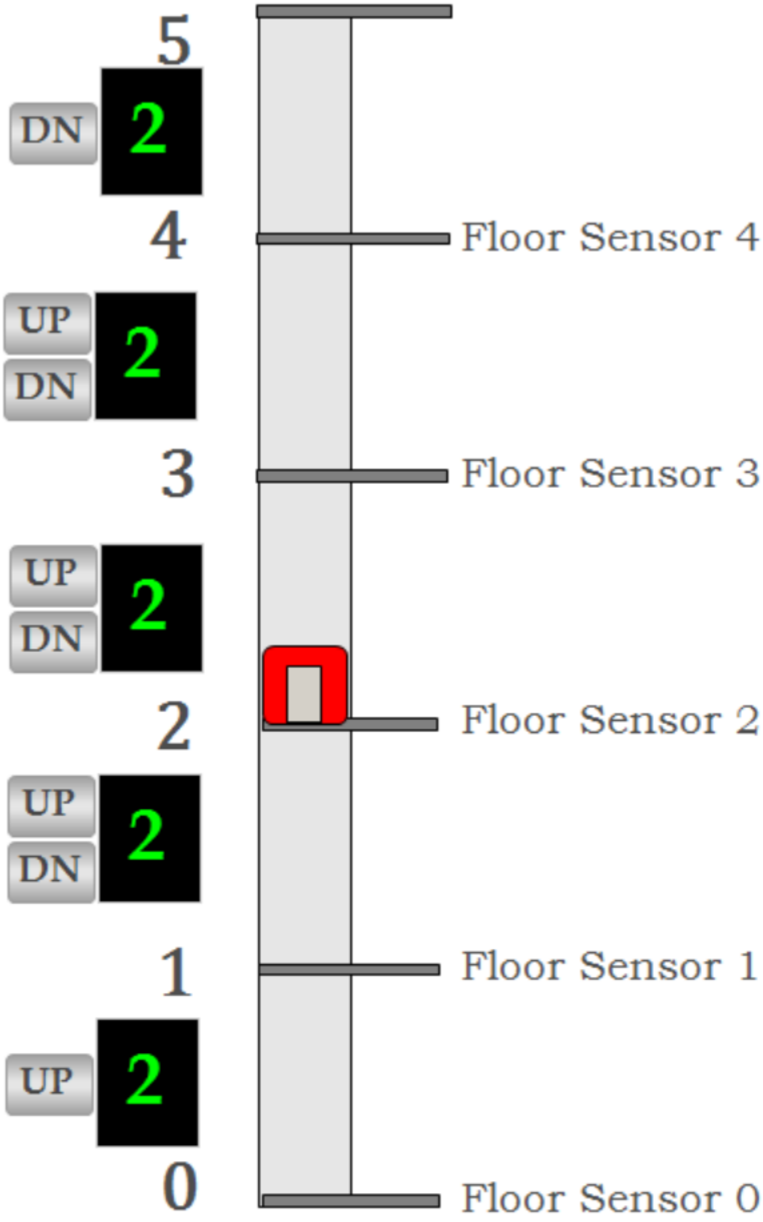
  If the lift is stationary on a particular floor (because people are moving into or out of the lift), then the output of that floor sensor is HIGH for an extended amount of time, equal to the duration for which the lift is on that floor.

  The lift door can be opened **only if** the elevator is stationary and on a particular floor.

- **Floor Display:** A variable that represents the current position of the elevator. This is displayed as a digit varying from 0 to 4, representing the floor where the lift is located. When the lift is moving, the last detected floor position is displayed until it reaches the next floor either above or below.


Elevator UP
Elevator DN
Elevator door Open / close
Elevator Simulator
Elevator position
Simulated Floor Sensors (FS0 – FS4)

A suggestive HMI is depicted below.



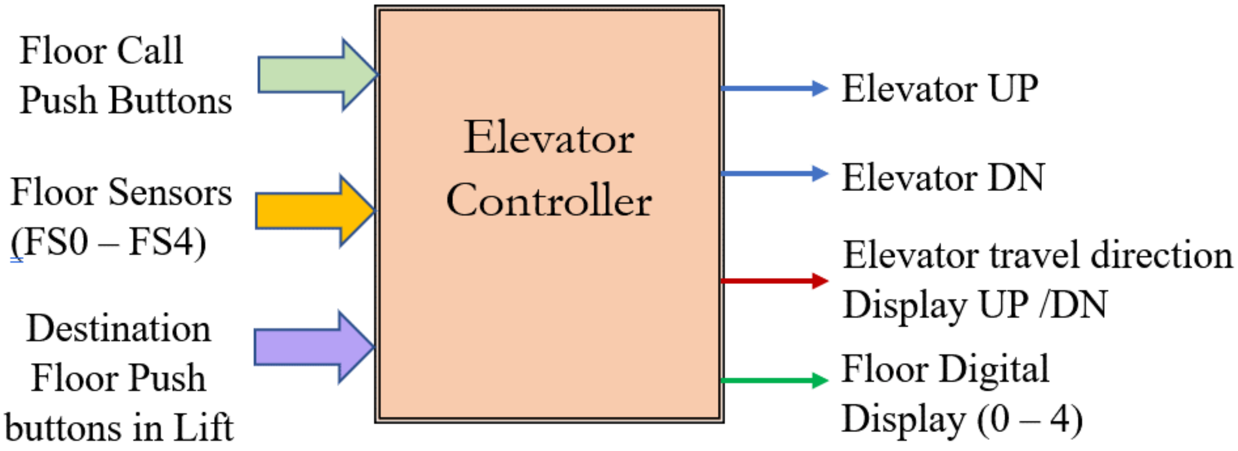
STEP 2: Develop a controller for the automatic operation of the elevator.

Inputs:

- Floor Sensors (FS0 – Fs4).
- Floor Call Push Buttons (on each floor).
- Destination Floor Push Buttons in elevator control panel.

Outputs:

- Elevator Control (Elevator UP).
- Elevator Control (Elevator DN).
- Elevator travel direction indicator (on each floor).
- Digital Floor Display (on each floor).


Floor Call Push Buttons
Floor Sensors (FS0 – FS4)
Destination Floor Push buttons in Lift
Elevator Controller
Elevator UP
Elevator DN
Elevator travel direction Display UP /DN
Floor Digital Display (0 – 4)

Floor Call Push Buttons & Elevator travel direction display

Destination Floor push buttons & Door open/close push buttons inside the Elevator

- Each floor contains two push buttons, commonly known as “floor calls” which when pressed ‘informs’ the elevator controller that a person is waiting for the elevator on a floor. There are two push buttons, one with an upward pointing arrow (or marked UP) and the other with a downward arrow (or marked DN). Pressing any one of these push buttons causes the lift to move to that floor. Pressing the push button marked with an upward arrow means that a person on that floor wants to travel to an upper floor and the push button with a downward arrow indicates that a person wants to travel to a lower floor. The ground floor has only one button pointed upwards and the top floor has one button that points downwards.
- Each floor also contains a digital indicator that displays the current position of the lift and the direction in which the left is presently travelling. A typical arrangement of push buttons and indicators, found on the floor is shown in the figure above.

- A lift also travels when a command is issued from the elevator by pressing a destination floor button. There is one button for each floor and the button pressed indicates the destination floor to which the lift travels. Multiple buttons may be pressed, in which case the elevator travels to all the floors commanded. A typical control panel of a lift is shown in the figure above. The two buttons at the bottom are for opening and closing the lift door manually.

Elevator Controller – operational logic:

The elevator is stationary if all the conditions below are true:

- It has not been called from any floor above or below its current position.
- No command has been issued from the panel within the lift.
- A lift can be stationary only when it is on a floor, it cannot be stationary when it is in between two floors.

The elevator moves upwards if any of the conditions below is true:

- It receives a "Floor call" from a floor above its current position.
- A "destination floor button" is pressed in the lift panel which is above the present position of the elevator.

While moving upwards, the elevator stops at a floor if any of the conditions below is true:

- The "Floor Call-UP" push button on the floor was pressed.
- It was asked to travel to this floor by pressing the "destination floor button" on the lift panel.
- It has reached the topmost floor.

While moving upwards, the elevator does not stop at a floor, even if it receives a call from this floor if:

- The "Floor Call-DN" push button on the floor was pressed as the person wants to travel to a lower floor. The elevator stops on this floor when it is moving downwards.
- The lift also does not stop in response to a floor call if the lift is filled to its full capacity, because it is unable to take in any more persons. However, to simplify the logic, we will not include this consideration in our controller.

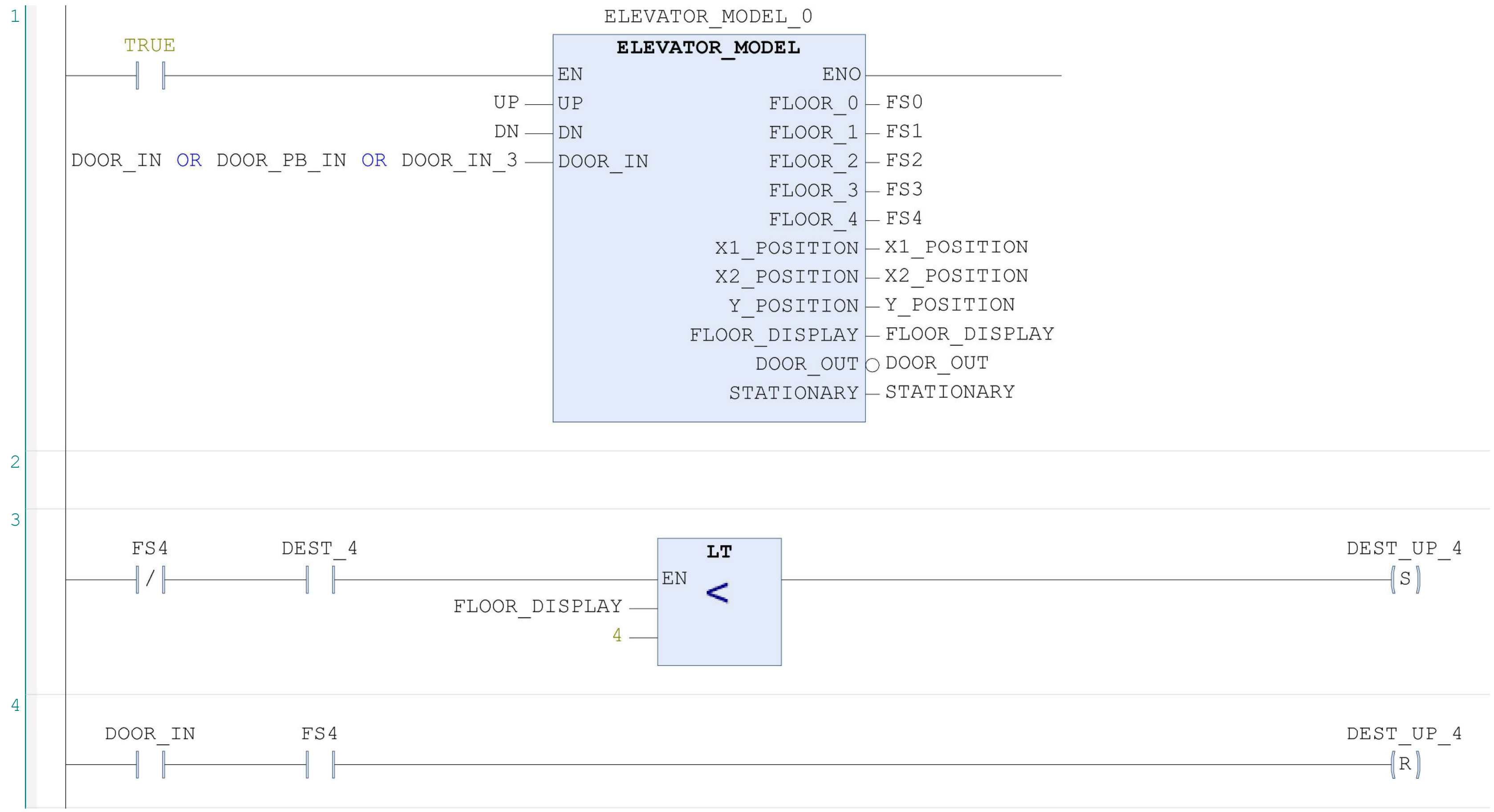
When a lift stops on any floor, the door automatically opens for 2 secs. After 2 secs the door closes and the lift resumes its journey. This duration can be increased in multiples of 2 seconds by pressing the open-door button on the lift panel. The lift door only opens when it is stationary on the floor. It does not open when the lift is travelling.

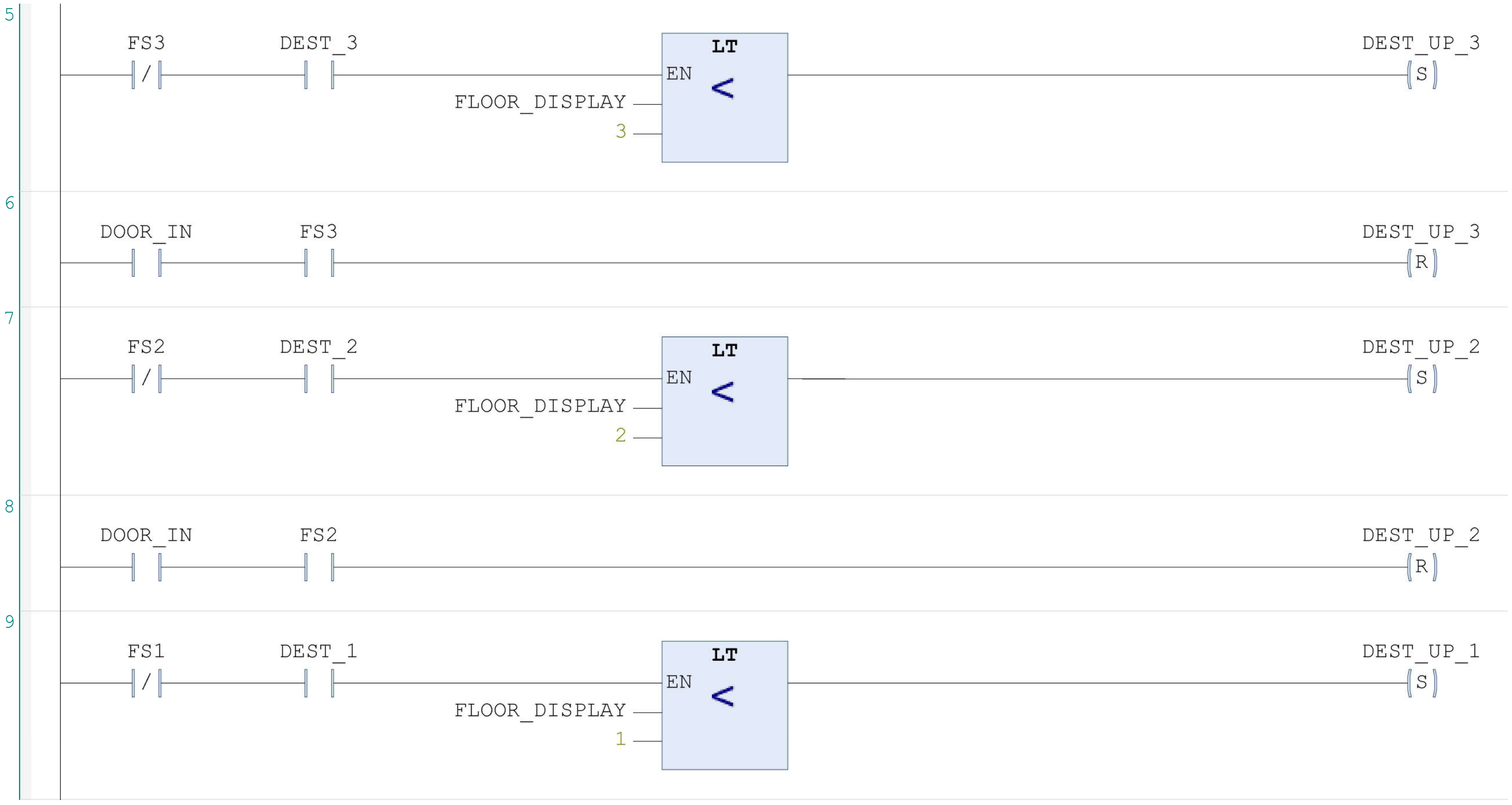
The lift moves downwards if……..

