بسم الله الرحمن الرحيم

Sudan University of science and technology
College of Graduate studies and scientific Research

Simulation of Passenger Lift By using BASCOM Programme

محاكاة مصعد ركاب باستخدام برامج BASCOM

Thesis submitted in partial fulfillment of the requirements for the award of degree of Master of Science in Mechatronic Engineering

By:
Almogdad Salah Eldeen Mohammed Hassan

Supervisor:
Dr. Abdelfttah Bellal

February 2015

الآيـــــة

﴿ وَقُل رَّبِّ زِدْنِي عِلْمًا ﴾

سورة طه -114

Dedication

I dedicate this research with much love and appreciation;

To the candles of my lives. My beloved mother who have always been there for me.

To my father who have always been the brick walls on whom me can learn and depend on forever.

To my brothers and sister who mean the world to me.

To my friends, family, colleagues and teachers in the Past and presents and to everyone that touch my heart.

Acknowledgement

Firstly, thanks to Allah, our creator above for being everything and for giving us the ability and strength to do anything.

I wish to express my deepest gratitude and appreciation for my supervisor for this research ***Dr. Abed alfattah Bilal*** *for his patience and continuous guidance, advice and supervision through this work.*

I would like to extend my gratitude to ***T. Braah Alfateh, ENG. Modthafer Ahmed*** *and* ***T. haram Mohammed Alfateh*** *for their supports and encouragement for the completion of this research.*

Also thank and gratitude to all our teachers who contributed too Ur education and to everyone who helped me in this study.

Mogdad

Contents

List of figures

List of tables

Abstract

In this work, control of elevator is accomplished using ATMEGA 16 (microcontroller), the control system is constructed to simulate as an actual lift in the real life which simulate by program via Bascom-AVR as programming language then Proteus was used to verify the code, and the microcontroller is used as the primary controller So the complexity of the system has been simplified which will result in reducing the cost. In the work was applied to a three-floors building in addition to the ground floor, it provides useful information to those who wish to carry out a lift control system. In the simulation the lift was placed in the ground floor and ordered to reach the first floor, the lift obeyed the instruction exactly. The lift was then ordered from the ground floor and the 2^{nd} floor at the same time, it resumed its motion to the 2^{nd} floor as excepted.

الملخص

في هذا العمل تم التحكم في المصعد باستخدام المتحكمة (ATMEGA 16) وذلك لمحاكاة عملية التحكم في مصعد حقيقي حيث استخدمت لغة البرمجة BASCOM لإنشاء البرنامج ومن ثم أستخدم برنامج المحاكاة للدوائر الإلكترونية PROTEUS للتأكد من صلاحيه البرنامج، والمتحكمة مستخدمة هنا كمتحكمة أساسيه وبذلك يتم تبسيط نظام التحكم في المصاعد مما يؤدي لتقليل التكلفة أيضا. في هذا العمل تم التطبيق على مبنى مكون من ثلاث طوابق بالإضافة إلى طابق أرضي، وهو يوفر معلومات قيمة ومفيد كذلك للذين يرغبون في تنفيذ نظام مراقبة وتحكم المصعد، المصعد قيد التحكم في مبنى ذو ثلاثة طوابق مضافا اليه الأرضي، ويوضح متى يتجاهل الحركة. في المحاكاة تم وضع المصعد في الطابق الأرضي وأمر للوصول إلى الطابق الأول ويقوم المصعد بتنفيذ الأوامر بدقة عالية جدا. ثم طلب المصعد من الطابق الأرضي والطابق الثاني في نفس الوقت، فإنه يستأنف حركته إلى الطابق الثاني كما هو متوقع.

Chapter One

INTRODUCTION

CHAPTER ONE

Introduction

1.1 General Overview

Lift or elevator, is a transport device that is very common to us nowadays because working space to high-rise buildings and areas below ground. It is used in every day to move peoples or goods vertically in high buildings such as hospitals, hotels, shopping centers, working offices, even domestic building and many more. A very useful device moves people to the desired floor in the shortest time in addition to minimizing human's effort.

The elevator as a device consist of mechanical, electrical and electronics systems. The heart of Elevator is the control System that is the focus of development in elevators industries.

