A 4-STORIES BUILDING ELEVATOR CONTROL SYSTEM

By

NEMAT HASSAN AHMED EL-HASSAN

INDEX NO. 064086

Supervisor

—Dr. Abdurrahman Ali Karrar

A REPORT SUBMITTED TO

University of Khartoum

In partial fulfillment of the requirements for the degree of

B.Sc. (HONS.) Electrical and Electronic Engineering

(ELECTRONICS ENGINEERING)

Faculty of Engineering

Department of Electrical and Electronics Engineering

July 2011

DEDICATION

To my beloved mother and father

My brothers and sisters for their inspiration, support and encouragement

My friends who made it easy being my self

To everyone helped making my life better in any way.

ACKNOWLEDGMENT

First of all, I would like to thank GOD for blessing me and giving me good health.

Thanks to Dr.A.A.Karrar, my supervisor, for his dedication in guiding me to complete this project and sharing his valuable time to educate me.

Thanks to my parents for being supporters and critics, for their pray and hope, thanks to all my family and friends for their support.

Thanks to Mr. Khalid Mohammed and Mr. Abbas, this project couldn't come so far without there precious help.

Thanks to Hadeel Omer Dafa'allah, it was a pleasure with her.

ABSTRACT

The purpose of this project is to design, program and build "prototype" 4-story building elevator control using PLC. The PLC used is S7200 with 14 inputs and 10 outputs. The elevator should operate in a power saving mode; therefore an intelligent design is used. The elevator control system is programmed in a ladder diagram format into the PLC. The operation of the programmed PLC is first tested in a computer environment using a simulation program "MICROwin" before wiring the PLC to the hardware "prototype". Some suggestions on how to extend this system to the control of more than four floors are also included.

المُسْتَخْلَصْ:

الهدف من المشروع هو تصميم برنامج و نموذَج لِنظام تحكُّم في مِصْعد لمبنى مُكوَّن من أربعَة طوابِقْ,و ذلك باستخدام المُتَحكِّمات المَنْطِقِيَّة القابِلة للبرمجة. المتحكمة المُستخدمة هي(سيمنز إس سفن) و تحتوي على 14اشارة دخل و 10 اشارات خرج. المصعد يجب ان يعمل في وضع حِفْظ الطَاقة, ولِذلك تَم استِخْدام التصميم الذَكي. تمت برمجة نظام التحكم الذكي المكتوب بلغة السلم في المتحكمة.

تم اختبار عمل المتحكمة أولاً ببرنامج محاكاة (مايكرو وِن) و ذلك قبل ان تثبت التوصيلات الكهربائية في النموذج الفعلي, ثم أُخْتُبِر عمليا في النموذج. في نهاية هذه الأطروحة تمت كتابة اقتراحات لتطوير و تحسين النظام.

Table of contents

Table of figures

CHAPTER I

Introduction

1.1 Overview

An elevator is a vertical transport equipment that efficiently moves people or goods between floors (levels, decks) of a building, vessel or other structure. Elevators are generally powered by electric motors that either drive traction cables or counterweight systems like a hoist, or pump hydraulic fluid to raise a cylindrical piston like a jack. [1]

The first reference to an elevator is in the works of the Roman architect Vitruvius, who reported that Archimedes (c. 287 BC – c. 212 BC) built his first elevator probably in 236 BC. In some literary sources of later historical periods, elevators were mentioned as cabs on a hemp rope and powered by hand or by animals. It is supposed that elevators of this type were installed in the Sinai monastery of Egypt. [1]

In 1000, the Book of Secrets by Ibn Khalaf al-Muradi in Islamic Spain described the use of an elevator-like lifting device, in order to raise a large battering ram to destroy a fortress. In the 17th century the prototypes of elevators were located in the palace buildings of England and France. [1]

Ancient and medieval elevators used drive systems based on hoists or winders. The invention of a system based on the screw drive was perhaps the most important step in elevator technology since ancient times, leading to the creation of modern passenger elevators. [1]

In 1853, American inventor Elisha Otis demonstrated a freight elevator equipped with a safety device to prevent falling in case a supporting cable should break. This increased public confidence in such devices. In 1846, Sir William Armstrong introduced the hydraulic crane, and in the early 1870s, hydraulic machines began to replace the steam-powered elevator. The hydraulic elevator is supported by a heavy piston, moving in a cylinder, and operated by the water (or oil) pressure produced by pumps.

Electric elevators came into to use toward the end of the 19th century. The first one was built by the German inventor Werner von Siemens in 1880. With the motor mounted at the bottom of the cab, this design employed a gearing scheme to climb shaft walls fitted with racks.

1.2 Problem Statement

So the aim of this project was to design an elevator control algorithm for 4-strorey building using PLC and build a prototype model with motorized door for demonstration.

1.3 Objective

The main objective of this project is to develop an intelligent motorized elevator system that is controlled by PLC and a set of push buttons and proximity switches. To accomplish this aim the following works are carried out:

- Development of control algorithm using ladder language.
- Program the code into the PLC.
- Development of the hardware to build a prototype.
- Test the system on the prototype.

1.4 Scope

The developed software is to be tested on the prototype. The elevator car has to go up and down based on calls and to determine whether to go up and down in a power saving manner.

1.5 Thesis Layout

This thesis illustrates the design principles of elevator systems in both software and hardware aspects.

Chapter 1 includes brief report of the history of elevators and there importance, and the steps taken to carry out the work of this project.

In chapter 2, the basic design principles of the two major types of elevator and the safety systems used in elevators are described, also description of PLCs, their uses and benefits over other controlling devices in this application is provided.

Chapter 3 is the Methodology part; it contains detailed description of the control algorithm and the hardware design the prototype.

Chapter 4 demonstrates the results obtained from simulating the control system, and results obtained from running the system on the prototype.

Lastly, Chapter 5 contains all the difficulties that faced this project and the alternatives used to overcome them, it also includes ideas to enhance both the model and the software design.

CHAPTER II

BACKGROUND (LITERATURE REVIEW)

2.1 Overview

The concept of an elevator is simple -- it's just a compartment attached to a lifting system. Tie a piece of rope to a box, and you've got a basic elevator.

Of course, modern passenger and cargo elevators are a lot more elaborate than this. They need advanced mechanical systems to handle the substantial weight of the elevator car and its cargo. Additionally, they need control mechanisms so passengers can operate the elevator, and they need safety devices to keep everything running smoothly. [2]

In this chapter, besides the concept of operation of elevators, types of designs, safety systems and elevator's sets of doors are reviewed. Also intelligent control, software and hardware components - including PLCs and their operation and programming- are reviewed.

2.2 Design

There are two major elevator designs in common use today: hydraulic elevators and roped elevators.

2.2.1 Hydraulic Elevators

Hydraulic elevator systems lift a car using a hydraulic ram, a fluid-driven piston mounted inside a cylinder. The cylinder is connected to a fluid-pumping system (typically, hydraulic systems like this use oil, but other incompressible fluids would also work). The hydraulic system has three parts:

- A tank (the fluid reservoir)
- A pump, powered by an electric motor

❖ A valve between the cylinder and the reservoir

The pump forces fluid from the tank into a pipe leading to the cylinder. When the valve is opened, the pressurized fluid will take the path of least resistance and return to the fluid reservoir. But when the valve is closed, the pressurized fluid has nowhere to go except into the cylinder. As the Hydraulic elevator system fluid collects in the cylinder, it pushes the piston up, lifting the car.

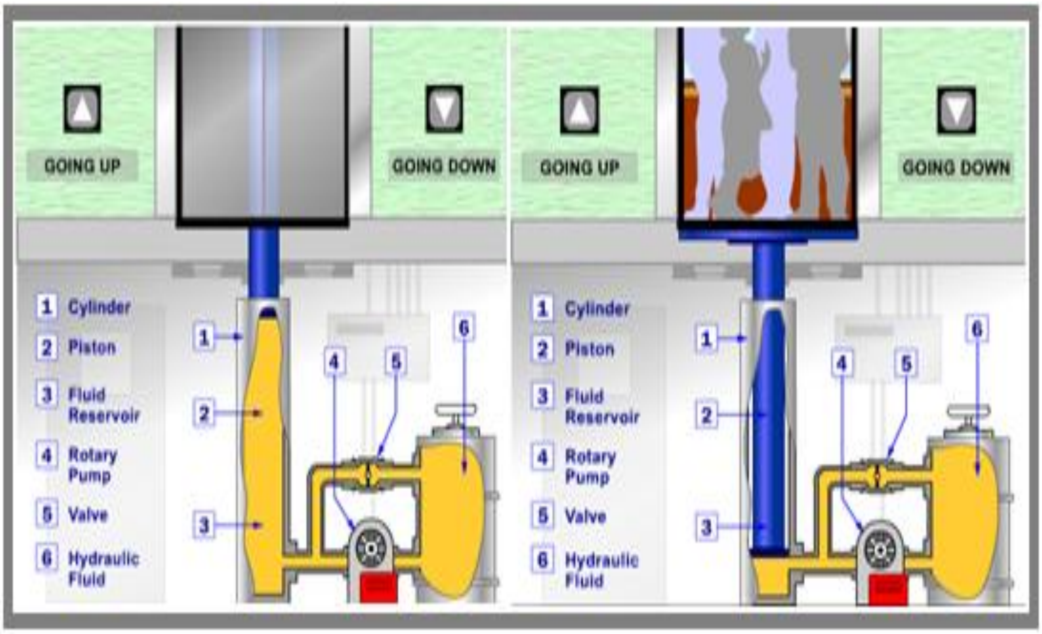


Figure 2:1 Hydraulic elevator system

When the car approaches the correct floor, the control system sends a signal to the electric motor to gradually shut off the pump. With the pump off, there is no more fluid flowing into the cylinder, but the fluid that is already in the cylinder cannot escape (it can't flow backward through the pump, and the valve is still closed). The piston rests on the fluid, and the car stays where it is.

To lower the car, the elevator control system sends a signal to the valve. The valve is operated electrically by a basic solenoid switch. When the solenoid opens the valve, the fluid that has collected in the cylinder can flow out into the fluid reservoir. The weight of the car and the cargo pushes down on the piston, which drives the fluid into the reservoir. The car gradually descends. To stop the car at a lower floor, the control system closes the valve again [3].

Pros and Cons of Hydraulics

The main advantage of hydraulic systems is that they can easily multiply the relatively weak force of the pump to generate the stronger force needed to lift the elevator car.

The hydraulic systems suffer from two major disadvantages:

The main problem is the size of the equipment. In order for the elevator car to be able to reach higher floors, you have to make the piston longer. The cylinder has to be a little bit longer than the piston, of course, since the piston needs to be able to collapse all the way when the car is at the bottom floor.

The problem is that the entire cylinder structure must be buried below the bottom elevator stop. This means you have to dig deeper as you build higher. This is an expensive project with buildings over a few stories tall. To install a hydraulic elevator in a 10-story building, for example, you would need to dig at least nine stories deep!

The other disadvantage of hydraulic elevators is that they're fairly inefficient. It takes a lot of energy to raise an elevator car several stories, and in a standard hydraulic elevator, there is no way to store this energy. The energy of position (potential energy) only works to push the fluid back into the reservoir. To raise the elevator car again, the hydraulic system has to generate the energy all over again [3].

2.2.2 The Cable System

The most popular elevator design is the roped elevator. In roped elevators; the car is raised and lowered by traction steel ropes rather than pushed from below [3]. The traction to raise and lower the car comes from the friction of the wire ropes against the grooved sheaves. The main sheave (3) is driven by an electric motor.

In a typical elevator, the car is raised and lowered by six to eight motor-driven wire ropes. [3]The ropes are attached to the elevator car, and looped around a sheave (3). A sheave is just a pulley with grooves around the circumference. The sheave grips the hoist ropes, so when you rotate the sheave, the ropes move too. The sheave is connected to an electric motor (2).When the motor turns one way, the sheave raises the elevator; when the motor turns the other way, the sheave lowers the elevator. In gearless elevators, the motor rotates the sheaves directly. In geared elevators, the motor turns a gear train that rotates the sheave. Typically, the sheave, the motor and the control system (1) are all housed in a machine room above the elevator shaft. The ropes that lift the car are also connected to a counterweight (4), which hangs on the other side of the sheave. [2] The counterweight adds accelerating force when the elevator car is ascending and provides a retarding effort when the car is descending so that less motor horsepower is required. A set of chains are looped from the bottom of the counterweight to the underside of the car to help maintain balance by offsetting the weight of the suspension ropes.

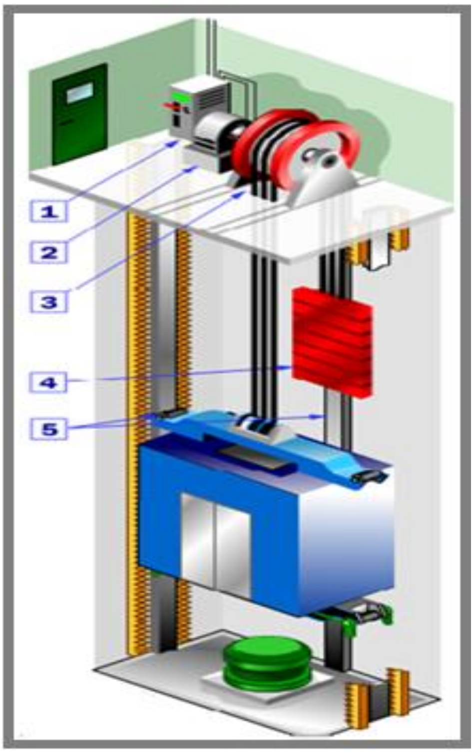


Figure 2:2 Cable system

The counterweight weighs about the same as the car filled to 40% capacity. In other words, when the car is 40% full (an average amount), the counterweight and the car are perfectly balanced.

The purpose of this balance is to conserve energy. With equal loads on each side of the sheave, it only takes a little bit of force to tip the balance one way or the other. Basically, the motor only has to overcome friction -- the weight on the other side does most of the work. To put it another way, the balance maintains a near constant potential energy level in the system as a whole. Using up the potential energy in the elevator car (letting it descend to the ground) builds up the potential energy in the weight (the weight rise to the top of the shaft). The same thing happens in reverse when the elevator goes up.

Both the elevator car and the counterweight ride on guide rails (**5**) along the sides of the elevator shaft. The rails keep the car and counterweight from swaying back and forth, and they also work with the safety system to stop the car in an emergency. [2] Rollers are attached to the car and the counterweight to provide smooth travel along the guide rails.

Roped elevators are much more versatile than hydraulic elevators, as well as more efficient. Typically, they also have more safety systems. [3]

Most elevators use a direct current motor because its speed can be precisely controlled to allow smooth acceleration and deceleration. Motor-generator (M-G) sets typically provide to dc power for the drive motor. Newer systems use a static drive control. The elevator controls vary the motor's speed based on a set of feedback signals that indicate the car's position in the shaftway. As the car approaches its destination, a switch near the landing signals the controls to stop the car at floor level. Additional shaftway limit switches are installed to monitor over travel conditions. [2]

2.2.3 Safety Systems

Elevators are built with several redundant safety systems that keep them in position. The first line of defense is the rope system itself. Each elevator rope is made from several lengths of steel material wound around one another. With this sturdy structure, one rope can support the weight of the elevator car and the counterweight on its own. But elevators are built with multiple ropes (between four and eight, typically). In the unlikely event that one of the ropes snaps, the rest will hold the elevator up.

Even if all of the ropes were to break, or the sheave system was to release them, it is unlikely that an elevator car would fall to the bottom of the shaft. Roped elevator cars have built-in braking systems, or safeties, that grab onto the rail when the car moves too fast. [3]

2.2.4 Types of breaking systems

Safeties are activated by a governor when the elevator moves too quickly. Most governor systems are built around a sheave positioned at the top of the elevator shaft. The governor rope is looped around the governor sheave and another weighted sheave at the bottom of the shaft. The rope is also connected to the elevator car, so it moves when the car goes up or down. As the car speeds up, so does the governor.

2.2.4.1 Flyweights breaking system

In this governor, the sheave is outfitted with two hooked flyweights (weighted metal arms) that pivot on pins. The flyweights are attached in such a way that they can swing freely back and forth on the governor. But most of the time, they are kept in position by a high-tension spring.