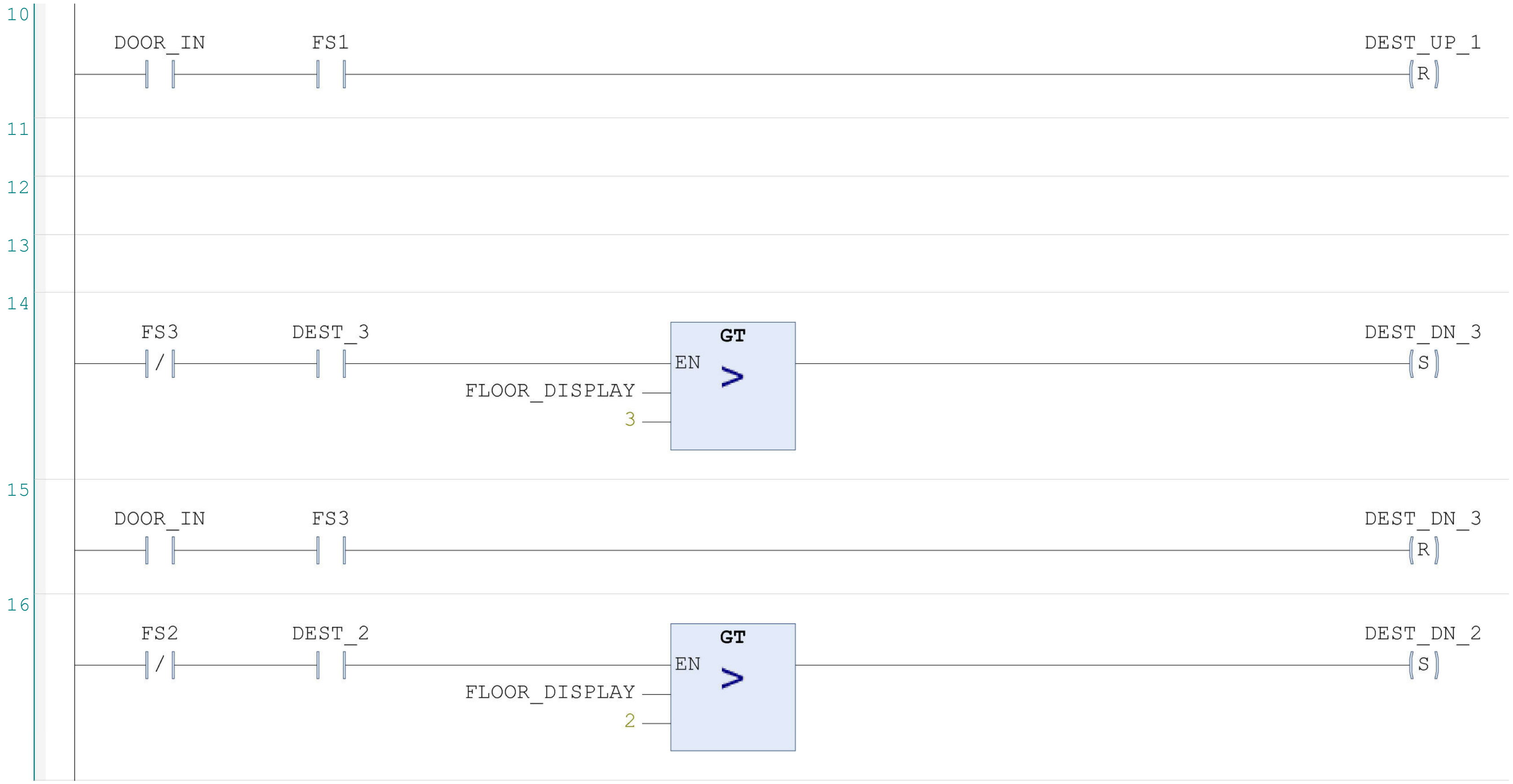
You can work out similar considerations as to when the lift moves downwards, and when the lift stops while moving downwards.

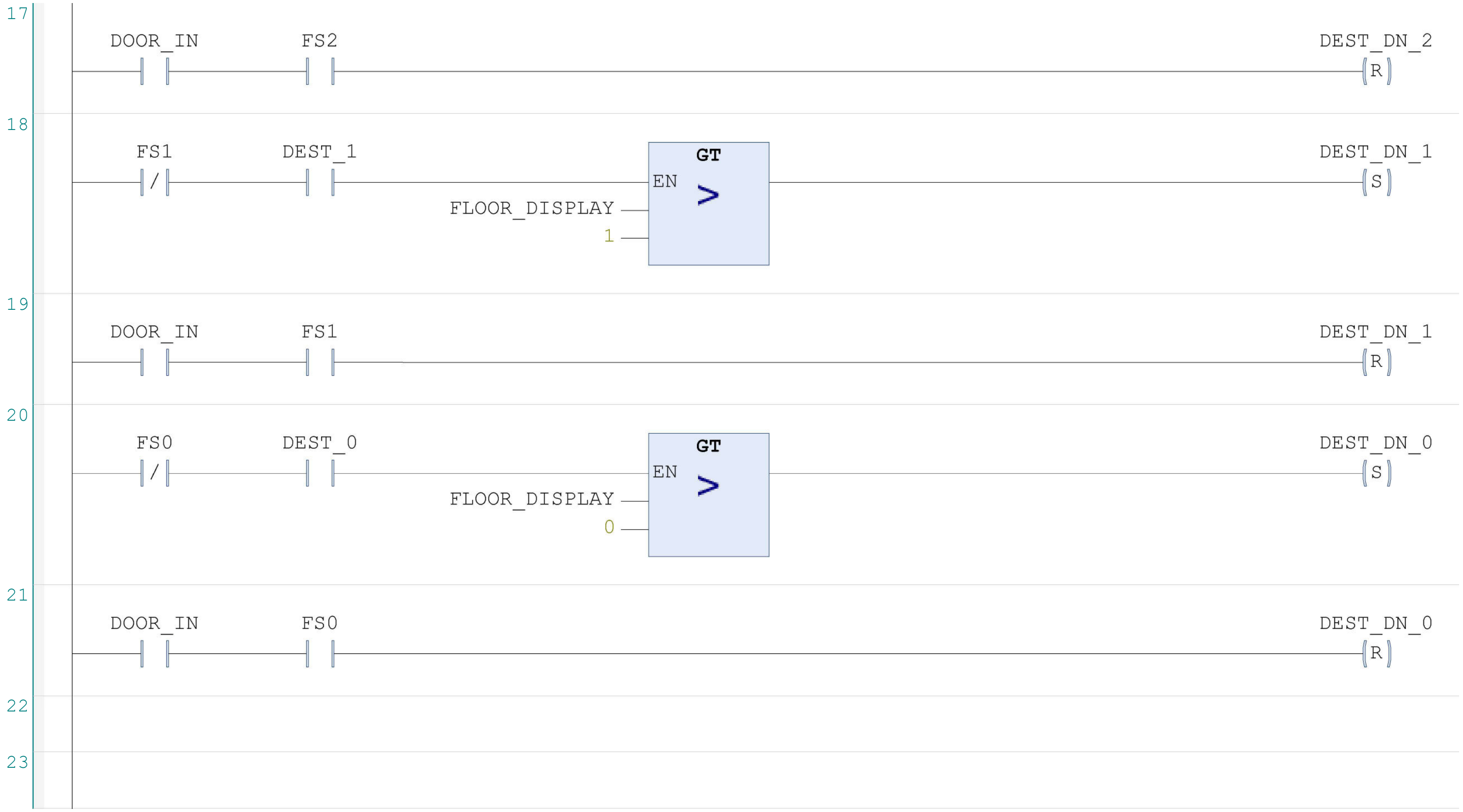
```
PROGRAM PLC_PRG
VAR
    ELEVATOR_MODEL_0 : ELEVATOR_MODEL ;
    UP : BOOL ;
    DN : BOOL ;
    FS0 : BOOL ;
    FS1 : BOOL ;
    FS2 : BOOL ;
    FS3 : BOOL ;
    FS4 : BOOL ;
    Y_POSITION : REAL := 0 ;
    FLOOR_DISPLAY : UINT ;
    X1_POSITION : REAL := 0 ;
    X2_POSITION : REAL := 0 ;


    DEST_0 : BOOL ;
    DEST_1 : BOOL ;
    DEST_2 : BOOL ;
    DEST_3 : BOOL ;
    DEST_4 : BOOL ;
    UP4 : BOOL ;
    UP3 : BOOL ;
    UP2 : BOOL ;
    UP1 : BOOL ;
    DN0 : BOOL ;
    DN3 : BOOL ;
    DN2 : BOOL ;
    DN1 : BOOL ;

    DOOR_IN : BOOL ;
    DEST_UP_4 : BOOL ;
    DEST_UP_3 : BOOL ;
    DEST_UP_2 : BOOL ;
    DEST_UP_1 : BOOL ;
```

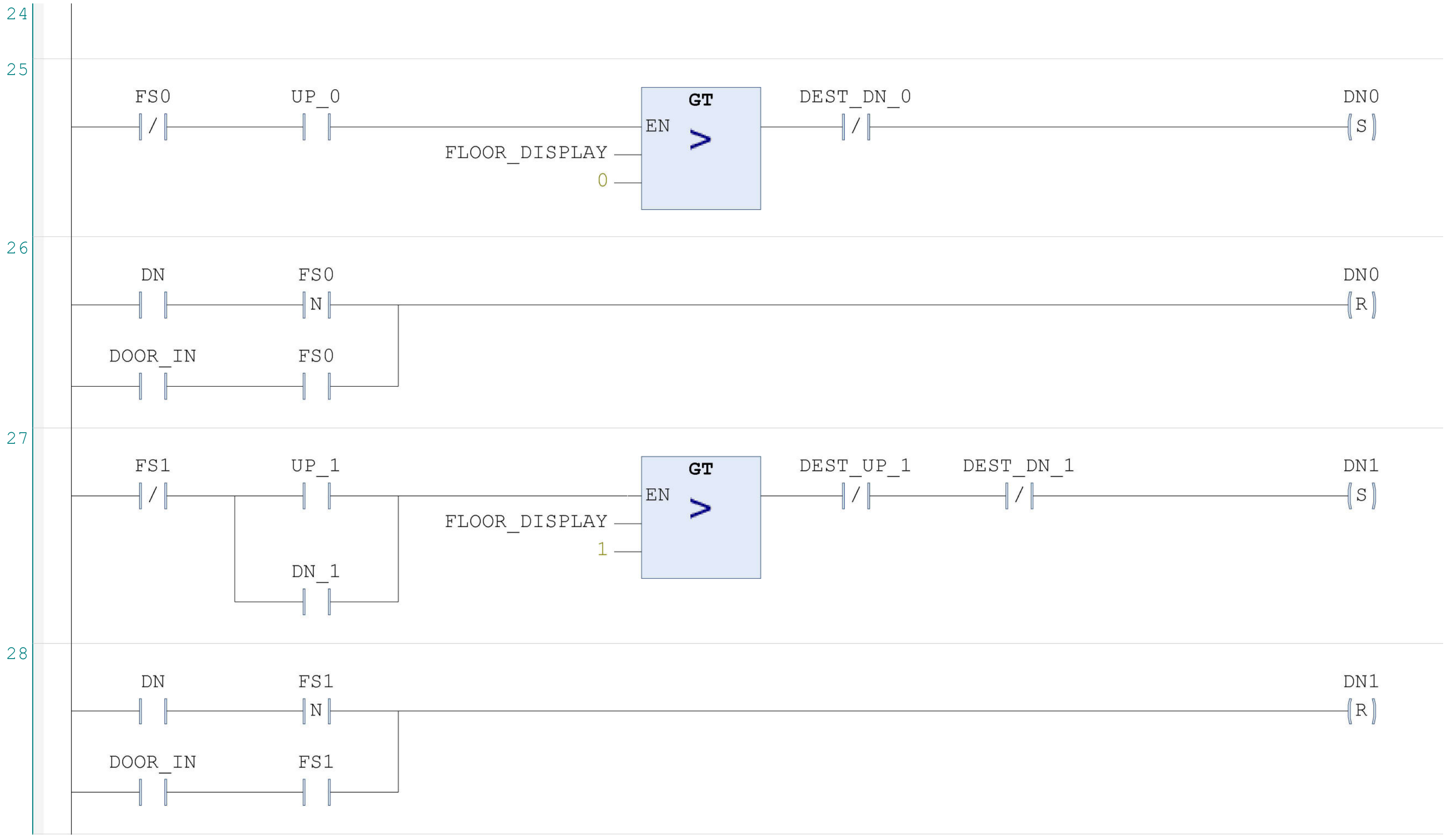
```
        DEST_DN_3 : BOOL ;
        DEST_DN_2 : BOOL ;
        DEST_DN_1 : BOOL ;
        DEST_DN_0 : BOOL ;
        UP_0 : BOOL ;
        UP_1 : BOOL ;
        UP_2 : BOOL ;
        UP_3 : BOOL ;
        DN_4 : BOOL ;
        DN_3 : BOOL ;
        DN_2 : BOOL ;
        DN_1 : BOOL ;
        DOOR_IN_1 : BOOL ;
        DOOR_IN_2 : BOOL ;
        UP_PRESSED_3 : BOOL ;
        UP_PRESSED_2 : BOOL ;
        UP_PRESSED_1 : BOOL ;
        UP_PRESSED_0 : BOOL ;
        DN_PRESSED_4 : BOOL ;
        DN_PRESSED_3 : BOOL ;
        DN_PRESSED_2 : BOOL ;
        DN_PRESSED_1 : BOOL ;
        DOOR : BOOL ;
        DOOR_PB : BOOL ;
        DOOR_PB_IN : BOOL ;
        DN_NEW : BOOL ;
        UP_NEW : BOOL ;
        DOOR_OUT : BOOL ;
        STATIONARY : BOOL ;
        DOOR_IN_3 : BOOL ;
    END_VAR

