University of Jordan
School of Engineering
Mechatronics Engineering Department

Automation and Process control Lab (0908538)

Final project: Design a 5-floors Elevator system.

Group Member	University ID
Rawan Hassan	0185947
Osama Obeid	0191909
Zuhour Alsaqqa	0195786
Nada Zeineddin	0195841

Date: 31/12/2023

Advisor:
Eng. Samer Salah
Eng. Haneen Alhaj Mustafa

Table of Content

Table of Figures

Table of tables

Project plan

#	Task	Responsibility	days						
			Sun 24/12	Mon 25/12	Tues 26/12	Wed 27/12	Thurs 28/2	Fri 29/12	Sat 30/12
1	I/O Assignment	1)Rawan							
2	System design	1)Zuhour 2)Nada							
3	Report and presentation	1)Osama							

Table 1: Project plan table

Abstract

In this project we will control an elevator system using Allen-Bradley PLC and simulate the process using "logixpro" application like we learn in the lab, also talking about some information about the elevator systems, and to be able to stimulate the process we are going to use the I/O Simulator screen from the app, finally mention the I/O Assignment we used during the design of the system and the designed system simulation.

Introduction

The automated systems become a big rule in the modern industry which can be defined as: an arrangement for automatic monitoring and control of the industrial process to get the desired results with minimum manual intervention, or in simpler way change the operation systems from being fully operated by humans to operated by robots and computers with a little help from the human, and to be able to do that a new type of small computers called "PLC" Programmer Logic controller, and there are a lot of systems controlled by the PLC such as: garage door, Silo system, Traffic Light Control and the elevator system, and we will focus our efforts in this project on Design a 5-floors Elevator system using the Logixpro I/O simulator screen.

Elevator

Elevators have been pivotal devices for transporting people and goods vertically within buildings or even underground in mines. The concept of elevators dates back to 1743 when the first human-powered, counter-weighted personal elevator was constructed for King Louis XV of France [3]. Since then, elevators have played a crucial role, particularly during the industrial revolution. Over time, they've undergone significant development.

The shift towards automated elevators wasn't surprising. In 1887, Alexander Miles introduced the concept of an automated elevator system. His design allowed the elevator to independently open and close its doors, as well as the doors of the shaft, upon arriving or departing from a particular floor [4]. This automation aimed to enhance user convenience and simplify design and control for manufacturers.

Elevator technology has diversified significantly, encompassing various types such as hydraulic, pneumatic, and passenger elevators. However, for the purposes of our project, we'll primarily focus on elevators as a product that facilitates vertical movement.

Automated elevator system and the Plc program

As we say before using the PLC we can automate any system and as a mechatronics engineers our duty is to design a program that do that using it, but because we don't have a real PLC or elevator obviously, we are going to use the Logixpro application to made a simulation for the real elevator, and that because of the perfect feature that the logixpro provide us that is the I/O Simulator screen which we can use it to design a system from the inputs and outputs that we need.
So for our design we need to design a 5-floors elevator system that have the following rules:
1) The priority also must be given to the bigger number of ordering the floor (if 5 people order the elevator to the 5-floor and 2 people order it to the 3-floor assuming the elevator is on the 4-floor the elevator will go up first and then go down).
2) The elevator doors will open only when the elevator is fully stopped.
3) When the elevator arrives the 5th-floor it can't go up more and if it is in the 1st-floor it will not go down.

I/O Assignment

We can see the I/O assignment in table 1 (it is the ports that the input and the output will connect to the PLC) I: input, O: Output, and we have also provided B: the temporary bits. Also, in figure 1 we can see the I/O simulator screen.

I/O Assignment					
Input	Address	output	Address	Memory	Address
First Floor	I:1/0	Lamp1	O:2/0	Floor 1 down	B3:0/1
Second Floor	I:1/1	Lamp2	O:2/1	Floor 2 down	B3:0/2
Third Floor	I:1/2	Lamp3	O:2/2	Floor 3 down	B3:0/3
Fourth Floor	I:1/3	Lamp4	O:2/3	Floor 4 down	B3:0/4
Fifth Floor	I:1/4	Lamp5	O:2/4	Floor 2 up	B3:0/5
External push button 1	I:1/5	Motor down	O:2/5	Floor 3 up	B3:0/6
Internal push button 1	I:1/6	Motor up	O:2/6	Floor 4 up	B3:0/7
External push button 2	I:1/7	Motor open	O:2/7	Floor 5 up	B3:0/8
Internal push button 2	I:1/8	Motor close	O:2/8	Stop 1	B3:0/9
External push button 3	I:1/9			Stop 2	B3:0/11
Internal push button 3	I:1/10			Stop 3	B3:0/12
External push button 4	I:1/11			Stop 4	B3:0/13
Internal push button 4	I:1/12			Stop 5	B3:0/14
External push button 5	I:1/13			Disabled Motor (up/down)	B3:0/10
Internal push button 5	I:1/14				

