

Implementation of a Four-Floor Programmable Logic Controlled Elevator System

KAMALU, U.A,
ugokamax@yahoo.com

OGUNLEKE, F.A
francakins@me.com

ABSTRACT: *This paper is on the design and implementation of a 4-Floor Elevator control system using Programmable Logic Controller (PLC). It involves developing a model of the Four-Floor and simulating the control using a SCADA system. The PLC in consideration is Siemens S7-200 with minimum of 14 inputs and 10 outputs, and the SCADA system in use is Intouch Wonderware, Schneider Electric product software. The elevator operates in a power saving mode; resulting in the use of an intelligent design model. The elevator control system is programmed in a ladder diagram format into the PLC. The design gives the basic structure, control principle and realization method of the PLC control system in details. It also presents the ladder diagram of the key aspects of the system. The system has simple peripheral circuit and the operation result showed that it enhanced the reliability and performance of the elevators. Some suggestions on how to extend this system to the control of more than four floors are also included. It contributes to the design and use of simple circuit, easy to understand, operate and use.*

Keyword: Controller, Elevator, Ladder, Logic, Programmable, SCADA

1 INTRODUCTION

With the development of architecture technology, the building structures are taller and elevators become important vertical transportation vehicles in high-rise buildings. For most people in urban cities, elevators have become an integral part of their life. Simply stated, an elevator is a hoisting or lowering mechanism, designed to carry passengers or freight, and is equipped with a car and platform that typically moves in a fixed guides and serves two or more landings. They are responsible to transport passengers, living/working or visiting in the building, comfortable and efficiently to their destinations. So the elevator control system is essential in the smooth and safe operation of each elevator. It tells the elevator in what order to stop at floors, when to open or close the door and if there is a safety-critical issue.

Hydraulic and roped elevators are the two types of elevators in use today. The main design considerations for choosing either electric traction drive or hydraulic for a particular project are the number of floors, the height of the building, the number of people to

be transported, desired passenger waiting times and frequency of use.

This paper elaborates the design and implementation of an elevator that is controlled using Programmable Logic Controller (PLC).

With latest developments in science and technology, PLC technologies have been widely applied in industrial control. A lot of factory production lines, processing equipment, ship loading and unloading device and the operation of elevators are controlled by the PLC, as long as the scheduled control tasks are compiled in the form of a string of instructions and stored in the memory. It is the job of PLC to drive the output devices in response to input signals present at its input ports.

PLC applications in elevator systems have been established to be one of the most potent means of elevator control. PLC acts as the master controller. It collects a variety of input signals of the elevator, including the location of the elevator, state of the signal of internal and external command button, the door lock

signal, the door zone signal, forced to slow down, the signal within the wells and fire signals. Based on the input signals to the elevator, which could be the elevator floor signal, speed signal, switch gate signal, directional relay signal or brake relay signal, the programmed logic runs a calculation to enable it control the operations of the elevator.

2. METHODOLOGY

The design of the elevator control system consists of two main parts; hardware and software design. The limit switches and push buttons are primary part of the system. They send signal to the PLC, which in turn; based on these input signals and the code implemented in it, generate the appropriate output to the pulley and door motors.

The important part of the system is the controller and the type of controller used here is PLC which is configured to control the operations of the elevator. Electromagnetic actuator and contact action were so slow, mechanical and electromagnetic components have large inertia. Therefore, the control precision of system was difficult. So there is a need to go for modern type of controllers like PLC.

In this type of control there is provision to monitor the entire system by means of SCADA (Supervisory Control and Data Acquisition) system.

2.1 Elevator Hardware Design

The design of the elevator system will be based on the calculation of certain parameters which defines the operation of an elevator. To meet the objectives of the elevator system, theoretical studies of its operation and the machine model of an ideal elevator was considered. Then to validate and to implement the simulated model as a prototype, certain calculations have been carried out to determine some critical parameters. The parameters include the size of the cabin, maximum number of persons that can be accommodated in the elevator, strength of the rope, base plate design, type of motor used, motors power rating and torque rating.

2.1.1 Design of Elevator Cabin

The elevator cabin is used to transport person and goods. It is a small enclosure with transparent window where persons enter and are protected before the elevator moves on. The elevator cabin is designed by having in mind the number of passengers it is going to accommodate. Also it should have the capacity to bear the weight of the passengers travelling in the elevator. The proposed elevator is having the capacity to accommodate approximately 10 persons of 65 kg each

Maximum No. of passengers that can be accommodated in the cabin = 10

Approximate weight of each person = 65 kg

Maximum weight that the cabin can handle = 650 kg

To accommodate persons the cabin has been designed with the following dimensions:

L x W x H1

This dimension (1.5 m x 1.5 m x 2 m) will give a total volume of 4.5 m³. This is assumed to accommodate a total of 10 persons with average volume occupancy of 0.45 m³. Also, considering the total weight of 650KG the cabin can handle, by introducing allowance of +50kg, a maximum weight of 700kg is in consideration. This total weight and the volume will guide in selecting the base support and the rope strength to carry the cabin and the passenger without failure.

2.1.2 Base Plate Support Calculation and Selection

The first and most important step towards successful selection of plate for use is the proper understanding of its application. Besides the calculation, the property requirements for plate applications in elevator designs are considered. The main properties include; corrosion resistance, oxidation resistance, strength and ductility, suitability for internal fabrication, stability in service, thermal conductivity, thermal expansion, electrical resistance and rigidity among others.

The required thickness of the base plate can be calculated as follows:

The cabin has been designed to take a maximum weight of 700kg. Having in mind the area coverage for 10 persons in the cabin which gives a volume of 4.5 m³ giving the area

as 2.25 m², and considering tensile strength that can withstand a force of 700kg x 9.8m/s² (6.86KN). According to plate selection table by Solid-Liquid Filtration and Separation Technology, the plate minimum thickness will be 10mm and the maximum thickness will be 30mm.

For the sake of design simplicity the entire system has reduced to a prototype of 1:20 ratio to the real model.

2.1.3 Rope Strength

The tension on the rope and its choice of selection can be determined with the following simple calculation. The force exerted on the rope can be found by using Newton’s formula.

$$F = \text{mass} \times \text{acceleration} = mg \dots\dots\dots 2$$

The expected entire weight of the elevator when fully loaded will be:

$$10 \text{ persons with } 65 \text{ kg each} = 650 \text{ kg}$$

$$\text{Cabin weight} = 100 \text{ kg}$$

$$\text{Counter weight} = 100 \text{ kg}$$

$$F = (650+100+100) \times 9.8 = 8330 \text{ N} = 8.33\text{kN}$$

The force acting on the string is 8.33kN. So a string with the capacity to bear at least 9kN has to be chosen on the standard rope strength in table 1.

Table 1: Rope Strength Table

Rope Diameter		Minimum Breaking Strength		Safe Load		Weight	
(in)	(mm)	(lb)	(kN)	(lb)	(kN)	(lb/ft)	(kg/m)
1/4	6.4	5480	24.4	1100	4.89	0.11	0.16
5/16	8	6520	29.3	1700	7.56	0.16	0.24
3/8	9.5	12200	54.3	2440	10.9	0.24	0.36
7/16	11.5	16540	73.6	3310	14.7	0.32	0.48
1/2	13	21400	95.2	4280	19.0	0.42	0.63
9/16	14.5	27000	120	5400	24.0	0.53	0.79
5/8	16	33400	149	6680	29.7	0.66	0.98
3/4	19	47600	212	9520	42.3	0.95	1.41
7/8	22	64400	286	12900	57.4	1.29	1.92
1	26	83600	372	16700	74.3	1.68	2.50
1 1/8	29	105200	468	21000	93.4	2.13	3.17
1 1/4	32	129200	576	25800	115	2.63	3.91
1 3/8	35	155400	691	31100	138	3.18	4.73
1 1/2	38	184000	818	36800	164	3.78	5.63
1 5/8	42	214000	952	42800	190	4.44	6.61
1 3/4	45	248000	1100	49600	221	5.15	7.66
1 7/8	48	282000	1250	56400	251	5.91	8.80
2	52	320000	1420	64000	285	6.72	10.0

From the rope manufacturer’s data sheet Table 1, to meet the design requirement, the cable with 10.9kN safe load capacity and thickness of 3/8” or 9.5 mm is considered.

2.1.4 Power Calculation for Motor

The motor used here is a capacitor start single phase induction motor. Based on the design specification the output power and the output torque of the motor are calculated by a simple calculation.