```

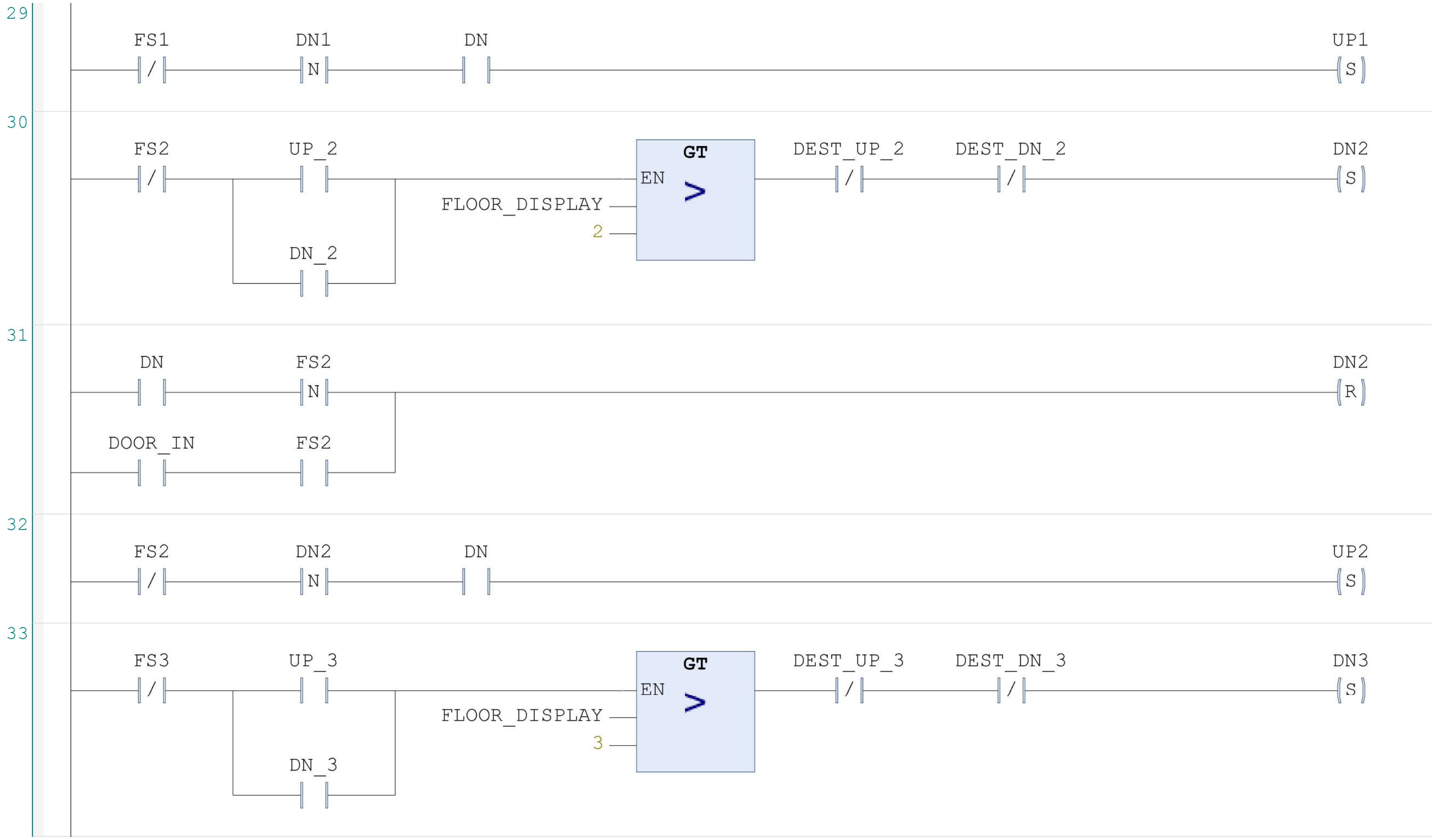

1
ELEVATOR_MODEL_0
TRUE
ELEVATOR_MODEL
EN
ENO
UP
UP
FLOOR_0
FS0
DN
DN
FLOOR_1
FS1
DOOR_IN OR DOOR_PB_IN OR DOOR_IN_3
DOOR_IN
FLOOR_2
FS2
FLOOR_3
FS3
FLOOR_4
FS4
X1_POSITION
X1_POSITION
X2_POSITION
X2_POSITION
Y_POSITION
Y_POSITION
FLOOR_DISPLAY
FLOOR_DISPLAY
DOOR_OUT
DOOR_OUT
STATIONARY
STATIONARY
2
3
FS4
/
DEST_4
LT
EN
<
FLOOR_DISPLAY
4
DEST_UP_4
S
4
DOOR_IN
FS4
DEST_UP_4
R

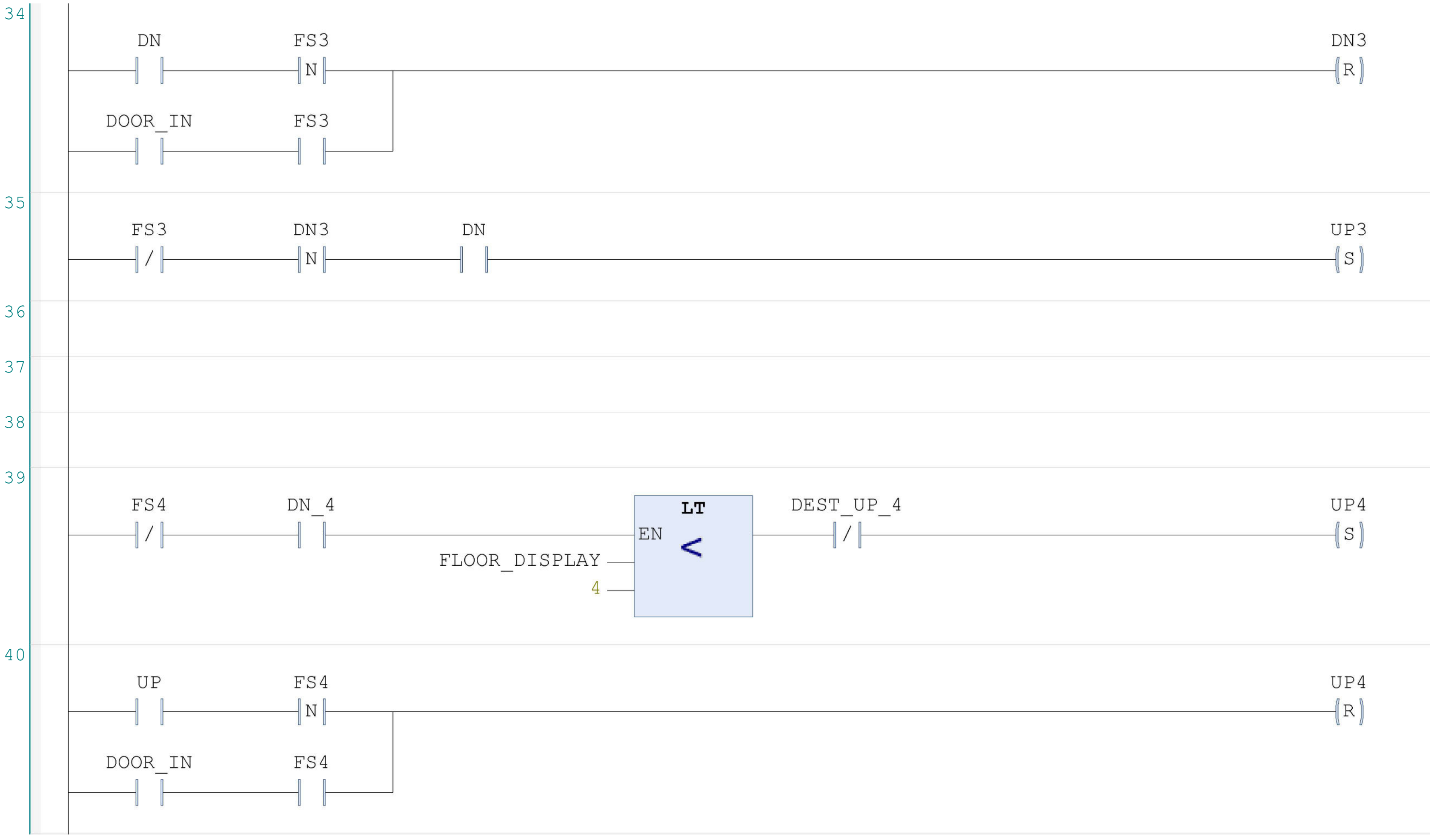

5
FS3
DEST_3
LT
EN
<
FLOOR_DISPLAY
3
DEST_UP_3
S
6
DOOR_IN
FS3
DEST_UP_3
R
7
FS2
DEST_2
LT
EN
<
FLOOR_DISPLAY
2
DEST_UP_2
S
8
DOOR_IN
FS2
DEST_UP_2
R
9
FS1
DEST_1
LT
EN
<
FLOOR_DISPLAY
1
DEST_UP_1
S

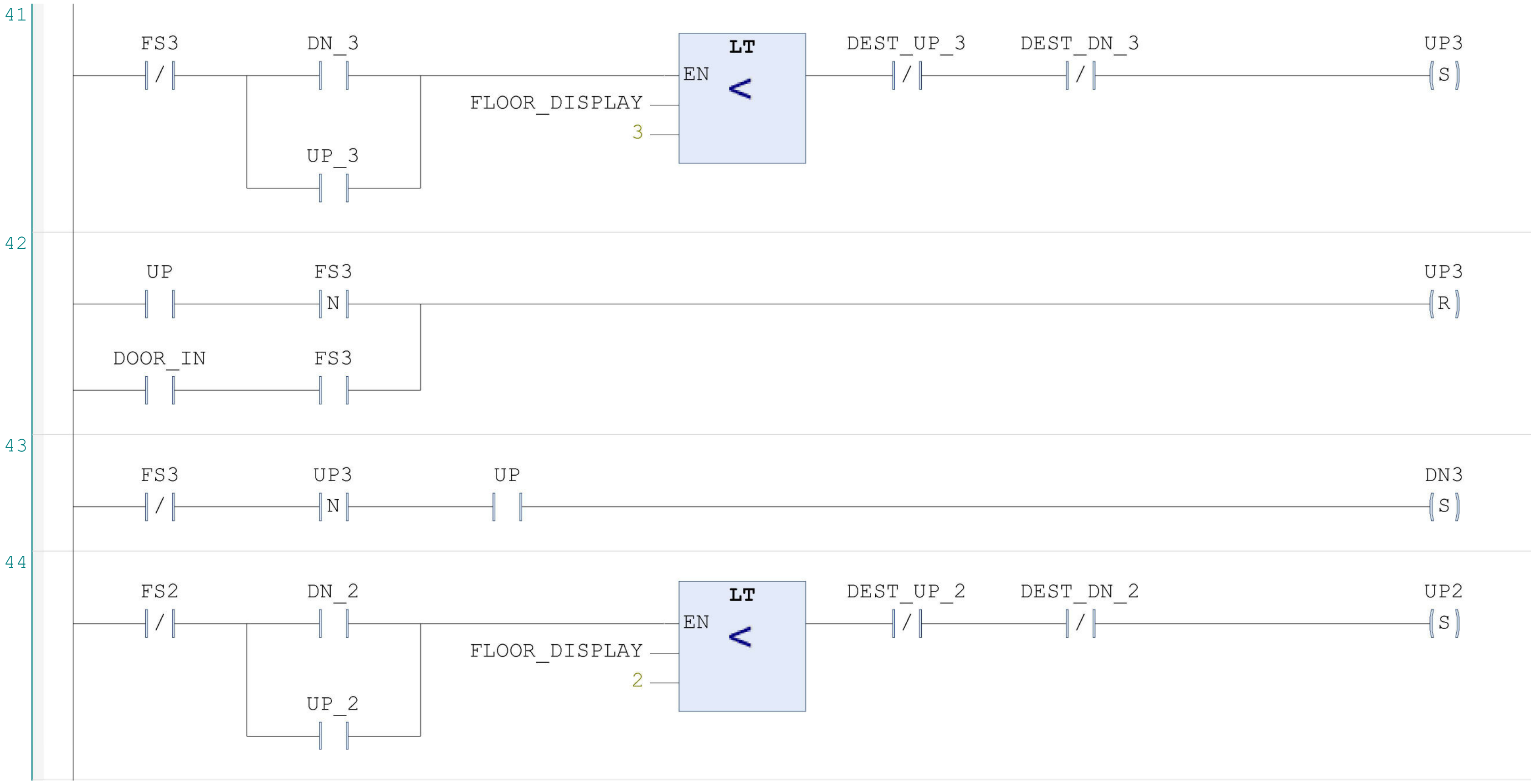

10
DOOR_IN
FS1
DEST_UP_1
R
11
12
13
14
FS3
/
DEST_3
GT
EN
>
FLOOR_DISPLAY
3
DEST_DN_3
S
15
DOOR_IN
FS3
DEST_DN_3
R
16
FS2
/
DEST_2
GT
EN
>
FLOOR_DISPLAY
2
DEST_DN_2
S

