BCIT SCHOOL OF ENERGY

Mechatronics and Robotics Engineering Technology

ROBT 4456:

PLC Applications

PROJECT REPORT:

ADVANCED FOUR-FLOOR ELEVATOR

Author:	<Enze Xu>
Date:	May 21, 2025

ACKNOWLEDGMENTS

I must express my thanks to the Mechatronics and Robotics faculty for their support and encouragement throughout the project development: Isaiah Regacho, Isaiah Regacho, and Isaiah Regacho. Completion of the project would have not been accomplished without the guidance from my instructor, Isaiah Regacho. I would also like to show appreciation for the Lab Instructor, Isaiah Regacho who was available to discuss PLC programming techniques outside of scheduled lab periods. Thanks to Isaiah Regacho for the report template...

ABSTRACT

This PLC project demonstrates the application of various PLC programming techniques. This project programs the lab bench mock elevator to behave as real elevator. To accomplish this, the project utilizes a finite state machine to control the process. Additional unique features were implemented which required advanced PLC programming topics such as Add-On Instructions, Networking, and HMI design...

PREFACE

The assignment of this project was provided in 9 distinct parts. Each part asks for additional functionality and must be completed sequentially. All 9 parts are complete and documented...

CONTENTS

FIGURES

DEFINITIONS

Table 0.1: List of Abbreviations

Abbreviation	Definition
PLC	Programmable Logic Controller
FPB	Floor Push Button
PB	Pushbutton
FLS	Floor Limit Switch
PBE5	Pushbutton Emergency
ILE5	Indicator Light Emergency
ONS	One Shot Instruction
FIL	Floor Indicator Light
IL	Indicator Light
CR	Control Relay
M	Motor
PBO	Pushbutton Open
PBC	Pushbutton Close
MCR	Master Control Reset

Table 0.2: List of Symbols

Symbol	Definition
V:	Virtual
!	Boolean operator: Logical NOT

1 INTRODUCTION

This document discusses the design and implementation of an Advanced Four-Floor Elevator PLC controller. As a PLC programming project, the hardware is provided by BCIT Mechatronics and Robotics.

1.1 PROJECT DESCRIPTION

The elevator project is defined in nine parts. Each part demands additional features on top of the previous parts. This report reflects the most advanced implementation.

1.2 PROJECT HARDWARE

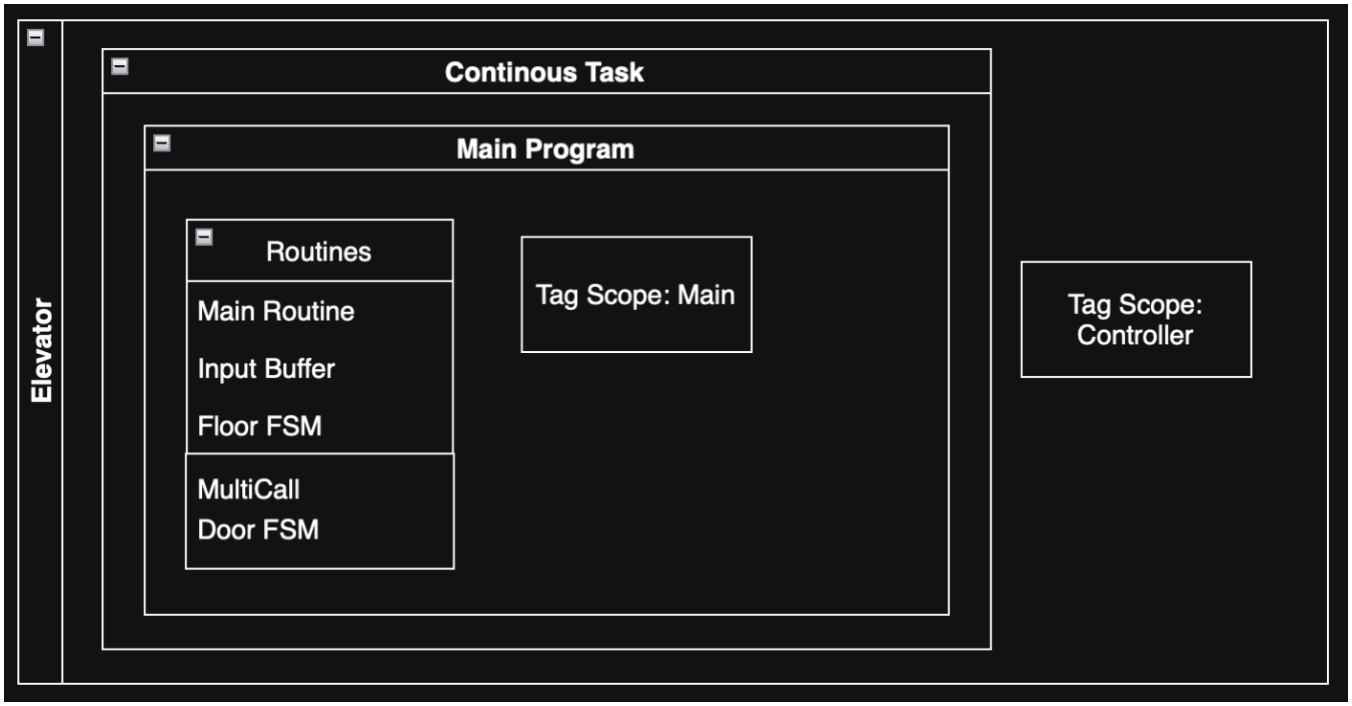
The hardware design and implementation is provided at the start of the project. The main PLC controller is an Allen-Bradley 1769 CompactLogix PLC. The I/O modules utilized are:

2 PROJECT OVERVIEW

The object of this project is to write a program that control a four-floor elevator. The program is to be written in 8 parts where each successive part is built upon the previous part. State Machine Method #3 should be used for every part. The parts should be completed sequentially as each added feature may introduce a new problem. Parts 7-9 can be added in any order after Part 6 is completed.
Each time a part is completed, save the program, and generate a .pdf for that part separately. For the final demo, be ready to present the programs saved. If the part demonstrated does not function adequately, the previous parts will be demonstrated until a working part is found.