1.2 Problem Statement

The civilian and architectural revaluation cause the necessity to using elevator widely. Elevators cost is too high due to importing all the components from outside. The control card is the most expensive part in this field so it is cost effective to produce locally.

The microcontroller is a good solution to achieve very low unit costs and low implementation costs. As such, the microcontroller is a very flexible electronic 'tool' to be applied in a very wide range of product and process applications.

For this, reason the programming of software to perform the operation logic is the main challenge.

1.3 Objective

To simulate a control system for 4 stops elevator (3 floor) by using microcontroller (ATMEGA 16) according to the real lift traffic as case study

1.4 Methodology

In this thesis BASCOM-AVR 2.0.5.0, programing Language was used to program the microcontroller (ATMEGA 16 from ATMEL® AVR®). PROTEUS 8.1 Professional software was used to verify the program and simulate the control system.

1.5 Project Layout

- Chapter One consists of brief introduction
- Chapter Two consists of theoretical background and literature reviews.
- Chapter Three consists of circuit design.
- Chapter Four consists of simulation and results.
- Chapter Five consists of conclusion and recommendations.

Chapter Two

THEORETICAL BACKGROUND & LITERATURE REVIEWS

CHAPTER TWO

Theoretical Background and Literature Review

2.1 Theoretical Background

Elevators are transported devices by far the most important transportation systems for handling both passenger and freight traffic in buildings. They provide safe, fast, and economical movement for people and goods vertically, In British and other Commonwealth countries; elevators are known more commonly as lifts, although the word elevator is familiar from Americans. 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, or The vacuum pump in pneumatic to generates higher and lower atmospheric pressures above or below the elevator car [1, 2].

There are three types of convoying methods in buildings

i. Elevators

ii. Escalators

iii. Ramps

Moreover, they have different dimensions and uses according to building type and number of users in it.

Elevators can be classified into:

1. Rope traction Elevators.
2. Hydraulic Elevators.
3. Pneumatic Elevators [1, 2].

2.2 Traction Elevators

This elevator type generally powered by electric motor that drive by traction cable and counterweight system like a hoist as shown in figure 2.1 all components of rope traction elevators), there are some details of component [1].