Power and torque calculation

$$\text{Weight of empty cabin} = 100 \text{ kg}$$

$$\text{Counter weight} = 100 \text{ kg}$$

$$10 \text{ persons with } 65 \text{ kg each} = 650 \text{ kg}$$

For constant speed operation of 1 m/s or 60 m/min

The power, work and force are calculated as follows:

$$\text{Power} = \text{Work} \times \text{Time} \dots\dots\dots 3$$

$$\text{Work} = \text{Force} \times \text{Distance} \dots\dots\dots 4$$

$$\text{Force, } F = mg \dots\dots\dots 5$$

$$\text{Power} = \text{Force} \times \frac{\text{distance}}{\text{time}} \dots\dots\dots 6$$

$$\text{Power} = \text{Force} \times \text{Velocity} \dots\dots\dots 7$$

$$\text{Force} = 650 \times 9.8 = 6370 \text{ N}$$

$$\text{Power} = 6370 \text{ N} \times 1 \text{ m/s} = 6370 \text{ W}$$

$$\text{But } 1 \text{ hp} = 745 \text{ W}$$

$$\text{Therefore } 6370 \text{ W will be } \frac{6370}{745} = 8.55 \text{ hp}$$

This can be approximated to 9hp or 6705 W

$$\text{Rotational Speed, } \omega = 1500 \text{ rpm}$$

The torque of the motor can be calculated by using the following expression:

$$\text{Torque of the motor (Nm)} = \frac{\text{Power in Watts}}{2\pi\omega} \dots\dots\dots 8$$

$$= \frac{6705}{2\pi \times 1500} = 31.5 \text{ Nm}$$

BUT with the prototype ratio of 1:20

$$\text{Weight of cabin} = 5 \text{ kg}$$

$$\text{Counter Weight} = 5 \text{ kg}$$

10 persons of 3.25 kg = 32.5 kg

For constant speed operation of 1 m/s = 60 m/min

The power, work and force are calculated using the equations 3 to 7.

$$F = 32.5 \times 9.8 = 318.5 \text{ W}$$

$$\text{But } 1 \text{ hp} = 745 \text{ W}$$

Therefore 318.5 W = 0.427 hp which is approximately 0.5 hp Or 372.5 W

Rotational Speed, $\omega = 1500 \text{ rpm}$

$$\begin{aligned} \text{Torque of the motor (Nm)} &= \frac{\text{Power in Watts}}{2\pi\omega} \\ &= \frac{372.5}{2\pi \times 1500} = 1.749 \text{ Nm} \end{aligned}$$

2.1.5 Gear Calculation

To achieve the desired speed of 1 m/s from the motor output of 1500 rpm, the diameter and the circumference of the gears has to be calculated thus;

$$\text{Cabin weight} = 100\text{kg}$$

$$\text{Counter weight} = 100\text{kg}$$

$$10 \text{ persons with } 65\text{kg} = 650\text{kg}$$

Speed of the elevator is 1 m/s

$$= 60\text{m/min} = 6000\text{cm/min}$$

Circumference of gear 2, $c_2 = 30\text{cm}$

$$\text{Therefore, diameter of gear 2, } d_2 = \frac{30}{\pi} = 9.5 \text{ cm}$$

To find the rpm of gear:

$$\text{rpm} = \frac{\text{velocity}}{\text{circumference}} \dots\dots\dots 9$$

$$\text{Speed of gear 2} = \frac{6000}{30} = 200 \text{ rpm}$$

However, the speed of gear 1 is 1500 rpm

Speed and length are inversely proportional

$$\text{Therefore, } \frac{200}{1500} = \text{Circumference of gear 1} = \frac{c_1}{30}$$

$$c_1 = 4\text{cm}$$

This implies, the diameter of gear 1, $d_1 = \frac{c_1}{\pi}$

$$= \frac{4}{\pi} = 1.273 \text{ cm}$$

For smooth meshing of gears, both the gears should have same sized teeth and invariably have the same pitch, p .

The pitch can be calculated using the equation (10)

$$P = \frac{2\pi r d_1}{N d_1} = \frac{2\pi r d_2}{N d_2} \dots\dots\dots 10$$

Take the number of teeth to be 8 (1 tooth for every 45 degree)

$$P = \frac{4}{8} = 0.5$$

From 'p' the value of $N d_2$ can be found out

$$N d_2 = \frac{2\pi r d_2}{p} = \frac{30}{0.5} = 60 \text{ teeth}$$

Therefore, the gear ratio is given by

$$R = N d_1 : N d_2 = 8:60 = 1:7.5$$

2.2 Controller Hardware Design and Configuration

When an engineer is embarking on any program of automation, it must be remembered that the controller is only a tool used to perform the necessary tasks. The actual task of implementing automation is therefore paramount, often involves many methodologies in its system design phase. Once the specifications are established, the job of selecting a suitable controller will then become most important as this would determine how at ease the automation program might continue.

There is a massive range of PLC systems available today, with new additions or replacements continually being produced with enhanced features of one types or another. Advances in technology are quickly adopted by manufacturers in order to improve the performance and market status of their products. However, irrespective of make, the majority of PLC in each size range is very similar in term of their control facilities. Where significant differences are to be found is in the programming methods and languages, together with differing standards of manufacturer support and backup. With the vast choice of equipment now available, the engineer can usually obtain similar systems from several original equipment manufacturers (OEM). When the

specification requires certain types of function or input/output, it is possible that one system from a single manufacturer standing out as more superior or cost effective than the other; but normally this is rarely the case.

To determine the most suitable PLC to be used in the automation task, there are several basic considerations to be made:

- Necessary input/output capacity;
- Types of I/O required;
- Size of memory required;
- Speed and power required of the CPU and instruction set
- Manufacturer's support and backup.

Based on these considerations; and having carefully considered features of most of the PLCs; Siemens STEP S7-200, CPU 224 PLC has been chosen. It has 14 input pins and 10 output pins and it matches the I/O requirements of this design work.

2.3 Elevator Control Algorithm and Software Development

The main aims of the elevator control system are:

- To bring the lift car to the correct floor.
- To minimize travel time.
- To maximize passenger comfort by providing a smooth ride.
- To accelerate/decelerate and travel within safe speed limits.

Figure 1 shows the control algorithm of a typical elevator control system. This can be applied to any system of elevator control

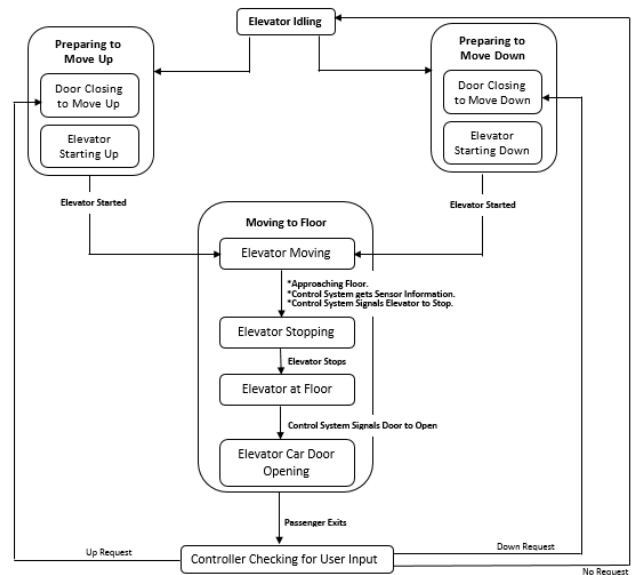


Figure 1 Elevator Control Algorithms

2.4 Software Design Logic

For a four-floor control system, the control software has been designed in such a way that each floor has its own control software block; in this way tasks will be performed in accordance with the particular location of the lift vehicle. This enables the design to be logically organized and, in addition, determine what routine to run in response to a particular input signal at different locations of the lift vehicle. The block diagram of the software design is shown in figure 2

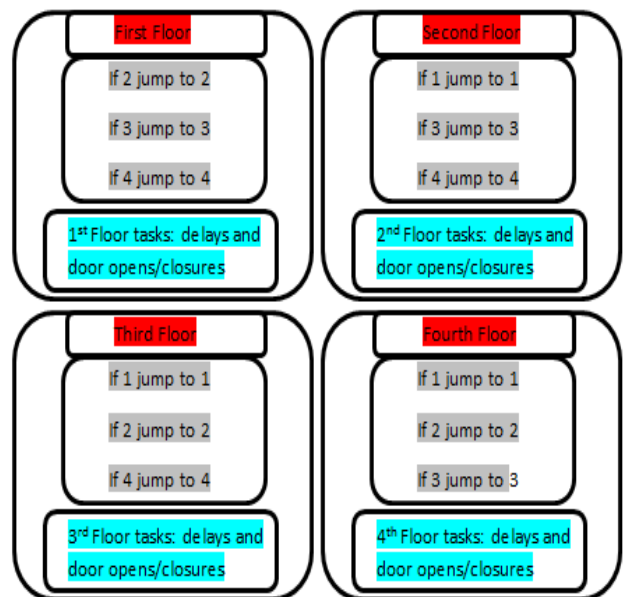


Figure 2 Software Design Block Diagram

Each block of the above mentioned blocks has been itself divided into three minor blocks and these are: Block register initialization, Destination scan and Push Button Scan.