Table 2:I/O assignment

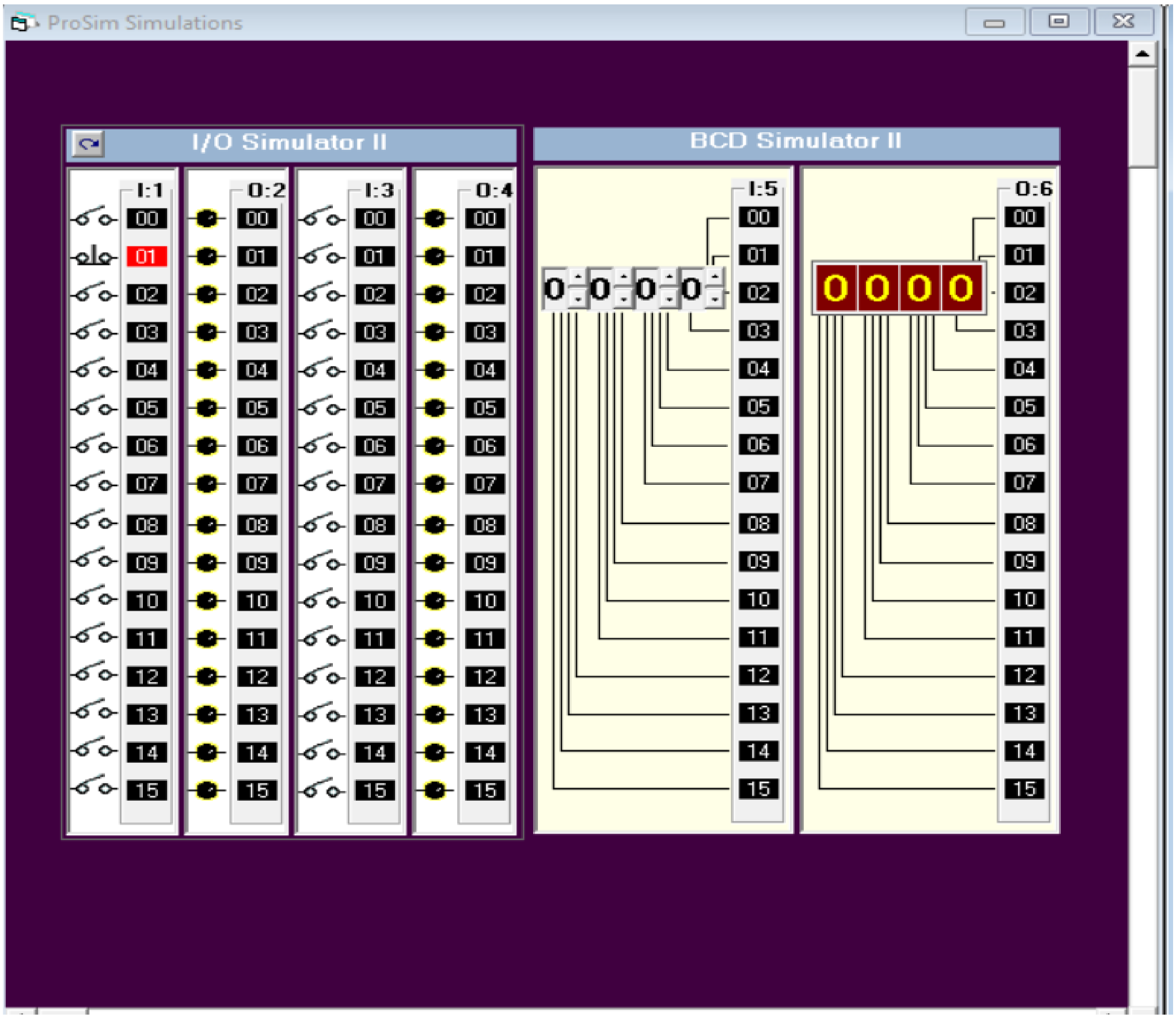

Figure 1: I/O simulator screen

System design (Logixpro Code)

1. We used limit switches at each floor to indicate the elevator spot using lamps.

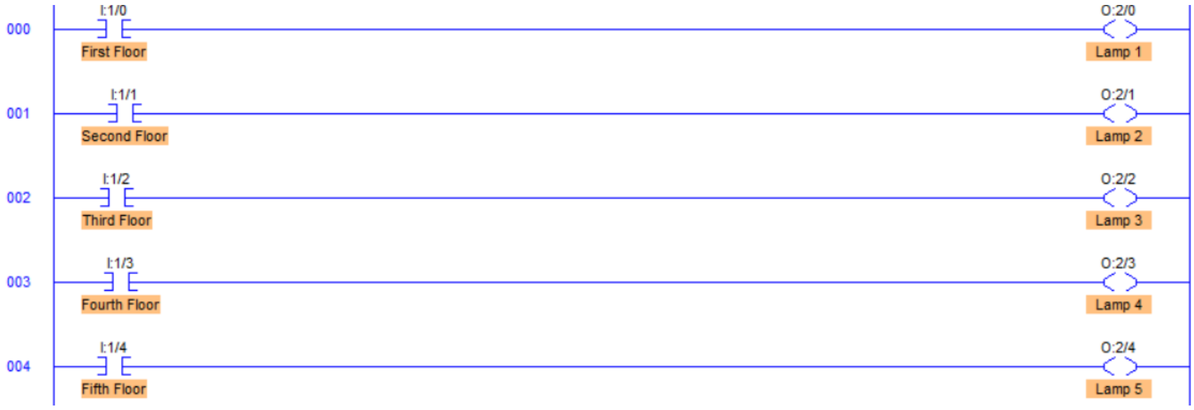

Figure 2: rungs 1-4 of the simulation.