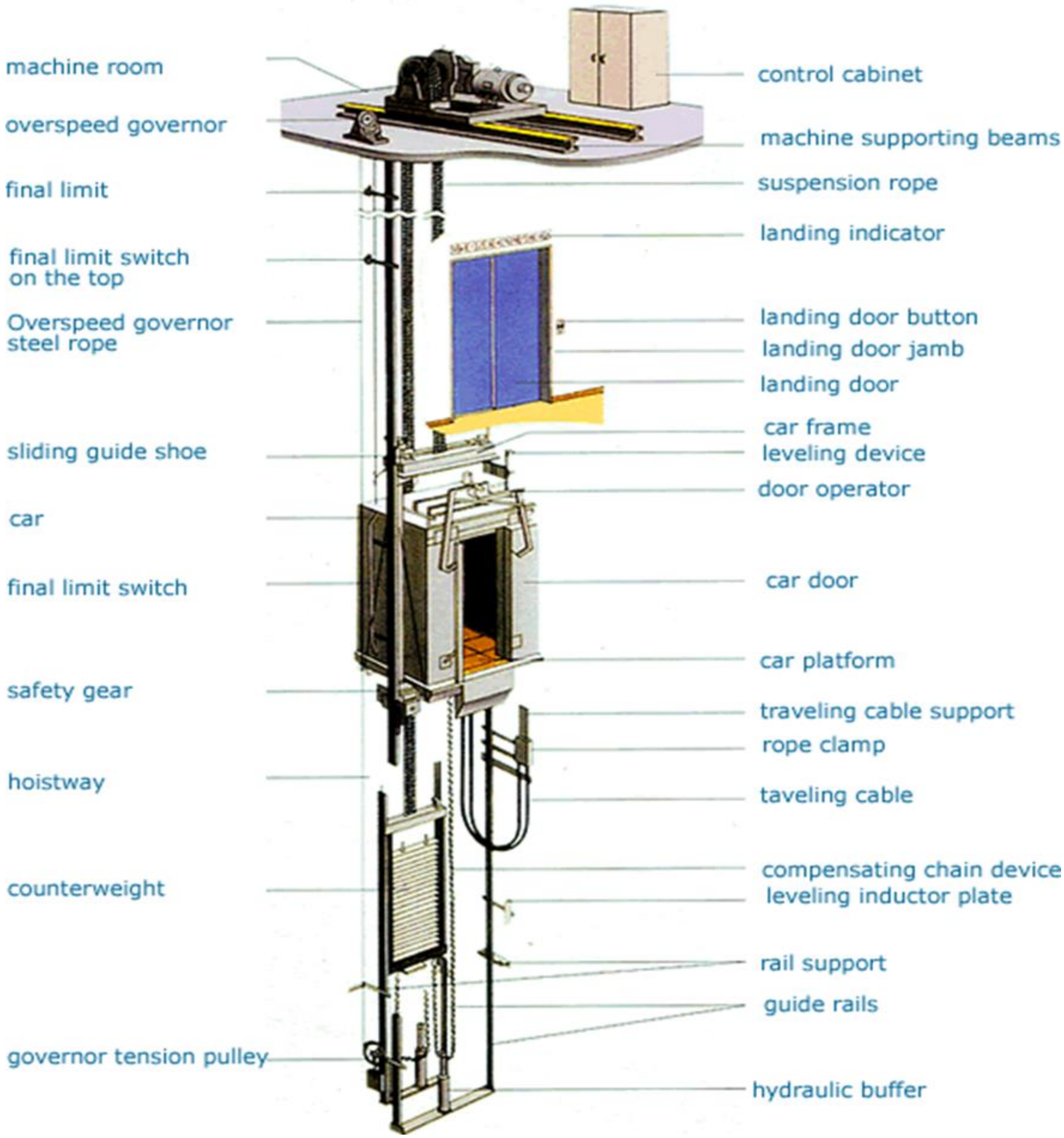


Figure 2.1 rope traction Components.

2.2.1 Car

Elevator Car is the vehicle that travels between the different elevators stops carrying passengers and/or goods; it is usually a heavy steel frame surrounding a cage of metal and wood, glass panels as shown in figure 2.2, the car has limited size, capacity, and speed these factors depend of passengers, and the numbers of people can be determine be:

Numbers of passengers = rated load /75 [2, 3].

Where 75 represent the average weight of a person in Kg [2, 4].

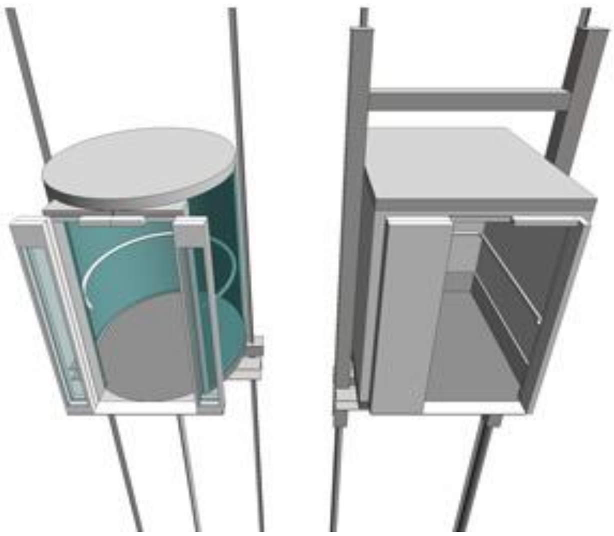


Figure 2.2 Elevator car/ cabin.