2.1 STUDIO 5000 PROJECT ORGANIZATION

The PLC program is organized to be modular.



2.2 CONTROLLER TAGS

Table 2.1: Controller Tags

Tag	Type	Task
Local:2:I.Data	DINT	InputBuffer
Local:3:I.Data	DINT	InputBuffer

Local:4:O.Data	DINT	OutputBuffer
Local:5:O.Data	DINT	OutputBuffer

2.3 Ladder Logic Structure

Each program adheres to an Input Buffer, Logic, and Output Buffer structure. The input buffer routine of a program ensures logical inputs are not change during a single scan. The output buffer only updates after the rest of the program has been scanned.

Each subroutine is responsible for a dedicated set of tags which may not be written to in other subroutines.

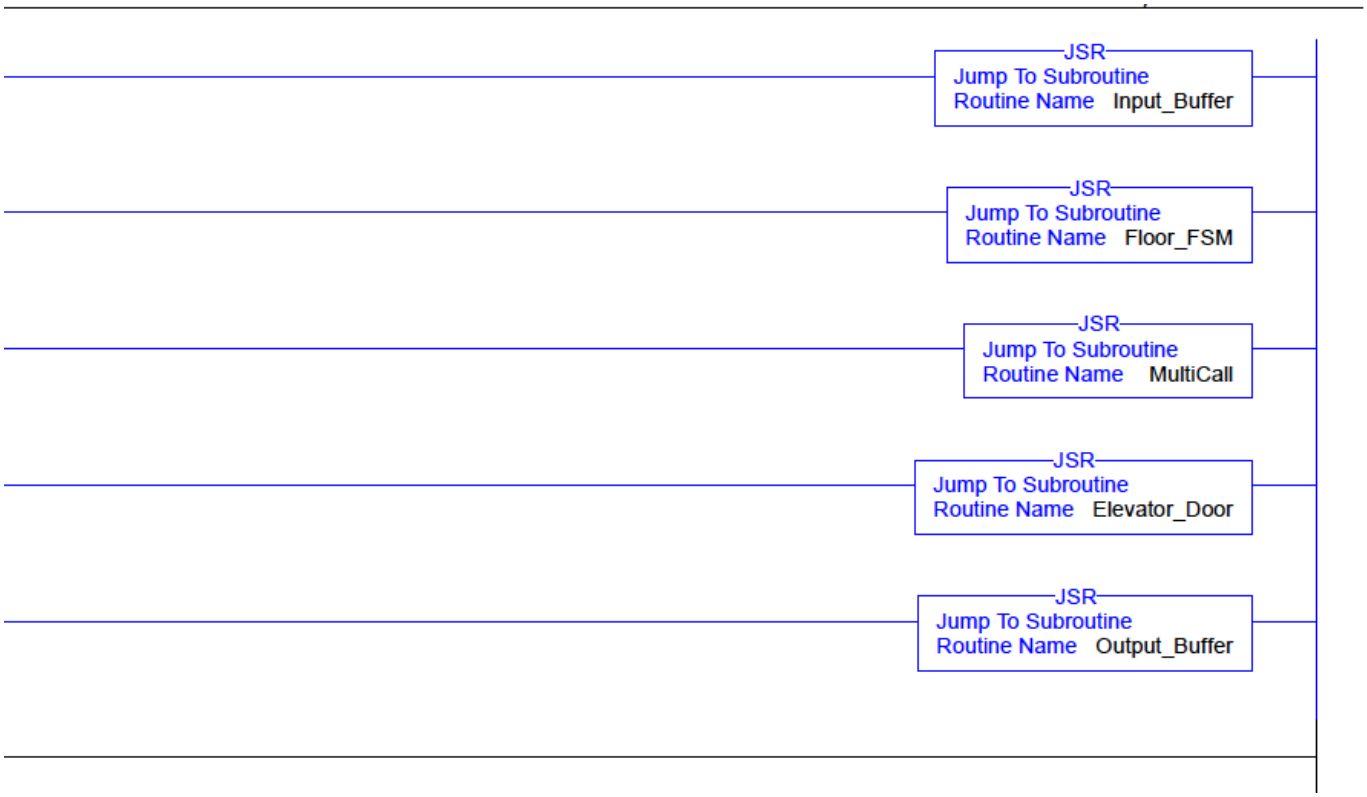


Figure 1 - Main Routine Structure

The main routine is programmed to only calls subroutines. Each subroutine starts with a SBR and end with a RET instruction. Floor_FSM is the subroutine controls the elevator movement, MultiCall indicates and stores any pending calls, Elevator_Door is a separate state machine, controls the door opening period and automatic open/close.

3 Main Program

The elevator operation is managed by a state machine. This state machine determines logical movement of the elevator by driving the motor and its direction.