2. Then we used the following 5-8 rungs to detects whether the elevator must go down using previous knowledge of its current place (lamps) and if someone requested the elevator and the floor where it has been requested to (internal and external PBs).

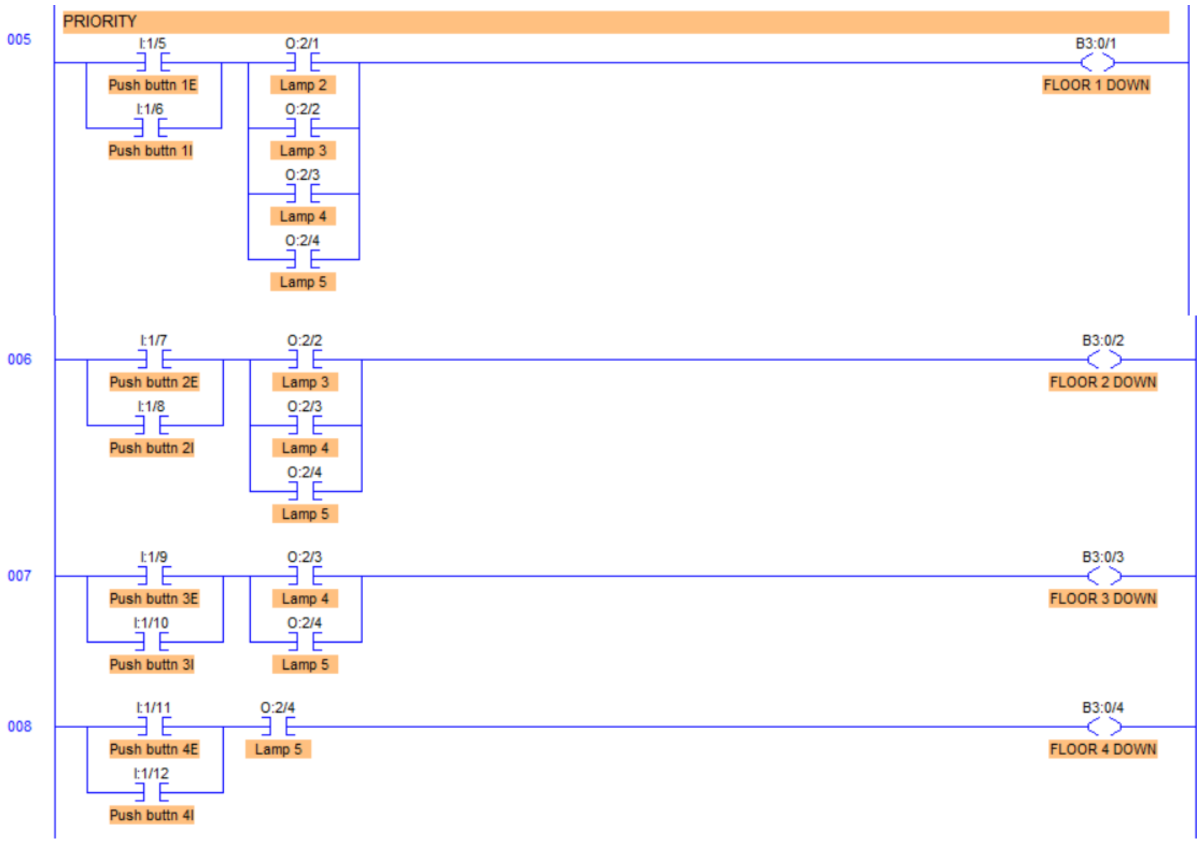

Figure 3: Rungs 5-8 from the simulator

3. Here we used 9-12 rungs to detects whether the elevator must go up using previous knowledge of its current place and where it is requested to go.

