



Write a self-contained 61131-3 structured text program for controlling an elevator. Every floor of the 5-floor building contains bottom and top limit switches. There is a 7 seconds timer for opening the elevetor door, and a 10 seconds timer for opening the door again if none of the buttons inside the elevator cabin are pressed. The elevator operates based on the current direction and the direction imposed by up and call button on each floor.



Here's a structured text program for controlling an elevator in a 5-floor building using IEC 61131-3. The program uses bottom and top limit switches on each floor, a 7-second timer for opening the elevator door, and a 10-second timer for opening the door again if none of the buttons inside the elevator cabin are pressed.

iechstCopy code

```
PROGRAM ElevatorControl
VAR
    floor: INT := 1;
    direction: INT := 0;
    callUp: ARRAY [1..5] OF BOOL;
    callDown: ARRAY [1..5] OF BOOL;
    cabinButton: ARRAY [1..5] OF BOOL;
    limitSwitchTop: ARRAY [1..5] OF BOOL;
    limitSwitchBottom: ARRAY [1..5] OF BOOL;
    doorOpen: BOOL := FALSE;
    doorTimer: TON;
    doorReopenTimer: TON;
    motorUp: BOOL := FALSE;
    motorDown: BOOL := FALSE;
END_VAR
```

iechstCopy code

```
METHOD UpdateFloor
FOR i := 1 TO 5 DO
    IF limitSwitchTop[i] AND limitSwitchBottom[i] THEN
        floor := i;
        EXIT;
    END_IF;
END_FOR;
END_METHOD
```

iechstCopy code

```
METHOD CheckButtons
FOR i := 1 TO 5 DO
    IF (callUp[i] AND direction = 1) OR (callDown[i] AND direction = -1) THEN
        direction := i - floor;
        EXIT;
    END_IF;
END_FOR;
END_METHOD
```

iechstCopy code

```
METHOD Updatedirection
IF floor = 1 THEN
    direction := 1;
ELSIF floor = 5 THEN
    direction := -1;
END_IF;
END_METHOD
```

iechstCopy code

```
METHOD ControlMotor
    motorUp := direction > 0;
    motorDown := direction < 0;
END_METHOD
```

iechstCopy code

```
METHOD OpenDoor
    doorOpen := TRUE;
    doorTimer(IN:=NOT doorTimer.Q, PT:=T#7s);
END_METHOD
```

iechstCopy code

```
METHOD CloseDoor
    doorOpen := FALSE;
    doorReopenTimer(IN:=NOT doorReopenTimer.Q, PT:=T#10s);
END_METHOD
```

iechstCopy code

```
METHOD CheckDoorReopen
    IF doorReopenTimer.Q THEN
        OpenDoor();
    END_IF;
END_METHOD
```

iechstCopy code

```
METHOD ResetButtons
FOR i := 1 TO 5 DO
    IF i = floor THEN
        callUp[i] := FALSE;
        callDown[i] := FALSE;
        cabinButton[i] := FALSE;
    END_IF;
END_FOR;
END_METHOD
```

iechstCopy code

```
METHOD Main
    UpdateFloor();
    CheckButtons();
    IF doorTimer.Q THEN
        CloseDoor();
    END_IF;
    IF doorOpen THEN
        CheckDoorReopen();
    ELSE
        Updatedirection();
        ControlMotor();
        IF (direction = 0) OR (direction > 0 AND callUp[floor]) OR (direction < 0 AND callDown[floor]) THEN
            OpenDoor();
            ResetButtons();
        END_IF;
    END_IF;
END_METHOD
```

iechstCopy code

```
Main();
END_PROGRAM
```

This program defines variables, methods for updating the floor, checking button presses, controlling the motor, opening and closing the door, and checking if the door needs to be reopened. The main method orchestrates these methods to control the elevator. Note that this



continue



code serves as a starting point and may need to be adapted to specific hardware and control systems. The program assumes that limit switches are normally open and close when the elevator is at the corresponding floor, and that button inputs are normally open and close when pressed.

To integrate this program into your elevator control system, you'll need to connect the variables to the appropriate inputs and outputs of your PLC. This includes limit switches, call buttons, cabin buttons, and door and motor control signals.

Additionally, this program does not implement advanced elevator control algorithms, such as optimized scheduling for multiple requests or handling emergency situations like power outages or system malfunctions. It is a basic example of an elevator control program that can be built upon and adapted for more complex systems.