2.4.1 Block Register Initialization

The function of this block is to make sure all the starting parameters are in the desired states when the car enters a particular floor block. As an illustration, when the elevator is on the first floor, location initialisation needs to be set to this floor. When the system is just powered on, the vehicle should be in a static state (not moving) and it does not make sense to say that the direction is up or down since there is no motion at all. The doors are sealed and at the moment the system has not been instructed to go anywhere. The above mentioned information has to be entered into the controller in form of instructions. These instructions have been organized into one set that is referred to as *Register Initialization block* since they are entered into the system by assigning values to registers.

2.4.2 Destination Scan

The Register Initialization block sole role is to tell the system where it is, or to set the starting point in the particular floor block. The next job is to check whether there are instructions to go anywhere. If there are any destination request the system should run routines to go to that particular destination. If no requests are available, the system should keep looping in this section, looking for a destination until it finds one.

2.4.3 Push-Button Scan

So far, the system knows its position or location and it knows how to look for a destination; this section is concerned with finding a way to give the system a destination. From the hardware design section signals are entered through Push-Buttons. Depending on the location of the lift vehicle, when a particular destination request PB is engaged, appropriate registers should be set. With the destination set, the Destination Scan block should find it and appropriate routines should be run.

2.5 Overview of Implementation Techniques

Any kind of programming style to be adopted must ensure that tasks are achieved by assigning values to the registers or setting/resetting bits in different registers. This design is not an exception to that. For example, the Register Initialization block is composed of a series of bit set/reset instructions or assignments of a full byte value to a particular register. Travels from floor to floor or door open/closure delays are achieved through Ladder Logic implementations.

The control algorithms developed in the process of implementing the software design are translated into ladder logic for effective design analysis. Based on these requirements, the logic has to be developed accordingly with the specifications of the PLC chosen. For the working model of an elevator, various logics for different operations have to be developed and it is discussed in the subsequent subtitles.

The flowchart in Figure 3 achieves the floors register initializations and destination scans.

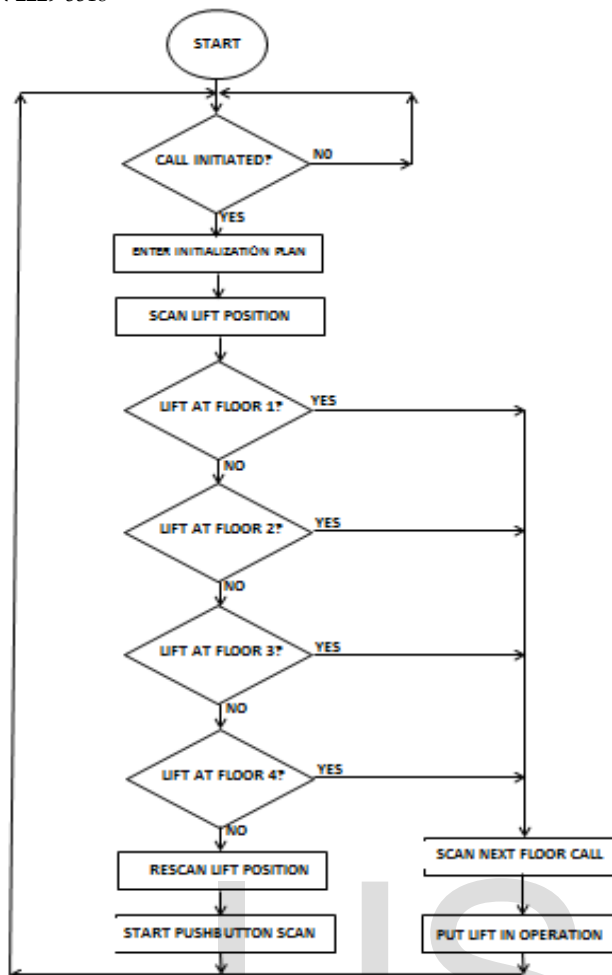


Figure 3 Initialization and Scan Flowchart

2.6 Ladder Logic Implementation

The Siemens SIMATIC micro/win program is used in the project to develop the logic which controls the PLC using the ladder diagram language. The following is detailed explanation for each and every network in the ladder diagram used to implement the elevator's software. However, the following highlights the addresses locations for the inputs, outputs and the memory assignments.

Input Assignment

- | | | |
|-----|------------------------|--------|
| 1. | FLOOR 1 LIMIT SWITCH | - I0.0 |
| 2. | FLOOR 2 LIMIT SWITCH | - I0.1 |
| 3. | FLOOR 3 LIMIT SWITCH | - I0.2 |
| 4. | FLOOR 4 LIMIT SWITCH | - I0.3 |
| 5. | FLOOR 1 CAR CALL | - I0.4 |
| 6. | FLOOR 2 CAR CALL | - I0.5 |
| 7. | FLOOR 3 CAR CALL | - I0.6 |
| 8. | FLOOR 4 CAR CALL | - I0.7 |
| 9. | FLOOR 1 HALL CALL UP | - I1.0 |
| 10. | FLOOR 2 HALL CALL UP | - I1.1 |
| 11. | FLOOR 2 HALL CALL DOWN | - I1.2 |
| 12. | FLOOR 3 HALL CALL UP | - I1.3 |
| 13. | FLOOR 3 HALL CALL DOWN | - I1.4 |
| 14. | FLOOR 4 HALL CALL DOWN | - I1.5 |

- | | | |
|-----|------------------------|--------|
| 15. | EMERGENCY STOP | - I1.6 |
| 16. | MANUAL DOOR OPEN/CLOSE | - I2.0 |

Output Assignment

- | | | |
|----|--------------------------|--------|
| 1. | PULLEY UP MOTOR SIGNAL | - Q0.0 |
| 2. | PULLEY DOWN MOTOR SIGNAL | - Q0.1 |
| 3. | MOTOR STOPS SIGNAL | - Q0.2 |
| 4. | DOOR OPEN SIGNAL | - Q0.3 |
| 5. | DOOR CLOSE SIGNAL | - Q0.4 |
| 6. | BCD TO 7 SEGMENT DISPLAY | - Q0.5 |
| 7. | BCD TO 7 SEGMENT DISPLAY | - Q0.6 |
| 8. | BCD TO 7 SEGMENT DISPLAY | - Q0.7 |
| 9. | BCD TO 7 SEGMENT DISPLAY | - Q1.0 |

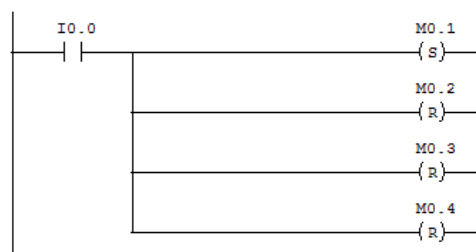
Memory Assignment

- | | | |
|-----|---------------------------|--------|
| 1. | FLOOR 1 MEMORY LOCATION | - M0.0 |
| 2. | FLOOR 2 MEMORY LOCATION | - M0.1 |
| 3. | FLOOR 3 MEMORY LOCATION | - M0.2 |
| 4. | FLOOR 4 MEMORY LOCATION | - M0.3 |
| 5. | FLOOR 1 LAST VISIT MEMORY | - M0.5 |
| 6. | FLOOR 2 LAST VISIT MEMORY | - M0.6 |
| 7. | FLOOR 3 LAST VISIT MEMORY | - M0.7 |
| 8. | FLOOR 4 LAST VISIT MEMORY | - M1.4 |
| 9. | FLOOR 1 VISIT MEMORY | - M1.0 |
| 10. | FLOOR 2 VISIT MEMORY | - M1.1 |
| 11. | FLOOR 3 VISIT MEMORY | - M1.2 |
| 12. | FLOOR 4 VISIT MEMORY | - M1.3 |

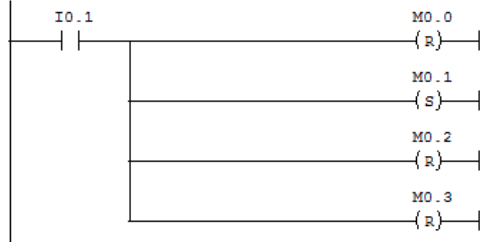
The program is analysed by networks arrangement and there are four major design considerations for the control parameters in this design.

2.6.1 Floor Limit Switch Sensor Design

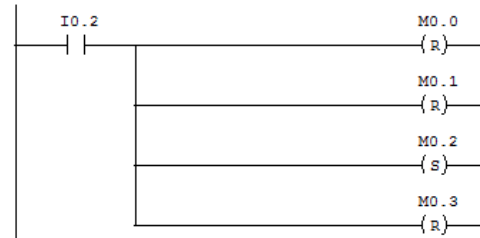
Each floor is designed with limit switch as sensor and each sensor is assigned a memory location for proper interlock in the process of design manipulations. The following networks describe the sensor designs for floor 1 to floor 4 respectively. The safety interlock design in each of the floor shows that whenever the limits switch memory of any floor is at "SET" position; all other memories are at "RESET".