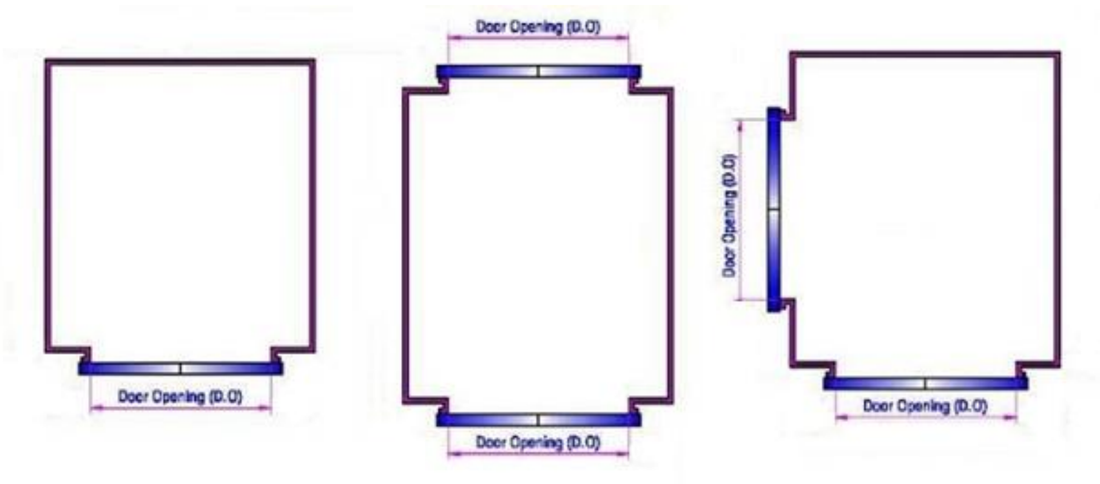
- **Elevator Car Types**

Elevator car can be classified according to the number of entrances and their locations as follows: [3].

i. Normal Cabin: There is only one door for entry/exit as shown in figure 2.3.a

ii. Open Through Cabin : There are two doors located opposite each other as shown in figure 2.3.b

iii. Diagonal Cabin: There are two doors located 90° (Diagonal) from each other as shown in figure 2.3.c [3].



a) Normal Cabin b) Open through Cabin c) Diagonal Cabin

Figure 2.3 a) Normal Cabin b) Open through Cabin c) Diagonal Cabin

2.2.2 Hoistway (shaft)

A simple definition for the hoistway is the shaft that encompasses the elevator cars as shown in figure 2.4, generally the Hoistway serving all floors of the building but in high-rise buildings hoistways may be banked with specific hoistways serving only the lower floors and others serving only middle or upper floors while traveling in a blind hoistway until reaching the floors that it serves. In addition, the blind hoistway has no doors on the floors that it does not serve [5].

- **Hoistway components**

A. Guide rails

Guide rails for both the car and counterweight are Steel Tracks in the form of a "T" that run the length of the hoistway, round, or formed sections with guiding surfaces to guide and direct the course of travel of an elevator car and elevator counterweights and usually mounted to the sides of the hoistway [3].

B. Counterweight

A tracked weight is suspended from cables and moves within its own set of guide rails along the hoistway walls.

- Counterweight is used for the following:

i. Balancing the mass of the complete car and a portion of rated load, and it will be equal to the dead weight of the car plus about 40- 50% of the rated load [2].

ii. Reducing the necessary consumed power for moving the elevator [3].

Figure 2.4 Hoistway.

C. Suspension (Hoisting) Ropes (Cables)

Suspension ropes means Suspension for car and counterweight, which are represented by steel wire ropes they are Used on traction type elevators, usually attached to the crosshead and extending up into the machine room looping over the sheave on the motor and then down to the counter weights and they related to design of speed.

D. Guide Shoes

Guide shoes are Devices used mainly to guide the car and counterweight along the path of the guide rails, it shall be fitted at the top and bottom of their frames and shall be spring loaded, self-lubricating with plastic inserts, and designed to give smooth and quiet running under all conditions of load [3] .

There are two types' first roller guides and sliding guides as shown in figure 2.5.

Figure 2.5 a) roller guides b) sliding guides

E. Landing (Hoistway) Doors

That door is seen from each floor of a building is referred to as the outer or hoistway door. This hoistway door is a part of the building (each landing). It is important to realize that the car door does all the work; the hoistway door is a dependent. These doors can be opened or closed by electric motors with car door, or manually for emergency incidents [3, 4].

The difference between the car doors and the hoistway doors is that the elevator car door travels through the hoistway with the car but the hoistway doors are fixed doors in each landing floor as shown in figure 2.6 [3, 4].