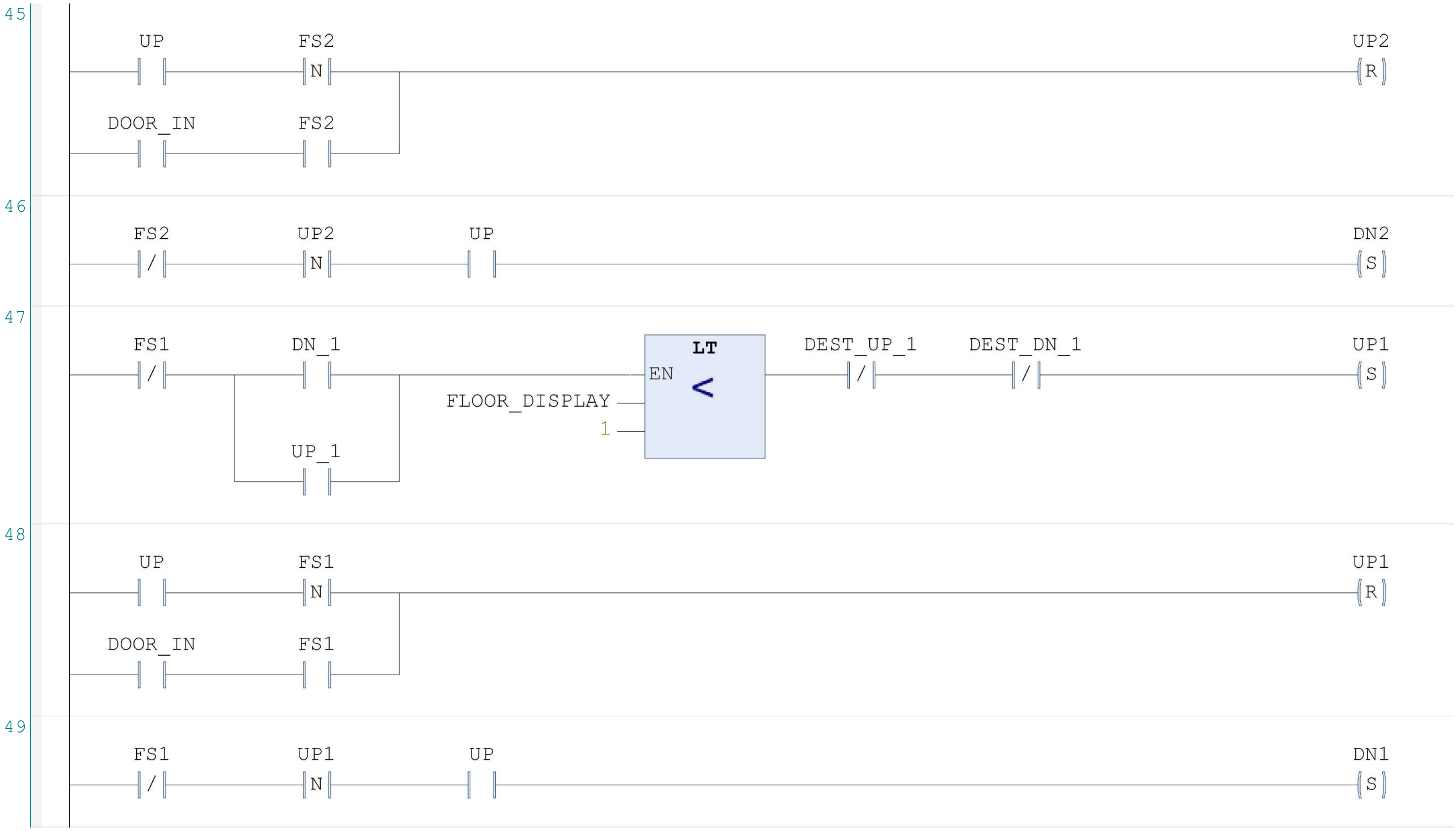

17
DOOR_IN
FS2
DEST_DN_2
R
18
FS1
/
DEST_1
GT
EN
>
FLOOR_DISPLAY
1
DEST_DN_1
S
19
DOOR_IN
FS1
DEST_DN_1
R
20
FS0
/
DEST_0
GT
EN
>
FLOOR_DISPLAY
0
DEST_DN_0
S
21
DOOR_IN
FS0
DEST_DN_0
R
22
23

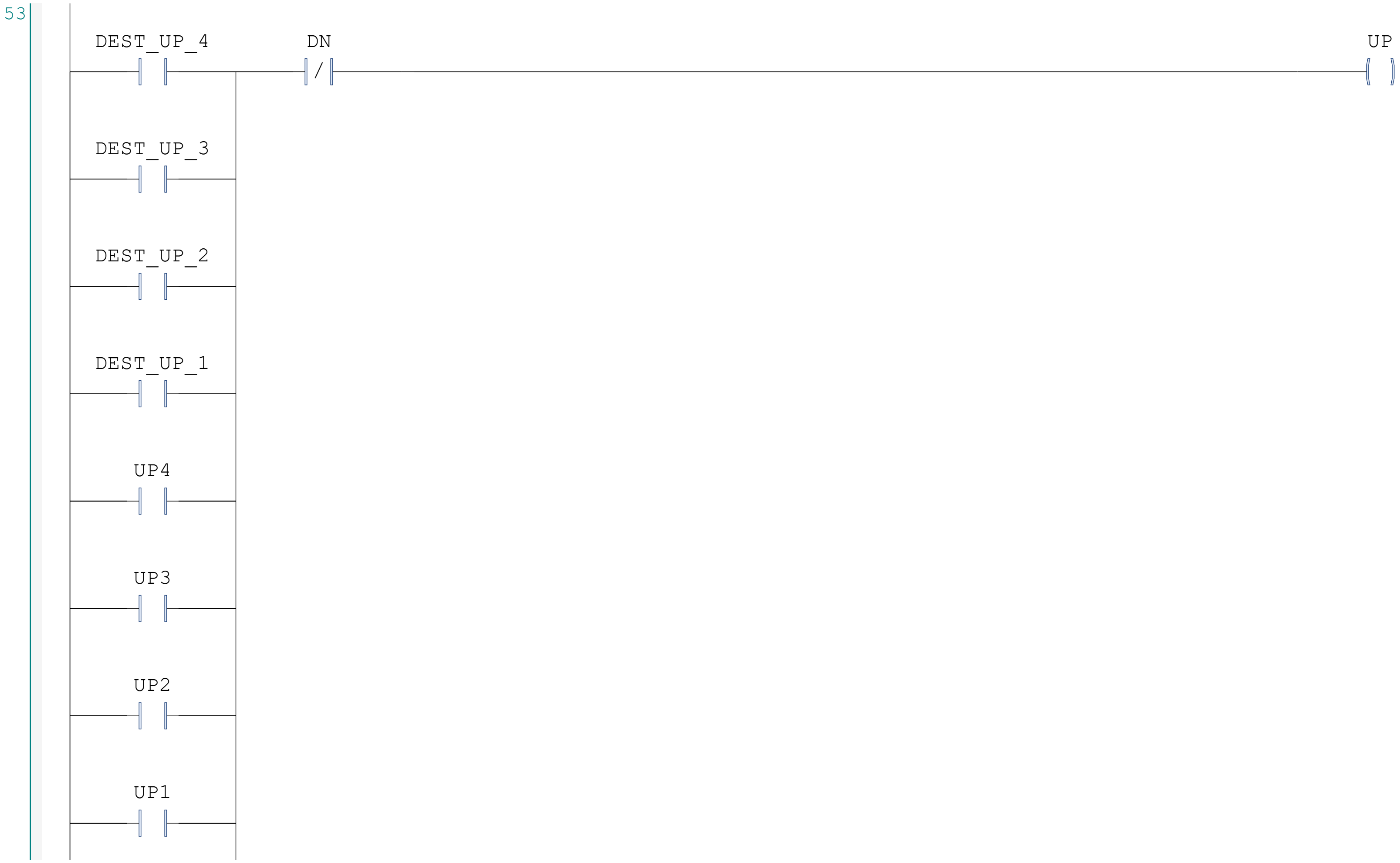

24
25
FS0
/
UP_0
GT
EN
>
FLOOR_DISPLAY
0
DEST_DN_0
/
DN0
S
26
DN
FS0
N
DN0
R
DOOR_IN
FS0
27
FS1
/
UP_1
DN_1
GT
EN
>
FLOOR_DISPLAY
1
DEST_UP_1
/
DEST_DN_1
/
DN1
S
28
DN
FS1
N
DN1
R
DOOR_IN
FS1

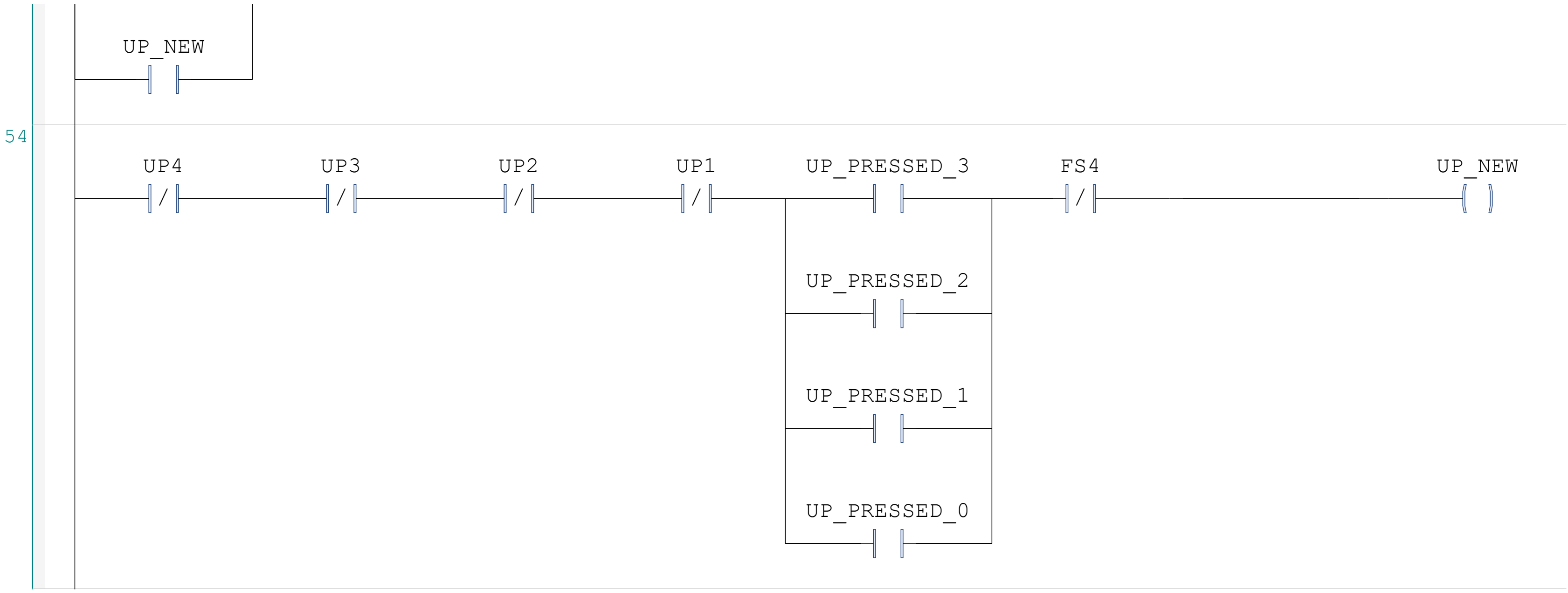

29
FS1
DN1
N
DN
UP1
S
30
FS2
UP_2
DN_2
FLOOR_DISPLAY
2
GT
EN
>
DEST_UP_2
DEST_DN_2
DN2
S
31
DN
FS2
N
DOOR_IN
FS2
DN2
R
32
FS2
DN2
N
DN
UP2
S
33
FS3
UP_3
DN_3
FLOOR_DISPLAY
3
GT
EN
>
DEST_UP_3
DEST_DN_3
DN3
S

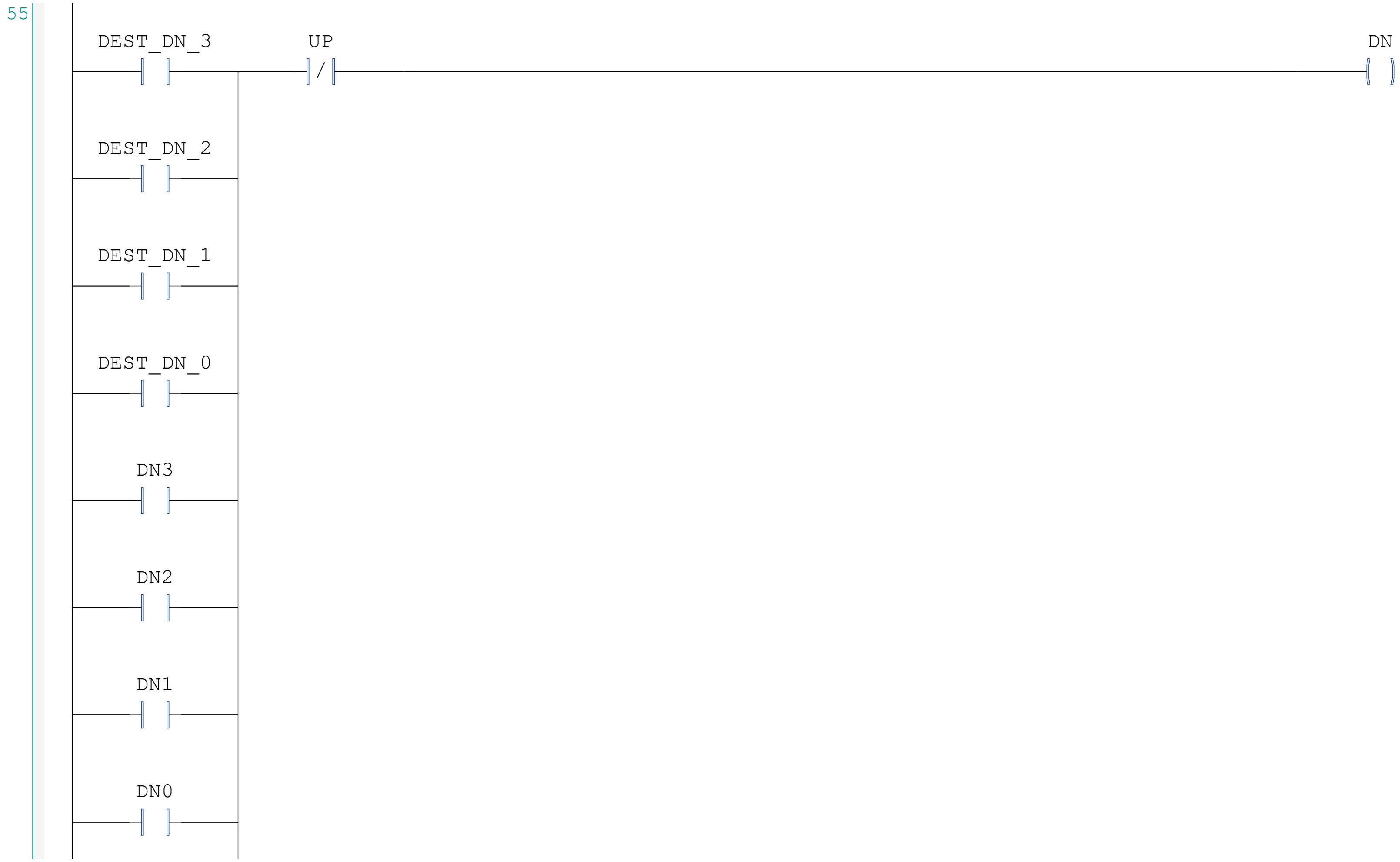

34
DN
FS3
N
DN3
R
DOOR_IN
FS3
35
FS3
/
DN3
N
DN
UP3
S
36
37
38
39
FS4
/
DN_4
LT
EN
<
FLOOR_DISPLAY
4
DEST_UP_4
/
UP4
S
40
UP
FS4
N
UP4
R
DOOR_IN
FS4