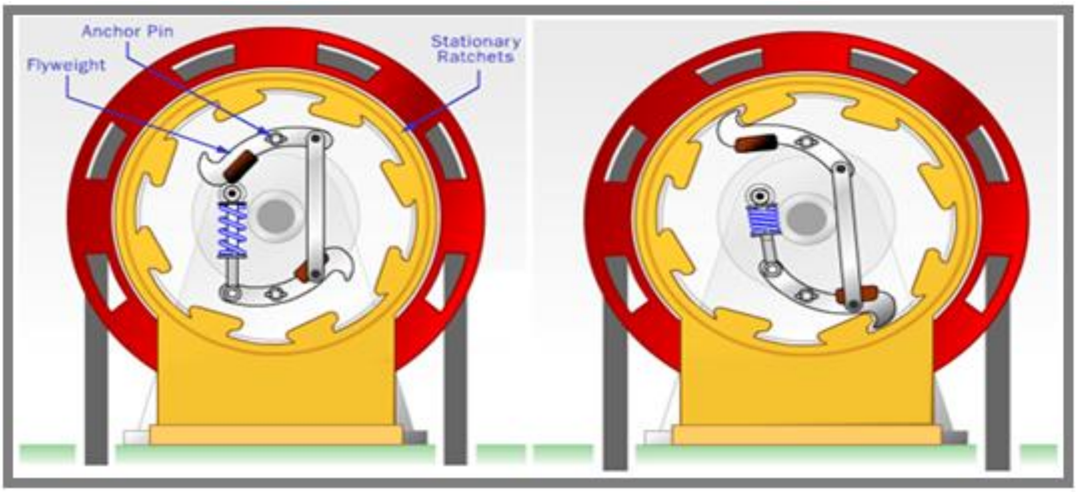


Figure 2:3Flyweights safety system

As the rotary movement of the governor builds up, centrifugal force moves the flyweights outward, pushing against the spring. If the elevator car falls fast enough, the centrifugal force will be strong enough to push the ends of the flyweights all the way to the outer edges of the governor. Spinning in this position, the hooked ends of the flyweights catch hold of ratchets mounted to a stationary cylinder surrounding the sheave. This works to stop the governor.

2.2.4.2 Actuator arm breaking system

The governor ropes are connected to the elevator car via a movable actuator arm attached to a lever linkage. When the governor ropes can move freely, the arm stays in the same position relative to the elevator car (it is held in place by tension springs).

But when the governor sheave locks itself, the governor ropes jerk the actuator arm up. This moves the lever linkage, which operates the brakes.

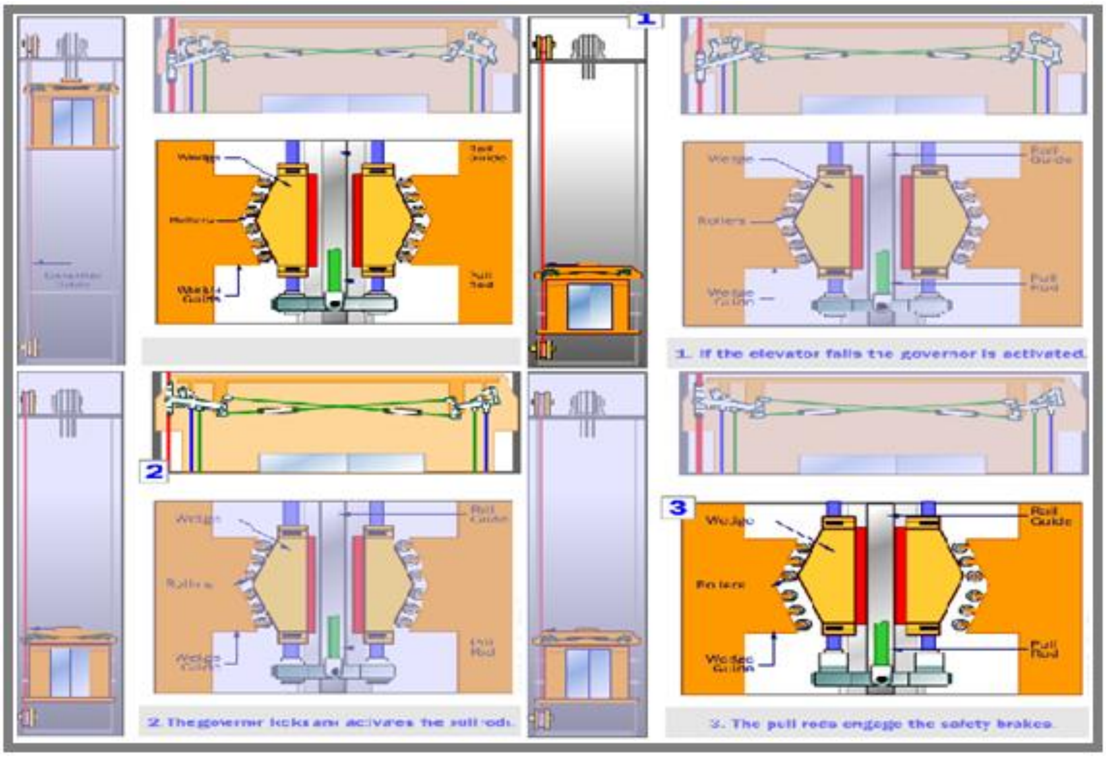


Figure 2:4 Actuator arm breaking system

In this design, the linkage pulls up on a wedge-shaped safety, which sits in a stationary wedge guide. As the wedge moves up, it is pushed into the guide rails by the slanted surface of the guide. This gradually brings the elevator car to a stop.

2.2.4.3 More backups

Elevators also have electromagnetic brakes that engage when the car comes to a stop. The electromagnets actually keep the brakes in the open position, instead of closing them. With this design, the brakes will automatically clamp shut if the elevator loses power

Elevators also have automatic braking systems near the top and the bottom of the elevator shaft. If the elevator car moves too far in either direction, the brake brings it to a stop.

If all else fails and the elevator did fall down the shaft, there is one final safety measure that will probably save the passengers. The bottom of the shaft has a heavy-duty shock absorber system -- typically a piston mounted in an oil-filled cylinder. The shock absorber works like a giant cushion to soften the elevator car's landing.

2.2.5 Making the Rounds

Any modern elevators are controlled by a computer. The computer's job is to process all of the relevant information about the elevator and turn the motor the correct amount to put the elevator car where it needs to be. In order to do this, the computer needs to know at least three things.

- Where people want to go
- Where each floor is
- Where the elevator car is

Finding out where people want to go is very easy. The buttons in the elevator car and the buttons on each floor are all wired to the computer. When one of these buttons is pressed, the computer logs this request.

There are lots of ways to figure out where the elevator car is. In one common system, a light sensor or magnetic sensor on the side of the car reads a series of holes on a long vertical tape in the shaft. By counting the holes speeding by, the computer knows exactly where the car is in the shaft. The computer varies the motor speed so that the car slows down gradually as it reaches each floor. This keeps the ride smooth for the passengers.

In a building with many floors, the computer has to have some sort of strategy to keep the cars running as efficiently as possible. In older systems, the strategy is to avoid reversing the elevator's direction. That is, an elevator car will keep moving up as long as there are people on the floors above that want to go up. The car will only answer "down calls" after it has taken care of all the "up calls." But once it starts down, it won't pick up anybody who wants to go up until there are no more down calls on lower floors. This program is highly inflexible.

More advanced programs take passenger traffic patterns into account. They know which floors have the highest demand, at what time of day, and direct the elevator cars accordingly. In a multiple car system, the elevator will direct individual cars based on the location of other cars.

In one cutting-edge system, the elevator lobby works like a train station. Instead of simply pressing up or down, people waiting for an elevator can enter a request for a specific floor. Based on the location and course of all the cars, the computer tells the passengers which car will get them to their destinations the fastest.

Most systems also have a load sensor in the car floor. The load sensor tells the computer how full the car is. If the car is near capacity, the computer won't make any more pick-up stops until some people have gotten off. Load sensors are also a good safety feature. If the car is overloaded, the computer will not close the doors until some of the weight is removed. [3]

2.2.6 Elevator doors configurations

Elevator doors protect riders from falling into the shaft. The most common configuration is to have two panels that meet in the middle, and slide open laterally. In a cascading configuration (potentially allowing wider entryways within limited space), the doors run on independent tracks so that while open, they are tucked behind one another, and while closed, they form cascading layers on one side. This can be configured so that two sets of such cascading doors operate like the center opening doors, allowing for a very wide elevator cab. In less expensive installations the elevator can also use one large "slab" door: a single panel door the width of the doorway that opens to the left or right laterally. Some buildings have elevators with the single door on the shaft way, and double cascading doors on the cab. [3]

2.2.6.1 Doors automation

The automatic doors in an elevator are absolutely essential. They are there to keep people from falling down an open shaft. [3]

Elevators use two different sets of doors: doors on the cars and doors opening into the elevator shaft. The doors on the cars are operated by an electric motor, which is hooked up to the elevator computer.

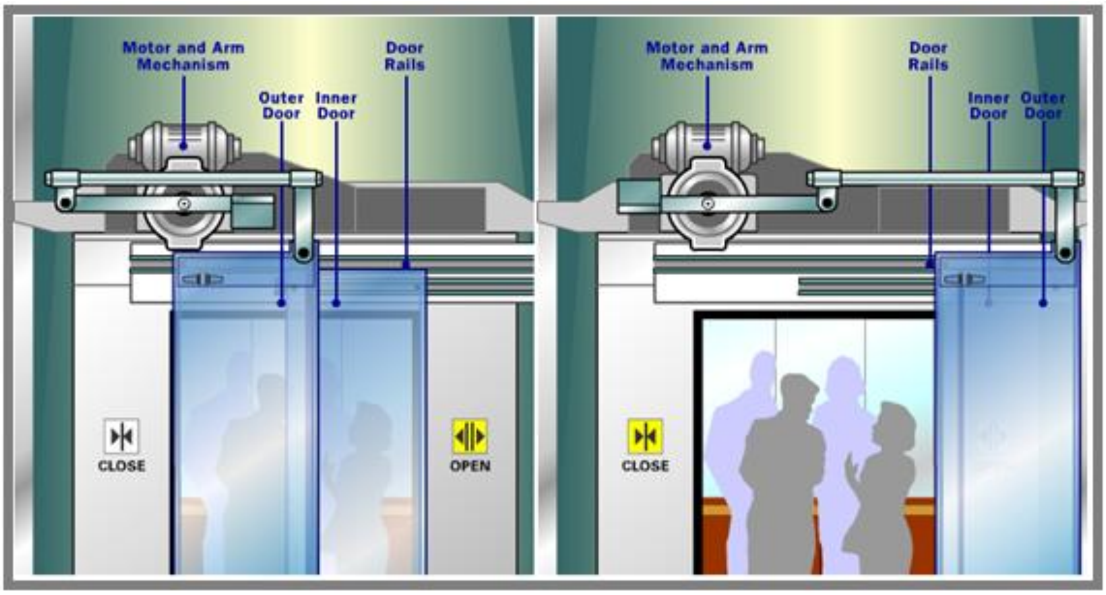


Figure 2:5 Doors automation

The electric motor turns a wheel, which is attached to a long metal arm. The metal arm is linked to another arm, which is attached to the door. The door can slide back and forth on a metal rail.

When the motor turns the wheel, it rotates the first metal arm, which pulls the second metal arm and the attached door to the left. The door is made of two panels that close in on each other when the door opens and extend out when the door closes. The computer turns the motor to open the doors when the car arrives at a floor and close the doors before the car starts moving again. Many elevators have a motion sensor system that keeps the doors from closing if somebody is between them.

The car doors have a clutch mechanism that unlocks the outer doors at each floor and pulls them open. In this way, the outer doors will only open if there is a car at that floor (or if they are forced open). This keeps the outer doors from opening up into an empty elevator shaft. [3]

2.3 Programmable logic controller PLC

A programmable logic controller (PLC) or programmable controller is a digital computer used for automation of electromechanical processes, such as control of machinery on factory assembly lines, amusement rides, or lighting fixtures. Unlike general-purpose computers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed or non-volatile memory. A PLC is an example of a hard real time system since output results must be produced in response to input conditions within a bounded time, otherwise unintended operation will result.

Many engineering systems, including elevator systems, are nowadays controlled by programmable logic controllers (PLCs). The reliance on these PLCs is due to their high reliability and efficiency. [4]

The PLC may be considered as a special-purpose computer with a basic architecture similar to that of any other known computer such as a central processing unit (CPU). It is based on a memory and a number of input and output terminals. The software used for PLC programming is based on a special language known as the ladder diagram. The ladder diagram is an easy programming language since it is based on Boolean logic functions. This makes the task of modifying any system much easier and more cost-effective. The size of the PLC is one of the factors considered when it is selected to control a process. [4]

2.3.1 Programming

Early PLCs, up to the mid-1980s, were programmed using proprietary programming panels or special-purpose programming terminals, which often had dedicated function keys representing the various logical elements of PLC programs. Programs were stored on cassette tape cartridges. Facilities for printing and documentation were very minimal due to lack of memory capacity. The very oldest PLCs used non-volatile magnetic core memory.

More recently, PLCs are programmed using application software on personal computers. The computer is connected to the PLC through Ethernet, RS-232, RS-485 or RS-422 cabling. The programming software allows entry and editing of the ladder-style logic. Generally the software provides functions for debugging and troubleshooting the PLC software, for example, by highlighting portions of the logic to show current status during operation or via simulation. The software will upload and download the PLC program, for backup and restoration purposes. In some models of programmable controller, the program is transferred from a personal computer to the PLC though a programming board which writes the program into a removable chip such as an EEPROM or EPROM.

While the fundamental concepts of PLC programming are common to all manufacturers, differences in I/O addressing, memory organization and instruction sets mean that PLC programs are never perfectly interchangeable between different makers. Even within the same product line of a single manufacturer, different models may not be directly compatible.

2.3.1.1 PLC programming languages

PLCs can be programmed using standards-based programming languages. A graphical programming notation called Sequential Function Charts is available on certain programmable controllers. Initially most PLCs utilized Ladder Logic Diagram Programming, a model which emulated electromechanical control panel devices (such as the contact and coils of relays) which PLCs replaced. This model remains common today.

IEC 61131-3 currently defines five programming languages for programmable control systems: FBD (Function block diagram), LD (Ladder diagram), ST (Structured text, similar to the Pascal programming language), IL (Instruction list, similar to assembly language) and SFC (Sequential function chart). These techniques emphasize logical organization of operations.

2.3.1.2 Ladder Logic Programming

Ladder diagram is a language which composes program using relay symbols as a base in an image similar to a hard-wired relay sequence, ladder diagram are combinations of relay symbols and function blocks The ladder diagram program is constructed by units called ‘rung’. A rung is defined as one network which is connected each other. The rung numbers are a series of numbers (decimal number) starting from 1, and cannot be skipped. There is no limit to the number of rungs. The size of any one rung is limited to 11 lines 12 columns.

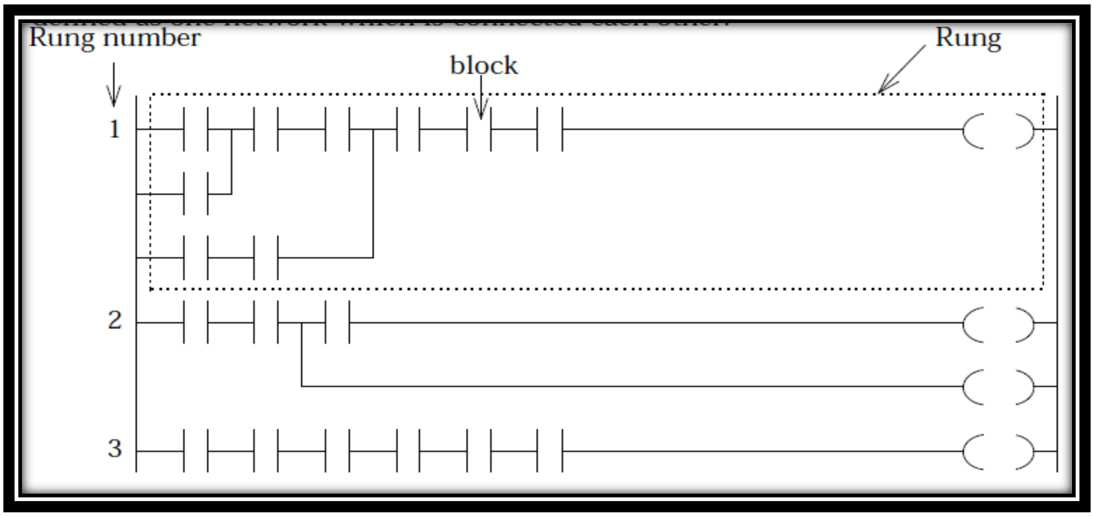


Figure 2:6 Ladder diagram

Program execution sequence

The instructions execution sequence is shown below.