Figure 2.6 landing (hoistway) door

F. Buffers in the pit

A Buffer is a device designed to stop a descending car or counterweight beyond its normal limit and to soften the force with which the elevator runs into the pit during an emergency. They may be spring or oil type in respect of the rated speed as shown in figure 2.7.

- There are two principle types of buffers in existence:
  - Energy accumulation: accumulate the kinetic energy of the car or counterweight.
  - Energy dissipation: dissipate the kinetic energy of the car or counterweight.

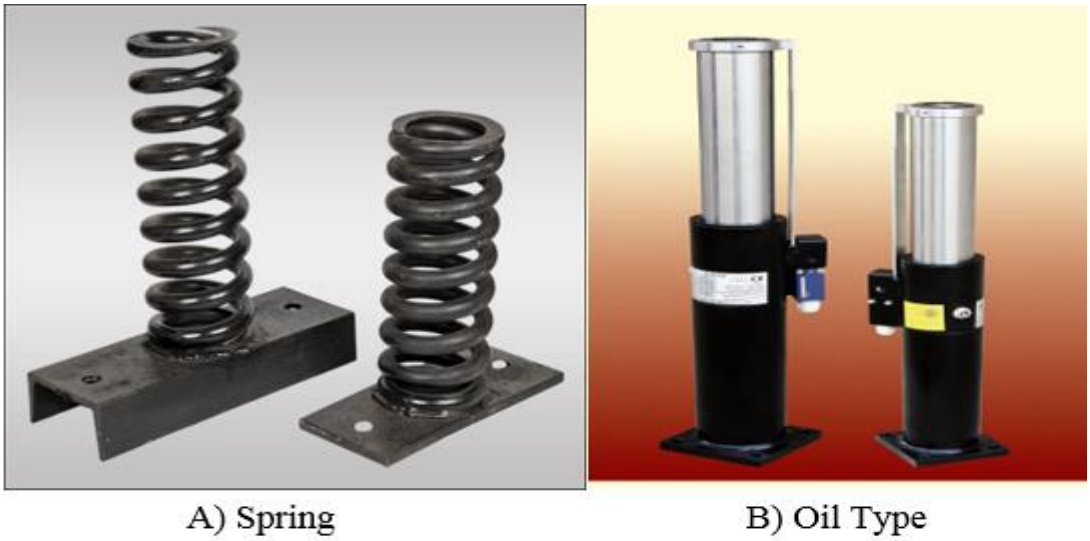


Figure 2.7 elevator buffers a) spring   b) oil type.

2.2.3 Machine/drive system

Driving machine this is the power unit of the elevator, and used to refer to the collection of components that raise or lower the elevator. These include the drive motor, brake, speed reduction unit, sheaves and encoders [5].

- For traction drive there are

i. Dc drive:

DC drives (contract speed of 2m/s or above); rugged and dependable, reliable and good control, good comfort, but DC generator needed, Imperial's line of DC Hoist motors are renowned in the industry.

ii. AC motor:

Two-speed (contract speed of around 1m/s or less); economical, poor comfort, difficult to control.

iii. Variable Voltage Variable Frequency (VVVF) the most widely used drive system, it relies on the fundamental principle that the speed of an induction motor is directly linked with the supply frequency applied to the stator windings. By varying the frequency and by keeping the voltage / frequency ratio constant. (contract speed of 1.5m/s or above); lower energy and cost than DC, faster floor-to-floor jump time, reasonably good comfort; used to be very expensive but with the advent of power electronics VVVF has become the prime workhorse nowadays [5].

- There are two types of driving machines provided for elevators:

I. Geared Machine

It used in low and mid-rise applications, Have a gearbox that is attached to the motor as shown in figure 2.8.a this design convert the initial electrical power (input power), which runs the motor (input speed), into mechanical power (out power), which can be used by the system. mechanical speed reduction by a worm gear system figure 2.8.b is made up of a worm and gear, it is not only decrease the RPM of the drive motor (input speed) to suit the required speed of the drive sheave and elevator (output speed), but also increasing the output torque therefore, having the ability to lift larger objects [5, 6].