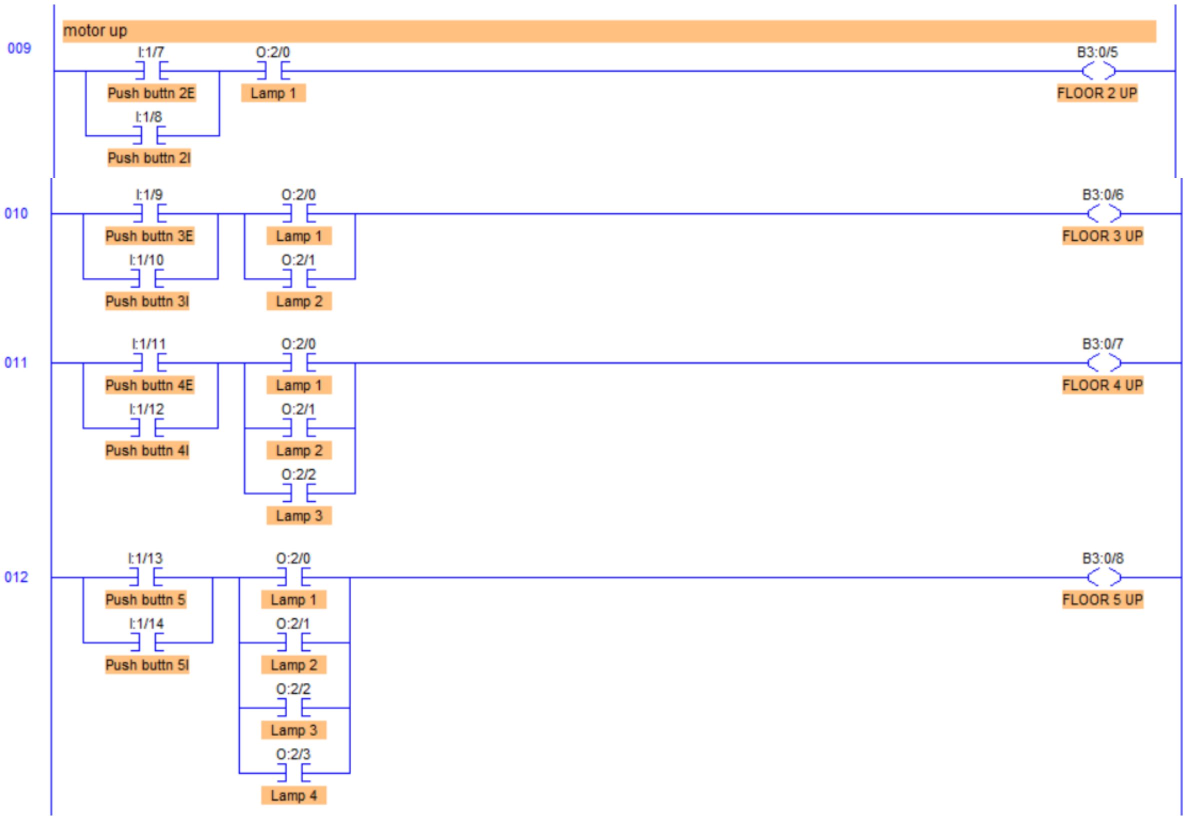

Figure 4: Rungs 9-12 from the simulator

4. Here we count every time a floor is requested as the elevator destination.

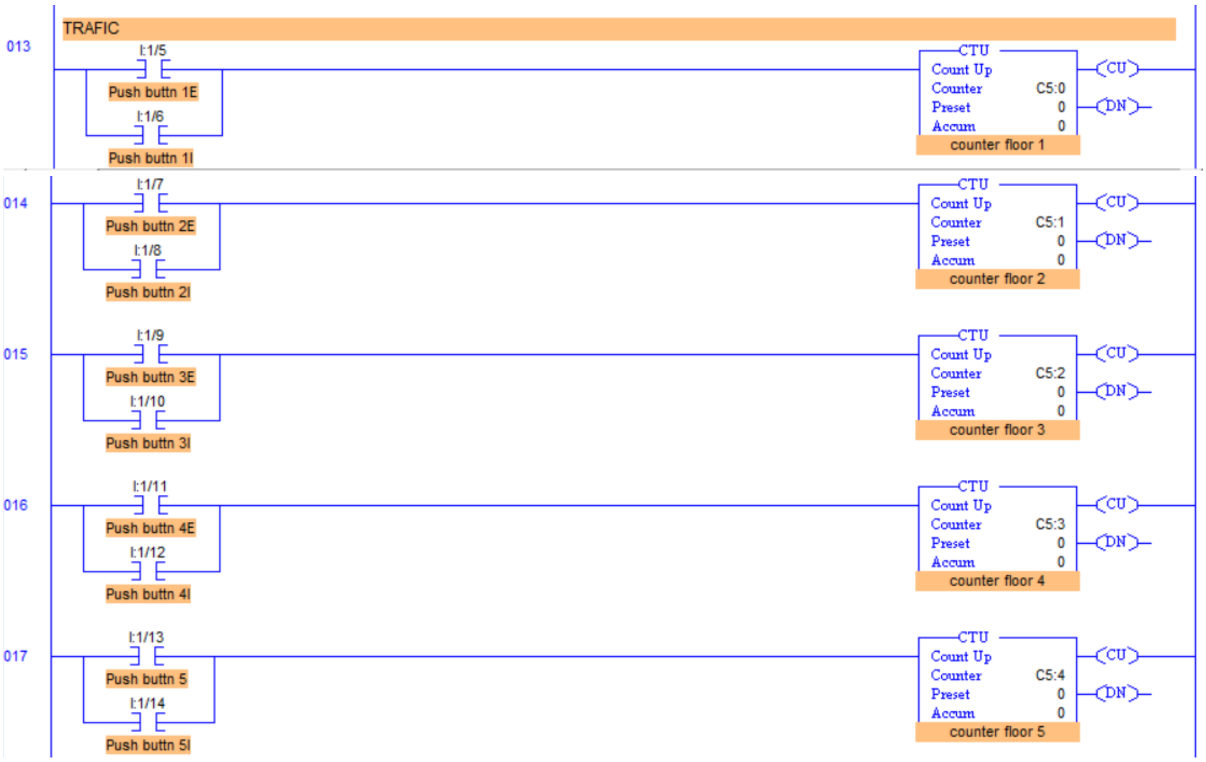

Figure 5:Rungs 13-17 from the simulator

5. The following two rungs saves the result for whether the motor must go up or down in a counter, such that, if it must go up, the counter add one to its accumulator value, if it is requested to go down the counter will subtract one from its accumulator value, and so the resultant value remains in the accumulator determines the elevator direction.