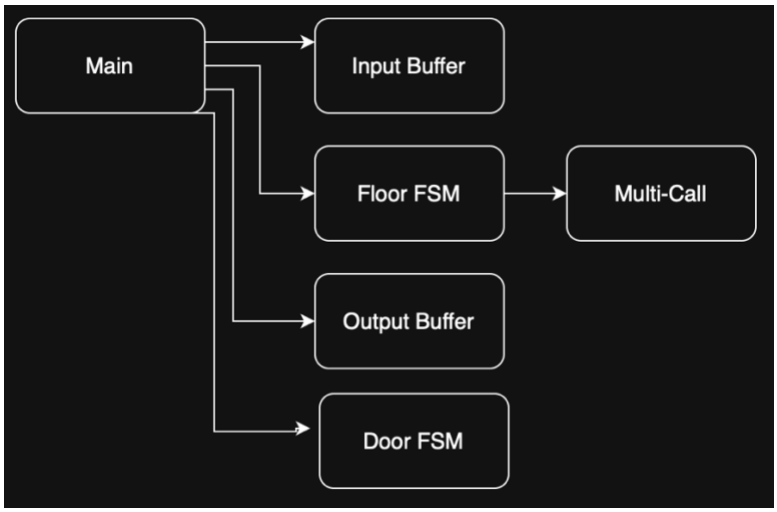


Figure 3.1: This diagram shows where each subroutine is called.

3.1 Program Tags

These tags are external to the Main program. Access to these tags is managed by the input and output buffer subroutines.

Table 3.1: Controller Tag Buffering

Tag	Mapping	Type	Subroutine
V_2I_Data	Local:2:I.Data	DINT	InputBuffer
V_3I_Data	Local:3:I.Data	DINT	InputBuffer
V_4O_Data	Local:4:O.Data	DINT	OutputBuffer
V_5O_Data	Local:5:O.Data	DINT	OutputBuffer

Table 3.2: Input Module 2 – InputBuffer Tags

Tag	Alias	Type	Description
V_PB4	V_2I_Data.0	BOOL	V: Pushbutton 4
V_PB3	V_2I_Data.1	BOOL	V: Pushbutton 3
V_PB2	V_2I_Data.2	BOOL	V: Pushbutton 2

V_PB1	V_2I_Data.3	BOOL	V: Pushbutton 1
V_PBO	V_2I_Data.4	BOOL	V: Pushbutton Open
V_PBC	V_2I_Data.5	BOOL	V: Pushbutton Close
V_PBE5	V_2I_Data.6	BOOL	V: Pushbutton Emergency 5
V_FPB4	V_2I_Data.8	BOOL	V: Floor Pushbutton 4
V_FPB3	V_2I_Data.9	BOOL	V: Floor Pushbutton 3
V_FPB2	V_2I_Data.10	BOOL	V: Floor Pushbutton 2
V_FPB1	V_2I_Data.11	BOOL	V: Floor Pushbutton 1
V_START	V_2I_Data.14	BOOL	V: START (N.O.)
V_STOP	V_2I_Data.15	BOOL	V: STOP (N.C.)

Table 3.3: Input Module 3 – InputBuffer Tags

Tag	**Alias**	**Type**	**Description**
V_FLS1	V_3I_Data.0	BOOL	V: Floor Limit Switch 1
V_FLS2	V_3I_Data.1	BOOL	V: Floor Limit Switch 2
V_FLS3	V_3I_Data.2	BOOL	V: Floor Limit Switch 3
V_FLS4	V_3I_Data.3	BOOL	V: Floor Limit Switch 4
V_DCLS	V_3I_Data.8	BOOL	V: Door Close Limit Switch
V_DOLS	V_3I_Data.9	BOOL	V: Door Open Limit Switch
V_TS2	V_3I_Data.10	BOOL	V: Toggle Switch 2
V_TS1	V_3I_Data.11	BOOL	V: Toggle Switch 1
V_TS0	V_3I_Data.12	BOOL	V: Toggle Switch 0

Table 3.4: Output Module 4 – OutputBuffer Tags

Tag	Alias	Type	Description
V_IL4	V_4O_Data.0	BOOL	V: Indicator Light 4
V_IL3	V_4O_Data.1	BOOL	V: Indicator Light 3
V_IL2	V_4O_Data.2	BOOL	V: Indicator Light 2
V_IL1	V_4O_Data.3	BOOL	V: Indicator Light 1
V_ILO	V_4O_Data.4	BOOL	V: Indicator Light Open
V_ILC	V_4O_Data.5	BOOL	V: Indicator Light Close
V_ILE5	V_4O_Data.6	BOOL	V: Indicator Light Emergency 5
V_FIL4	V_4O_Data.8	BOOL	V: Floor Indicator Light 4
V_FIL3	V_4O_Data.9	BOOL	V: Floor Indicator Light 3
V_FIL2	V_4O_Data.10	BOOL	V: Floor Indicator Light 2
V_FIL1	V_4O_Data.11	BOOL	V: Floor Indicator

3.2 ELEVATORSTATEMACHINE

This subroutine is responsible for the main elevator operation. The elevator actuators are only controlled by the logic within this subroutine and is also the only subroutine which may call the door state machine subroutine. The logic within this subroutine implements a finite state machine.