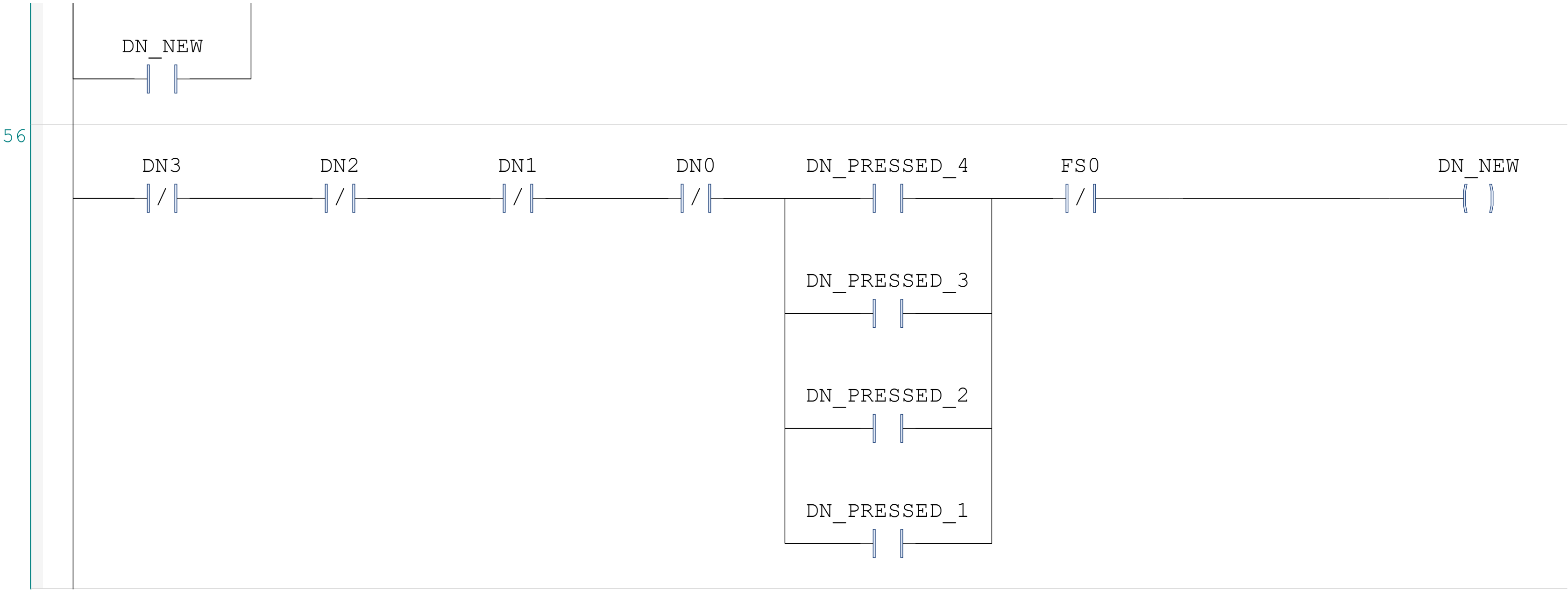

41
FS3
DN_3
LT
EN
<
DEST_UP_3
DEST_DN_3
UP3
/
S
FLOOR_DISPLAY
3
UP_3
42
UP
FS3
N
UP3
R
DOOR_IN
FS3
43
FS3
UP3
UP
DN3
/
N
S
44
FS2
DN_2
LT
EN
<
DEST_UP_2
DEST_DN_2
UP2
/
S
FLOOR_DISPLAY
2
UP_2


45
UP
FS2
N
UP2
R
DOOR_IN
FS2
46
FS2
/
UP2
N
UP
DN2
S
47
FS1
/
DN_1
LT
EN
<
DEST_UP_1
/
DEST_DN_1
/
UP1
S
FLOOR_DISPLAY
1
UP_1
48
UP
FS1
N
UP1
R
DOOR_IN
FS1
49
FS1
/
UP1
N
UP
DN1
S