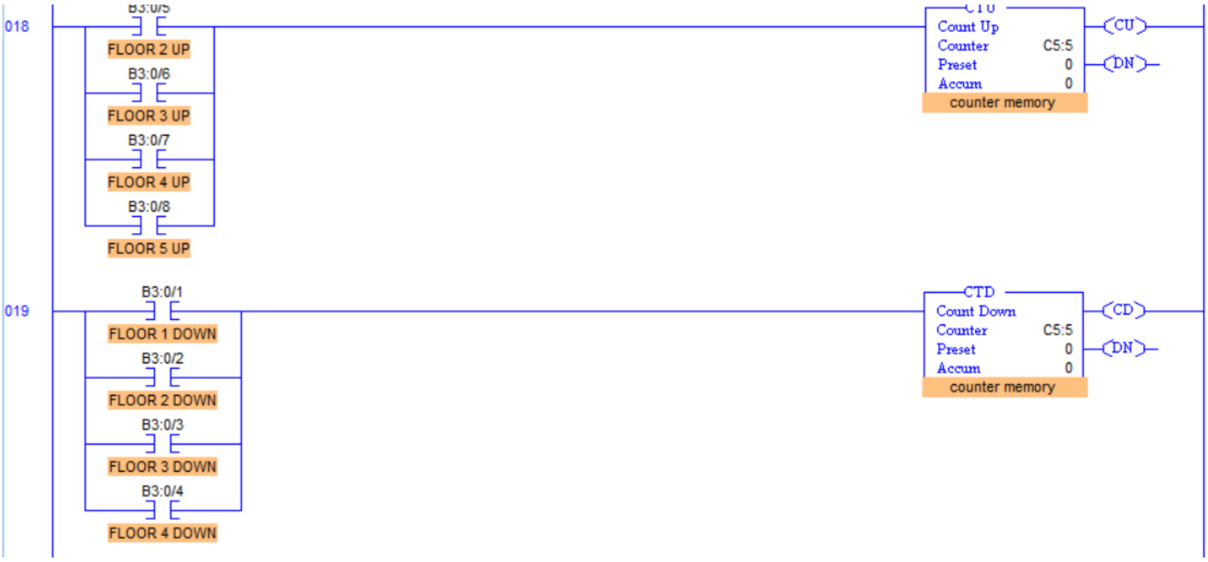

Figure 6: Rungs 18-19 from the simulator

6. Here we command the elevator's motor to move with the right direction, that can be known from the last counter's accumulator, if it holds a positive number, the elevator will go up, if it holds a negative one, the elevator go down.

Figure 7: Rungs 20-21 from the simulator

7. If the number of requests for the elevator to go up is the same as the requests number to go down, then the counter accumulator will have a value of zero, but we can Distinguish between having the same number of requests and having no request at all by latching the requests until the request is fulfilled. In this case, we give the priority to the elevator to go down if it wasn't moving up.