Table 3.5: ElevatorStateMachine Tags

Tag	Alias	Description	Type	Read By
See Tags.PDF	See Tags.PDF	See Tags.PDF	See Tags.PDF	See Tags.PDF

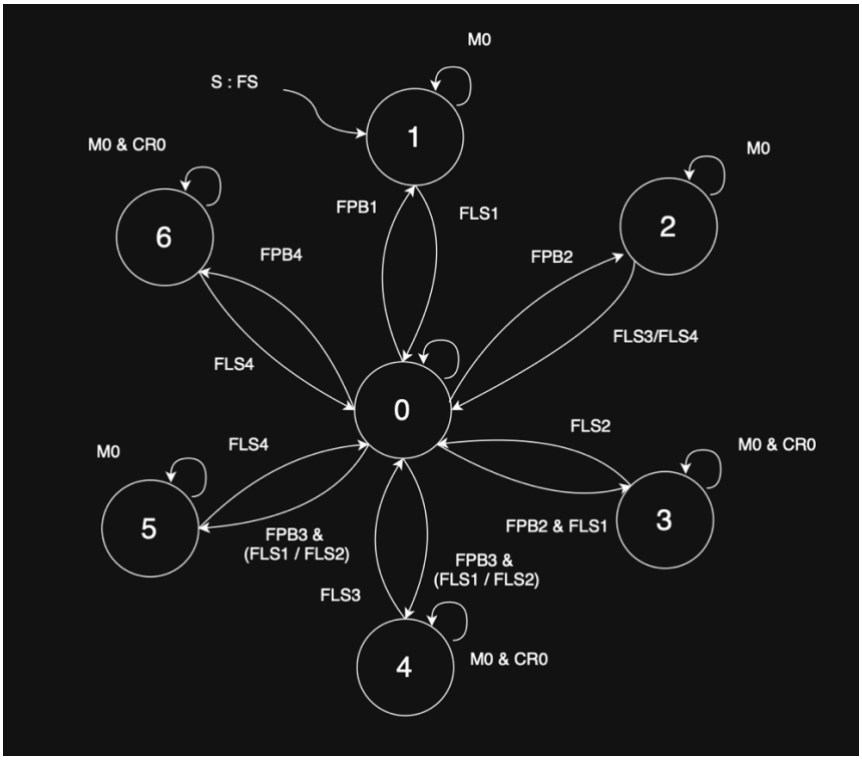


Figure 3 - Floor FSM for Outside Panel FPBx

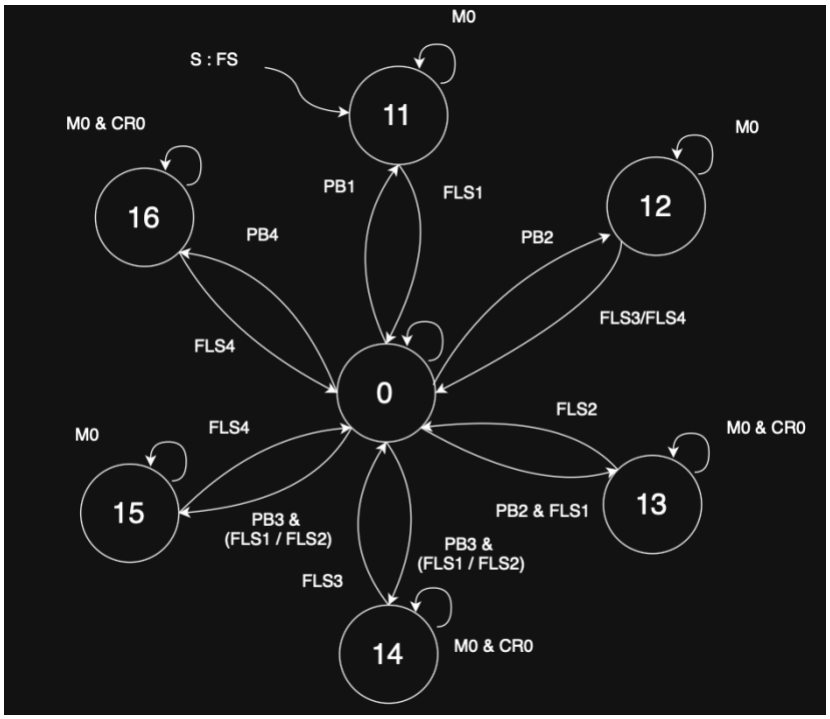


Figure 4 - Floor FSM for Inside Panel PBx

These two figures show a complete state machine diagram for both outside and inside panel floor request, and the elevator actuator behavior for any four-floors. Each state has unique tag value, avoid conflicts in state transition. Specifically, based on the pushbutton input, the elevator will actuate the motor M0, and decide the direction it's going by turning on/off the control relay 0 module CR0. The decision is based on the current floor sensed by the FLS sensor.