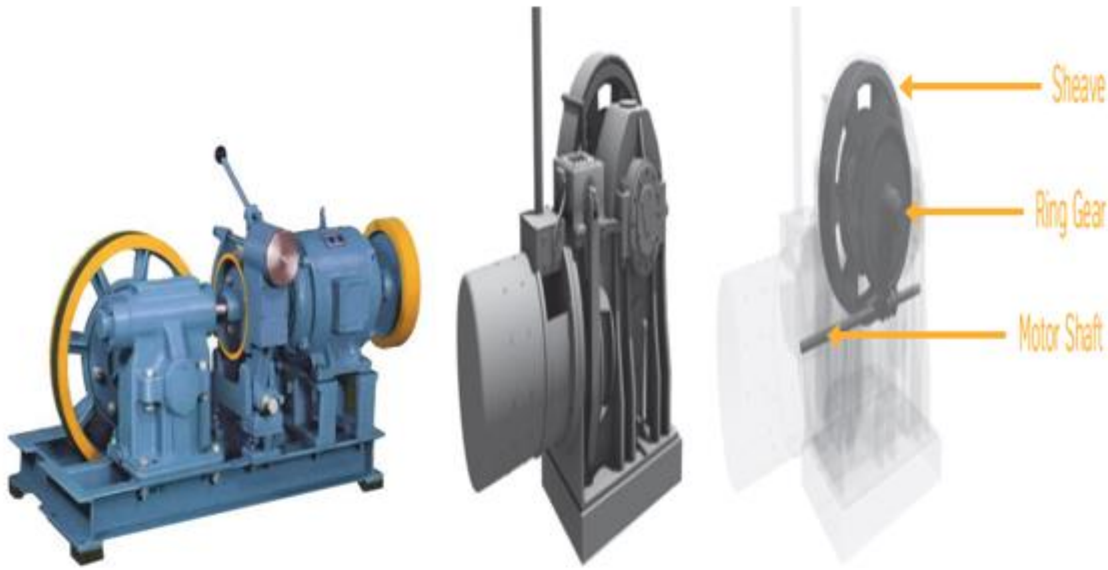
Generally, the gear reduction ratios typically vary between 12:1 and 30:1; geared machines are used for speeds between 0.1 m/s and 2.5 m/s and are suitable for loads from 5 Kg up to 50,000 Kg and above.

- Types of geared machine drive according to location of installation:

  a. Overhead traction.

  b. Basement Traction.

  c. Offset Traction.



a) Geared machine b) worm gear system in geared machine

Figure 2.8 a) geared machine b) worm gear system in geared machine.

II. Gearless Machine

A gearless traction machine consists of a DC or AC motor, which driving machine is a direct-drive system in which there is no reduction gear between the motor and the drive sheave as shown if figure 2.9. That is, the drive sheave is connected directly to the motor and brake. Gearless designs to use in the world’s tallest structures. They are efficient and used for driving high speeds between 2.5 m/s to 10 m/s [5, 6].

Figure 2.9 Gearless Machine.

Gearless machine introduce a new concept in elevators industry that called Machine Room-Less (MRL), Machine-Room-Less Elevators are traction elevators that does not have a machine room above the elevator shaft unlike the traditional designs as shown in figure 2.10.

Usually located Gearless machine above the hoistway in a penthouse or one meter above the highest floor it serves, but may be in the basement if overhead space is unavailable.

Machine-room-less elevators are becoming more common; however, many maintenance departments do not like them due to the hassle of working on a ladder as opposed to within a room [5, 6].