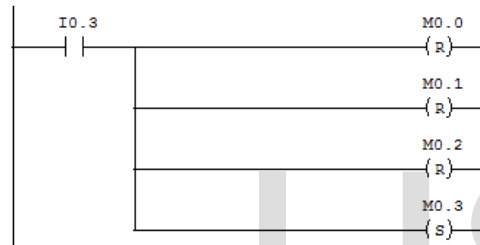
Network1: Floor 1 Limit Switch Sensor design



Network2: Floor 2 Limit Switch Sensor design



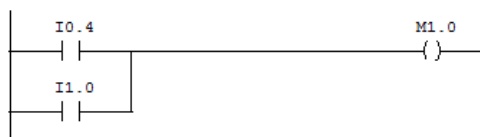
Network 3: Floor 3 Limit Switch Sensor design



Network4: Floor 4 Limit Switch Sensor design

2.6.2 Hall and Car Call Design

There are two kinds of calls in the elevator they are hall call and car call. The hall call is when the pushbutton in the control panel outside the elevator cabin is pressed. When the pushbutton on the control panel which is inside the elevator car is pressed that is the car call. The inputs addresses denoted by I0.4, I0.5, I0.6 and I0.7 are for Car call request. Similarly, the inputs addresses denoted by I1.0, I1.1, I1.2, I1.3, I1.4 and I1.5 are for Hall call request which can either be upward call or downward call as indicated in the ladder representation.



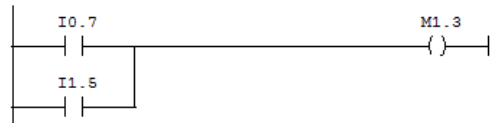
Network 5: First Floor Call



Network 6: Second Floor Call

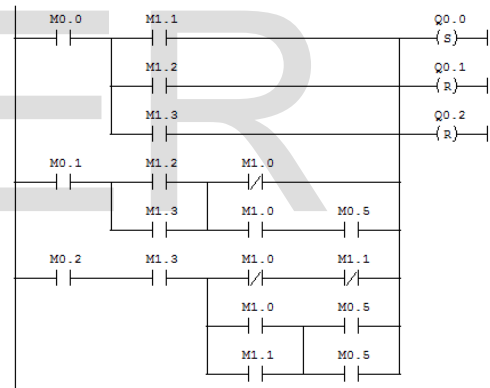


Network 7: Third Floor Call



Network 8: Fourth Floor Call

2.6.3 Motor Pulley Up and Down Movement Design



Network 9: Pulley up – Car Upward Direction

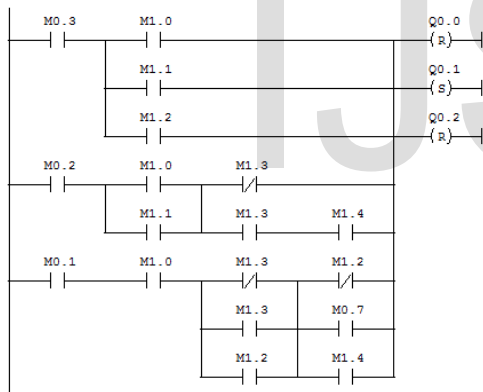
The ladder network above describes the situations in which a signal to move the pulley up is made (Q0.0=1), which are:

- If the car was currently residing at the **first floor** (M0.0=1) and a call for the second, third, or fourth floor (M1.1=1, M1.2=1, or M1.3=1) respectively is made.
- If the car was at the **second floor** (M0.1=1), AND a call for the third floor (M1.2=1) OR the fourth floor (M1.3=1), AND there was no call for the first floor (M1.0=0) OR if there was a call to the first floor (M1.0=1) BUT the last visited floor was the first

floor (M0.5) then the pulley has to move up for the sake of conserving power.

- If the car was at the **third floor** (M0.2=1), AND the fourth floor called for the car (M1.3=1), AND there was no call from either the first floor (M1.0=0) or the second floor (M1.1=0) the pulley goes up. OR if there was a call to the first floor (M1.0=1) OR the second floor (M1.1=1) BUT the last visited floor was the first floor (M0.5=1) OR the second floor (M0.6=1) then the car has to answer the request of the higher floors and continue moving up to reduce power consumption.

When the pulley up signal is set (Q0.0=1), the pulley down signal(Q0.1) and the motor stop signal(Q0.2) are reset (to 0) to ensure that the motor driving the pulley stops and reverse direction if the pulley was not at standstill position and was moving in the opposite direction.



Network 10: Pulley Down – Car Downward Direction

The ladder network above describes the situations in which a signal to move the pulley down is made (Q0.1=1), which are:

- If the car was currently residing at the **fourth** (M0.3=1) and a call for the first, second, OR third floor (M1.0=1, M1.1=1, or M1.2=1) respectively is made.
- If the car was at the **third floor** (M0.2=1), AND a call for the second floor (M1.1=1) OR the first floor (M1.0=1), AND there was no call for the fourth floor (M1.3=0) OR if there was a call to the fourth floor (M1.3=1)

BUT the last visited floor was the fourth floor (M1.4) then the pulley has to move down for the sake of conserving power.

- If the car was at the **second floor** (M0.1=1), AND the first floor called for the car (M1.0=1), AND there was no call from either the fourth floor (M1.3=0) or the third floor (M1.2=0) the pulley goes down. OR if there was a call to the fourth floor (M1.3=1) OR the third floor (M1.2=1) BUT the last visited floor was the fourth floor (M1.4=1) OR the third floor (M0.7=1) then the car has to answer the request of the first floor and continue moving up to reduce power consumption.

When the pulley up signal is set (Q0.0=1), the pulley down signal(Q0.1) and the motor stop signal(Q0.2) are reset (to 0) to ensure that the motor driving the pulley stops and reverse direction if the pulley was not at standstill position and was moving in the opposite direction.



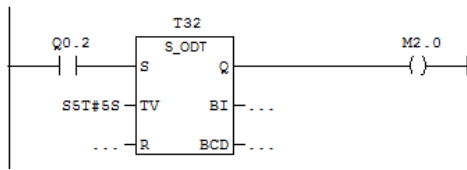
Network 11: Motor Stop Reset

The purpose of ladder network above is to reset the motor pulley up and pulley down signals whenever the motor stop signal is set; it is a locking mechanism to ensure that the motor does not move while the car is loading and unloading passengers.

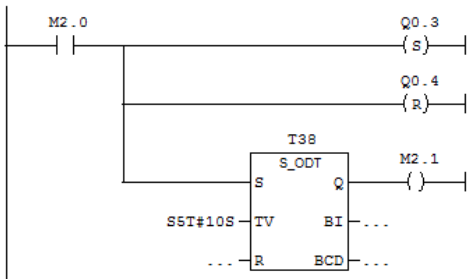
2.6.4 Door Automation Design

The networks in this section describe the door opening and closing actions. When the motor stop signal is set (Q0.2=1) a timer is set (T32) which counts 5 seconds then sets a signal (T32) after that. The need for this counter is first to ensure the car has come to stand still position since there will be car movement even after the motor stops for few seconds. This movement is taken into account and therefore the motor is signalled to stop once it passes by the limit switch which is fixed at a location few millimetres below the actual floor level. The time that takes the car to stop from the moment the motor is signalled to stop is

enough to make the car reach the actual floor level.

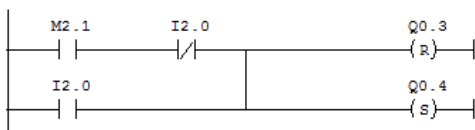


Network 12: Motor Stop/Door Open Time Delay



Network 12: Door Open Signal

After T32 start signalling (5 seconds after motor stop) it is made sure that the car is still and a signal to the door motor to run and open the door is made (Q0.3) another timer (T38) is set after that signal (that timer defines how long the door will remain open before a signal to close the door is made). When the door open signal is high (Q0.3=1), the door closing signal is reset (Q0.4=0) to ensure that the two signals are not sent simultaneously causing serious design and operation errors.



Network 13: Door Close Signal

If the timer defining the time period during which the door should remain open start signalling (10 seconds after the door open signal was set high) AND if the door open button was not pressed (I2.0=0) OR if the door close button was pushed (I2.1) even before the 10 seconds specified are over, the door close signal is sent to the door motor to start running on the opposite direction and close the car's door. Consequently, setting the door opening signal resets the door close signal and conversely, setting the door close signal resets the door open signal.

3 RESULTS AND DISCUSSIONS

To obtain results for this system in a readable way; several scenarios covering all potential usage of the system were tested in both software simulators and hardware implementation. Software testing, involves only test for sequential events, but concurrent events (events that typically occur on normal usage of the elevator) could not be tested by simulation since the simulator used cannot provide this feature. But both responses moving pulley up and down are tested using sequence events (test cases).

In hardware testing, scenarios for testing intelligent design are provided. Pictures of the results for each case test were taken and attached to these scenarios as shown in simulations test cases results.

3.1 Simulation Results

Two major simulation software packages are used in the test cases results, namely; PLCSIM and SCADA Simulators.