(1) They are executed in the sequence from block 1 through the final block which contains the END instruction.

(2) They are executed in the sequence from rung 1 through the final rung in a block (or the END instruction).

(3) They are executed according to the following rules in any one rung.

- ❖ When there is no vertical connection, they are executed from left to right.
- ❖ When there is an OR connection, the OR logic portion is executed first.
- ❖ When there is a branch, they are executed in the order from the upper line to the lower line.

2.3.2 Features

The main difference from other computers is that PLCs are armored for severe conditions (such as dust, moisture, heat, cold) and have the facility for extensive input/output (I/O) arrangements. These connect the PLC to sensors and actuators. PLCs read limit switches, analog process variables (such as temperature and pressure), and the positions of complex positioning systems. Some use machine vision. On the actuator side, PLCs operate electric motors, pneumatic or hydraulic cylinders, magnetic relays, solenoids, or analog outputs. The input/output arrangements may be built into a simple PLC, or the PLC may have external I/O modules attached to a computer network that plugs into the PLC.

2.3.2.1 System scale

PLCs come in various sizes and different capabilities; the sizes range from small controllers with limited inputs and outputs used for controlling small processes, to very large ones with more inputs and outputs provided which are used to control much larger processes and operations. Determining the appropriate PLC to be used is based on analyzing the process to be controlled and accordingly identifying the number of inputs and outputs required. [4]

A small PLC will have a fixed number of connections built in for inputs and outputs. Typically, expansions are available if the base model has insufficient I/O.

Modular PLCs have a chassis (also called a rack) into which are placed modules with different functions. The processor and selection of I/O modules is customized for the particular application. Several racks can be administered by a single processor, and may have thousands of inputs and outputs. A special high speed serial I/O link is used so that racks can be distributed away from the processor, reducing the wiring costs for large plants.

2.3.3 User interface

PLCs may need to interact with people for the purpose of configuration, alarm reporting or everyday control.

A Human-Machine Interface (HMI) is employed for this purpose. HMIs are also referred to as MMIs (Man Machine Interface) and GUIs (Graphical User Interface).

A simple system may use buttons and lights to interact with the user. Text displays are available as well as graphical touch screens. More complex systems use programming and monitoring software installed on a computer, with the PLC connected via a communication interface.

2.3.4 Communications

PLCs have built in communications ports, usually 9-pin RS-232, but optionally EIA-485 or Ethernet. Modbus, BACnet orDF1 is usually included as one of the communications protocols. Other options include various fieldbuses such as DeviceNet orProfibus.

Most modern PLCs can communicate over a network to some other system, such as a computer running a SCADA (Supervisory Control and Data Acquisition) system or web browser.

PLCs used in larger I/O systems may have peer-to-peer (P2P) communication between processors. This allows separate parts of a complex process to have individual control while allowing the subsystems to co-ordinate over the communication link. These communication links are also often used for HMI devices such as keypads or PC-type workstations.

2.3.5 PLC compared with other control systems

PLCs are well-adapted to a range of automation tasks. These are typically industrial processes in manufacturing where the cost of developing and maintaining the automation system is high relative to the total cost of the automation, and where changes to the system would be expected during its operational life. PLCs contain input and output devices compatible with industrial pilot devices and controls; little electrical design is required, and the design problem centers on expressing the desired sequence of operations. PLC applications are typically highly customized systems so the cost of a packaged PLC is low compared to the cost of a specific custom-built controller design. On the other hand, in the case of mass-produced goods, customized control systems are economic due to the lower cost of the components, which can be optimally chosen instead of a "generic" solution, and where the non-recurring engineering charges are spread over thousands or millions of units.

For high volume or very simple fixed automation tasks, different techniques are used. For example, a consumer dishwasher would be controlled by an electromechanical cam timer costing only a few dollars in production quantities.

A microcontroller-based design would be appropriate where hundreds or thousands of units will be produced and so the development cost (design of power supplies, input/output hardware and necessary testing and certification) can be spread over many sales, and where the end-user would not need to alter the control.

PLCs may include logic for single-variable feedback analog control loop, a "Proportional integral derivative" or "PID controller". A PID loop could be used to control the temperature of a manufacturing process, for example. Historically PLCs were usually configured with only a few analog control loops; where processes required hundreds or thousands of loops, a distributed control system (DCS) would instead be used. As PLCs have become more powerful, the boundary between DCS and PLC applications has become less distinct.

PLCs have similar functionality as Remote Terminal Units. An RTU, however, usually does not support control algorithms or control loops. As hardware rapidly becomes more powerful and cheaper, RTUs, PLCs and DCSs are increasingly beginning to overlap in responsibilities, and many vendors sell RTUs with PLC-like features and vice versa. The industry has standardized on the IEC 61131-3 functional block language for creating programs to run on RTUs and PLCs, although nearly all vendors also offer proprietary alternatives and associated development environments.

The PLC has many advantages over other control systems. It is known for its flexibility, lower cost, operational speed, reliability, ease of programming, security, and it is easy in implementing changes and correcting errors. One of the applications using PLCs is the control of elevator systems. [4]

2.4 S-7 200

SIEMENS micro PLC model S7-200, CPU 224XP

It is a digitally operating electronic apparatus which uses a programmable memory for the internal storage of instructions by implementing specific functions, such as logic, sequencing, timing, counting and arithmetic to control through digital or analog I/O modules various types of machines or processes.

Siemens makes several PLC products in the SIMATIC S7 family, S7-200 is referred to as micro PLC and because of its small size it is used in small and less complex applications.

RS-458/PPI Multi-Master Cable

It is a standard for serial binary data signals connecting between a DTE (Data Terminal Equipment) and a DCE (Data Circuit-terminating Equipment). It is commonly used in computer serial ports.

2.5 STEP 7- Micro/WIN, version 4.0

STEP 7- Micro/WIN is programming software package for the S7 PLC provides a user-friendly environment to develop, edit, and monitor the logic needed to control applications.

2.6 S7-200 simulator

It is a program for simulation through PC for the programmable robots of SIEMENS S7_200; it is a useful application that was designed in order to provide a way to simulate in a PC the operation of the programs created for robots.

2.7 Relay

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. [5]

Solid-state relay

A solid state relay (SSR) is a solid state electronic component that provides a similar function to an electromechanical relay but does not have any moving components, increasing long-term reliability. [5]

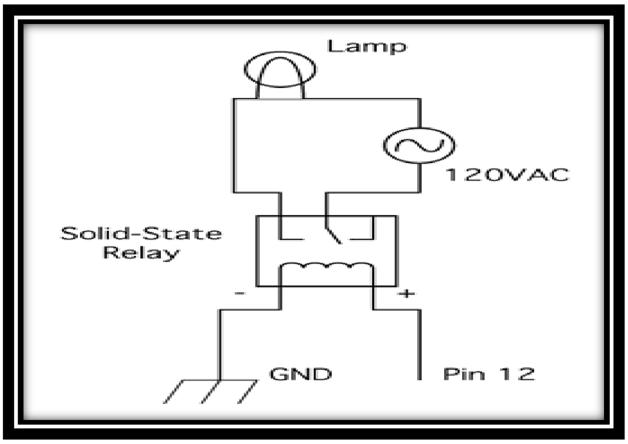


Figure 2:7 Solid state relay

2.8 Limit switch

A limit switch is used to stop the motion of a machine slide or element once it reaches a fixed point. The basic limit switch has a spring-loaded push button connected to an electrical circuit. When the push button (actuator) is pushed inside it trips or activates the electrical circuit. The electrical circuit then signals a machine element to stop or start its motion. Once the pressure on the push button is removed, it comes back to its original position and is ready for the next cycle. [6]

CHAPTER III

HARDWARE AND SOFTWARE DEVELOPMENT

3.1 Overview

The design of the elevator control system consists of two main parts; software and hardware 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.

In this chapter detailed explanation of the code implemented (software) on the PLC and the hardware used is provided.

3.2 Hardware Design

In the hardware design each and every hard were component was selected carefully to fit in the project requirements and limitations; the following is description of the hardware components one by one:

Elevator shaft and car

Iron is the material used to build the prototype of the car and the shaft. Iron bars were cut and welded with shaft and car dimensions selected so that the shaft could fit the car, limit switches, and all the wiring related. At the top of the shaft; the pulley and its driving motor were fixed. And at the top of the car the door motor is welded. At two sides of the cars rolling wheels were attached to smooth the movement of the car inside the shaft and to limit it to certain positions.

Figure 3:1 Elevator shaft dimensions

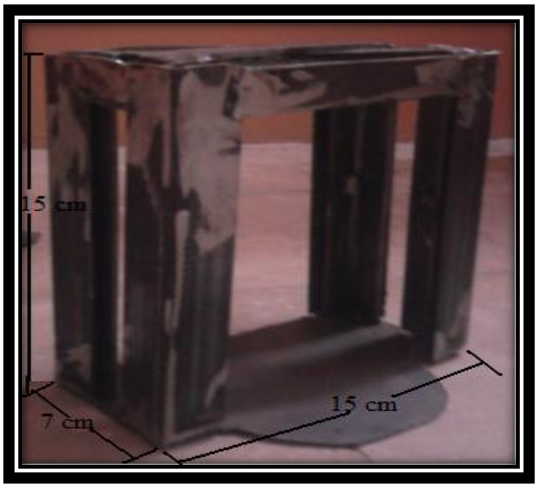


Figure 3:2Elevator car dimension Figure 3:3 Bearings attached to car

Pulley

The pulley is made using aluminum rod. The rod is lathed to fit the shaft of the motor, and bearings were attached to it, grooves were made to limit the rope movement.

The diameter of the pulley had to large enough to minimize the rotation the motor had to apply to cross a given distance; 2" was found to be the appropriate diameter.

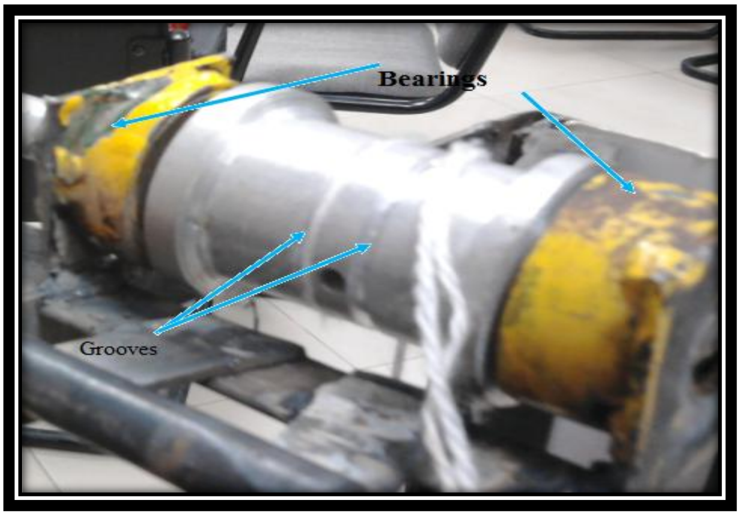


Figure 3:4 Pulley used

Rope

A plastic rope was used, the plastic rope was preferred to other materials because it was flexible compared to the metal rope first tried and it was more consistent and had more strength compared to the wool rope that snapped when was attached.

The length of the rope was chosen based on the height of the shaft; the rope had to be long enough to connect both the car and counter weight making only to rounds around the pulley; and it length had to be exactly that much, nothing more or less to insure smooth movement to bottom and top of the shaft (1^{st} to 4^{th} floors).

Pulley & door motors

DC motor (24V, 105mA) is used to rotate the pulley to lift the car up and bring it down. It was borrowed from a printer; so its torque was sufficient (high) and its speed was law which perfectly matched this elevator application.

The pulley motor was then mounted at the top of the shaft.

Figure 3:5Pulley DC motor Position

While a DC motor (12V) was used to provide the movement of the sliding doors, this motor also was mounted at the top of the car.

Figure 3:6 Cars DC motor position

Relays

A solid state relay is used to change the rotation of the motor based on the output signal from the PLC. Each motor enquires two relays to control its rotation direction and the following circuit was used to wire the relays with the DC motors.

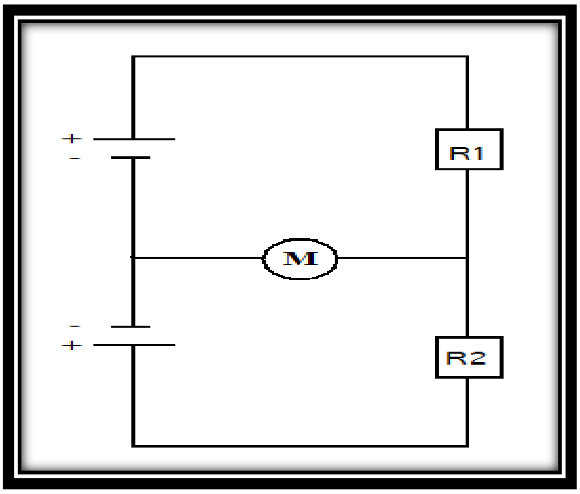


Figure 3:7 Relay connection circuit

Figure 3:8 SSR used

Counter weight

Iron was the material used to create the counter weight; its exact weight was measured with the following equation:

Weight = 0.4(car maximum weight)

Weight = $0.4\times(1.16\text{ kg}+583\text{g})\approx 700\text{gm}$

The cars maximum weight is the cars weight plus the maximum weight of passengers; the counter weight is used to balance the car and helping the motor in moving the car.

The movement of the counter weight it self is controlled with a rail welded inside the body of the shaft to insure that the weight does not get in the way of the cars movement troubling it.

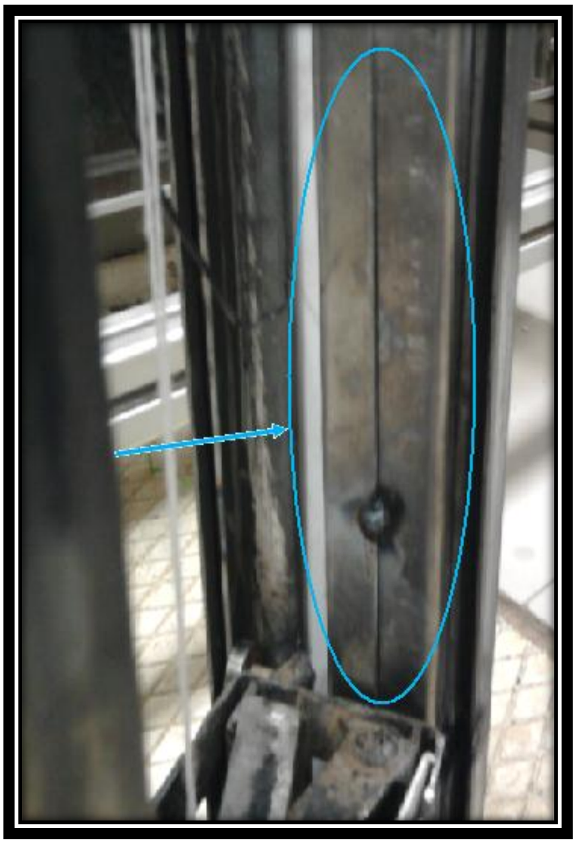


Figure 3:9 Counter weight railing

Sliding doors

To make the doors slide to open; the exact same mechanism used to make the radio channels change is used. A gear that rotates on straight line that has teeth just like the ones in the circular gear; once the circular gear starts rotating driven by the door motor the line will start moving to the left or the right depending on the direction of the gears rotation; that way the circular movement of the gear is transformed into linear movement of the door.

The doors were maid of crok; due to its light weight.