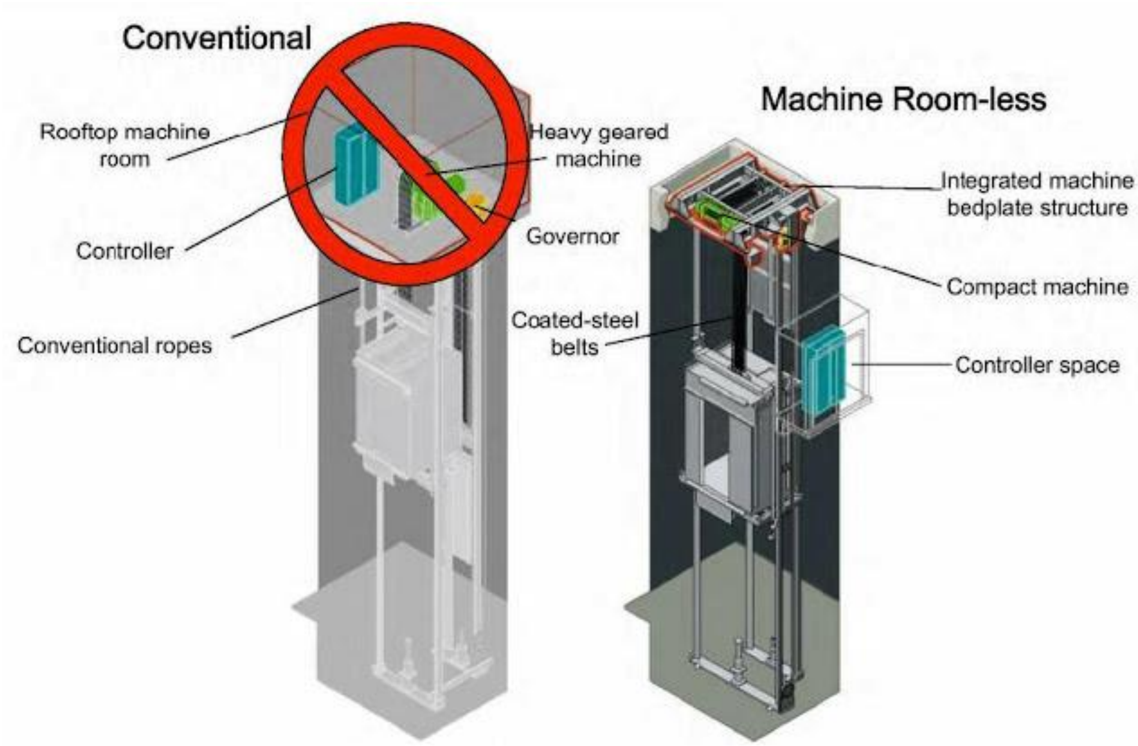


Figure 2.10 shows the difference between Machine Room elevators and MRL.

2.2.4 Safety Systems

Ensuring a safe and dependable ride every time is the goal and the most important system in every elevators installed, and safety system safe operation for both passengers and equipment in this system can be divided the safety system into two categories as below [2, 4] :

1. **Mechanical safety system**

This can be in Ropes and Overspeed.

i. **Ropes**

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 as shown in figure 2.11, one rope can support the weight of the elevator car and the counterweight. However, elevators are built with multiple ropes (between 4 ~ 8, typically). In the unlikely event that one of the ropes snaps, the rest will hold the elevator up [2].

Figure 2.11 Rope cross Section

ii. Overspeed governor:

Safeties are activated by a governor when the elevator moves too quickly (if the car speed exceeds 110% of its rated value as the car speeds up, so does the governor). Most governor systems ropes are attached to the safeties on the underside of the car so it moves when the car goes up or down, called the governor rope. This governor rope is looped around the governor sheave in machine room and another weighted sheave at the bottom of the shaft. The governor it is situated either in the machine room or in the headroom. Its works on the floating principle with a cam curve and roller guided rocker as shown in figure 2.12 [2].

Figure 2.12 Overspeed governor.

2. Safety Circuits

Before energize the machine the controller must check the state of following safety circuits to decide to move the cabin or not [3, 4]:

i. Landing and car Doors

The state of landing and car doors can be monitored throw a contact switches to specify if they are closed or not, Should the doors be forced open, the interlock circuit will be broken, causing the elevator to immediately stop, because that will be a risk for people [4].

They are also interconnected electrically to prevent operation of the elevator if any of the elevator's hoistway doors are open.

Landing doors can be capable of being unlocked from the outside with special key (called Hoistway Emergency Door Keys) for emergency (e.g. firing, maintenance …) [4].

ii. Infrared sensor (IR Sensor)

It a sensor located at the car door in both side (sender and receiver) as shown figure 2.13 to detect if anybody in the zone of the door the controller decide to reopen the door to allow the passenger to pass [2].