S7-PLCSIM is a SIMATIC Engineering tool for simulating any program implementation in Siemens STEP-7 environment. S7-PLCSIM can execute and test any STEP 7 user program in a simulated programmable logic controller (PLC).

Similarly, InTouch Wonderware SCADA software is a Schneider Electric product, a component of Wonderware Factory Suite which can create powerful, full-featured applications that exploit the key features of Microsoft Windows including ActiveX controls, OLE, graphics, networking and many more applications.

3.1.1 PLCSIM Simulation Results

At the start of the simulation module and the Ladder programme run mode, the following figure depicts the overview.

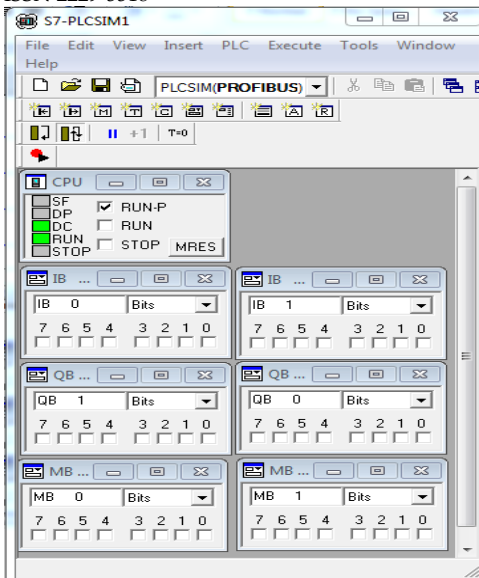


Figure 5 Initial overview at Start up

3.1.2 Test Case 1: Elevator at Floor 1

From figure 6 below, the ladder configuration shows the Floor 1 memory location is “SET”. Also from the PLCSIM module, the memory bit M0.0 and M0.5 confirmed that the elevator is at floor 1 and the last visit floor is floor 1 respectively.

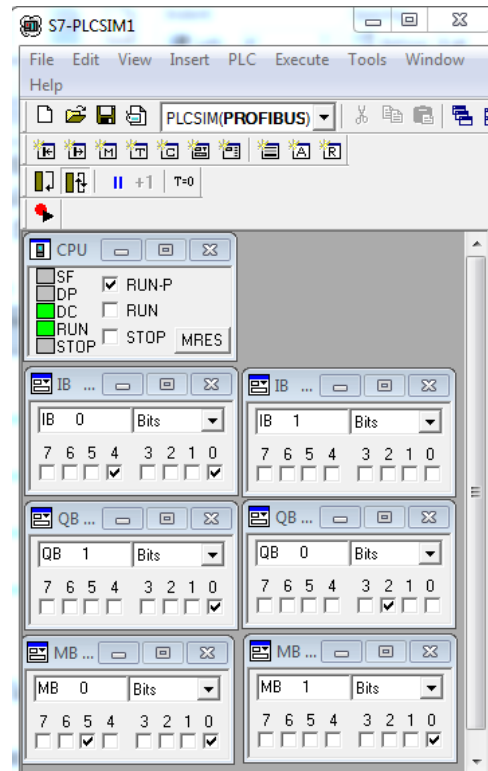
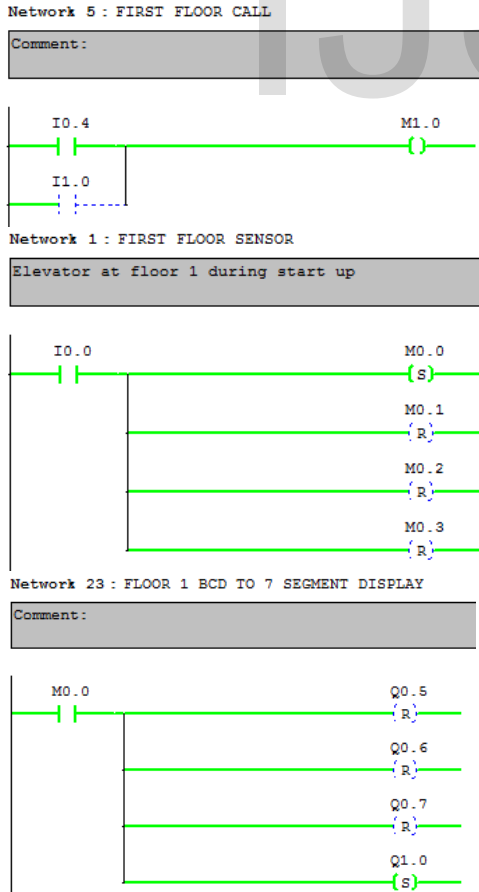
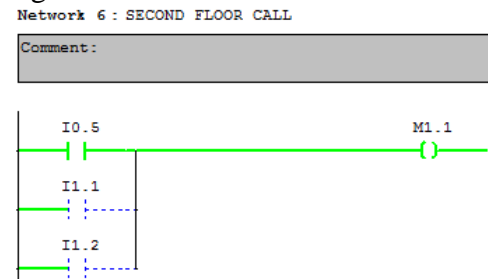


Figure 6 Elevator at Floor 1

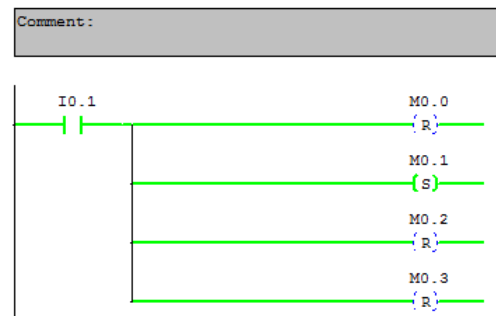
The car is at first floor and the output to the 7-segment is set to drive it to show decimal number [1] equivalent of [0001] binary. The output array shows that only Q1.0 is “high” and Q0.5, Q0.6, Q0.7 set “low”. The Floor 1 BCD-7 Segment Display is shown in figure 6 above.

3.1.3 Test Case 2: Moving from Floor 1 to Floor 2

While at the first floor, a call to the second floor occurred, which makes 2nd floor call input to be set high (M0.1 = SET), as shown in figure 7.



Network 2 : SECOND FLOOR SENSOR



Network 24 : FLOOR 2 BCD TO 7 SEGMENT DISPLAY

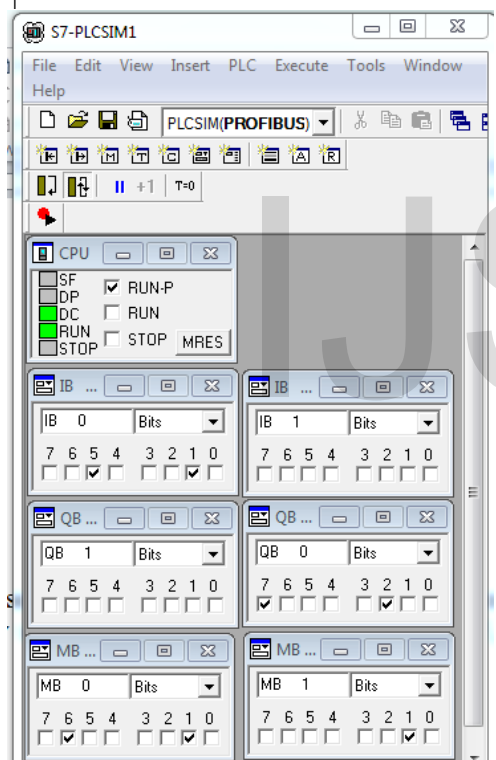
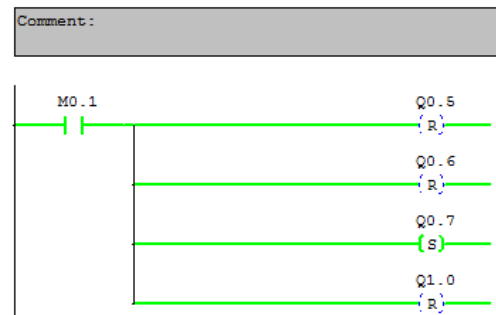


Figure 7 Floor 2 Call

The car is at second floor and the output to the 7-segment is set to drive it to show decimal number [2] equivalent of [0010] binary. The output array shows that only Q0.7 is “high” and Q0.5, Q0.6, Q1.0 set “low”. The Floor 2 BCD-7 Segment Display is shown in figure 7 above.