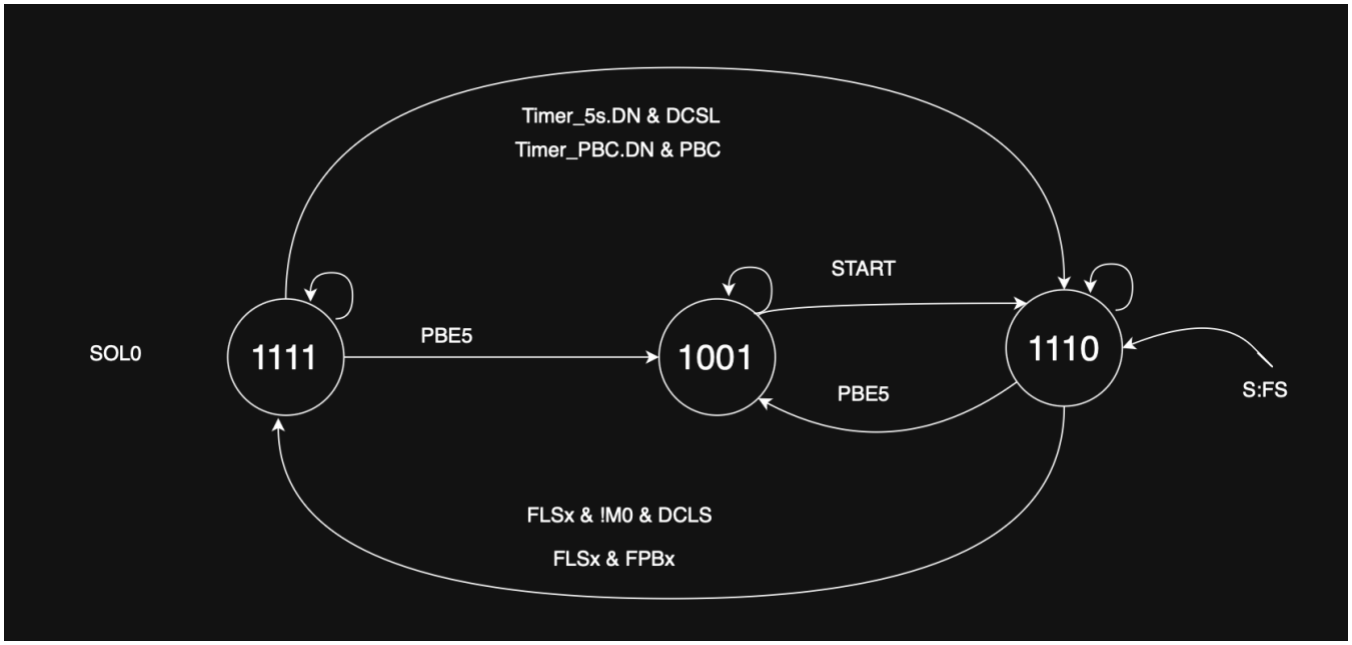


Figure 5 - Door State Machine

This door state machine is programmed to work separately with the floor FSM, an emergency state 1001 could be triggered by pressing PBE5 pushbutton. The door is expected to remain open during this state. The normal opening and closed state are triggered either by timer or the FPBx button.

The state machine forces the elevator to behave a specific way. Modifying the transitions can modify the overall behaviour of the system.

Table 3.6: Floor FSM State Descriptions

State	Entry Points	Exit Points	Actions	Logic
0: Idle	No condition	Entry Points (1-6)	No actions.	START: CTU(Counter_1)

1: Floor 1	FPB1 First Scan	FLS1	Turn on M0 !CR0	FPB1: OTE (M0) S:FS : OTE (M0)
2: Floor 2	FPB2 FLS3/FLS4	FLS2	Turn on M0 !CR0	FPB2 & (FLS3/FLS4) : OTE (M0)
3: Floor 2	FPB2 & FLS1	FLS2	Turn on M0 CR0	FPB2 & FSL1 : OTE (M0, CR0)
4: Floor 3	FPB3 & (FLS1/FLS2)	FLS3	Turn on M0 CR0	FPB3 & (FLS1/FLS2) : OTE (M0, CR0)
5: Floor 3	FPB3 & FLS4	FLS3	Turn on M0 !CR0	FPB3 & FLS4 : OTE (M0)
6: Floor 4	FPB4	FLS4	Turn on M0 CR0	FPB4: OTE (M0,CR0)
11: Floor 1	PB1 First Scan	FLS1	Turn on M0 !CR0	PB1: OTE (M0) S:FS : OTE (M0)
12: Floor 2	PB2 FLS3/FLS4	FLS2	Turn on M0 !CR0	PB2 & (FLS3/FLS4) : OTE (M0)
13: Floor 2	PB2 & FLS1	FLS2	Turn on M0 CR0	PB2 & FSL1 : OTE (M0, CR0)
14: Floor 3	PB3 & (FLS1/FLS2)	FLS3	Turn on M0 CR0	PB3 & (FLS1/FLS2) : OTE (M0, CR0)
15: Floor 3	PB3 & FLS4	FLS3	Turn on M0 !CR0	PB3 & FLS4 : OTE (M0)
16: Floor 4	PB4	FLS4	Turn on M0 CR0	PB4: OTE (M0,CR0)

Table 3.7 - Door FSM State Description

State	Entry Points	Exit Points	Actions	Logic
1110: Door Close	First Scan START Timer_5s.DN & DCLS Timer_PBC.DN & PBC	PBE5 FLSx & !M0 & DCLS FLS & FPBx	No actions.	IDLE
1111: Door Open	FLSx & !M0 & DCLS FLS & FPBx	PBE5	Turn on SOL0	PBE5: OTE (SOL0)
1001: Door Emergency	PBE5	START	Turn on SOL0	START & FLSx: OTE (SOL0)

3.3 Call and Request Manager

Passengers press buttons at a floor to call the elevator to that floor. Once inside the elevator, the passenger may request a floor to be dropped off at. This subroutine accepts the calls and requests and generates the next floor the elevator will go to.

A floor call condition is used through to drive the transition between states of the elevator. Whichever FPBx or PBx button is pressed, the corresponding floor-call condition is activated and latched, until the elevator reached the requested. This method is useful for multi-call/requests feature as it stores the user requests.

See MultiCall PDF [Page.3] for the ladder diagram

3.4 Door Routine

This routine manages the door controls and opens and closes while the elevator is servicing floors.
Requirements:

- The elevator door opens for 5 seconds when servicing a floor.
- After 2 seconds, the door can be closed by pressing PBC on the inside panel.
- Pressing PBO should hold the door open for another 3 seconds.