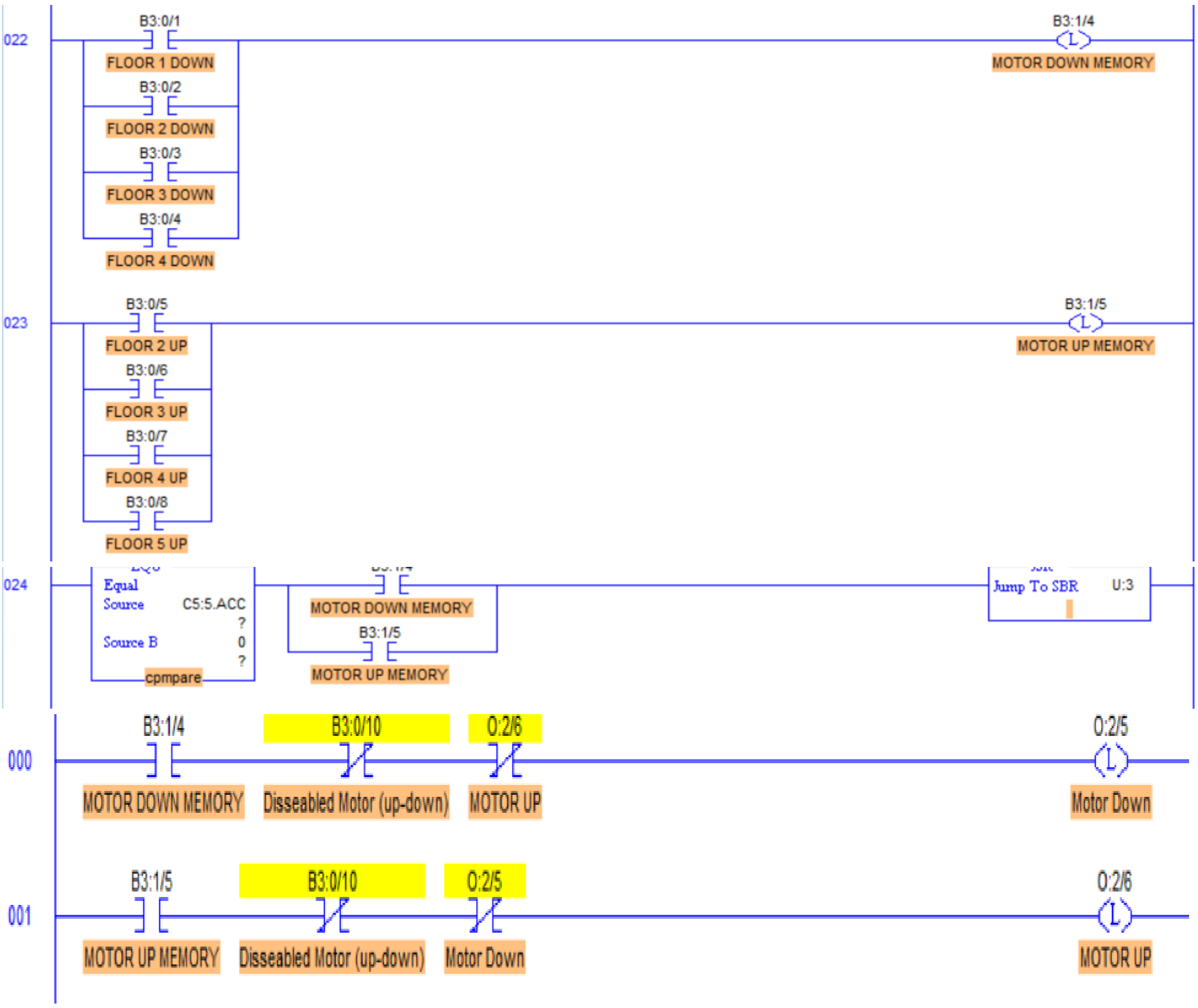

Figure 8.rungs from 22-24 along with subroutine3.

8. The following ten rungs detects whether the motor must stop at the current floor, or it should continue moving, to do so, we use the position indicator (lamps) and the requests counter, we also move the counter accumulator to a register (N7:0) before clearing the counter accumulator, since the request has been fulfilled.

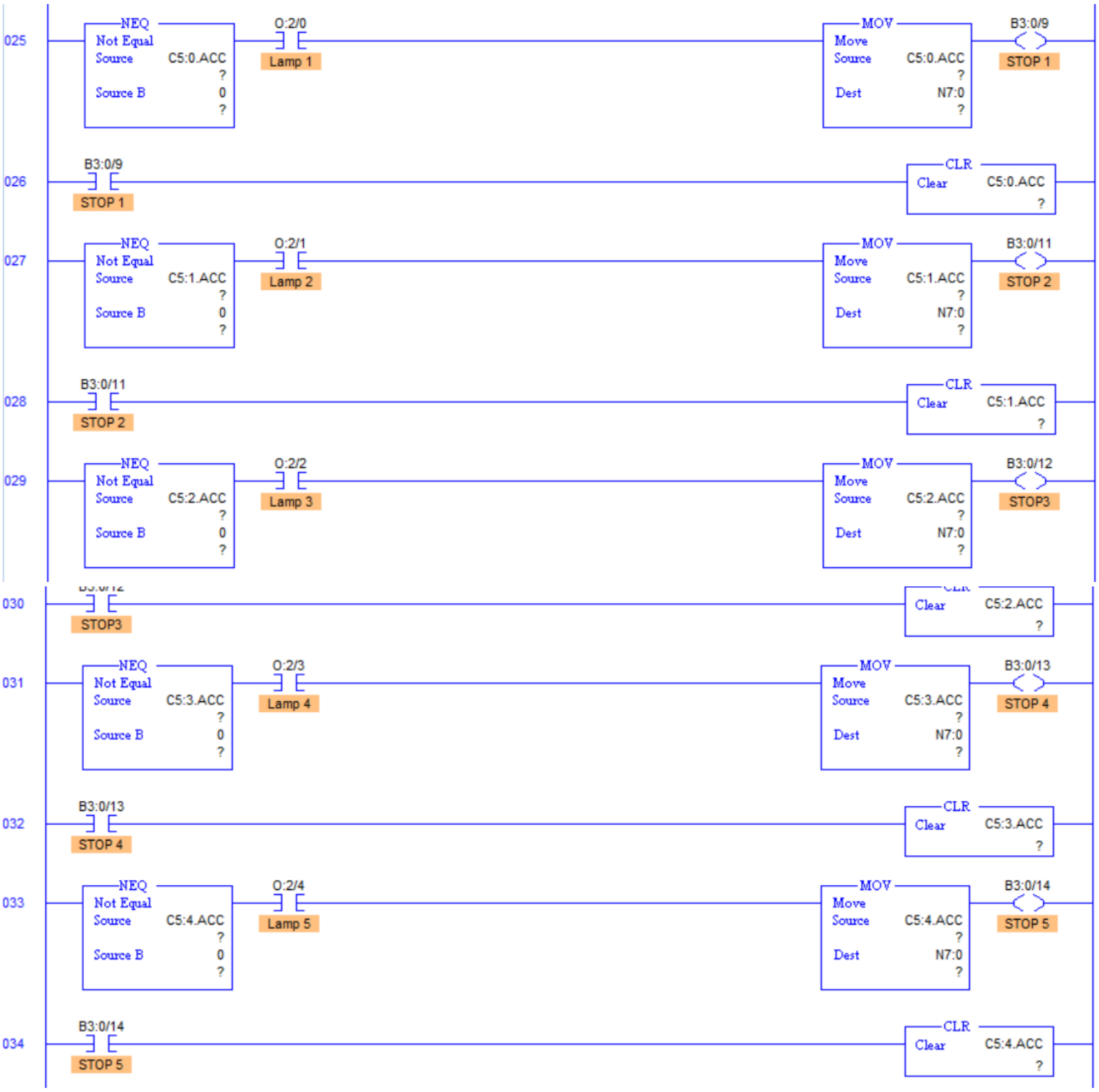

Figure 9: Rungs 25-34 from the simulator

9. The stop signal is then latched until the elevator door is closed again and the elevator is allowed to move again. Once the up/down motors are unlatched the main counter should subtract or add -depending on the elevator direction of moving- the value of the counter's accumulator of the current floor we stopped at.