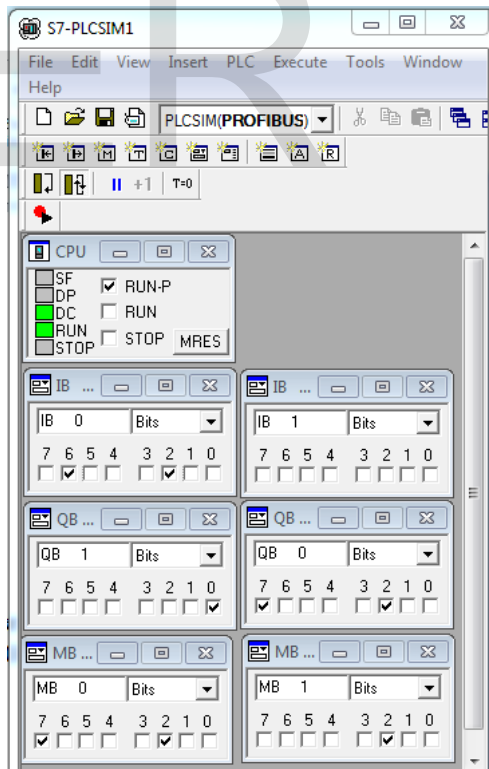
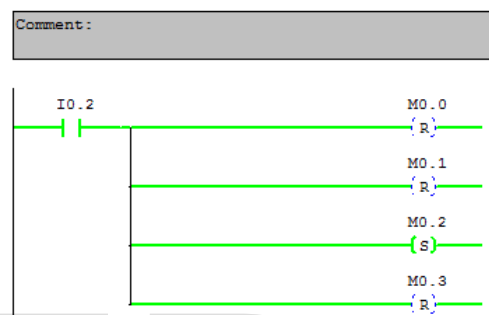
3.1.4 Test Case 3: Moving from Floor 2 to Floor 3

While at the second floor, a call to the third floor occurred, which makes 3rd floor call input to be set high (M0.2 = SET) as shown in figure 8.

Network 7 : THIRD FLOOR CALL



Network 3 : THIRD FLOOR SENSOR



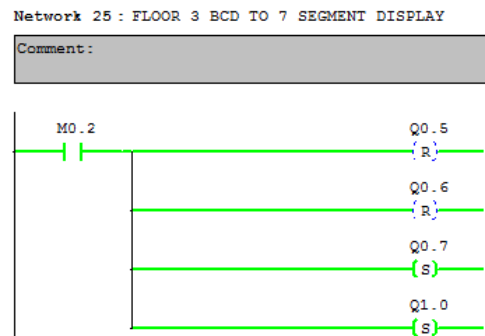


Figure 8: Floor 3 Call

The car is at third floor and the output to the 7-segment is set to drive it to show decimal number (3) equivalent of [0011] binary. The output array shows that Q0.7, Q1.0 set “high” and Q0.5, Q0.6, set “low”. The Floor 3 BCD-7 Segment Display is shown in figure 8 above.

3.1.5 Test Case 4: Moving from Floor 3 to Floor 4

While at the third floor, a call to the fourth floor occurred, which makes 4th floor call input to be set high (M0.3 = SET) as shown in figure 9.

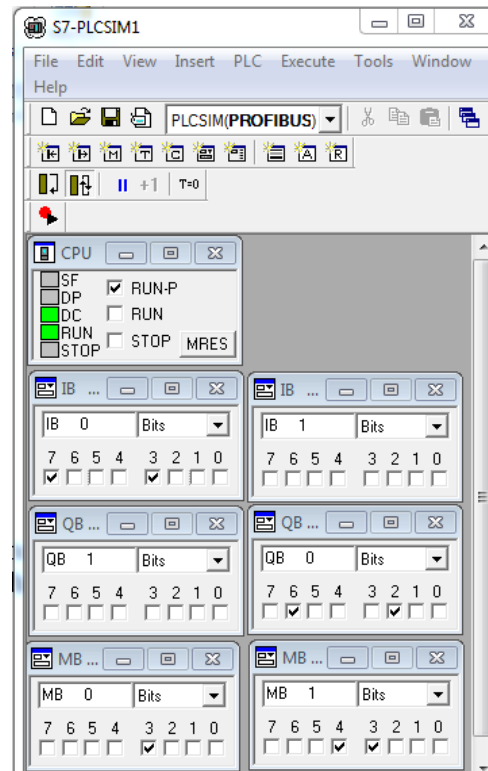
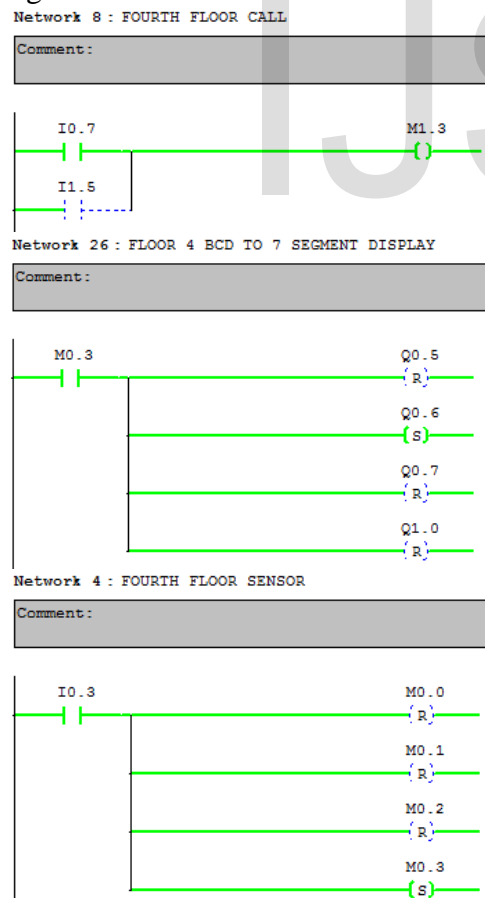


Figure 9: Floor 4 Call

The car is at fourth floor and the output to the 7-segment is set to drive it to show decimal number (4) equivalent of [0100] binary. The output array shows that only Q0.6, set “high” and Q0.5, Q0.6, Q1.0 set “low”. The Floor 4 BCD-7 Segment Display is shown in figure 9 above.

However, a lot of other simulations can be carried out depending on the scenario to be implemented.

3.2 Creating Window Script for SCADA Simulation

The simulation of the design was best done by creating window application script. The scripts were created from the Ladder Network described in sub-section 2.8 above.

The window scripts developed are divided into two parts: the “on show” and the “while showing”. The “on show” part is necessary to give the initialization condition while the “while showing” part implements the flow of the network.

The following window reports were generated for the “on show” and “while showing”.

Window Scripts for "Elevator Control":

```
Database entries used in "Elevator Control"
CB ELEVATOR CABIN
D1 FLOOR 1 DOOR FC1 FLOOR 1 CALL
D2 FLOOR 2 DOOR FC2 FLOOR 2 CALL
D3 FLOOR 3 DOOR FC3 FLOOR 3 CALL
D4 FLOOR 4 DOOR FC4 FLOOR 4 CALL
LS1 FLOOR 1 LIMIT SWITCH TM1 AUTO TIMER 1
LS2 FLOOR 2 LIMIT SWITCH TM1 AUTO TIMER 2
LS3 FLOOR 3 LIMIT SWITCH TM1 AUTO TIMER 3
LS4 FLOOR 4 LIMIT SWITCH TM1 AUTO TIMER 4
M05 FLOOR 1 VISIT MEMORY M10 MEMORY FOR FC1
M06 FLOOR 2 VISIT MEMORY M11 MEMORY FOR FC2
M07 FLOOR 3 VISIT MEMORY M12 MEMORY FOR FC3
M14 FLOOR 4 VISIT MEMORY M13 MEMORY FOR FC4
Q00 MOTOR OUTPUT FORWARD DIRECTION
Q01 MOTOR OUTPUT REVERSE DIRECTION
Q02 MOTOR STOP INTERLOCK OUTPUT
```

Script on show:

```
LS1=0;LS2=0;LS3=0;LS4=0;D1=0;D2=0;D3=0;D4=0;
FC1=0;FC2=0;FC3=0;FC4=0;
M05=0;M06=0;M07=0;M14=0;M10=0;M11=0;M12=0;M13=0;
Q00=0;Q01=0;Q02=0; CB=0; TM1=0; TM2=0; TM3=0; TM4=0;
```