- If the elevator is waiting for a floor call, the door remains closed.
- Program the door with a separate state machine.

This routine uses a TON TIMER_5s to ensure the door will be closed after opening for 5 seconds. And another timer TON TIMER_PBC to time for 2 seconds after the door opens, useful to prevent the door opening when the elevator has just arrived at a floor. PBO will change the TIMER_5s.ACC to 2 seconds every time it's been pressed.

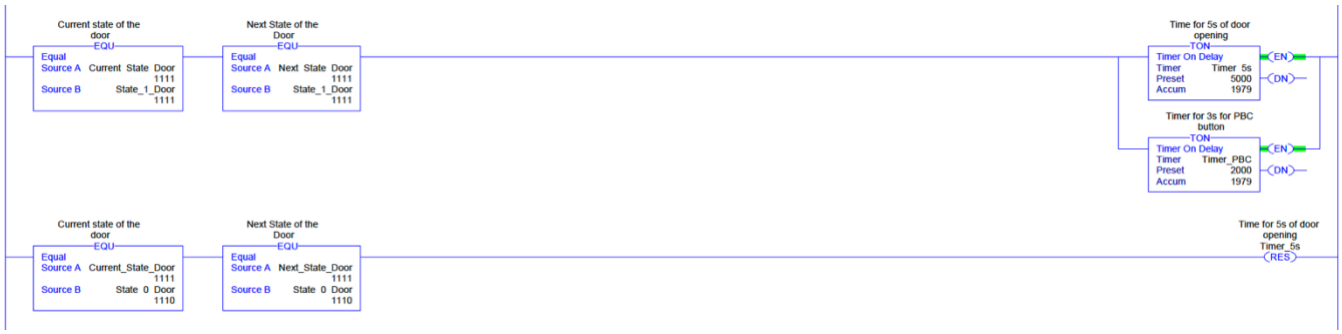


Figure 6 - Ladder Logic of Door Operation

Table 3.8 - Door FSM Tag Description

Tag/State Name	Description
Timer_5s	5 seconds of door opening
Timer_PBC	Enable PBC after 2 seconds of door opening
State_1_Door	Door open state
State_0_Door	Door close state

See Door PDF [Page.2] for the ladder diagram.

3.5 EMERGENCY STOP

This routine stops the operation of the elevator when an emergency happens.
Requirements:

- Pressing the emergency stop button PBE5 should stop the operation of the elevator.
- An indicator light ILE5 indicates the elevator has been stopped due to an emergency.
- The elevator should not respond to calls or requests.
- If an emergency occurs with the door open, the door remains open. The door can be opened and must remain open if the elevator is at a floor.

- A reset pushbutton, START, must be pressed once it is safe.
- After a reset, the elevator returns to the 1st floor with the door closed.

Once the PBE5 button is pressed, the elevator stops any current on-going operation, all outputs are de-energized. This is achieved by adding two MCR instructions in the Floor FSM, that will evaluate any rung as false during the Emergency Stop, except for the rung containing the START button, which will be pressed to initialize the elevator when it's safe.
Once the START is pressed, the elevator state will transition to State_1_IN, returns to the 1st floor with the door closed.