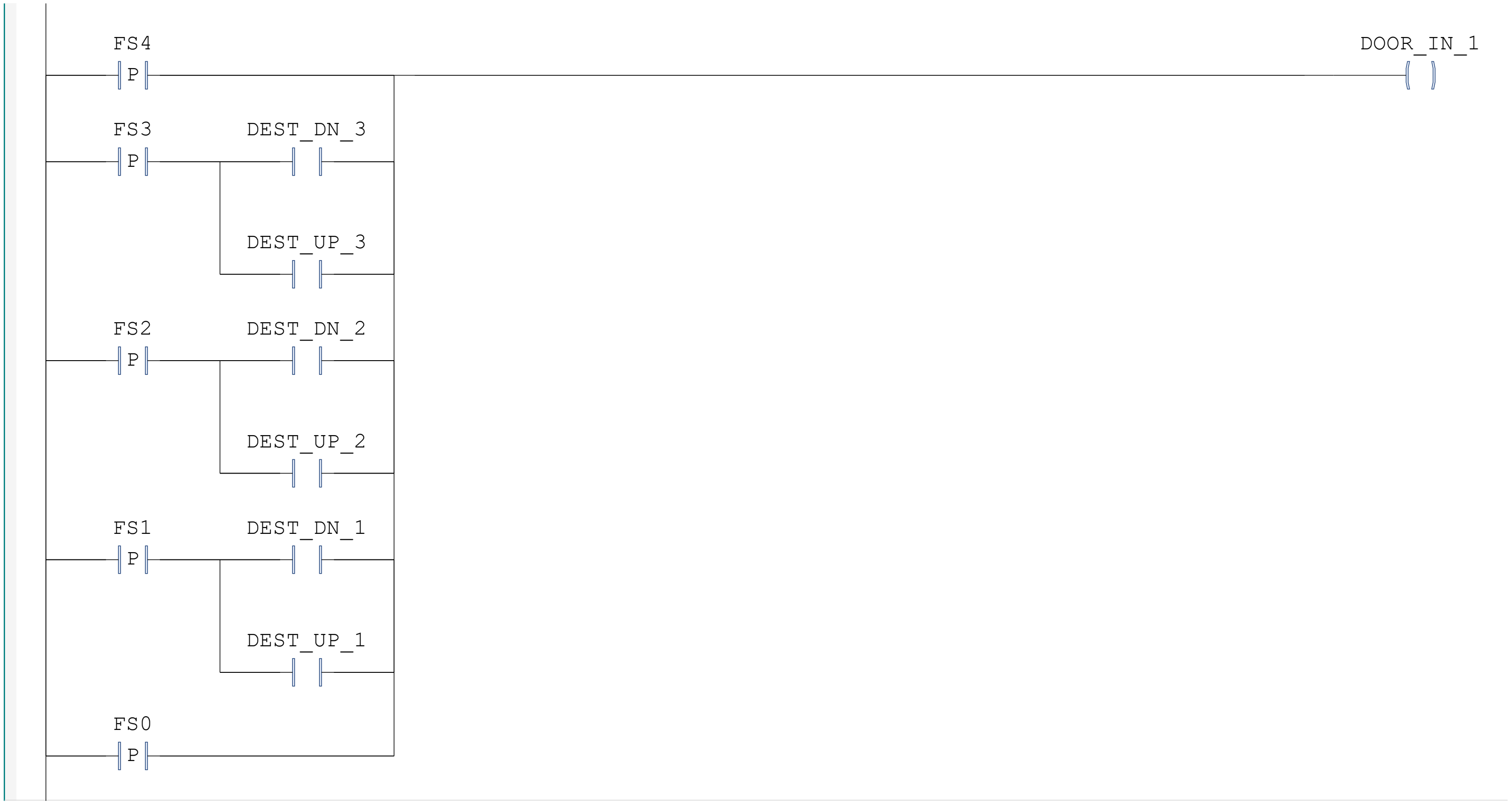

53
DEST_UP_4
DN
UP
DEST_UP_3
DEST_UP_2
DEST_UP_1
UP4
UP3
UP2
UP1

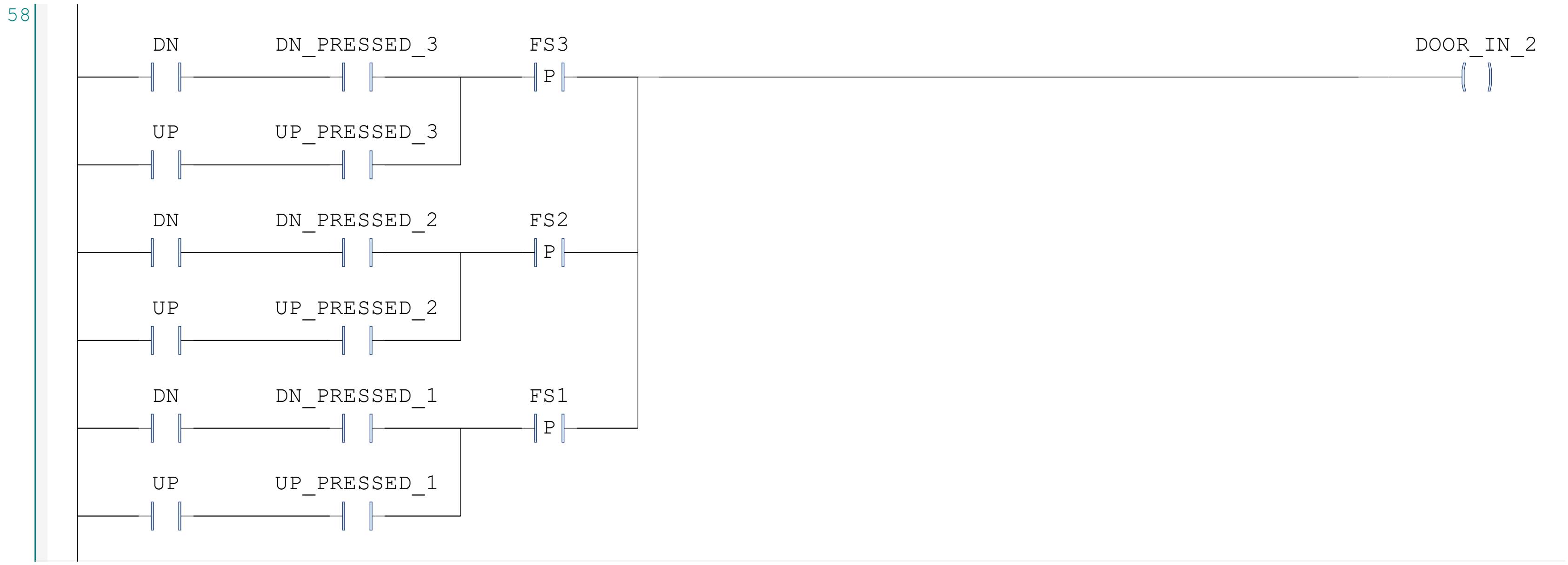

UP_NEW
54
UP4
UP3
UP2
UP1
UP_PRESSED_3
FS4
UP_NEW
UP_PRESSED_2
UP_PRESSED_1
UP_PRESSED_0

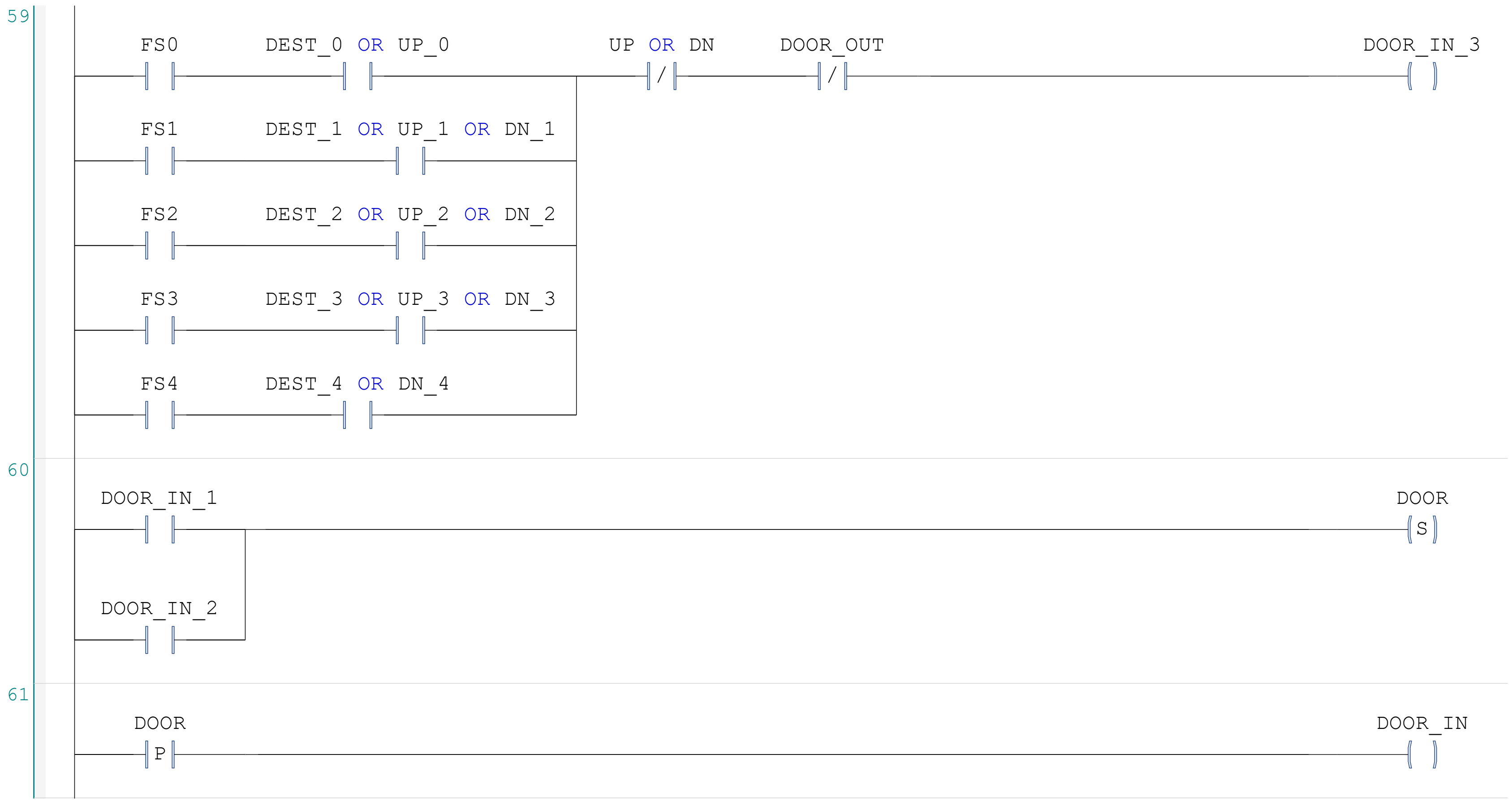

55
DEST_DN_3
UP
DN
DEST_DN_2
DEST_DN_1
DEST_DN_0
DN3
DN2
DN1
DN0

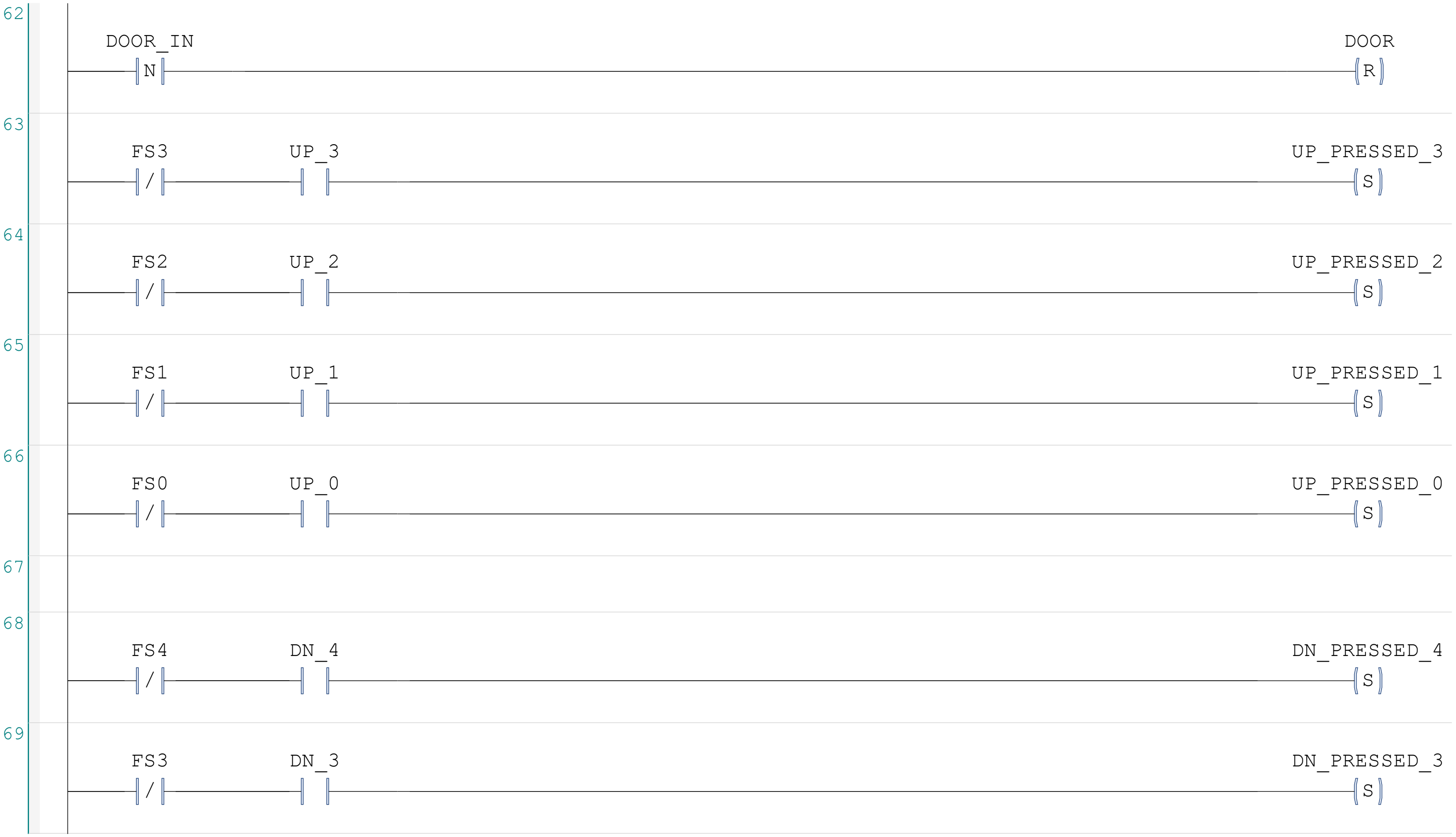

DN_NEW
56
DN3
DN2
DN1
DN0
DN_PRESSED_4
FS0
DN_NEW
DN_PRESSED_3
DN_PRESSED_2
DN_PRESSED_1

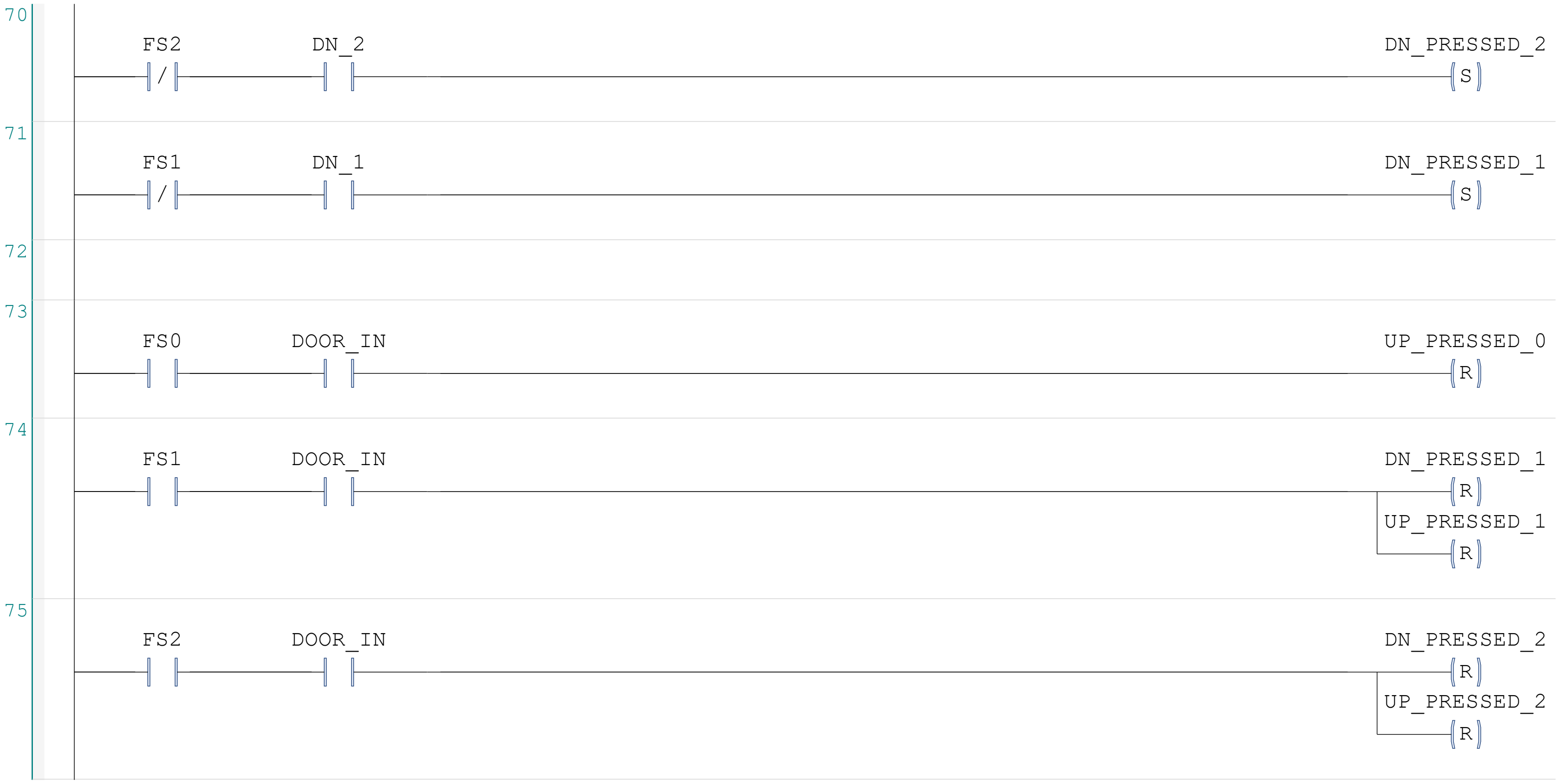
57

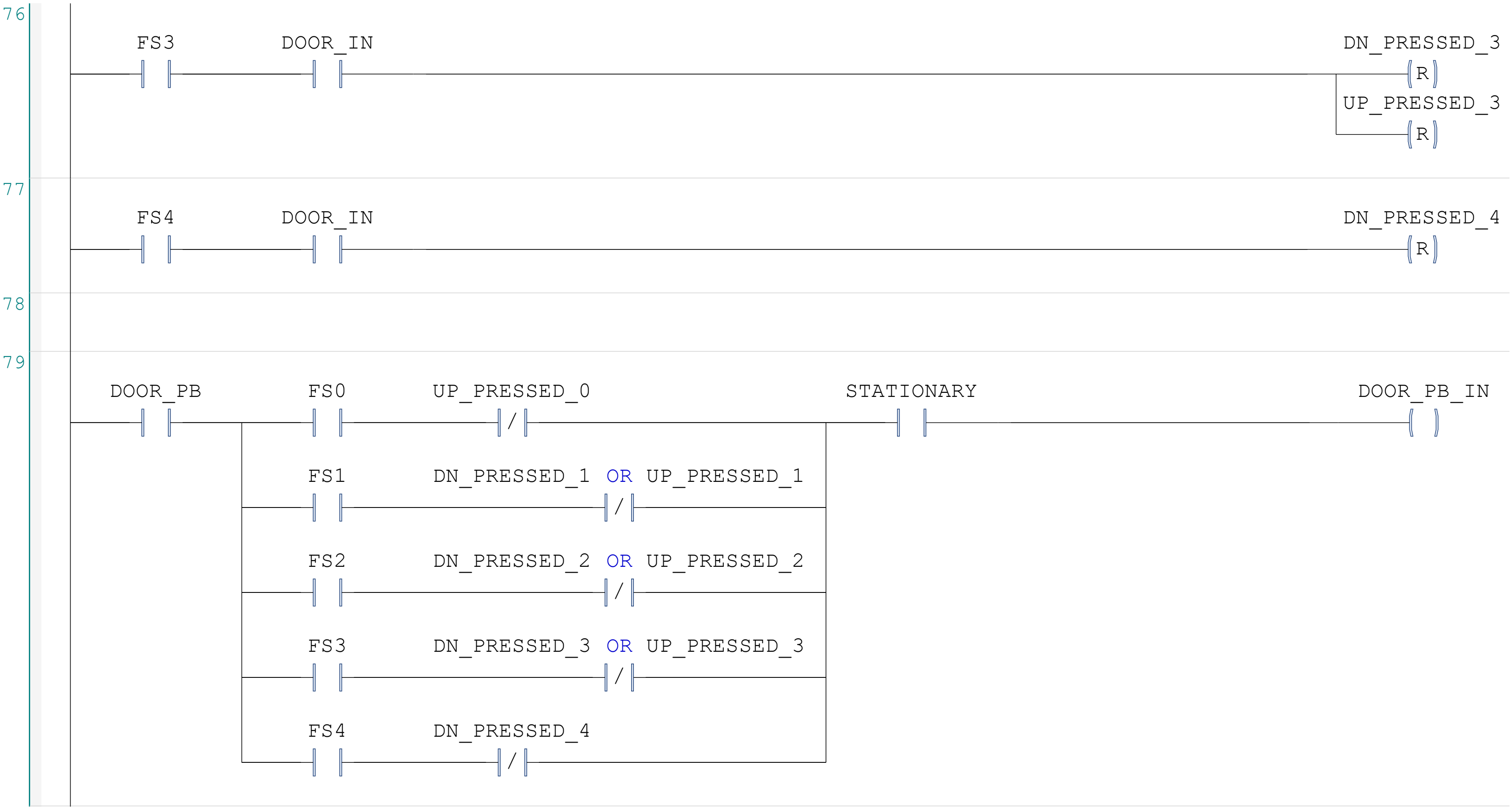

FS4
P
DOOR_IN_1
FS3
P
DEST_DN_3
DEST_UP_3
FS2
P
DEST_DN_2
DEST_UP_2
FS1
P
DEST_DN_1
DEST_UP_1
FS0
P


58
DN
DN_PRESSED_3
FS3
P
DOOR_IN_2
UP
UP_PRESSED_3
DN
DN_PRESSED_2
FS2
P
UP
UP_PRESSED_2
DN
DN_PRESSED_1
FS1
P
UP
UP_PRESSED_1


59
FS0
DEST_0 OR UP_0
UP OR DN
DOOR_OUT
DOOR_IN_3
FS1
DEST_1 OR UP_1 OR DN_1
FS2
DEST_2 OR UP_2 OR DN_2
FS3
DEST_3 OR UP_3 OR DN_3
FS4
DEST_4 OR DN_4
60
DOOR_IN_1
DOOR
S
DOOR_IN_2
61
DOOR
P
DOOR_IN