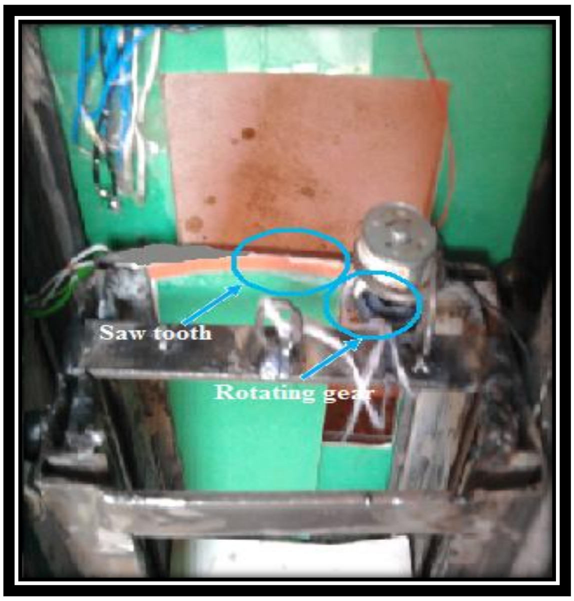


Figure 3:10 Sliding doors configuration

Figure 3:11 Sliding door gear

Limits witches (17 pcs)

Four limit switches were used for determining the position of the car. Each floor had a limit switch fixed at a position just below the floor, once the car passes by the limit switch it presses it and continue moving. Pressing the limit switch cause it to send a signal to the PLC which in turn use it to determine the current position of the car and to decide the direction of the car movement.

Figure 3:12 Limit switch used

LEDs (13 pcs)

LEDs were used to make the elevator system user friendly. One leg of the led is connected to the push button while the other leg is grounded so the LED will illuminate every time a push button is pressed.

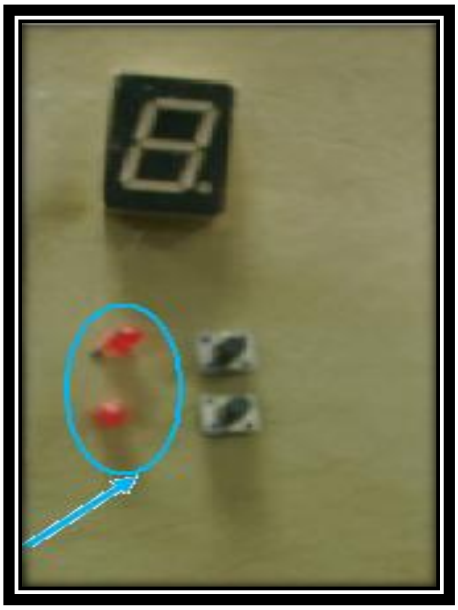


Figure 3:13 LEDs used

Seven-segment display (5 pcs)

Seven-segment displays were used to indicate the current position of the car to people at all floors and to people riding the elevator. A driver chip is used to convert from BCD to 7-segment display, this driver chip operates on 5 volts so a voltage divider circuit (5kΩ&20kΩ) was used to step down the source 24V voltage. The displays were used to make the system user friendly.

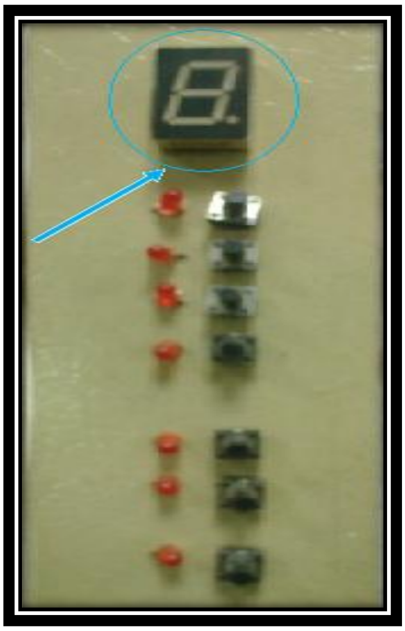


Figure 3:14 7-segment display position

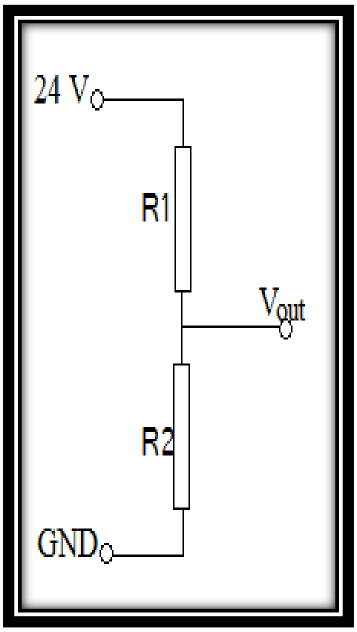


Figure 3:15 voltage divider

Push buttons (13 pcs)

Push buttons were used as input sources to the PLC. Each floor has push buttons that signal the PLC for a floor call, the car has 7 push buttons in it; each floor has a push button assigned to it that is pushed to send a request for visiting that specific floor, two buttons were dedicated to signal a request to keep the elevators door open or closed, and a seventh button is fixed to be bushed on emergencies; pressing that button will make the car stop!.

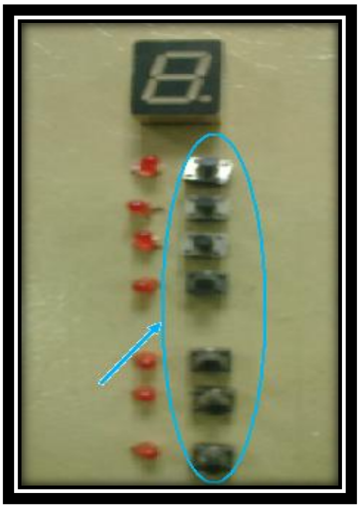
 

Figure 3:16Push buttons

SIEMENS micro PLC model S7-200, CPU 224XP

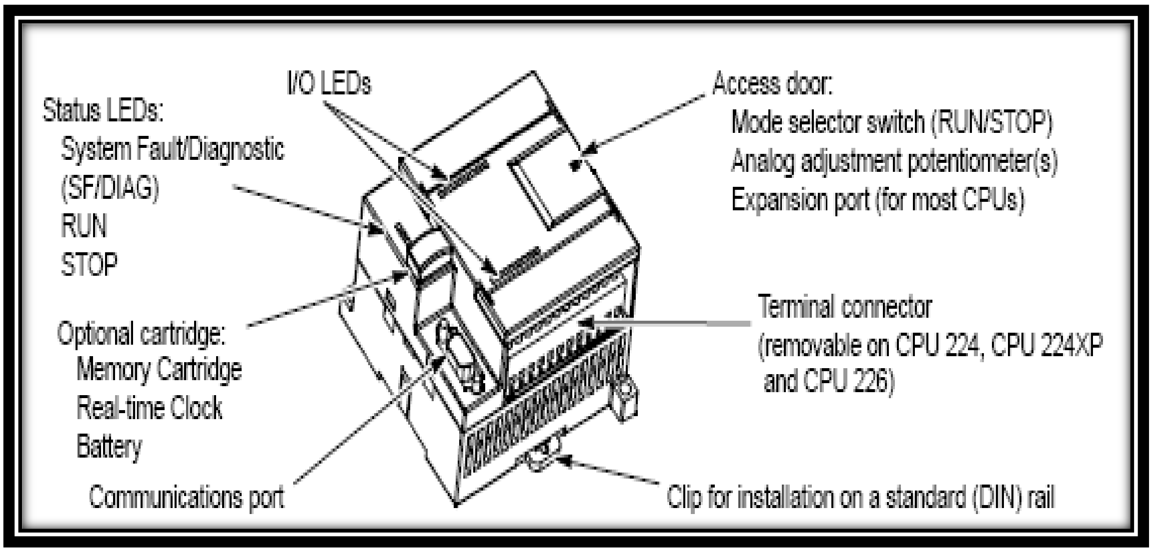


Figure 3:17 S7-200 PLC

Siemens makes several PLC products in the SIMATIC S7 family, S7-200 is referred to as micro PLC and because of its small size it is used in small and less complex applications.

Siemens S7-200 CPU 224 was the PLC used in this project; it has 14 inputs and 10 outputs.

RS-458/PPI Multi-Master Cable

It is a standard for serial binary data signals connecting between a DTE (Data Terminal Equipment) and a DCE (Data Circuit-terminating Equipment). The PPI cable was used to transfer the logic written by the ladder diagram from the PC to the PLC.

Figure 3:18 RS-458/PPI Multi-Master Cable

3.3 Software Design

The 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.

Figure 3:19 First floor sensor

M0.0 represents the memory location in which the input from the first floor limit switch (I0.0)is stored. M0.1, M0.2, and M0.3 are the memory locations of the second, third, and forth floor limit switches(I0.1, I0.2, I0.3)respectively.

When the first floor limit switch is pressed by the car the limit switch inputs that signal to the PLC which stores it in M0.0, the functionality of network 1 is to reset M0.1, M0.2, & M0.3 when M0.0 is set. The point of the resetting action is to make sure that only correct and one floor is indicated as the present location of the car.

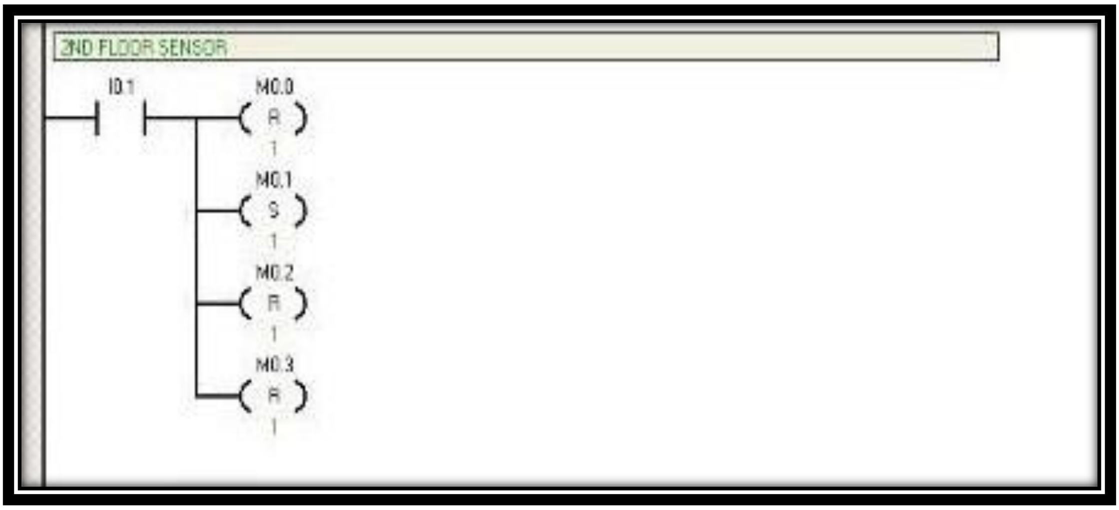


Figure 3:20 Second floor sensor

M0.1 is set if the car presses the second floor limit switch and the memory locations indicating the other floors are reset.

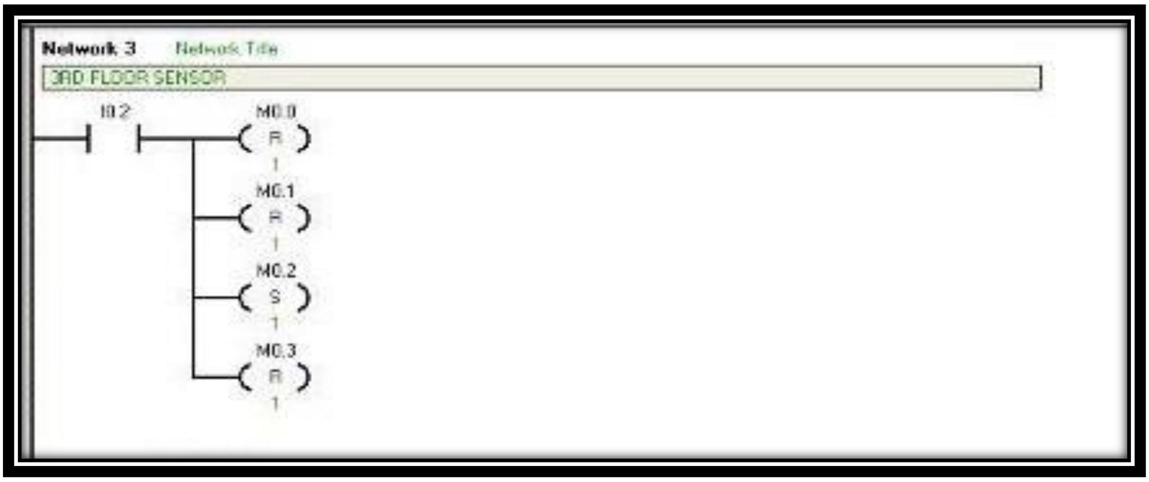


Figure 3:21 Third floor sensor

M0.2 is set if the car presses the third floor limit switch and the memory locations indicating the other floors are reset.

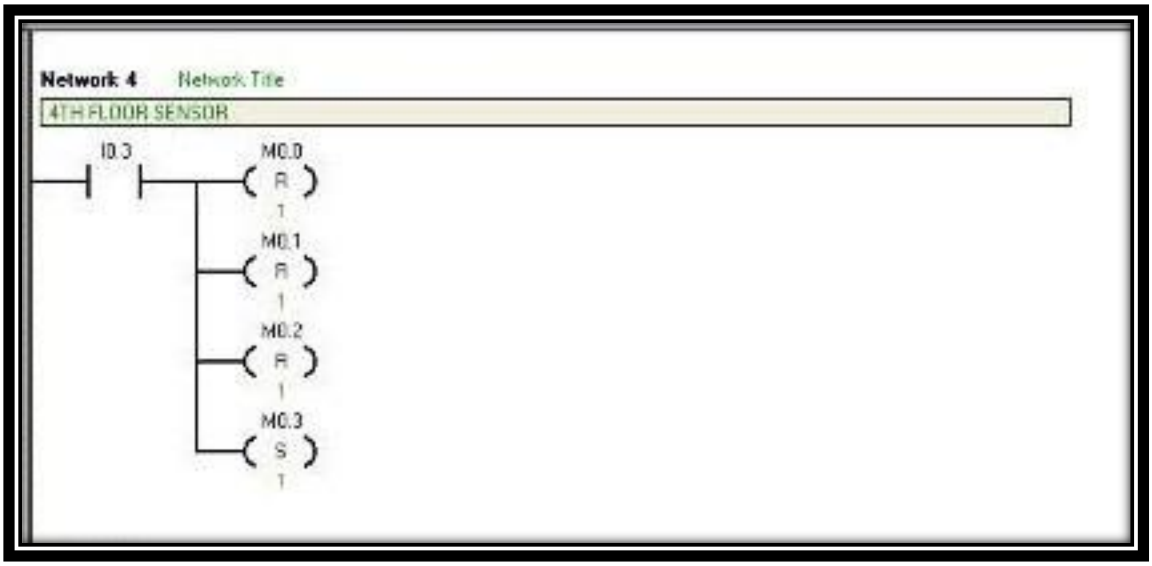


Figure 3:22 Forth floor sensor

M0.3 is set if the car presses the forth floor limit switch and the memory locations indicating the other floors are reset.

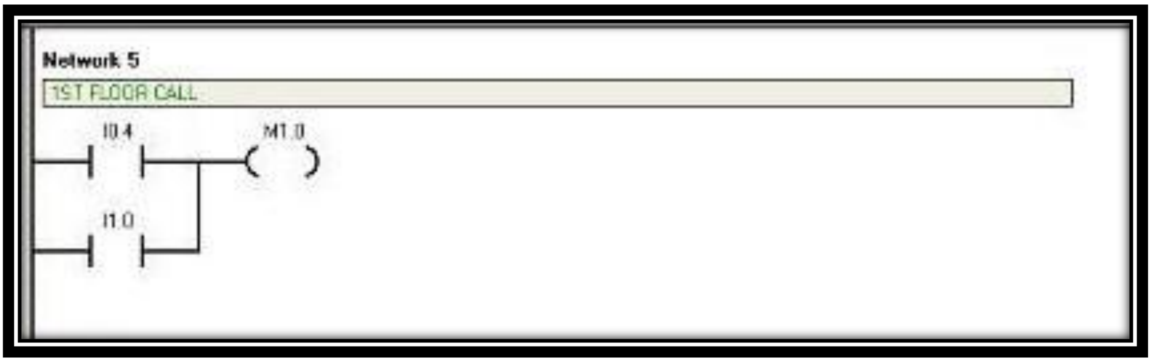


Figure 3:23 First floor call

I1.0 represents the input from the push button at the first floor for a request to go to upper floors. While I0.4 represent a signal from the limit switch inside the elevator car indicating a request to go down to the first floor. If either one of the signals occur a memory location (M1.0) indicating a call to visit the first floor is set.