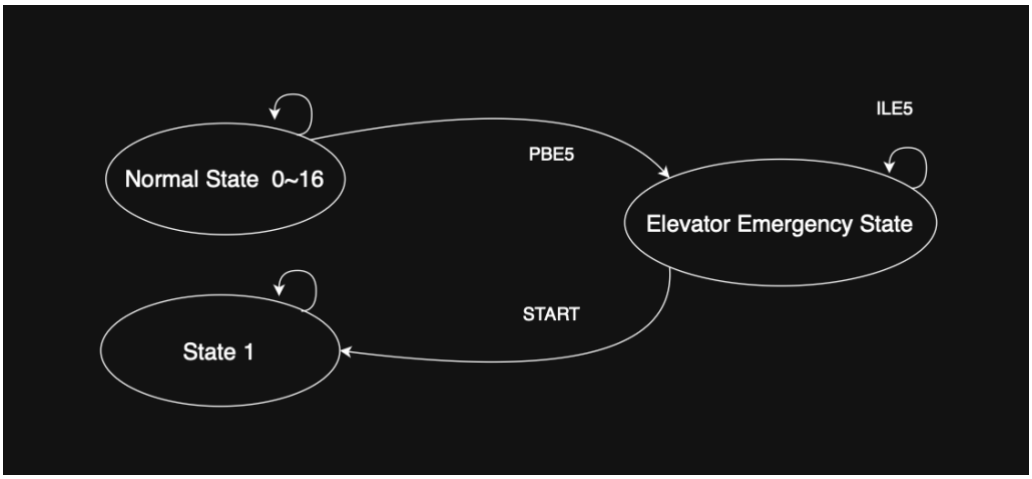


Figure 7 - Emergency Stop State Machine

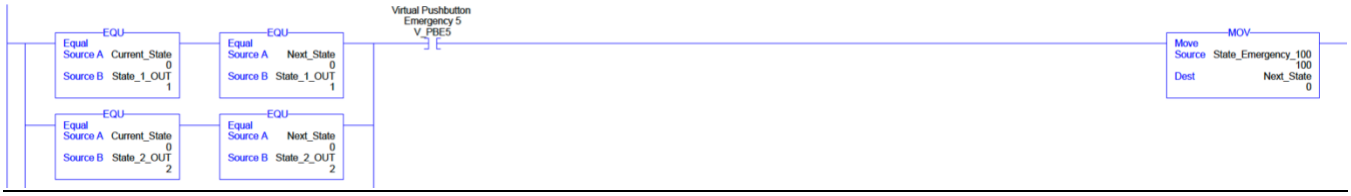


Figure 8 – Ladder Logic of Emergency Stop

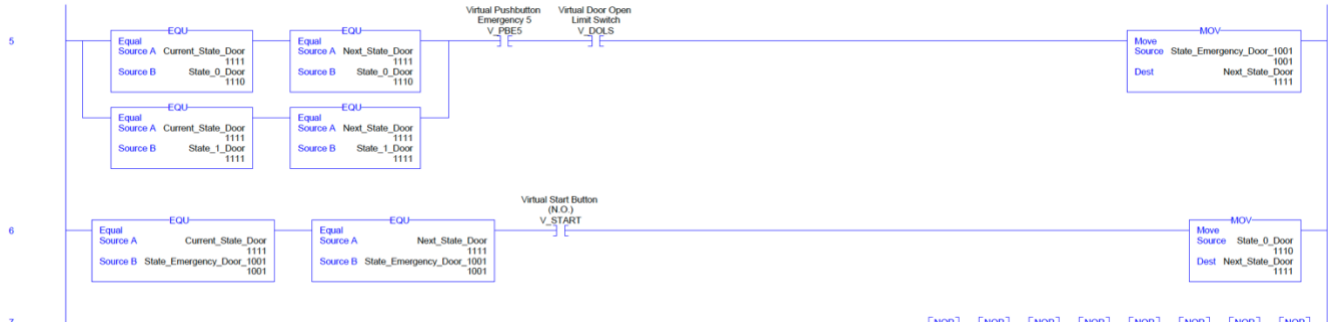


Figure 9 - Ladder Logic of Door Emergency State

Table 3.9 - Door FSM Emergency Tag/State Description

Tag/State Name	Description
State_Emergency_100	Elevator Emergency, stop Operation.
State_Emergency_Door_1001	Door Emergency, keep open
State_1_Door	Door open state
State_0_Door	Door close state

See Floor FSM [Page.1] for Ladder Diagram

3.5 THREE FLOOR ELEVATOR

Program the PLC elevator to respond to calls made from floors 1-3.

Requirements:

- Once run, the elevator should move to the Floor 1 and wait.
- The elevator can be called to a corresponding floor with FPB1, FPB2, and FPB3.
- FIL1, FIL2, and FIL3 should notify the user that the call is accepted.
- The elevator should stop at the floor it was called to and wait for the next call.
- The elevator never moves below Floor 1 or above Floor 3.

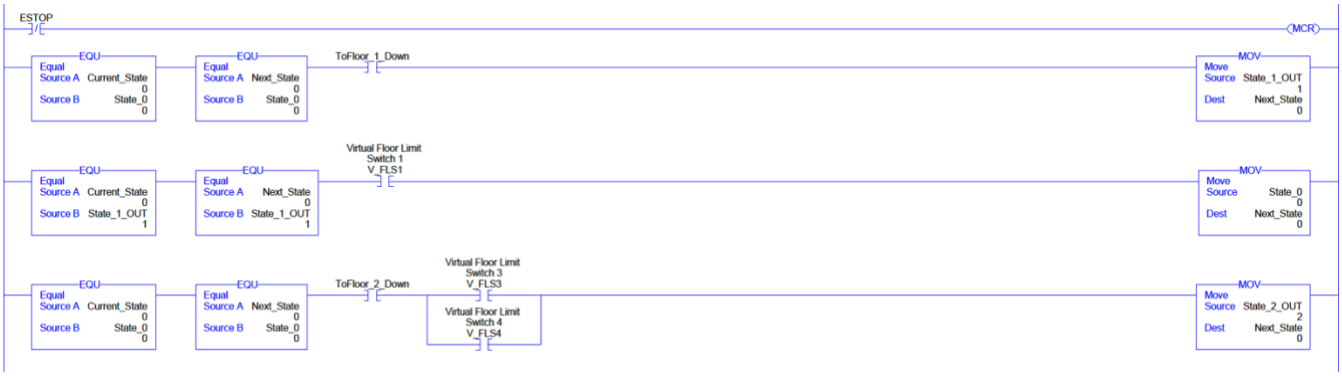


Figure 10 - Ladder logic of three floor elevator

This example shows how the state transition happens between different floor calls. All state machines are programed in Method 3.

Table 4.0 – Three floor FSM Tag/State Description

Tag/State Name	Description
ToFloor_1_Down	Elevator going to floor 1_Down direction
ToFloor_2_Down	Elevator going to floor 2_Down direction
State_0	Elevator Idle, waiting for user request
State_1_OUT	Elevator state 1: going down to floor 1, request from outside
State_2_OUT	Elevator state 2: going down to floor 2, request from outside

3.6 INSIDE PANEL

Program the PLC elevator to respond to requests to move to one of three floors.
Requirements:

- Passengers can request to go to a corresponding floor with PB1, PB2, and PB3.
- IL1, IL2, and IL3 should notify the passenger the request is accepted.