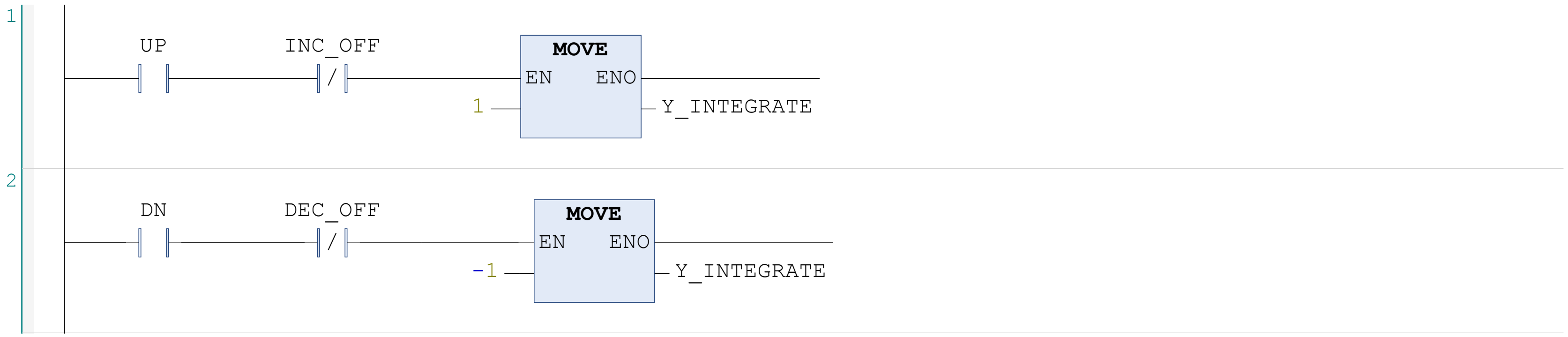

62
DOOR_IN
N
DOOR
R
63
FS3
/
UP_3
UP_PRESSED_3
S
64
FS2
/
UP_2
UP_PRESSED_2
S
65
FS1
/
UP_1
UP_PRESSED_1
S
66
FS0
/
UP_0
UP_PRESSED_0
S
67
68
FS4
/
DN_4
DN_PRESSED_4
S
69
FS3
/
DN_3
DN_PRESSED_3
S

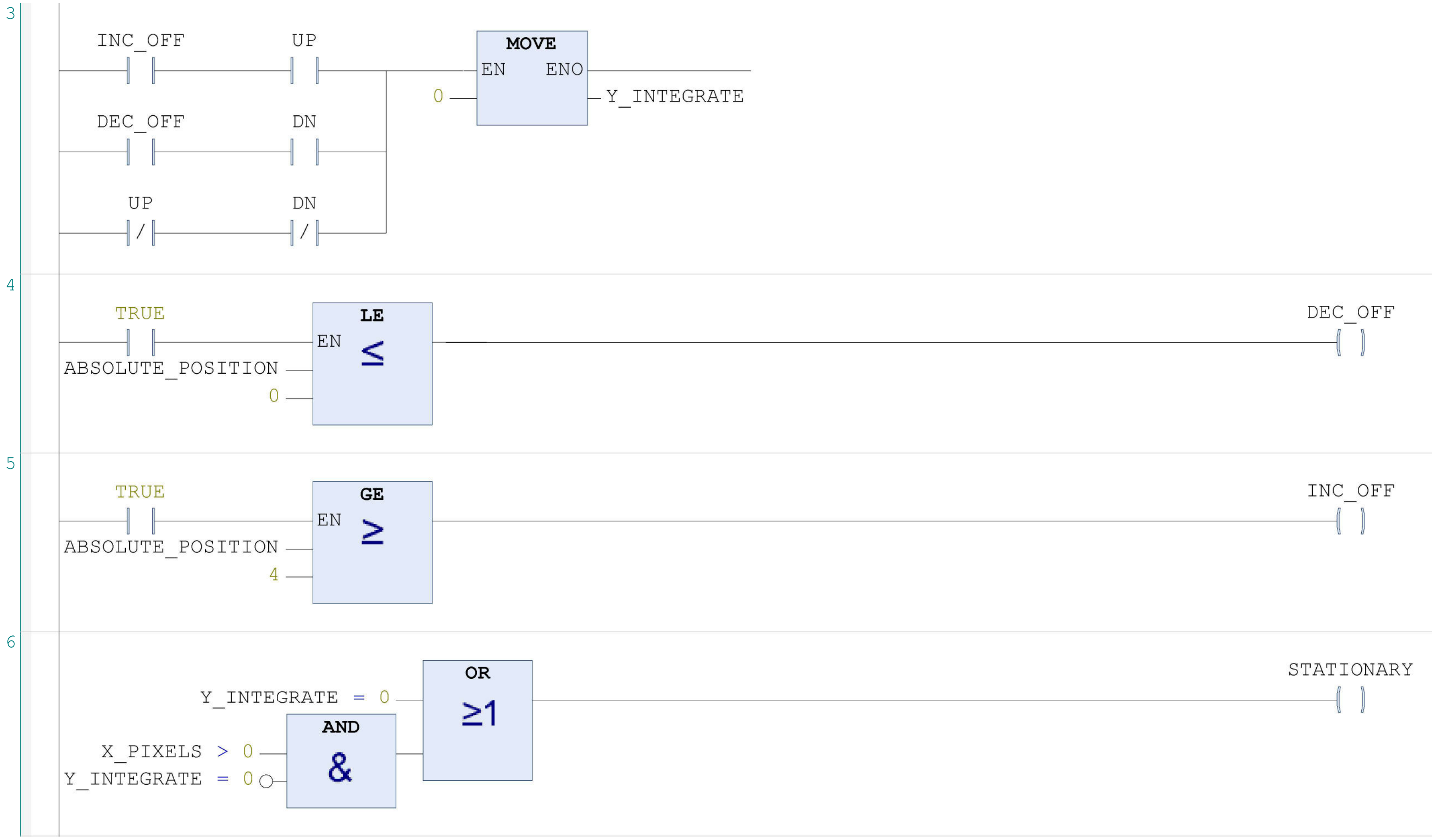

70
FS2
DN_2
DN_PRESSED_2
/
S
71
FS1
DN_1
DN_PRESSED_1
/
S
72
73
FS0
DOOR_IN
UP_PRESSED_0
R
74
FS1
DOOR_IN
DN_PRESSED_1
R
UP_PRESSED_1
R
75
FS2
DOOR_IN
DN_PRESSED_2
R
UP_PRESSED_2
R

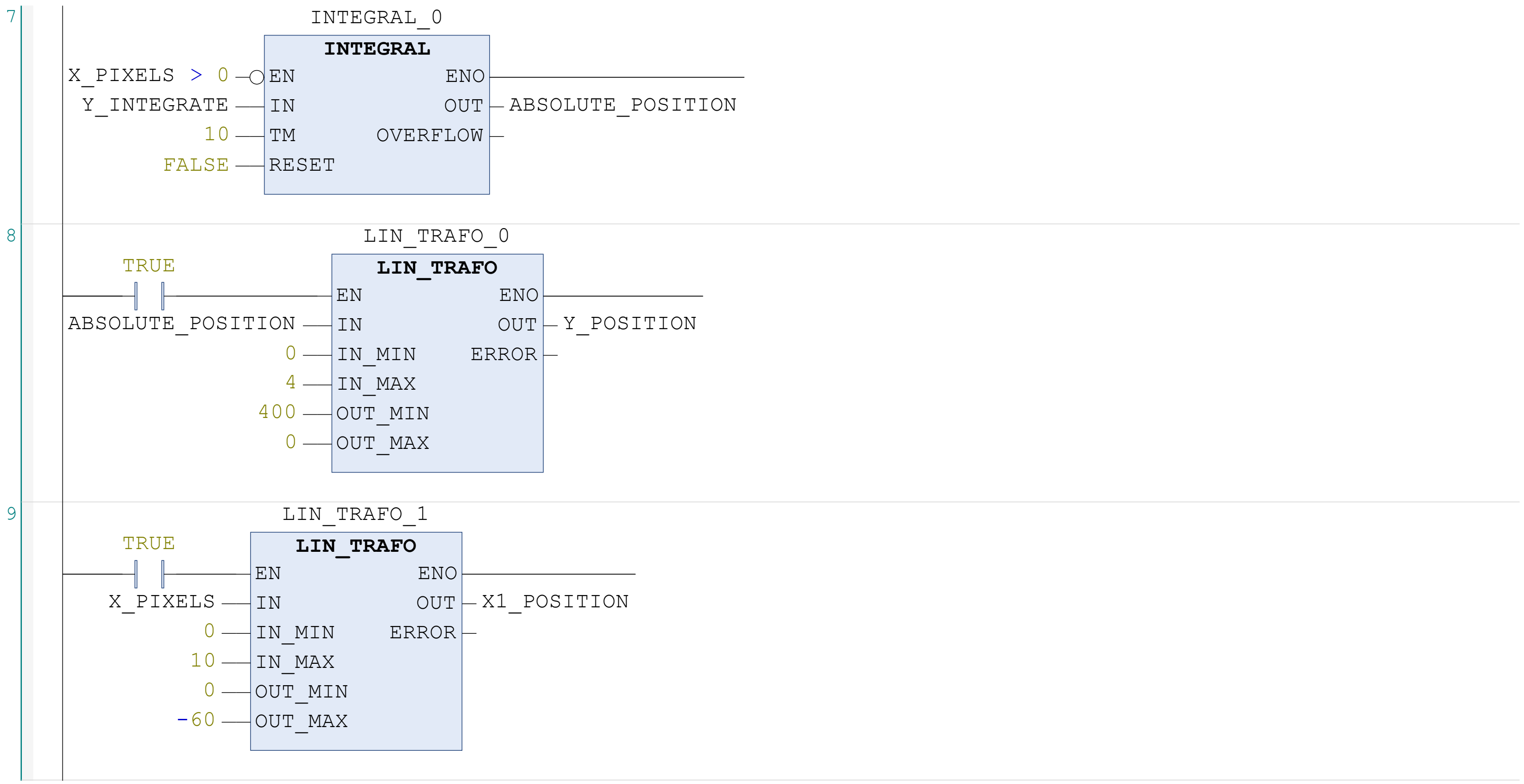

76
FS3
DOOR_IN
DN_PRESSED_3
R
UP_PRESSED_3
R
77
FS4
DOOR_IN
DN_PRESSED_4
R
78
79
DOOR_PB
FS0
UP_PRESSED_0
STATIONARY
DOOR_PB_IN
FS1
DN_PRESSED_1 OR UP_PRESSED_1
FS2
DN_PRESSED_2 OR UP_PRESSED_2
FS3
DN_PRESSED_3 OR UP_PRESSED_3
FS4
DN_PRESSED_4

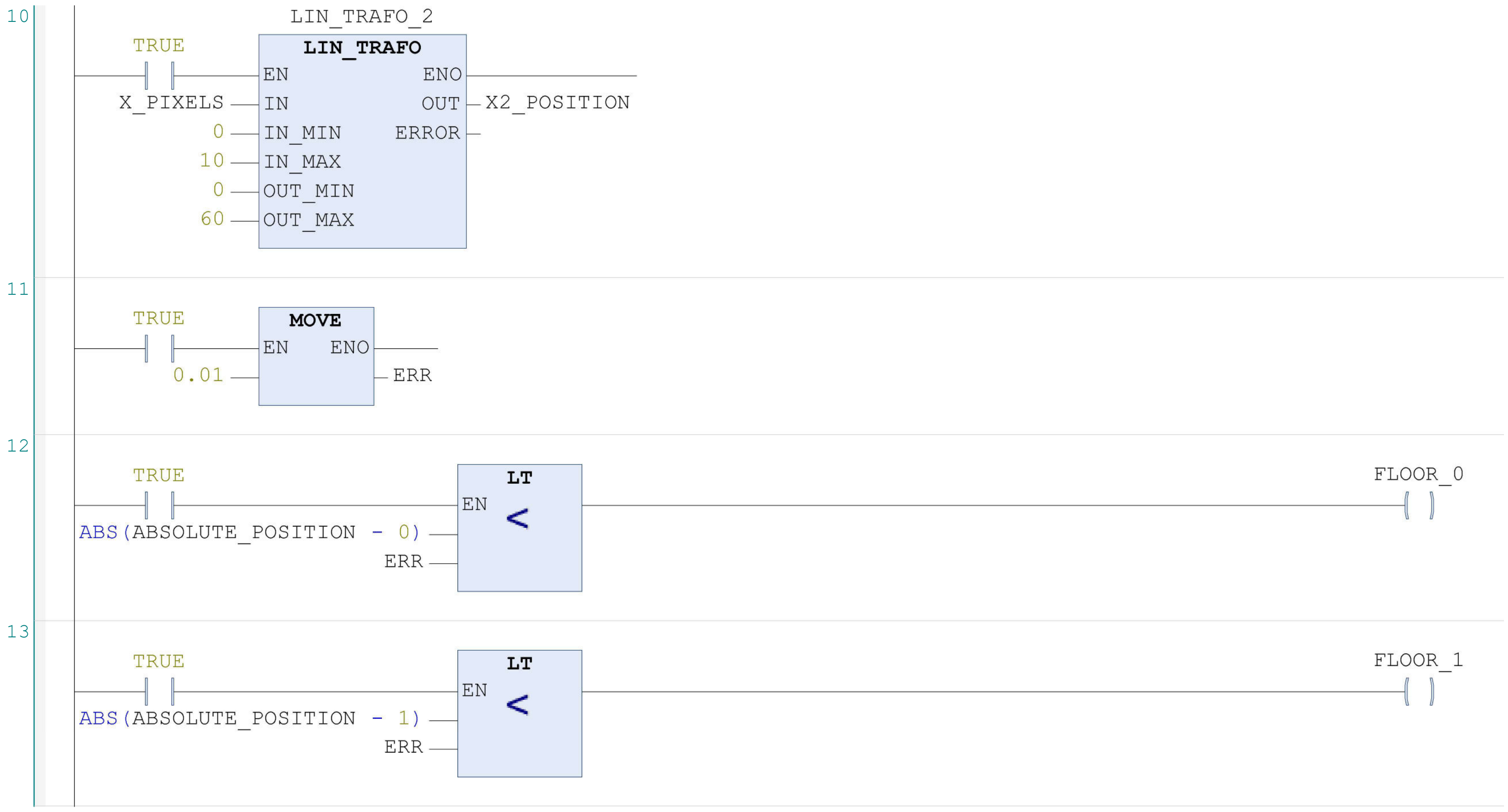
```
FUNCTION_BLOCK ELEVATOR_MODEL
VAR_INPUT
    UP : BOOL ;
    DN : BOOL ;

    DOOR_IN : BOOL ;
END_VAR
VAR_OUTPUT
    FLOOR_0 : BOOL ;
    FLOOR_1 : BOOL ;
    FLOOR_2 : BOOL ;
    FLOOR_3 : BOOL ;
    FLOOR_4 : BOOL ;
    X1_POSITION : REAL ;
    X2_POSITION : REAL ;
    Y_POSITION : REAL ;
    FLOOR_DISPLAY : UINT ;

    DOOR_OUT : BOOL ;
    STATIONARY : BOOL ;

END_VAR
VAR
    INC_OFF : BOOL ;
    DEC_OFF : BOOL ;
    Y_INTEGRATE : INT ;
    INTEGRAL_0 : INTEGRAL ;
    LIN_TRAFO_0 : LIN_TRAFO ;
    LIN_TRAFO_1 : LIN_TRAFO ;
    LIN_TRAFO_2 : LIN_TRAFO ;
    TOF_0 : TOF ;
    X_PIXELS : REAL ;
    X_PIXELS_1 : REAL ;
    ABSOLUTE_POSITION : REAL ;
    TOF_1 : TOF ;
```