Figure 3:24 Second floor call

I0.5 is the input from the limit switch inside the car that indicates a request to go visit the second floor, I1.1 & I1.2 are input signal from the push buttons at the second floor the former is for a going up request and the latter is for a going down reaquest . if any of these three signals set; a call to visit the second floor is made by setting the memory bit M1.1.

Figure 3:25 Third floor call

I0.6 is the input from the limit switch inside the car that indicates a request to go visit the third floor, I1.3 & I1.4 are input signal from the push buttons at the second floor the former is for a going up request and the latter is for a going down reaquest . if any of these three signals set; a call to visit the third floor is made by setting the memory bit M1.2.

Figure 3:26 Forth floor call

I1.5 represents the input from the push button at the forth floor for a request to go down to lower floors. While I0.7 represent a signal from the limit switch inside the elevator car indicating a request to go up to the forth floor. If either one of the signals occur a memory location (M1.0) indicating a call to visit the first floor is set.

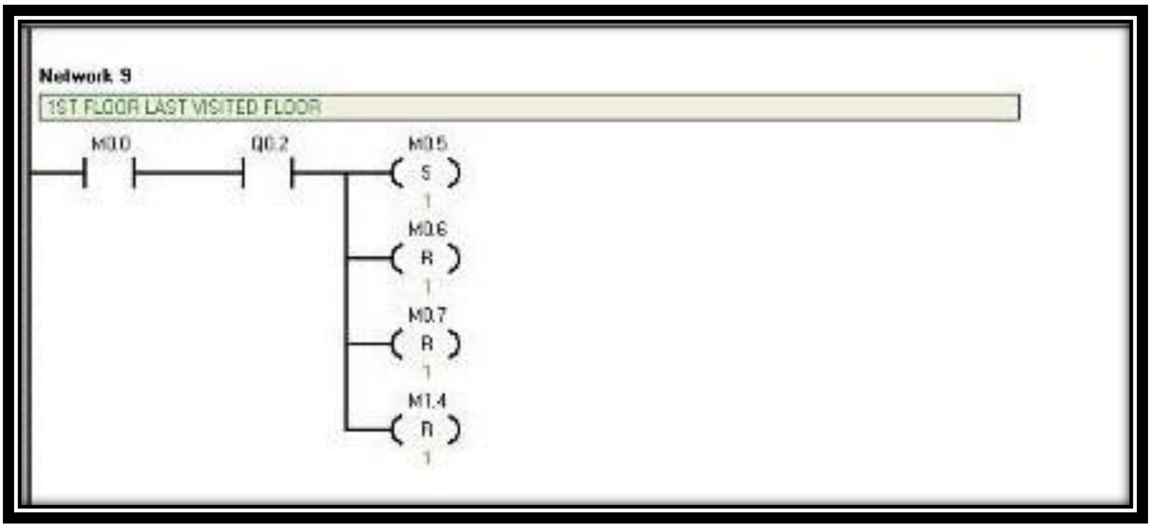


Figure 3:27 1st floor last visited floor

Networks 9-12 are dedicated to set memory locations (M0.5, M0.6, M0.7, and M1.4) that indicate which floor was the last visited floor. The purpose of this indication is to direct the elevator car to move in a power saving manner; i.e. if the last movement of the elevator car was to go up and there are multiple request from different floors for the car; it will continue moving upward answering the calls of higher floors before it goes down to answer the calls of lower floors and vice versa.

M0.5 is set when the memory location indicating that the car's location is currently the first floor is set (M0.0=1) and the car door opening signal (Q0.2) is on. That way it's insured that the car was actually visiting the first floor and not just passing by it.

To insure that only one floor is indicated as the last visited floor, memory locations that indicate for the other floors (M0.6,M0.7,M1.4)are reset (to 0) once the first floor is set as the last visited floor; the resetting action is performed to the rest of the floors each time a floor is set as the last visited floor.

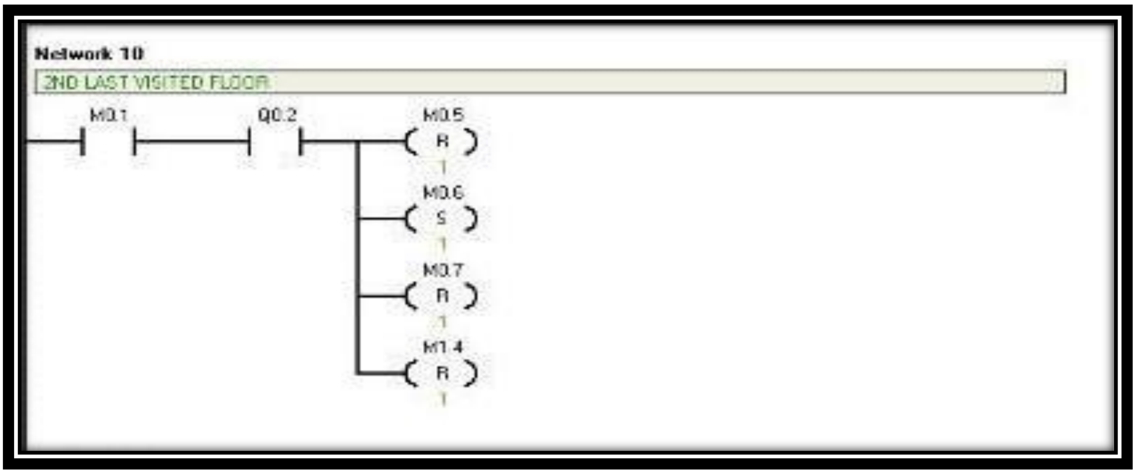
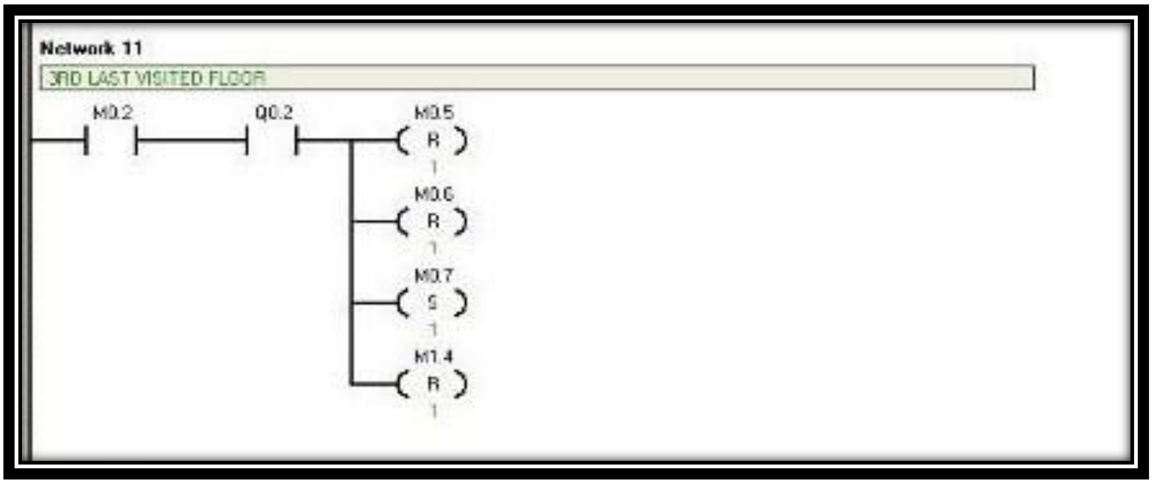


Figure 3:28 2nd floor last visited floor

M0.6 is the memory location that tells if the last visited floor was the second floor ; it is set when the memory location indicating that the car's location is currently the second floor is set (M0.1=1) and the car door opening signal (Q0.2) is on. That way it is insured that the car was actually visiting the second floor and not just passing by it.

Also memory locations indicating that any of the other floors was visited last (M0.5, M0.7, M1.4) are reset (to 0) once the second floor is set as the last visited floor.



3:29 3rd floor last visited floor

M0.7 is the memory location that tells if the last visited floor was the second floor ; it is set when the memory location indicating that the car's location is currently the third floor is set (M0.2=1) and the car door opening signal (Q0.2) is on. That way it is insured that the car was actually visiting the third floor and not just passing by it.

Also memory locations indicating that any of the other floors was visited last (M0.5, M0.6, M1.4) are reset (to 0) once the second floor is set as the last visited floor**.**

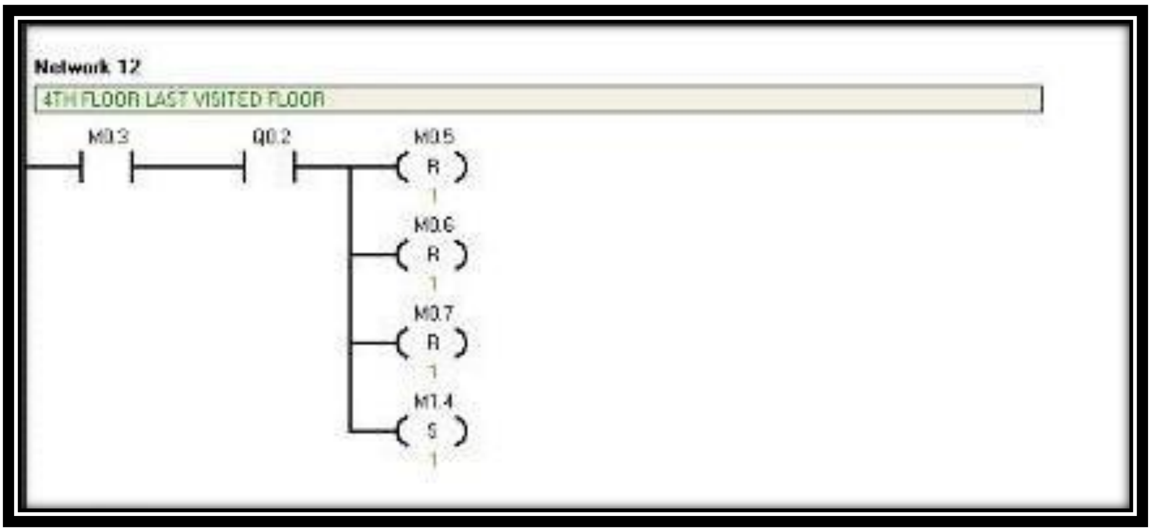


Figure 3:30 4th floor last visited floor

M1.4 is the memory location that tells if the last visited floor is the 4th floor; it is set when the memory location indicating that the car's location is currently the 4th floor is set (M0.3=1) and the car door opening signal (Q0.2) is on. That way it is insured that the car was actually visiting that floor and not just passing by it.

Also memory locations indicating that any of the other floors was visited last (M0.5, M0.6, M0.7) are reset (to 0) once the 4th floor is set as the last visited floor**.**

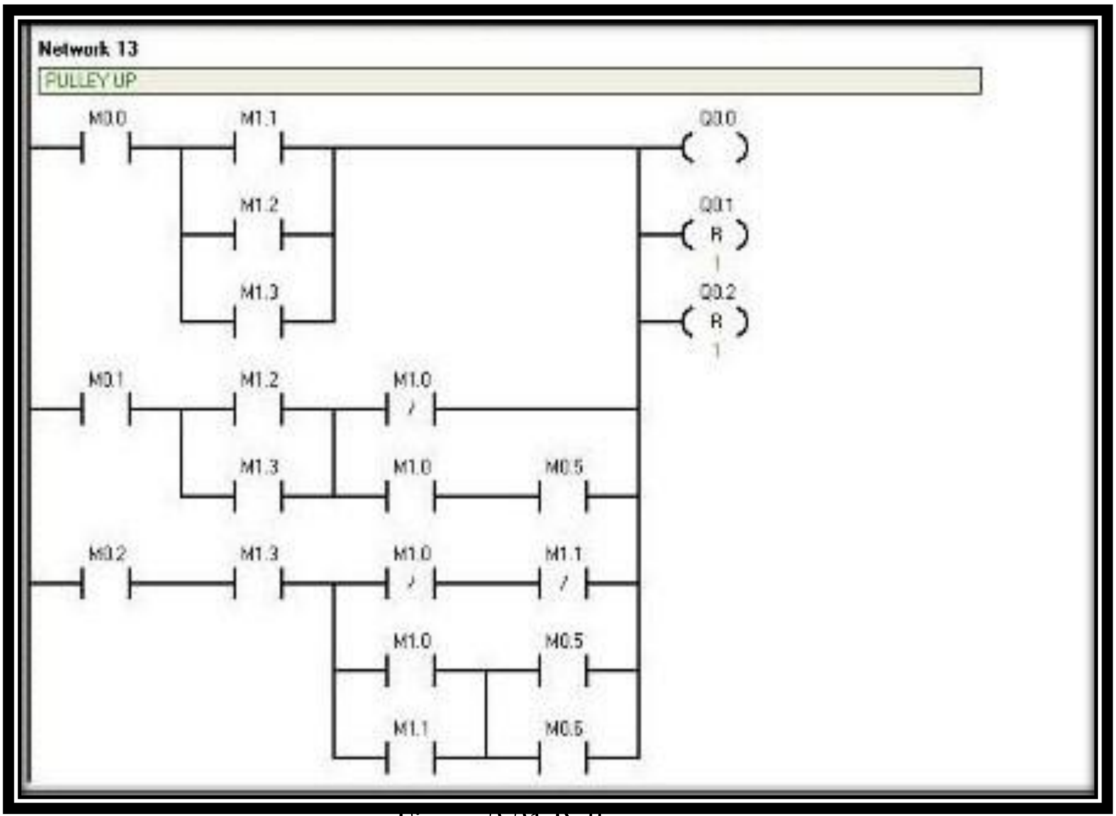


Figure 3:31 Pulley up

Network 13 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 forth 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 forth 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 forth 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 insure that the motor driving the pulley stops and reverse direction if the pulley wasn't at still position and was moving in the opposite direction.

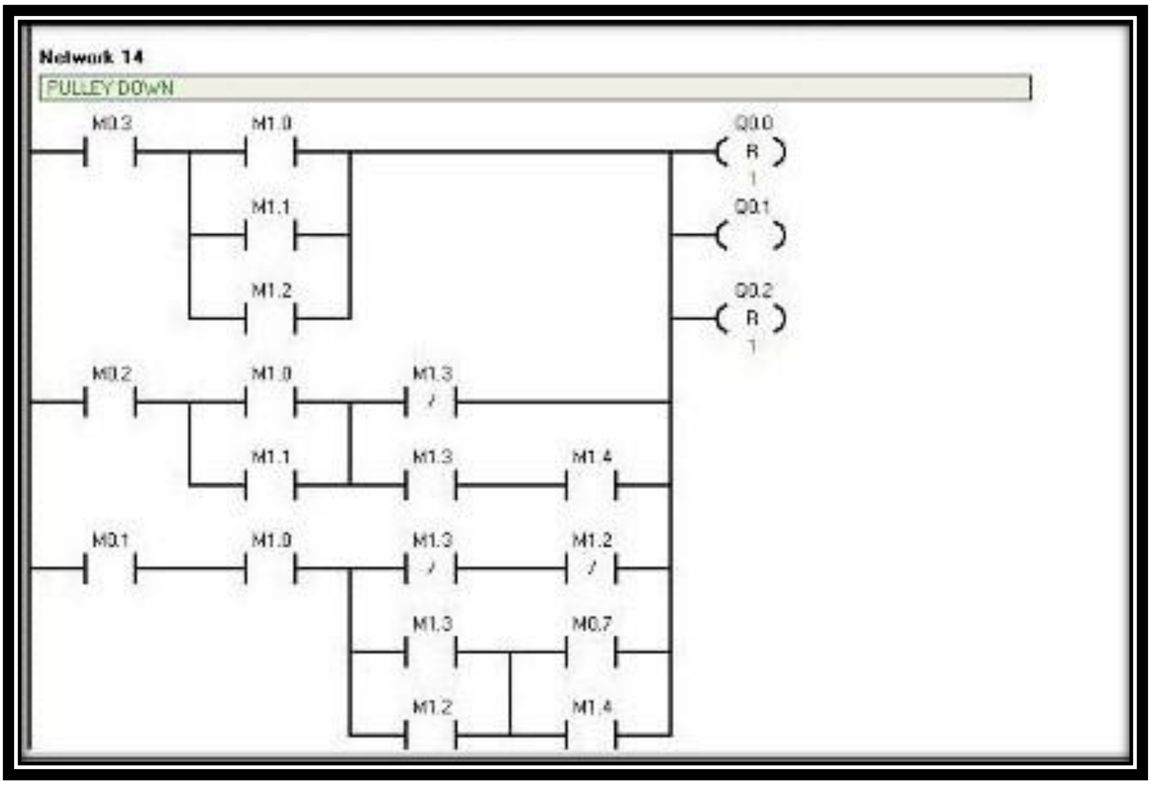


Figure 3:32 Pulley down

Network 14 in contrast to 13, 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 **forth** (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 forth floor (M1.3=0) OR if there was a call to the forth floor

(M1.3=1) BUT the last visited floor was the forth 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 forth floor (M1.3=0) or the third floor (M1.2=0) the pulley goes down. OR if there was a call to the forth floor (M1.3=1) OR the third floor (M1.2=1) BUT the last visited floor was the forth 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 insure that the motor driving the pulley stops and reverse direction if the pulley wasn't at still position and was moving in the opposite direction.