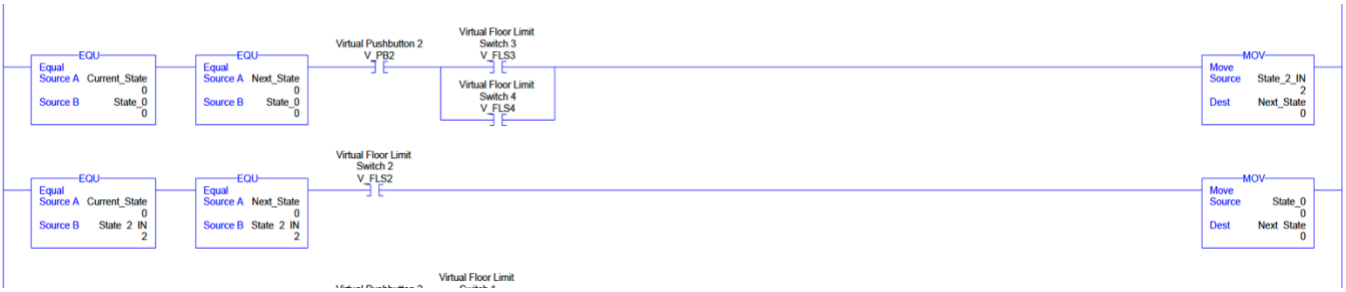


Figure 11 - Ladder logic of inside panel operation.

This example shows the use of inside panel for floor request.

Table 4.1 – Inside Panel FSM Tag/State Description

Tag/State Name	Description
State_1_IN	Elevator state 1: going down to floor 1, request from inside
State_2_IN	Elevator state 2: going down to floor 2, request from inside

3.7 FOUR FLOOR ELEVATOR

Program the PLC elevator to service all four floors.

- The elevator must be called to floor 4 with FPB4 / PB4
- FIL4 / IL4 notice the passenger the request is accepted.

Figure 12 - Ladder logic of four floor elevator operation.

Based on the program the three-floor elevator, floor 4 is added to current state machine. This introduced a change on 2nd and 3rd floor sate transition. The request on both of these two floors have to consider the direction it's going. State_5_OUT/IN, State_6_OUT/IN were added for this modification.

Table 4.2 – Four floor FSM Tag/State Description

Tag/State Name	Description
State_5_OUT	Elevator state 1: going down to floor 3, request from outside
State_6_OUT	Elevator state 2: going up to floor 4, request from outside
ToFloor_3_Down	Elevator going to floor 3_Down direction
ToFloor_3_Up	Elevator going to floor 3_Up direction
ToFloor_4_Up	Elevator going to floor 4_Up direction

3.8 MULTI-CALL/REQUESTS

Program the PLC the response to multiple calls or requests of any four-floor

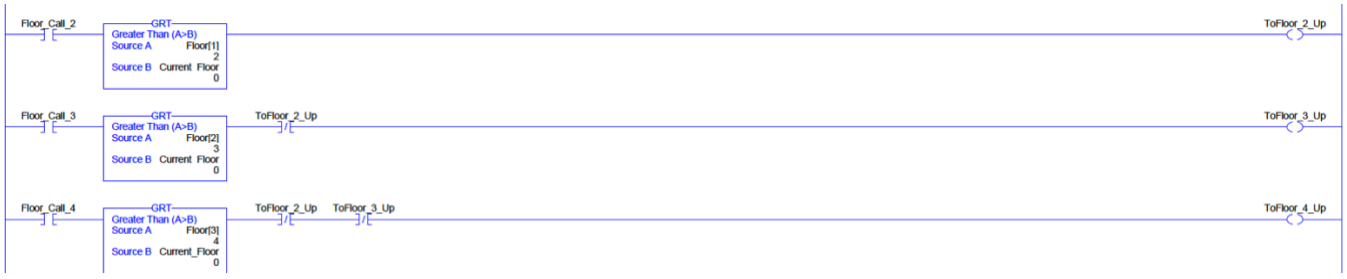


Figure 13 - Ladder logic of multi-call/requests operation.

As stated above, the Floor_Call_x will be latched until it's reached the requested floor. It will compare the current floor number with the one that's requested, energizing any output condition based on the comparing result. The elevator will prioritize the request floor on its current moving direction. This example shows a multi-call structure in UP Mode.

Table 4.3 – MultiCall/Request FSM Tag/State Description

Tag/State Name	Description
Floor_Call_1/2/3/4	Floor request to 1/2/3/4 floor (button pressed)
ToFloor_2/3/4_Up	Elevator going to floor 2/3/4_Up direction
ToFloor_1/2/3_Down	Elevator going to floor 1/2/3_Down direction

4 PROGRAM TAGS

Program executes within the Continuous task. This program perform control of all the actuator and read pushbutton input.

4.1 TAGS

See PDF Tags.

5 CONCLUSION

The project is successful, major issue resides in the MultiCall subroutine, where it causes the elevator ignoring the intermediate floor when it has multiple calls requested. The elevator also memorise the previous floor calls after it's being reset by START button. The possible causes were possibly raised by the structure of the FSM, which programs each floor as a state, so it could not change the "next-state" when it's transitioning to the one called before.

Improvement should be made on modifying state machine structure, which only advances the state by comparing the difference between the current floor and the request.

6 Appendix: Project Deliverables (Handout)

7 Ladder Logic PDF

List the program PDFs and where to find relevant ladder logic.

Input Buffer – Input_Buffer.pdf

Output Buffer – Output_Buffer.pdf

Three Floor Elevator – Floor_FSM.pdf [page 1-9]

Inside Panel - Floor_FSM.pdf [page 9-13]

Door Routine – Door.pdf

MultiCall/Request – MultiCal.pdf

Emergency Stop – Floor_FSM.pdf [page 1, page 13]