Script while showing, every 2 msec:

```
IF FC1==1 THEN M10=1;ELSE M10=0;ENDIF;
IF FC2==1 THEN M11=1;ELSE M11=0;ENDIF;
IF FC3==1 THEN M12=1;ELSE M12=0;ENDIF;
IF FC4==1 THEN M13=1;ELSE M13=0;ENDIF;
IF CB==0 THEN LS1=1;ELSE LS1=0;ENDIF;
IF CB==80 THEN LS2=1;ELSE LS2=0;ENDIF;
IF CB==160 THEN LS3=1;ELSE LS3=0;ENDIF;
IF CB==240 THEN LS4=1;ELSE LS4=0;ENDIF;
IF CB==0 AND TM1<20 THEN TM1=TM1 +1;ENDIF;
IF CB==0 AND TM1==20 THEN D1=1;ENDIF;
IF CB==0 AND D1==1 AND TM1<100 THEN TM1=TM1 +1;ENDIF;
IF CB==0 AND TM1==100 THEN D1=0;ENDIF;
IF CB==0 THEN TM2=0; TM3=0; TM4=0;ENDIF;
IF CB==80 AND TM2<20 THEN TM2=TM2 +1;ENDIF;
IF CB==80 AND TM2==20 THEN D2=1;ENDIF;
IF CB==80 AND D2==1 AND TM2<100 THEN TM2=TM2 +1;ENDIF;
IF CB==80 AND TM2==100 THEN D2=0;ENDIF;
IF CB==80 THEN TM1=0; TM3=0; TM4=0;ENDIF;
IF CB==160 AND TM3<20 THEN TM3=TM3 +1;ENDIF;
IF CB==160 AND TM3==20 THEN D3=1;ENDIF;
IF CB==160 AND D3==1 AND TM3<100 THEN TM3=TM3 +1;ENDIF;
IF CB==160 AND TM3==100 THEN D3=0;ENDIF;
IF CB==160 THEN TM1=0; TM2=0; TM4=0;ENDIF;
IF CB==240 AND TM4<20 THEN TM4=TM4 +1;ENDIF;
IF CB==240 AND TM4==20 THEN D4=1;ENDIF;
IF CB==240 AND D4==1 AND TM4<100 THEN TM4=TM4 +1;ENDIF;
IF CB==240 AND TM4==100 THEN D4=0;ENDIF;
IF CB==240 THEN TM1=0; TM2=0; TM3=0;ENDIF;
IF LS1==1 AND FC1==1 OR LS2==1 AND FC2==1 OR LS3==1 AND FC3==1
OR LS4==1 AND FC4==1 THEN Q02=1;ELSE Q02=0;ENDIF;
IF FC1==1 AND CB>0 THEN CB=CB -10;ENDIF;
IF FC2==1 AND CB>80 THEN CB=CB -10;ENDIF;
IF FC3==1 AND CB>160 THEN CB=CB -10;ENDIF;
IF FC4==1 AND CB>240 THEN CB=CB +10;ENDIF;
IF FC3==1 AND CB<160 THEN CB=CB +10;ENDIF;
IF FC2==1 AND CB<80 THEN CB=CB +10;ENDIF;
```

```
IF LS1==1 AND Q02==1 THEN M05=1; M06=0; M07=0; M14=0; ELSE IF
LS2==1 AND Q02==1 THEN M06=1; M05=0; M07=0; M14=0; ELSE IF LS3==1
AND Q02==1 THEN M07=1; M05=0; M06=0; M14=0; ELSE IF LS4==1 AND
Q02==1 THEN M14=1; M05=0;M06=0;M07=0; ENDIF;ENDIF;ENDIF;
IF LS1==1 AND M11==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS1==1 AND M12==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS1==1 AND M13==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS2==1 AND M12==1 AND M10==0 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS2==1 AND M13==1 AND M10==0 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS2==1 AND M13==1 AND M10==1 AND M05==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS3==1 AND M13==1 AND M10==0 AND M11==0 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS3==1 AND M13==1 AND M10==1 AND M05==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS3==1 AND M13==1 AND M10==1 AND M06==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS3==1 AND M13==1 AND M11==1 AND M05==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS3==1 AND M13==1 AND M11==1 AND M06==1 THEN Q00=1;Q01=0;Q02=0;ENDIF;
IF LS4==1 AND M10==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS4==1 AND M11==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS4==1 AND M12==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS3==1 AND M10==1 AND M13==0 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS3==1 AND M11==1 AND M13==0 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS3==1 AND M10==1 AND M13==1 AND M14==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS3==1 AND M11==1 AND M13==1 AND M14==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M13==0 AND M12==0 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M13==0 AND M07==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M13==0 AND M14==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M13==1 AND M07==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M13==1 AND M14==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M12==1 AND M12==0 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M12==1 AND M07==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF LS2==1 AND M10==1 AND M12==1 AND M14==1 THEN Q01=1;Q00=0;Q02=0;ENDIF;
IF CB=0 AND Q02==1 OR CB=80 AND Q02==1 OR CB=160 AND Q02==1 OR CB=240
AND Q02==1 THEN Q00=0;Q01=0;ENDIF;
```

3.3 SCADA SIMULATION RESULTS

The PLC operation was first tested with the PLSIM software in the SIMATIC manager as described in sub section 3.1.1. The Network Logic parameters are now translated to the scripts programming in InTouch Wonderware.

The following figure shows simulation layout and results.

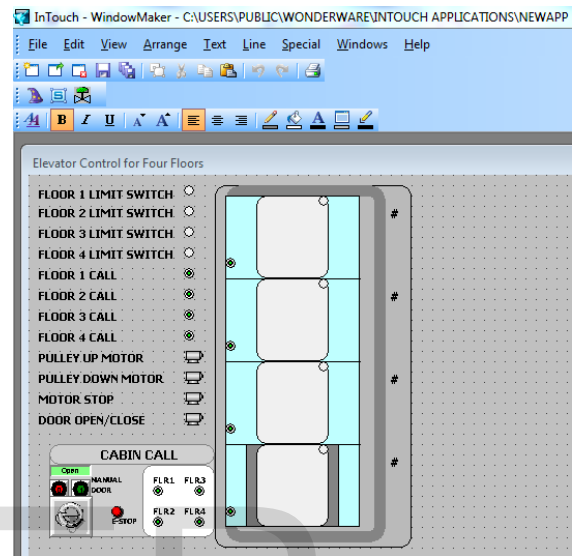


Figure 10 WindowMaker Design layout

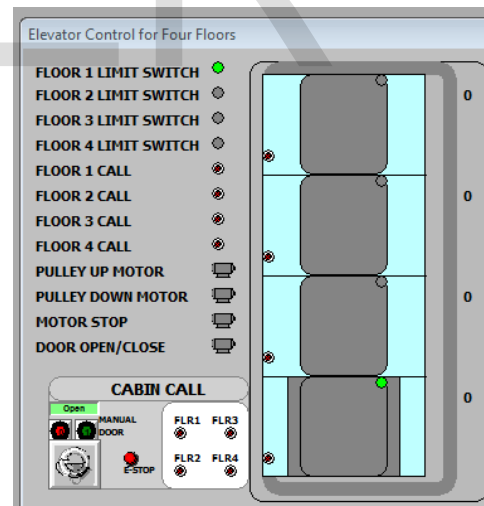


Figure 11 Simulation Start up Runtime Layout

3.3.1 Elevator Simulation status at each floor

The figures below show the status of the elevator at each floor. When the elevator is at any of the floors 1 to 4, the door is opened and remained opened for the set time (10s in this simulation). While the door remained opened, the motor stop signal is "set" and the door

animation signal is “set”. This is a safety measure to make sure the elevator is not set to move while the door is opened. Therefore, at any floor when the door is opened or opening, the motor pulley up and down are set at “0” while the motor stop and door animation signals are set at “1”.

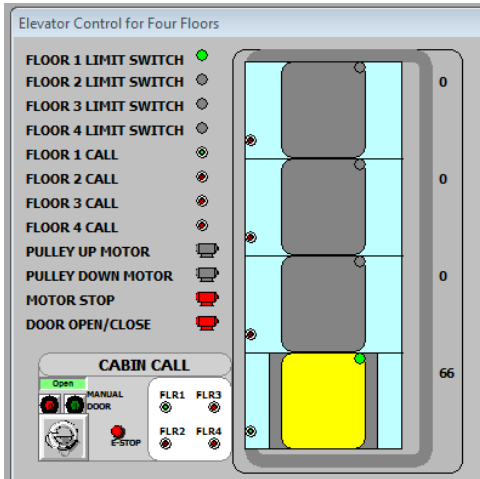


Figure 12 Elevator at floor 1

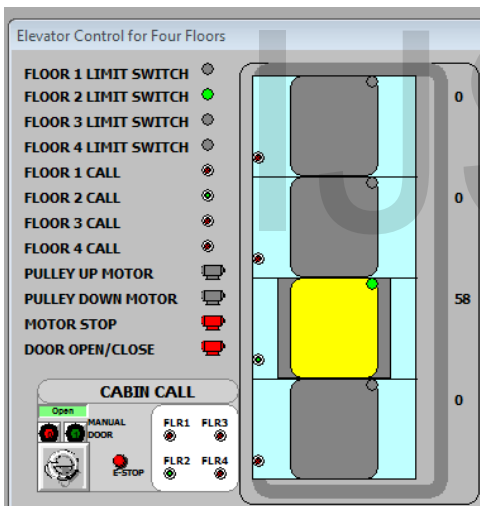


Figure 13 Elevator at floor 2

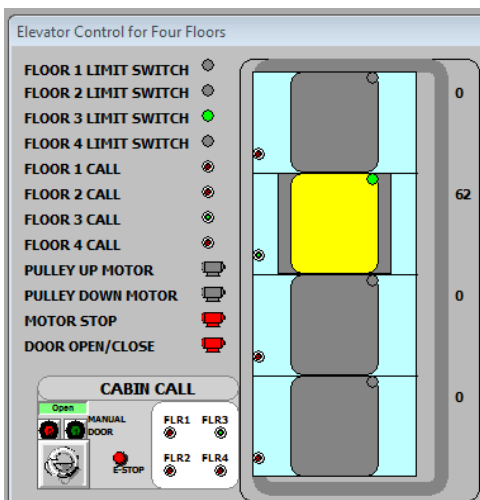


Figure 14 Elevator at floor 3

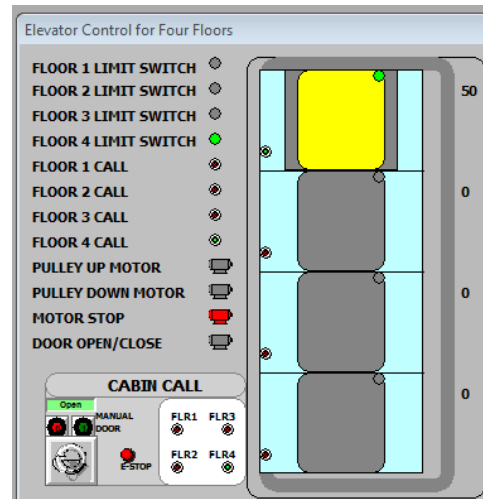


Figure 15 Elevator at floor 4

3.3.2 Elevator upward call to floors

When a call is initiated from a lower floor to the floor above it, as the elevator is moving up, the pulley up motor signal is set “high” while the pulley down motor, motor stop and door animation signals are set to “0”.