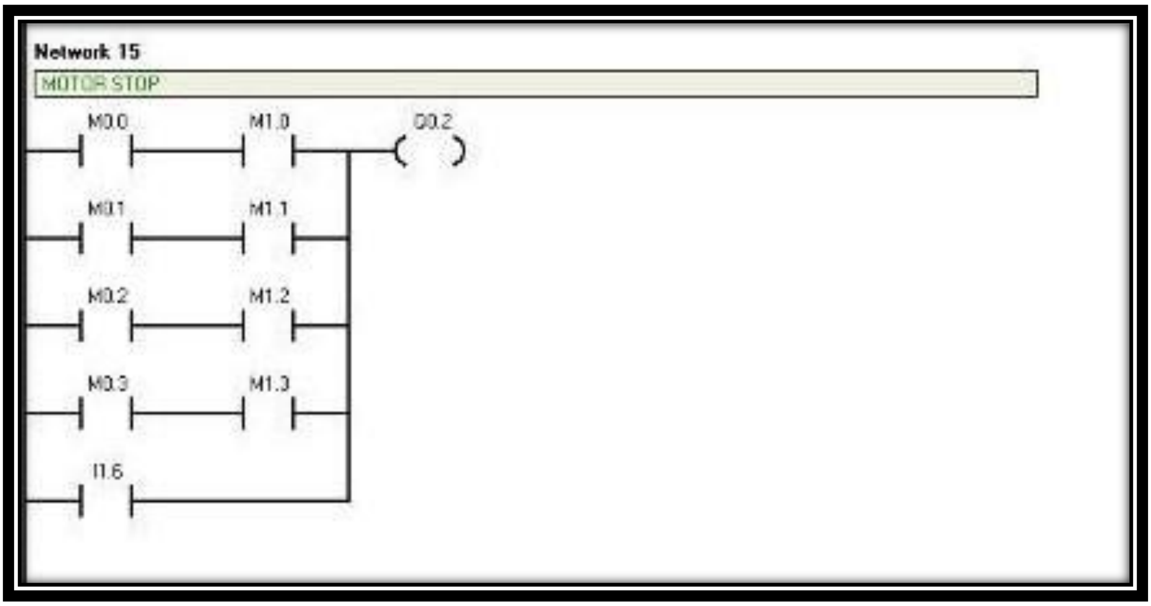


Figure 3:33 Motor stop

The motor stop signal is set (Q0.2) when ever a floor call matches the current position of the car. Or if the emergency stop button is pressed (I1.6=1).

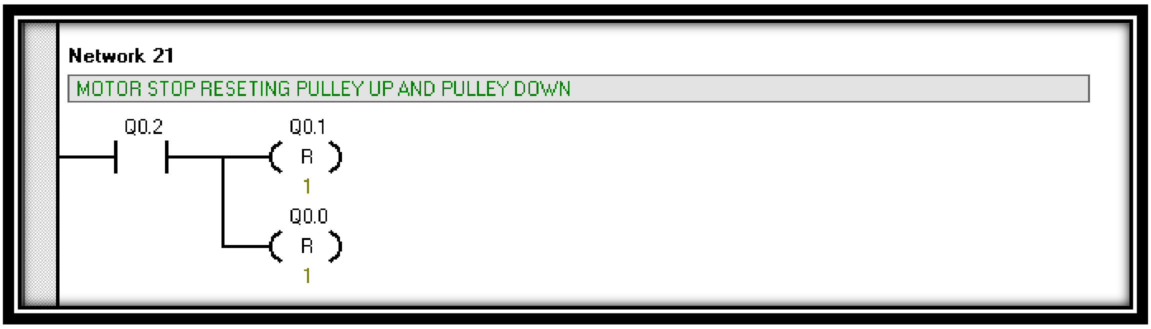


Figure 3:34 Motor lock

The purpose of network 16 is to reset the motor pulley up and pulley down signals whenever the motor stop signal is set ; it's a locking mechanism to insure that the motor doesn't move while the car is loading and unloading passengers.

Figure 3:35 Motor stop

There is a master stop signal, when ever that signal is set the entire elevating system has to come to a stop. This signal could be set for any reason; power blackout or emergency shutting down or even shutting down for maintenance actions.

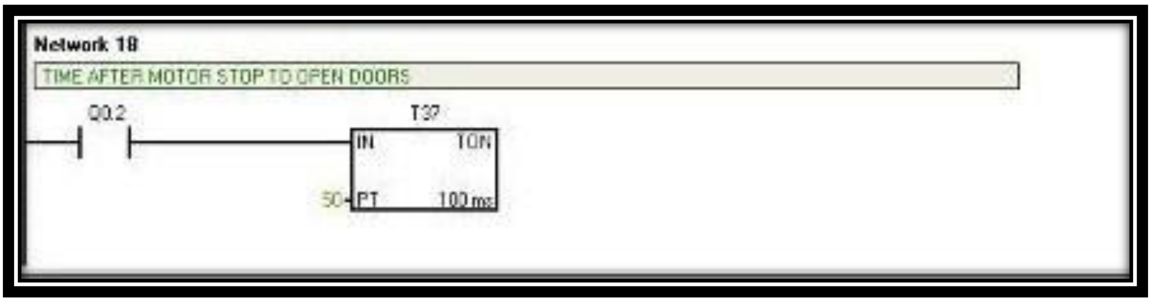


Figure 3:36 Timer to open doors after motor stops

Networks 18-21 specify 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 insure 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 since there for the motor is signaled to stop once it passes by the limit switch which is fixed at a location few millimeters below the actual floor level. The time that takes the car to stop from the moment the motor is signaled to stop is enough to make the car reach the actual floor level.

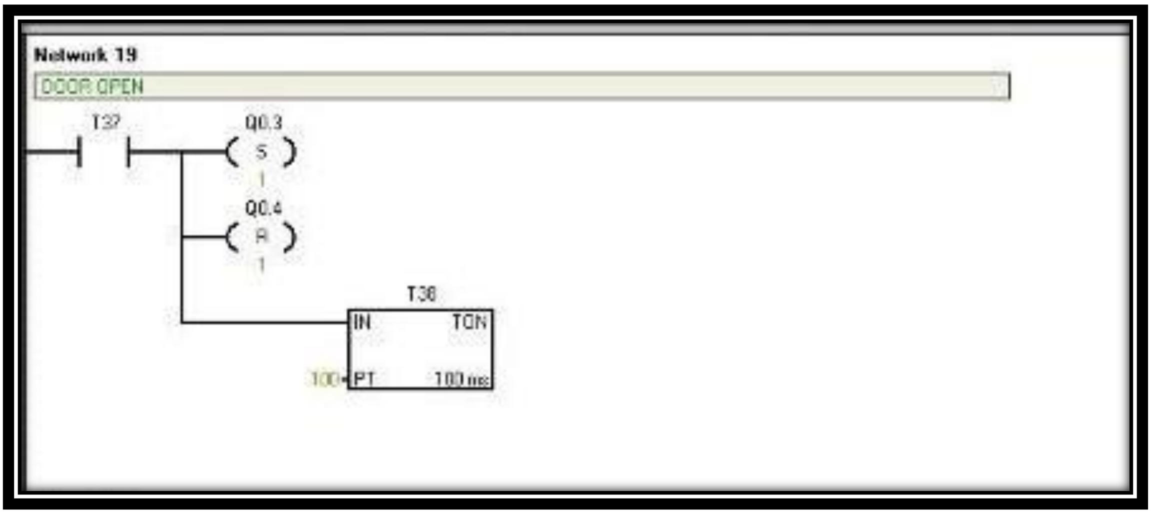


Figure 3:37 Door open

After T32 start signaling (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 insure that the two signals are sent simultaniously causing serious design and operation errors.

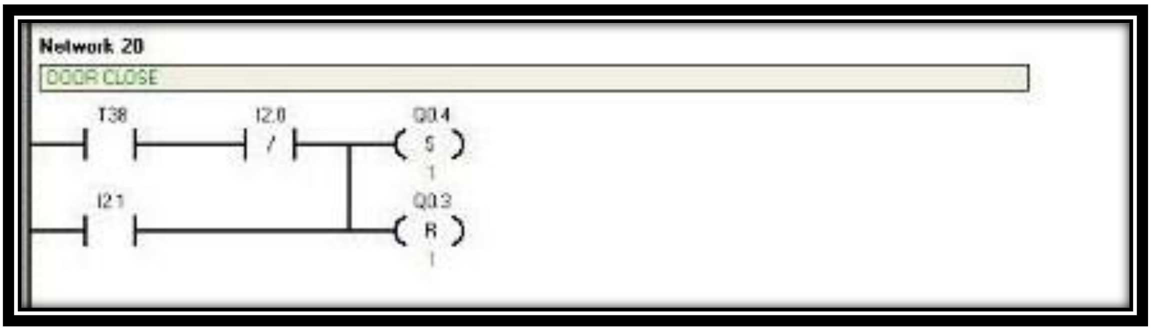


Figure 3:38 Door close

If the timer defining the time period during which the door should remain open start signaling (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.and for the same reseaon setting the door openening signal resets the door close signal, setting the door close signal resets the door open signal.

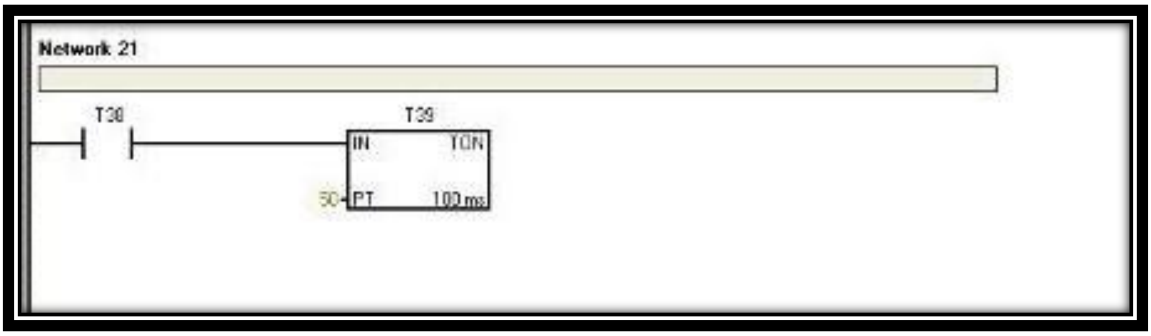


Figure 3:39 Motor movement timer

Scince there was no sensor or limit switch used inb this project to signal the motor to stop running when the door reaches the edge; a timier was used for that purpose. Given that the door will reah the edgeof the car and be fulley closed after 5 seconds; T39 is a timer that counts 5 seconds after the door close signal is set high and send a signal once the 5 seconds have passed.

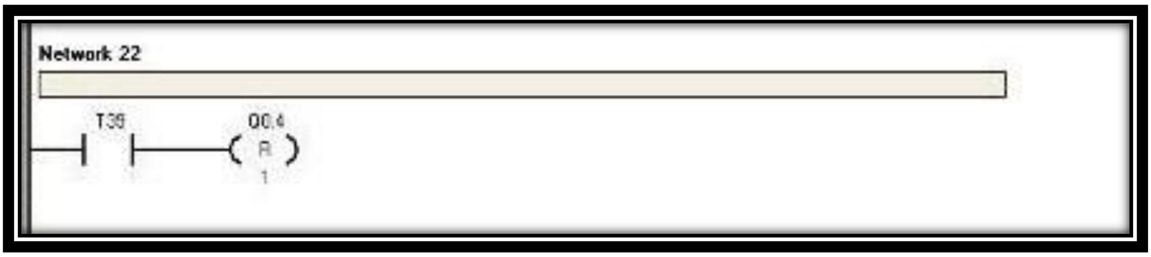


Figure 3:40 Door motor stop

Once T39 starts sending a high signal the door close signal is reset to stop the door motor from continuing to run and force the door into a brick wall or the metal edge of the car! .

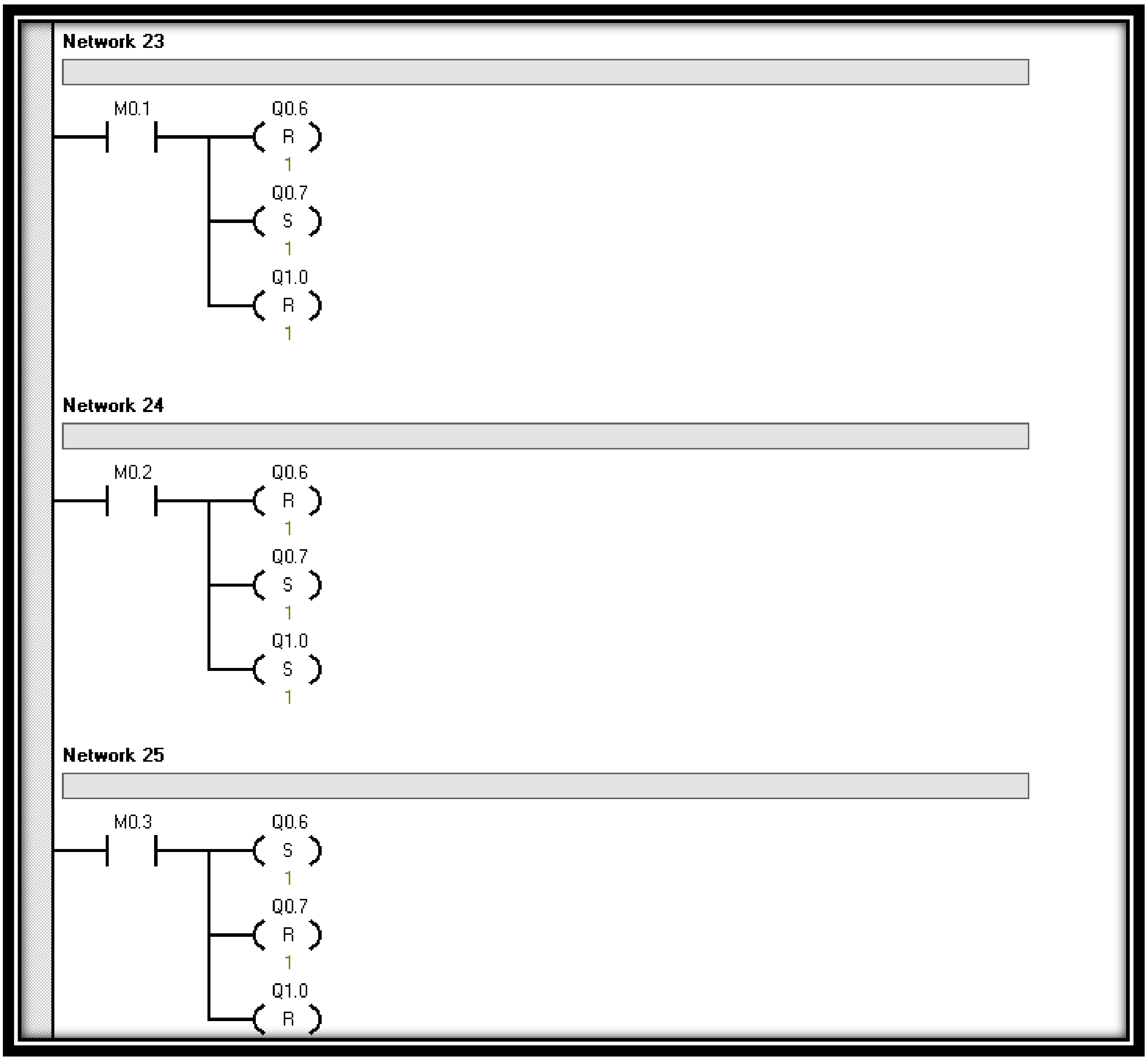
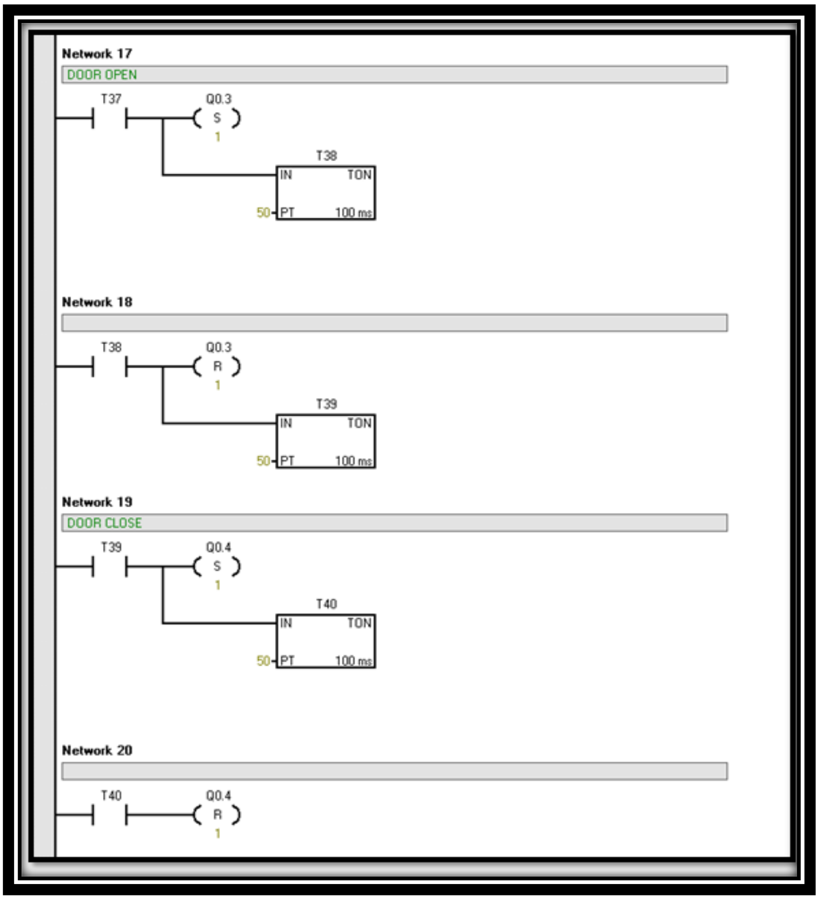