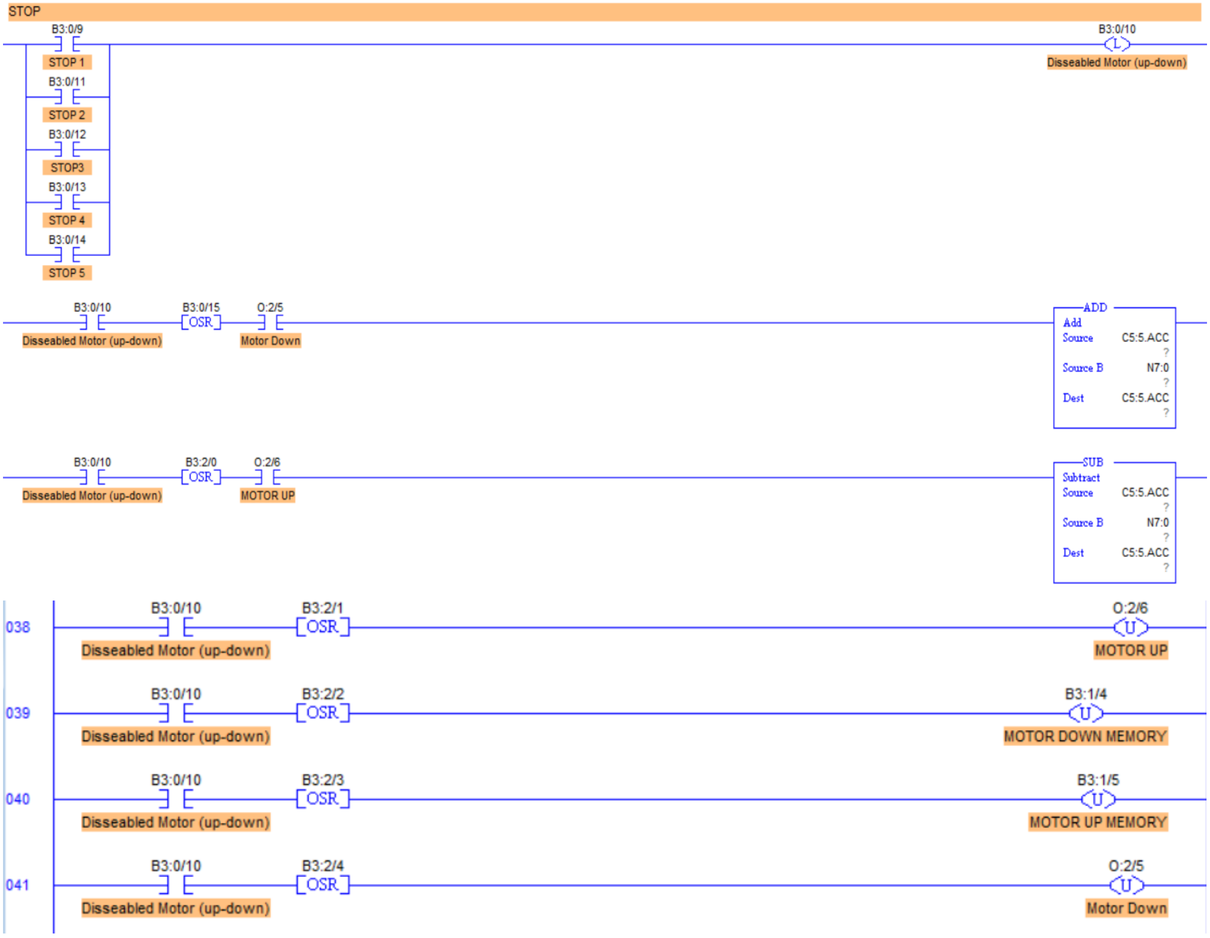

Figure 10: Rungs 35-41 from the simulator

10. The following 6 rungs represents the sequence ocurring after stopping the up/down motors, first, the elevator start opening for 10seconds, and then remain idle for 20seconds waiting for passengers to enter/exit, followed by another 10seconds for the elevator to close its doors. Then the elevator is allowed to move again.

Figure 11: Rungs 42-48 from the simulator

Code explanation

For the code to be able to follow the rule we mention the design will be divided into 6 main parts as shown in the figures before, but first we are going to talk briefly about the code:

Our design consider a 5 floor elevator that can be ordered from the outside panel as shown in Figure 12 and from inside the elevator as shown in figure 13, and the lamps as shown in figure will be on when the elevator is in there floor and to be able to do that we but a limit switches in every floors to light these lamp, and according to where the elevator is and where the order come from the up or down motor will turn on and other specifications we will talk about it below.

Figure 12: The external buttons for the elevator

Figure 13: internal panel buttons for the elevator

1) The lamps assignment

The first 5 rungs is responsible to assign every light to its floor (I:1/0 for the first floor limit switch will turn the lamp O:2/0 which is the first floor lamp).