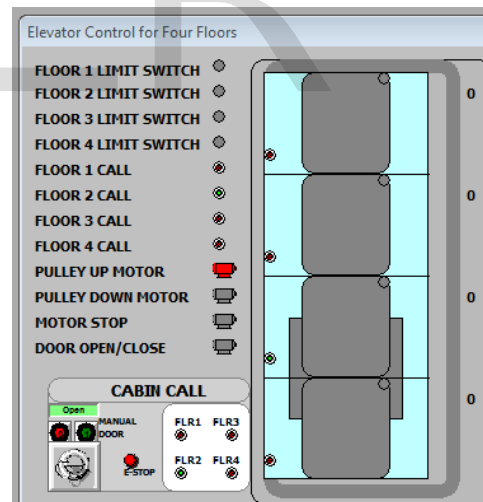


Figure 16 Elevator call from floor 1 to 2

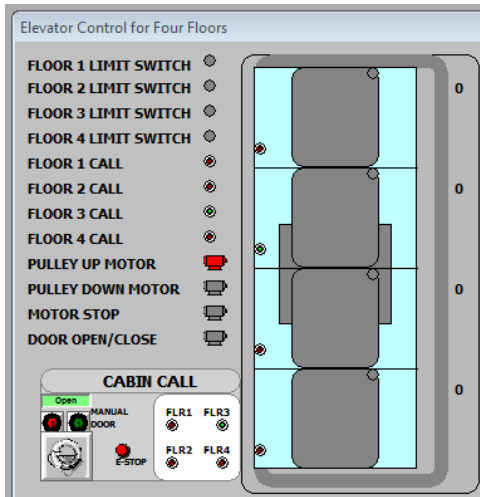


Figure 17 Elevator call from floor 2 to 3

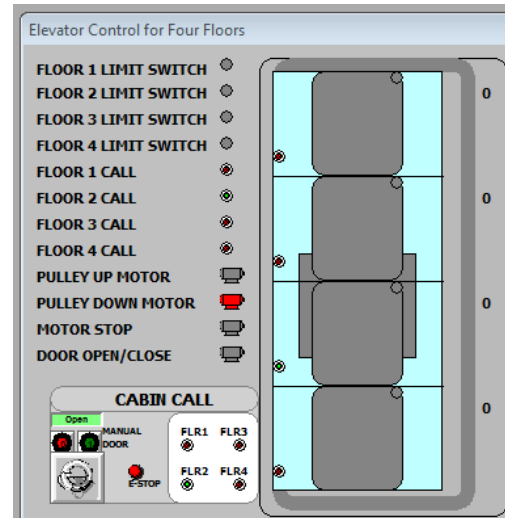


Figure 20 Elevator call from floor 3 to 2

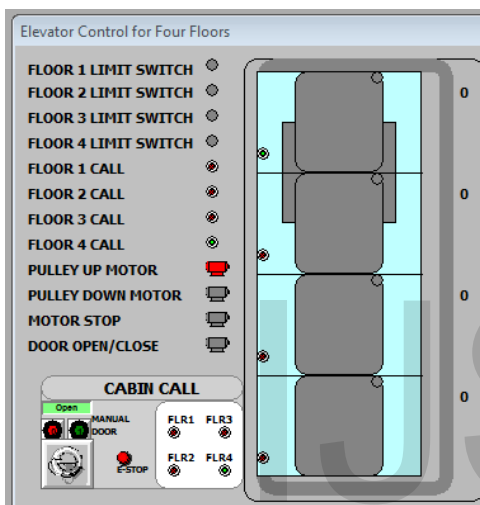


Figure 18 Elevator call from floor 3 to 4

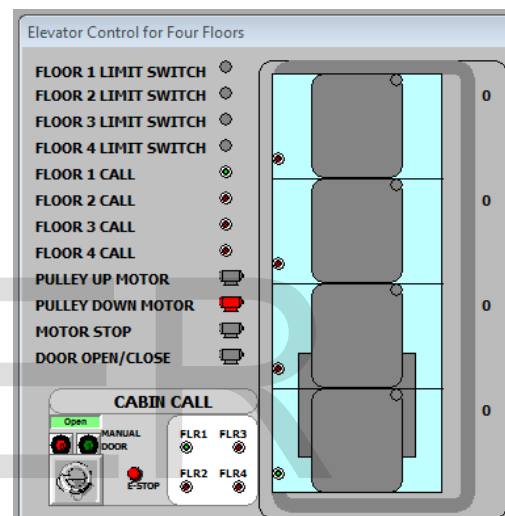


Figure 21 Elevator call from floor 2 to 1

3.3.3 Elevator downward call to floors

When a call is initiated from a higher floor to any floor below it, as the elevator is moving down, the pulley down motor signal is set “high” while the pulley up motor, motor stop and door animation signals are set to “0”.

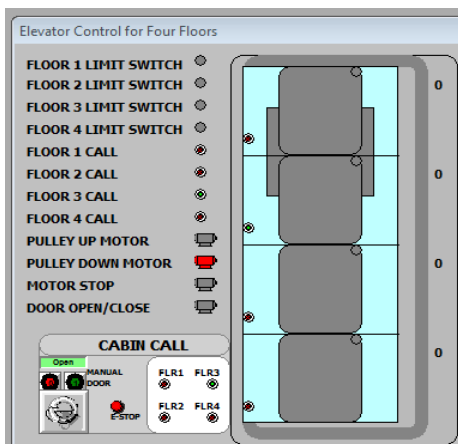


Figure 19 Elevator call from floor 4 to 3

4 CONCLUSION

This paper describes implementation for four floors elevator with SCADA system using S7-200 PLC. Also, sensors, pushbuttons, actuators connection and their configuration at different floors are effectively incorporated into the ladder logic network to implement the design simulation. The developed ladder logic has been implemented by Ladder diagram programming. The required inputs and outputs of elevator for moving upward and reverse, door opening and closing, and motors operation have been included in the logic and interpreted effectively through the simulation methods (PLCSIM and SCADA) implemented.

REFERENCES

- Batten, G.L., “*Programmable Controllers: Hardware, Software, and Applications*”, Second Edition, McGraw-Hill, 1994.
- Hugh, J., “*Automating Manufacturing Systems with PLCs*”, Free Software Foundation, Version 4.7, November 2005.
- Rohner, P., “*Automation with Programmable Logic Controllers*”, International Specialized Book Service, 1996.
- Stenerson, J., “*Fundamentals of Programmable Logic Controllers, Sensors and Communications*”, Prentice Hall, 1998.
- Darshil, S. and Sharma, S. A., (2008) “*Development of a PLC Based Elevator System with Colour Sensing Capabilities for Material Handling in Industrial Plant*”, Joint International Conference on Power System Technology and IEEE Power Conference , pp 1-7.
- Gieras, J.F. and Wing, M. (1993) “*Control of an elevator drive with a single-sided linear induction motor*”, Fifth European Conference on Power Electronics and Applications, IET Conference Publications, p 353-358.
- Hong, S. L. and Krishnan, R. (2007) “*Ropeless Elevator with Linear Switched Reluctance Motor Drive Actuation Systems*”, IEEE Transactions on Industrial Electronics, vol. 54, pp 2209-2218.
- Knezevic, B. and Marcetic, D. (2011) “*Model of elevator drive with jerk control*”, XXIII International Symposium on Information, Communication and Automation Technologies, IEEE Conference Publications, pp 1-5.
- Lewis, R.W., Antsaklis, P.J., “*Programming Industrial Control Systems Using IEC 1131-3 (IEE Control Engineering, No. 59)*”, Inspec/IEE, 1995.

<http://www.electrical-knowhow.com/2017/05/elevator-control-system.html>