Figure 3:41 7-segment display signals

The 7-segment displays the floor at which the car is resedent whether it will make a stop at it or not; the signals to drive the 7-segment display are taken directly from the limit switches indicating the floors passed.

Modifications

Networks 19 and 20, figures 37 and 38 respectivly ware modified due to limitations in PLC inputs; the modifications ware as follows:



These networks replaces networks 19 and 20 they performed the same task of the previous networks but the door hold and close options were excluded.

After the motor stops and T37 output goes high, the door open signal goes high at the same time a timer T38 counts the time required to open the door.

When T38 output goes high it indicates that the door has reached the end of its openning position, the door open signal is reset. And another timer T39 start counting the time the door has to be open for passengers to load and unload.

Finally after timer T39 output goes high the door close signal goes high.

3.4 The final product

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Overview

To obtain results for this system in a readable way; several scenarios and use cases covering all potential usage of the system were tested in both software simulators and hardware implementation.

In software testing, you can 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.

In this chapter pictures of the results of each use case test were taken and attached to there scenarios.

4.2 Software results

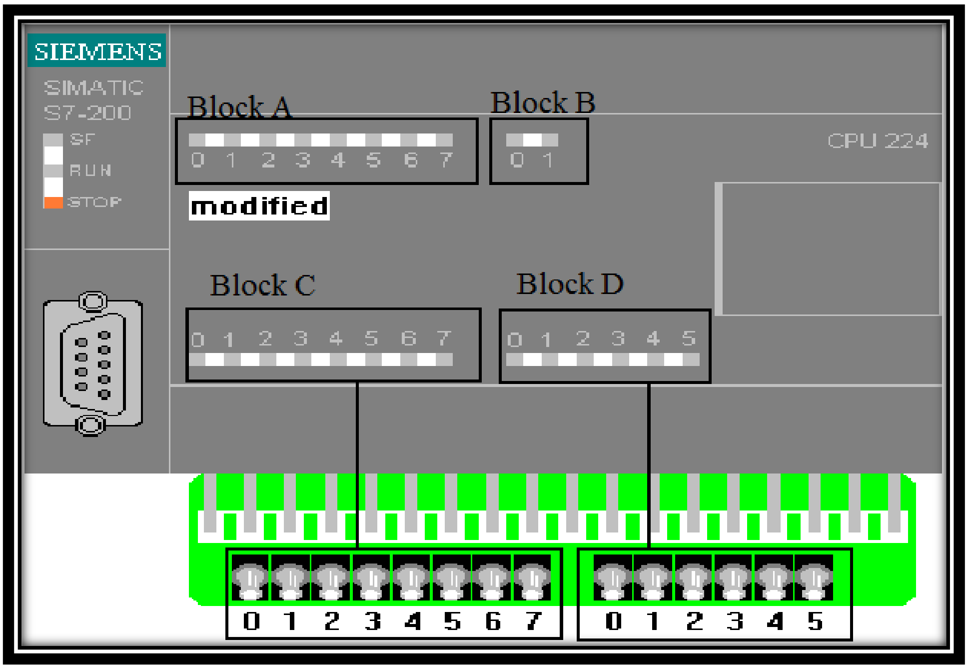


Figure 4:1 PLC simulator window

Block A(outputs):

0: Pulley up signal

1: Pulley down signal

2: Motor stop signal

3: Door open signal

4: Door close signal

6: Bit 2 in BCD (floor number)

7: Bit 1 in BCD (floor number)

Block B (outputs):

0: Bit 0 in BCD (floor number)

Block C (inputs):

0: 1st floor limit switch

1: 2nd floor limit switch

2: 3rd floor limit switch

3: 4th floor limit switch

4: inner call to 1st floor

5: inner call to 2nd floor

6: inner call to 3rd floor

7: inner call to 4th floor

Block D (inputs):

0: call from 1st floor (going up)

1: call from 2nd floor (going up)

2: call from 2nd floor (going down)

3: call from 3rd floor (going up)

4: call from 3rd floor (going down)

5: call from 4th floor (going down)

Test case 1:

Moving from 1st to 2nd floor

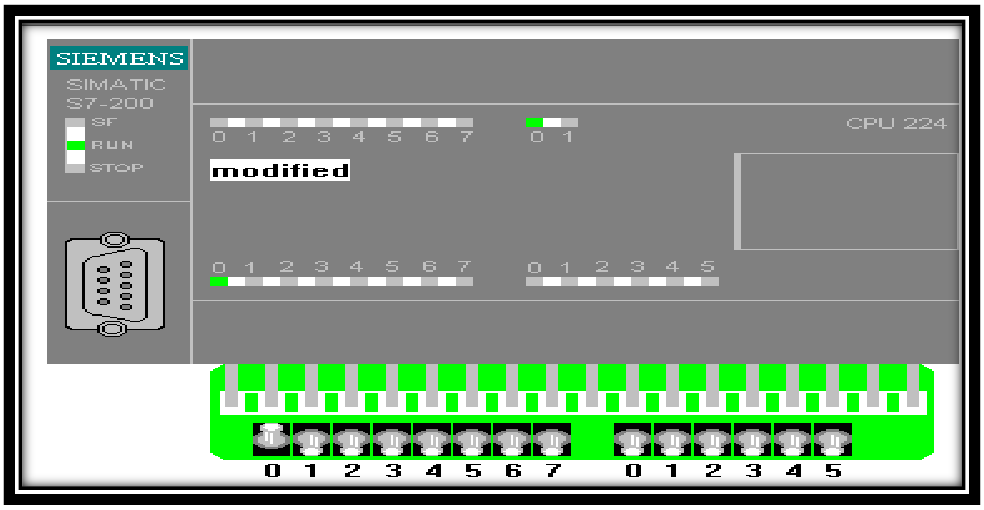


Figure 4:2 Case 1-1

The car is at 1st floor so the first floor limit switch input is high and the output to the 7-segment is set to drive it to show (1) [001] binary.

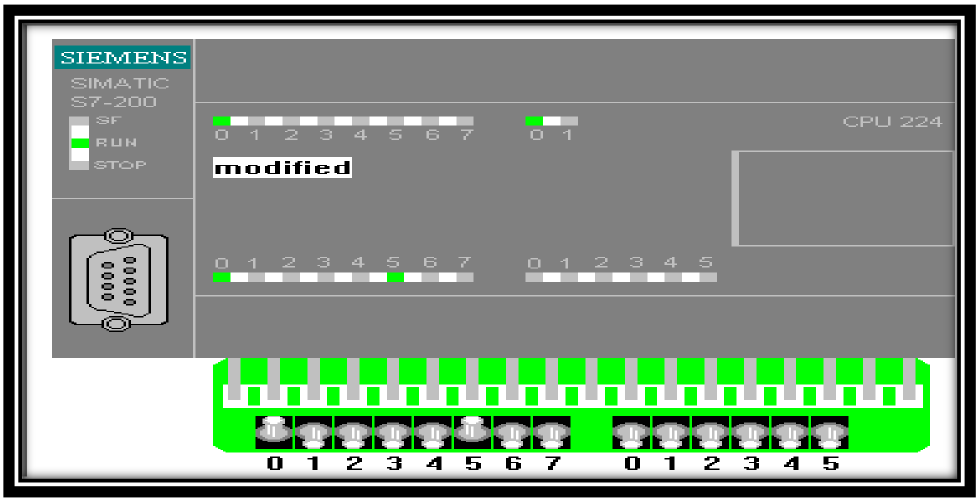


Figure 4:3 Case 1-2

While at the 1st floor, a call to the 2ndfloor occurred; 2nd floor call input is set high. The PLC set the pulley up signal high. The 7-segment still display floor No.1 as the floor at which the car resides.

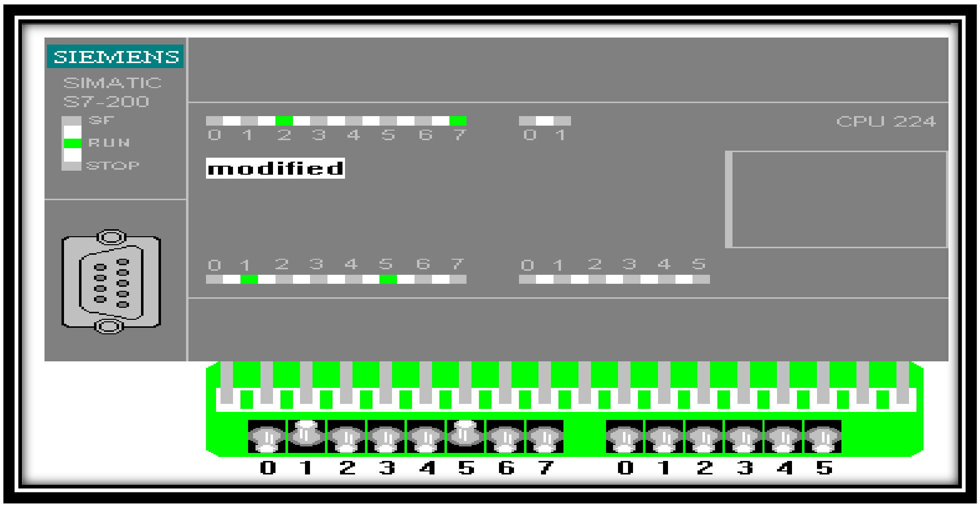


Figure 4:4 Case 1-3

The car reached the 2^{nd} floor so the limit switch signaled the PLC; the n the PLC signaled the pulley motor to stop.The 7-segment now indicates that the car is at the 2^{nd} floor; showing (2) [010] binary.

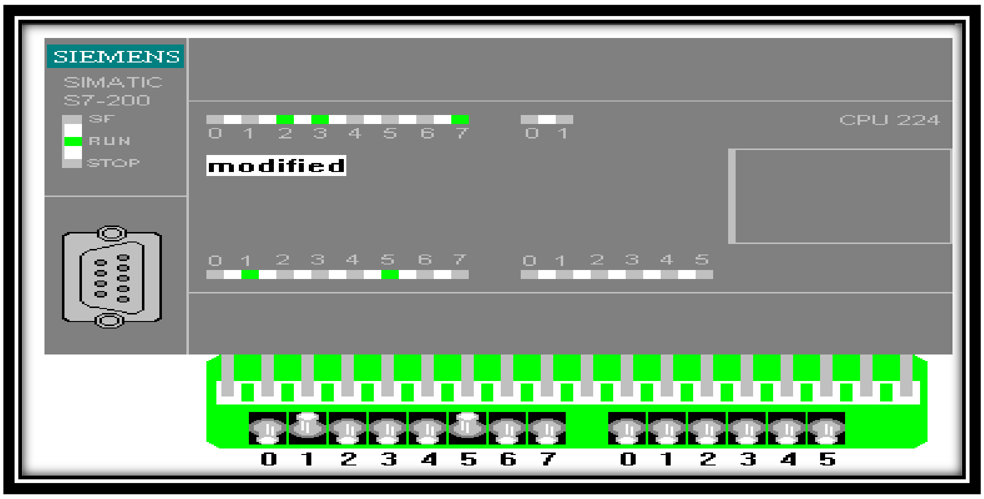


Figure 4:5 Case 1-4

After the motor stops and the first counter indicating the time needed for the car to rest after the motor stops signals, the PLC sets the door open signal high, and the second counter indicating the time needed to open the door starts counting. The 7-segment still indicates that the car is currently at the 2^{nd} floor.

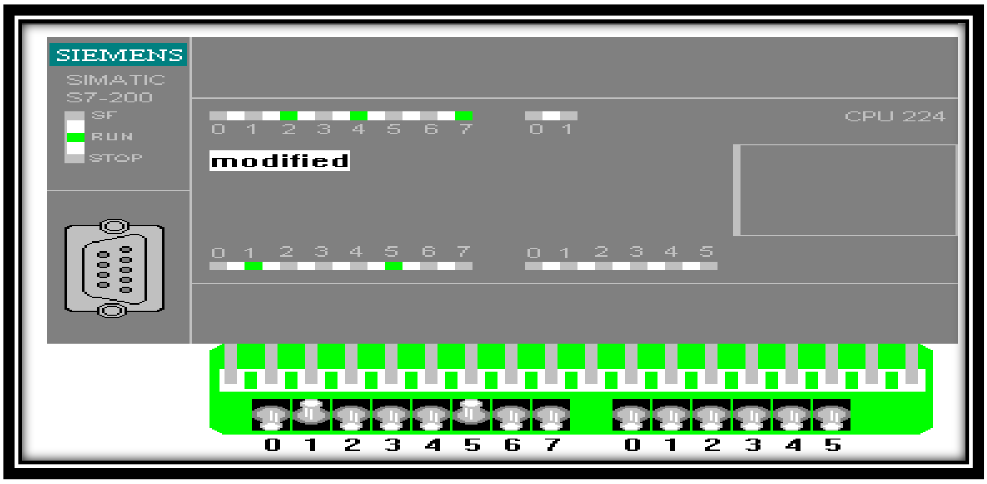


Figure 4:6 Case 1-5

The moment the second counter signals the PLC signals the door to close. A third counter starts counting the time needed for the door to close, once it signals the pulley can move if a call occurred. These counters insure that the pulley doesn't move while the car in loading or unloading passengers.

Test case 2:

Moving from 2nd to 3rd floor

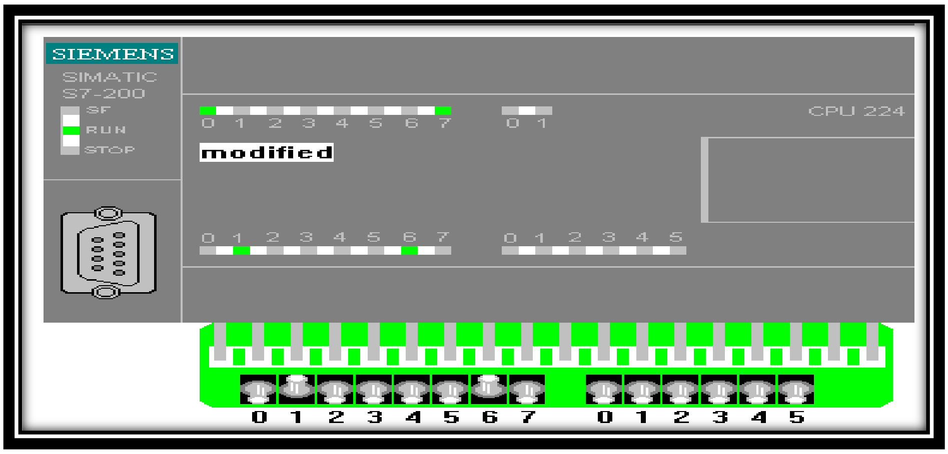


Figure 4:7 Case 2-1

The car is currently at the 2nd floor as the 7-sement display indicates [010];And since a call to the 3rd floor is asserted the PLC signals the pulley motor to go up.