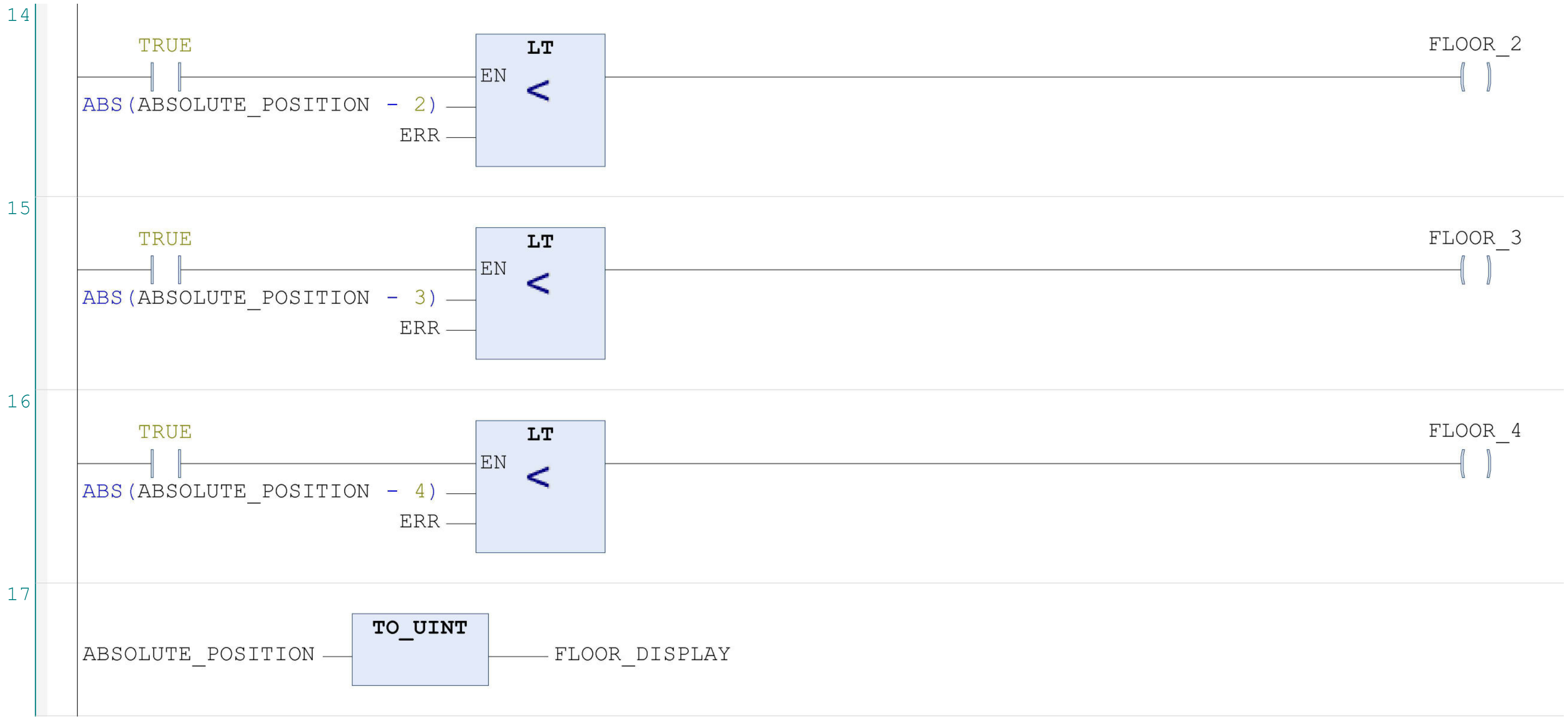
```
        ERR :  REAL ;
        STOP_INTEGRATION :  BOOL ;
        DOOR_PB_IN :  BOOL ;
        TOF_2 :  TOF ;
        DOOR_PB_PRESSED :  BOOL ;
        X_TMP :  INT ;
        X_TIME :  INT ;
        INTEGRAL_1 :  INTEGRAL ;
        DOOR_REAL :  REAL ;
        T1 :  BOOL ;
        T2 :  BOOL ;
        DOOR :  BOOL ;
    END_VAR

```

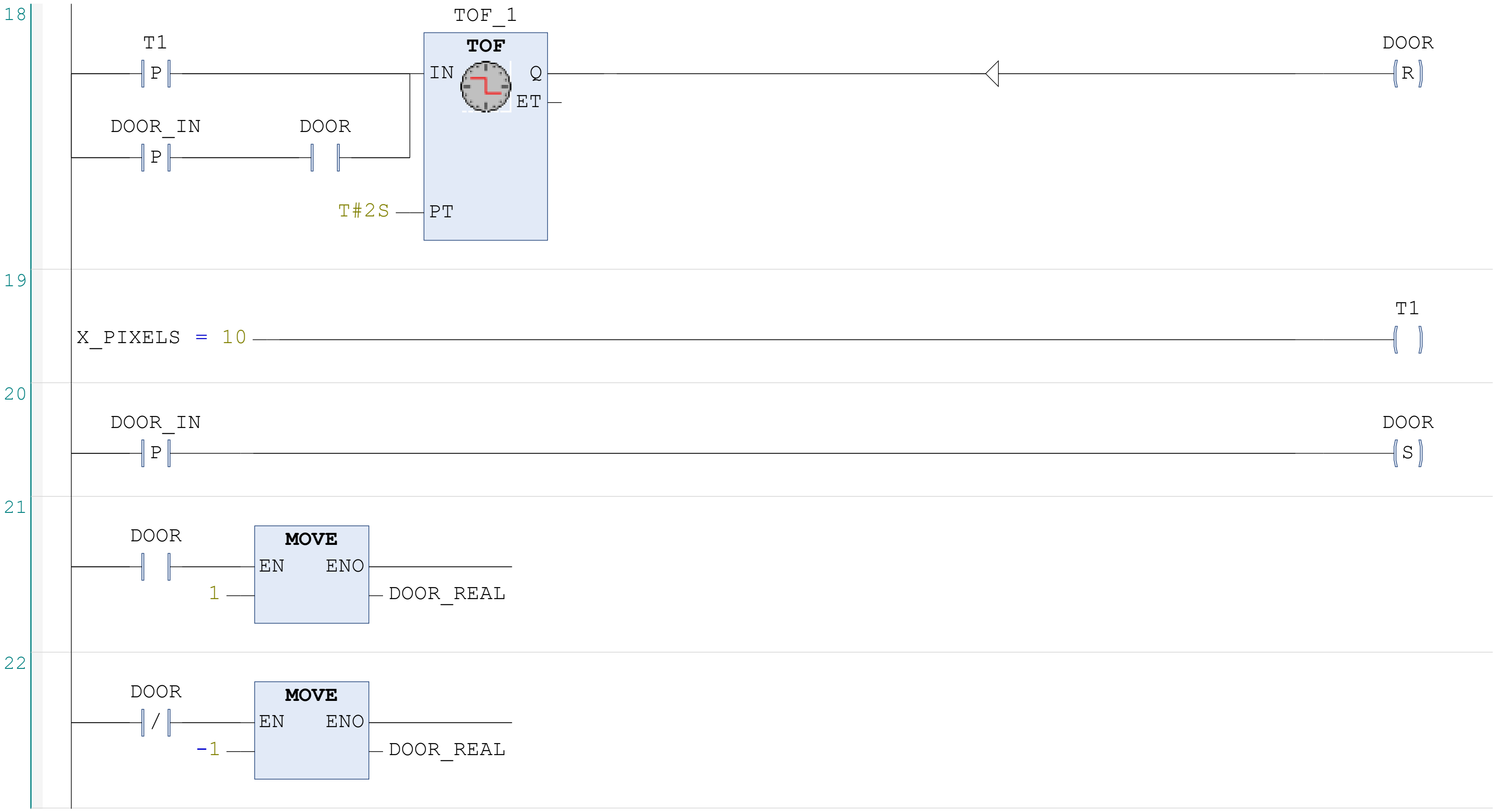


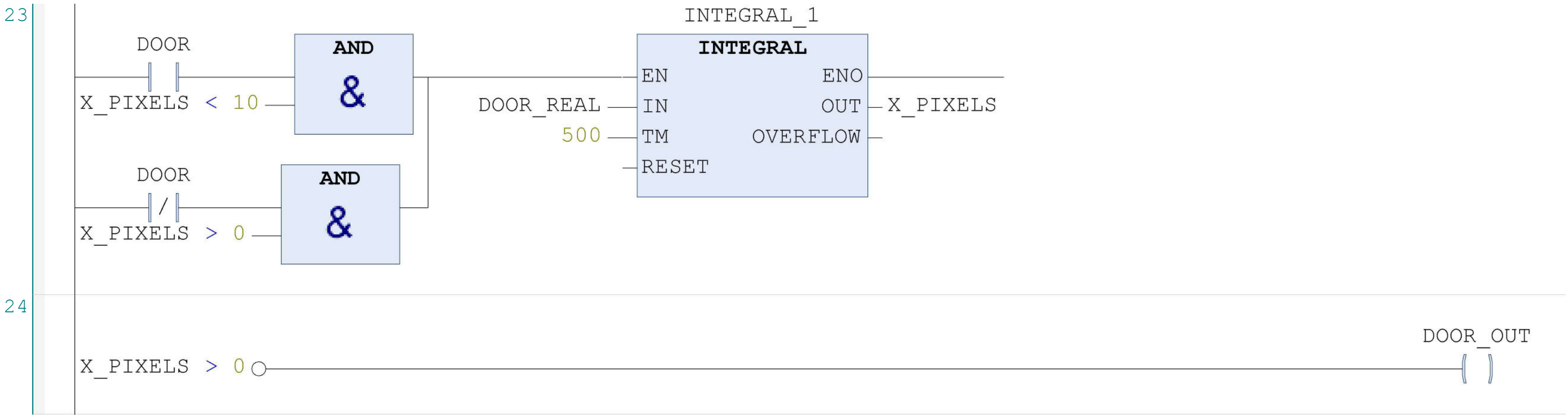

3
INC_OFF
UP
MOVE
EN
ENO
0
Y_INTEGRATE
DEC_OFF
DN
UP
DN
4
TRUE
LE
EN
≤
ABSOLUTE_POSITION
0
DEC_OFF
5
TRUE
GE
EN
≥
ABSOLUTE_POSITION
4
INC_OFF
6
OR
≥1
STATIONARY
Y_INTEGRATE = 0
AND
&
X_PIXELS > 0
Y_INTEGRATE = 0

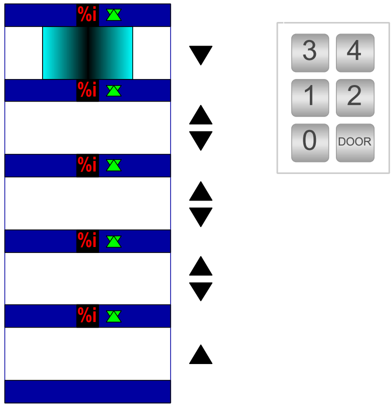

7
INTEGRAL_0
INTEGRAL
X_PIXELS > 0
EN
ENO
Y_INTEGRATE
IN
OUT
ABSOLUTE_POSITION
10
TM
OVERFLOW
FALSE
RESET
8
LIN_TRAFO_0
LIN_TRAFO
TRUE
EN
ENO
ABSOLUTE_POSITION
IN
OUT
Y_POSITION
0
IN_MIN
ERROR
4
IN_MAX
400
OUT_MIN
0
OUT_MAX
9
LIN_TRAFO_1
LIN_TRAFO
TRUE
EN
ENO
X_PIXELS
IN
OUT
X1_POSITION
0
IN_MIN
ERROR
10
IN_MAX
0
OUT_MIN
-60
OUT_MAX


10
LIN_TRAFO_2
TRUE
LIN_TRAFO
EN
ENO
X_PIXELS
IN
OUT
X2_POSITION
0
IN_MIN
ERROR
10
IN_MAX
0
OUT_MIN
60
OUT_MAX
11
TRUE
MOVE
EN
ENO
0.01
ERR
12
TRUE
LT
EN
<
ABS(ABSOLUTE_POSITION - 0)
ERR
FLOOR_0
13
TRUE
LT
EN
<
ABS(ABSOLUTE_POSITION - 1)
ERR
FLOOR_1


14
TRUE
LT
EN
<
ABS(ABSOLUTE_POSITION - 2)
ERR
FLOOR_2
15
TRUE
LT
EN
<
ABS(ABSOLUTE_POSITION - 3)
ERR
FLOOR_3
16
TRUE
LT
EN
<
ABS(ABSOLUTE_POSITION - 4)
ERR
FLOOR_4
17
ABSOLUTE_POSITION
TO_UINT
FLOOR_DISPLAY


18
TOF_1
T1
P
TOF
IN
Q
ET
DOOR
R
DOOR_IN
P
DOOR
T#2S
PT
19
X_PIXELS = 10
T1
20
DOOR_IN
P
DOOR
S
21
DOOR
MOVE
EN
ENO
1
DOOR_REAL
22
DOOR
/
MOVE
EN
ENO
-1
DOOR_REAL


23
DOOR
X_PIXELS < 10
AND
&
DOOR
/
X_PIXELS > 0
AND
&
INTEGRAL_1
INTEGRAL
EN
ENO
DOOR_REAL
IN
OUT
X_PIXELS
500
TM
OVERFLOW
RESET
24
DOOR_OUT
X_PIXELS > 0



```
General
  Visualization size algorithm version: Respecting scrollbar location
Background
  Use background color: False
  Background color: 16777215
Interface
  VAR_IN_OUT

  END_VAR
Visualization Element List
  Rectangle Id: 227
    Element name: GenElemInst_192
    Tab Order: default
    Static optimized: True
    Type of element: Rectangle
    Position
      X: 10
      Y: 20
      Width: 220
```