2) Motor down

The rungs from 5-8 is responsible of let the elevator know when the motor is going to go down, but it will only turn the temporary bit that will be used in later rungs (when either the internal and external first floor button and the elevator is in any floor expect the first floor it will go down to the first floor etc.).

3) Motor up

The rungs from 9-12 is responsible of let the elevator know when the motor is going to go up, but it will only turn the temporary bit that will be used in later rungs (when either the internal and external fifth floor button and the elevator is in any floor expect the first floor it will go up to the fifth floor etc.).

But what will happened if two button is pressed at the same time where do the elevator will go, we took that in mind and the next rungs will take care of it.

4) Traffic

The rungs from 13-19 is responsible of dealing with problem above by using 1 counter C5:5 as a counter up and counter down to decide where the bigger number of orders and according to that it will go to that floor, and to do that we will use the temporary bits that we decide to keep for later and every time the button is pressed it will cause a pulse in the signal and its counter C5:0-C5:4 will increase once for every pressed button for that floor(if the elevator is in the fourth floor and 5 people press the fifth floor button and 2 press the second floor button the counter C5:5 will be 3 (5-2)).

5) Move up and down

The rungs from 20-34 is responsible for the actual movement of the elevator up or down and that by taking the value of the counter C5:5 if it's more than 0 the elevator will go up and if its less than 0 it will go down but when it is equal to zero (the number of orders in the two floors are equal) in this case the subroutine 3 that shown in figure14 will make the elevator go down. Then after we decide where the elevator will go up or down, we must decide the floor that it will stop in it, and also when it arrives, its counter must become zero (all the people are getting on the elevator) and that is the function for rungs in figure 15.

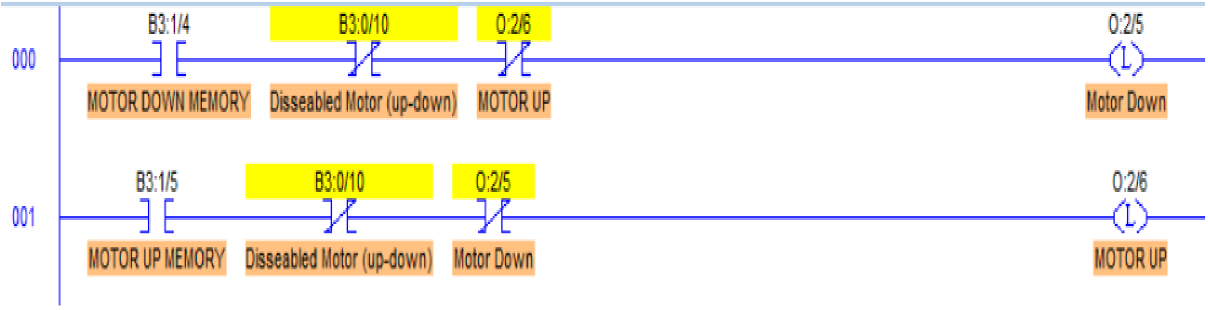

Figure 14: Subroutine for the motor down

When the lamp O:2/0 is on and the Counter C5:0 accumulator is not zero that value will turn the bit B3:0/9 that is responsible of stopping the elevator as we are going to see in the next rungs and will clear the counter of the first floor.

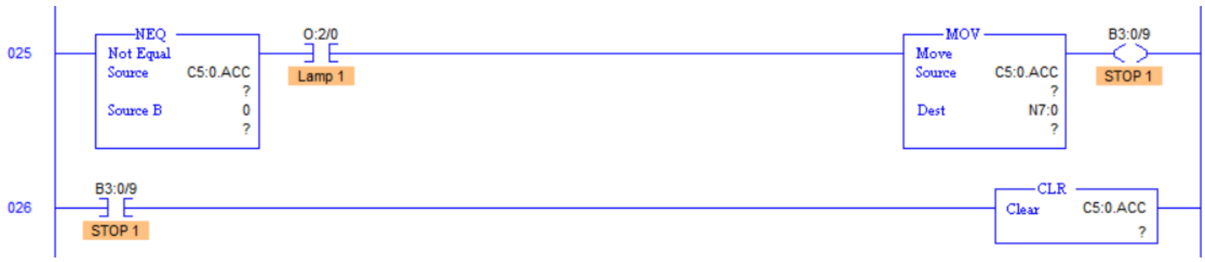

Figure 15: the Rungs for choosing the first floor for the elevator to stop at.

6) Stop

From the name the rungs from 35-47 is responsible of stopping the motor when the stop bit for the floor is initiate and it will keep the motor closed until the door is opened and fully closed (for safety reasons) and by using timers for opining the door, entering the elevator and closing the door the motor must be turned off, and these timers are T4:0,1,2 with timing off 10s 20s 10s respectively.

Conclusion

In the end of the project we discovered that using the logixpro simulation is a good idea to make sure that our design is fully functional to avoid wasting the time and money if we design it using a real parts, also it will save some lives if something happened in a real working place that can cause some accidents but when using the simulator not thing will happened and nobody will be in danger, also it's a good method for new engineers or any other human being to be able to learn the programming language and use it to make a lot of different projects, and for the elevator the automated version is easier to deal with so it is more popular in these days, and it is the perfect use for our experience in the lab.

References

1) Automation and process control.
2) Automation and process control lab.
3) https://www.elevators.com/a-brief-history-of-elevators
4) https://blog.ansi.org/Brown