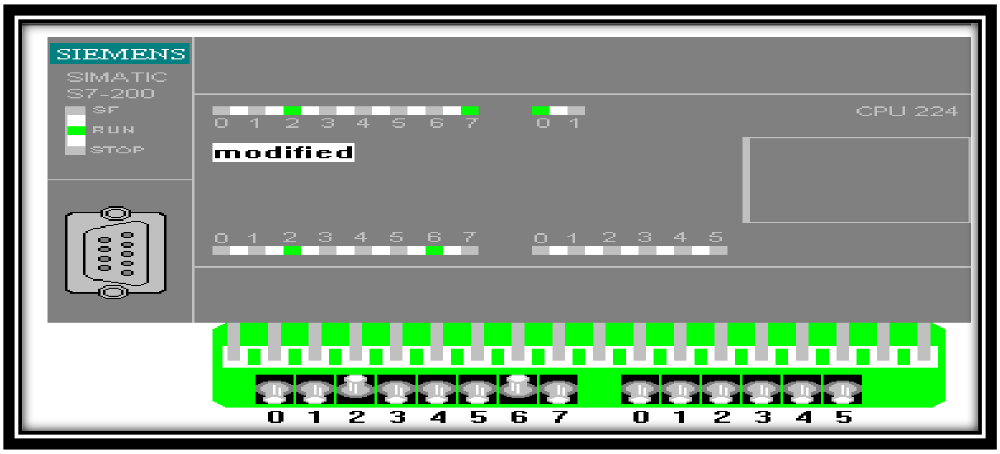


Figure 4:8 Case 2-2

The car reached the 3rd floor so the limit switch signals the PLC; then the PLC signals the pulley motor to stop. The 7-segment now indicates that the car is at the 3rd floor; showing (3) [011] binary.

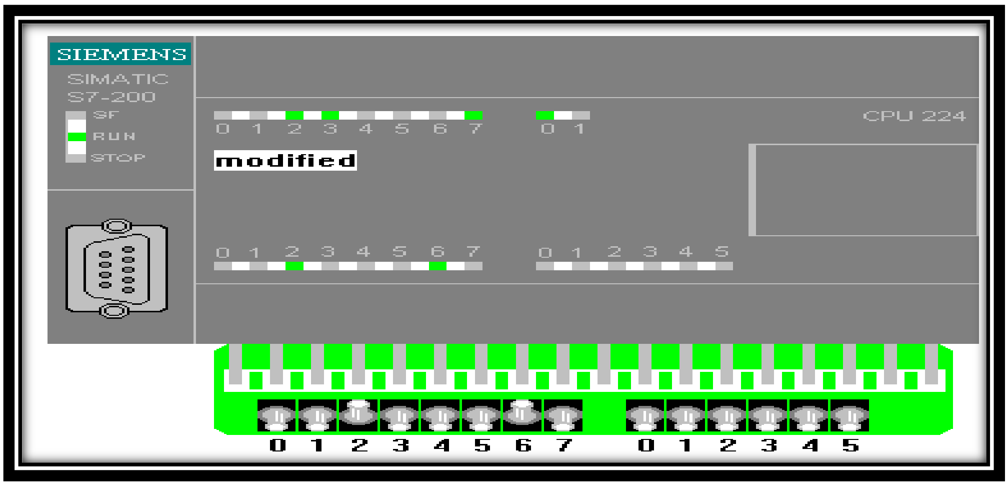


Figure 4:9 Case 2-3

After the motor stops and the first counter indicating the time needed for the car to rest after the motor stops signals, the PLC sets the door open signal high, and the second counter indicating the time needed to open the door starts counting. The 7-segment still indicates that the car is currently at the 3rdfloor.

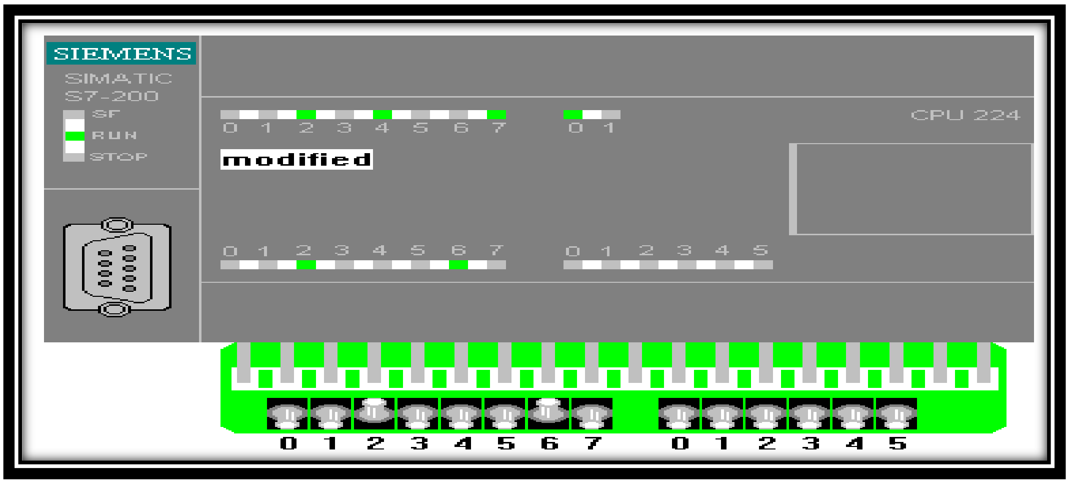


Figure 4:10 Case 2-4

Similar to case (1-5) the moment the second counter signals the PLC signals the door to close. A third counter starts counting the time needed for the door to close, once it signals the pulley can move if a call occurred. These counters insure that the pulley doesn't move while the car in loading or unloading passengers. The 7-segment still indicates that the car is currently at the 3rd floor.

Test case 3:

Moving from 3rd to 2nd floor

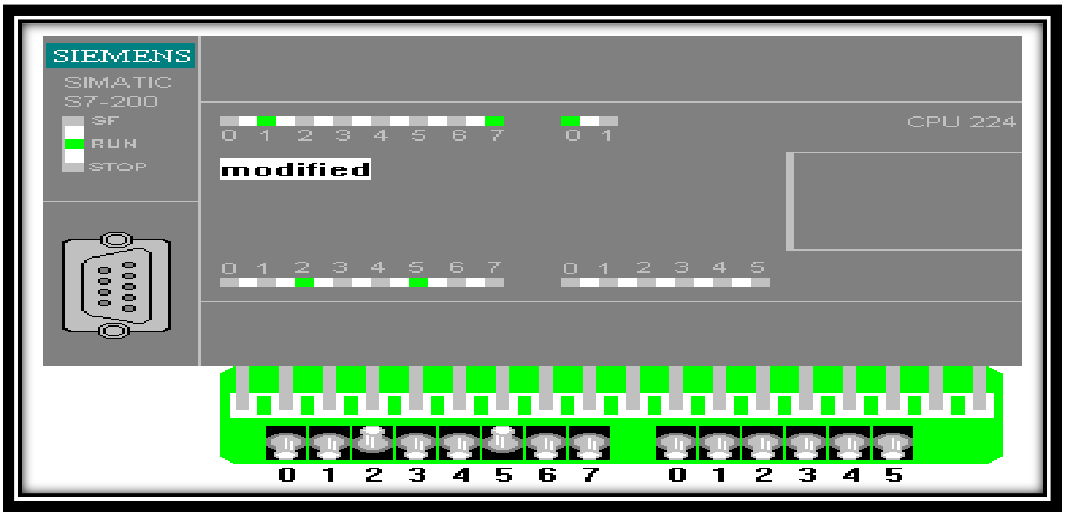


Figure 4:11 Case 3-1

While at the 3rd floor, call to the second floor occurred; 2nd floor call input is set high. The PLC set the pulley down signal high. The 7-segment still display floor No.3 as the floor at which the car resides.

Once the car reaches the 2ndfloor, the limit switch signal the PLC and the motor stop, door open & door close sequence is generated, as in figures(4.4),(4.5) and (4.6) respectively.

Test case 4:

Moving from 2nd to 1st floor

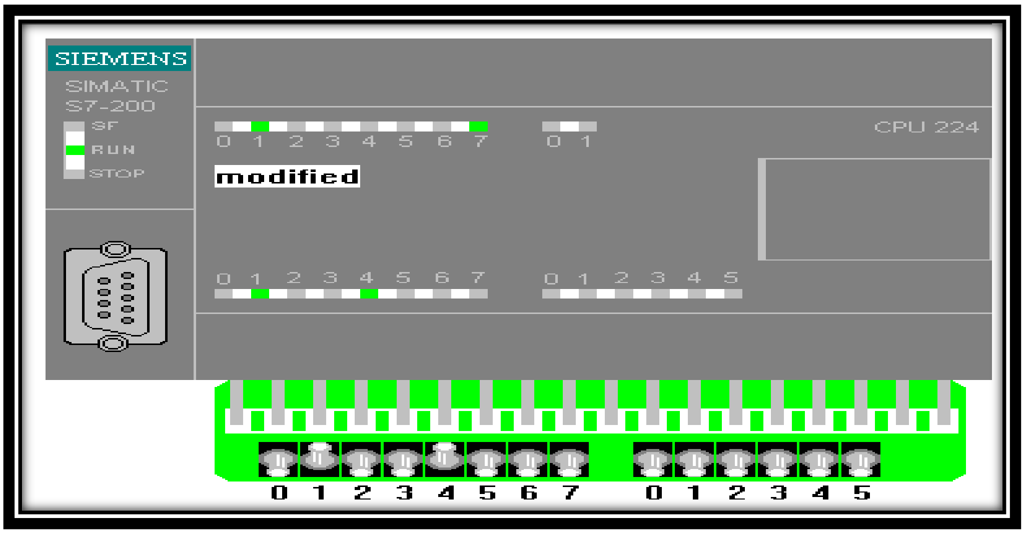


Figure 4:12 Case 4-1

While at the 2nd floor, a call to the 1st floor occurred; 1st floor call input is set high. The PLC asserts the pulley down signal high. The 7-segment at this point displays floor No.2 as the floor at which the car resides.

Once the car reaches the 1stfloor, the limit switch signal the PLC and the motor stop, door open & door close sequence is generated, as in previous cases.

4.3 Hardware results

The PLC operation was first tested without wiring it to the system and it operated properly; each system component including limit switches, LEDs, push buttons, 7-segment displays, pulley and door motors and relays were also tested separately before wiring them to form the system.

All these components performed well before wiring except the limit switches which showed poor performance.

After wiring the system the PLC and associated components -excluding limit switches- were once again tested with the same sequential use cases tested in software simulation, and the system performed as expected except the door which showed poor performance.

CHAPTER 5

CONCLUSION

5.1 Performance

The software system performance was satisfactory; it performed all required tasks successfully during testing.

The prototype built showed relatively poor performance during testing due to low equipment reliability. The limit switches malfunctioned, and the door motor torque was law and created difficulties in configuring the door.

The pulley DC motor showed an impressive performance. It had very high torque and law speed, the high performance of this motor originates from the gears used to reduce its speed. Reduced speed led to the very required high torque.

5.2 Difficulties

5.2.1 Hardware

5.2.1.1 Mechanical problems

- Finding the appropriate motor for the prototype:

  The following motors were tried and both the torque and speed were taken into consideration:

  9V motor with high speed but poor torque

  12V motor with high speed and better but still poor torque

  12V motor with high speed and suitable torque (dish washer motor) but its driving current was too high (3A) that non of the available power supplies could drive it.

  Finally 24V, 105mA printer motor was found to be the appropriate motor in terms of both torque and speed.

The torque and speed relate with the following equation:

Power = Torque * Speed

Pulley material and diameter:

- ❖ A pulley with grooves was first tried but did not fit in the design because it was fixed to be only hanged down, and the design requires the pulley to be fixed on top of the shaft.
- ❖ An aluminum rode was shaped, and used as a pulley that met the designs requirements.
- ❖ The diameter of the pulley was controversial mater because the diameter of the pulley controls the speed of the elevators car; which is a vital part of the success of the system. But there was no strict constrains on that speed it only had to be slow enough for the limit switch to signal, the motor to stop before the car reaches the next floor and fast enough for the movement to be sufficient and smooth the diameter was chosen to be 2 inches. The first pulley used diameter was 1 inch. It required more rounds to lift the car the same length than the ones taken by the pulley with 2 inches diameter.
- ❖ The technique used to limit the movement of the counter weight and prevent it from swaying back and forth was to cage it in a box, it only stopped it from interfering with the cars movement, unfortunately bad welding added friction which obstructed the system movement.

5.2.1.2 Electrical problems

Proximity switches were the first choice to sensor floor levels; but they were not available in the market which forced the use of available limit switches, they fitted in the application and performed the same task. But there reliability was questionable and they had poor performance; therefore they were dropped of the project and during testing there task was performed manually.

5.2.2 Software

The project aim was to develop an elevator control system with at least 24 inputs. This aim was met at software level; but when it was time to apply the system on an actual PLC (not simulation), financial difficulties came into picture; and the only PLC available at universities labs (S7-200 CPU 224) had 14 inputs.

As a result the algorithm was modified to accommodate the new restriction and take in only 14 inputs; a bypassing circuit was used to accommodate the inputs taken out from the algorithm.

5.3 Future work

Expand the system to control elevators for more than four story buildings, and with more features than those offered in this system; to do so few changes are required:

Use a PLC with more inputs than the one used -with only 14 inputs-to be able to add features such as door hold within the system without the need for an external circuitry like the one used.

Control motor speed basing using sensors and gears; this will improve the performance of the elevator system for multistory buildings -minimize the service time-.

Replacing the limit switches with single light sensor or magnetic sensor on the side of the car and a series of holes on a long vertical tape in the shaft; when the sensors passes by holes it counts them to determine which floor is the car at. This is more economic and also minimizes the number of inputs required from the PLC.

Use a load sensor that will set on a signal whenever the weight of the passengers exceeds the limit set by the designer; as long as the overload signal is set the car does not move and the doors do not close-this will maintain the gears, the pulley and the motors in shape and it will minimize the failures of the system due to hardware problems.

Use motion sensors to prevent the door from closing while there is an object that is not entirely in the car -this increases the safety of the system-.

Use more advanced logic that takes passenger traffic patterns into account. To know which floors have the highest demand, at what time of day, and direct the elevator cars accordingly. Elevator traffic simulation software can be used to model complex traffic patterns and elevator arrangements that cannot necessarily be analyzed by RTT (round trip time) calculations.

Use two sets of doors; one attached to the car (inner) and one at each floor (outer). The car doors have a clutch mechanism that unlocks the outer doors at each floor and pulls them open. In this way, the outer doors will only open if there is a car at that floor (or if they are forced open). This keeps the outer doors from opening up into an empty elevator shaft. Also front and rear sets of doors could be used.

A breaking system must be added to insure safety and quality of services.

References:

1. Elevator. *wikipedia.* [Online] http://en.wikipedia.org/wiki/Elevator.

2. belevator. *inventors about.* [Online] September 2010. [Cited: November 3, 2010.] http://inventors.about.com/library/inventors/blelevator.htm.

3. Elevators. *How stuff work.* [Online] 2010. [Cited: November 2, 2010.] http://science.howstuffworks.com/transport/engines-equipment/elevator.htm.

4. *Control of a four-level elevator system using a programmable logic controller.* **L Cheded, Al-Mulla, Ma'an.** Manchester : International Journal of Electrical Engineering Education, 2002.

5. Relay. *Wikipedia.* [Online] June 2011. http://en.wikipedia.org/wiki/Relay.

6. Limit-switch. *partminer.* [Online] 2005. http://www.partminer.com/glossaryhtml/limit_switch.html.