Figure 2.13 IR Multiple-Beam

iii. Overload sensor

Any elevator when it designed the passengers maximum load it takes into account because it specifies the power of the machine and other criteria like counterweight, speed and suspension…etc. for all these reasons an overload sensor is used to prevent exceeding the maximum capacity which cause unstable operation. It is mounted on the lower transom to sense the nearness of car floor during loading of car isolation springs as shown in figure 2.14 when overload existed the control shock motions and send signal like alarm buzzer and electronic message on the car indicator to reduce the weight (passenger) [3].

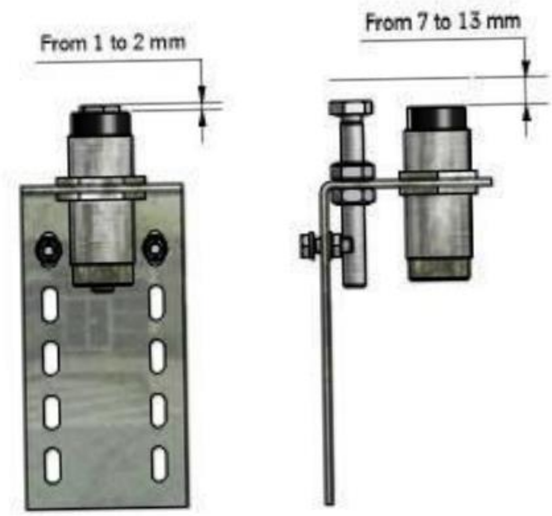


Figure 2.14 spring overload sensors

iv. Final Limit switch

A simple limit switch attached to the elevator car to prevent exceeding the terminal floors (the highest or lowest landing of lifts). Without risk of accident as shown in figure 2.15 [4].

In addition, this device reset the direction moving the car to opposite.

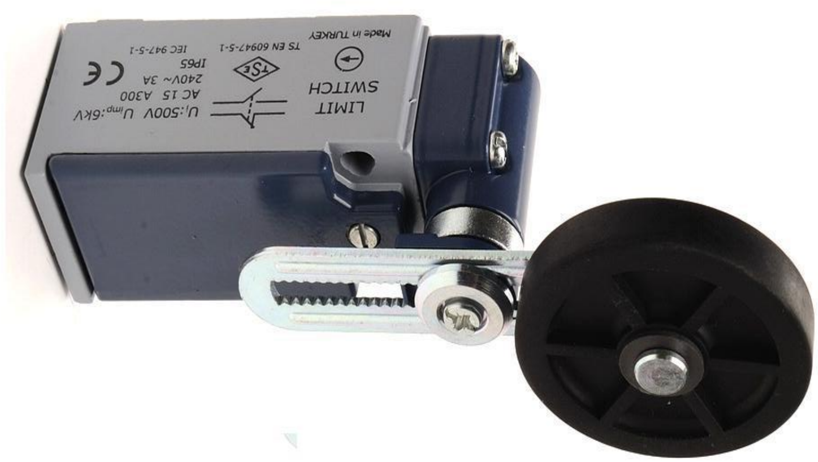


Figure 2.15 final Limit switch.

2.2.5 Elevator Control System

Elevator Control System is the system aims to [4]:

- To bring the lift car to the correct floor.
- Door opening speed and delay, reopen and closing.
- Leveling and hall lantern signals.
- To minimize travel time.
- To maximize passenger comfort by providing a smooth ride.
- To accelerate, decelerate and travel within safe speed limits.

It accepts all inputs signal (e.g. button signals) and produces outputs (elevator cars moving, doors opening, etc...), according to input.

- **Elevator control system components**

The elevator as a control system has a numbers of components; such as the inputs, the outputs and the controllers [7].

I. Hall Buttons

These buttons (inputs to controller) are on a button panel on the outside of the elevator hoistway and are used by passengers to call an elevator car to get to the floor that the pressed summon button is located on. There are two Hall buttons on each floor as shown in figure 2.16 – one for up to mean the passenger goes up floors from his floors, another for down vice versa , except on the top floor where there is only down and on the bottom floor where there is only up. The controller interacts with these buttons by receiving press and release signals indicating the requested direction and floor number. It also sends light on/off signals to indicate the status of the buttons.

Figure 2.16 Hall Buttons

Elevators acknowledge calls by lighting up the tell-tale LED, which is usually building into the hall call button. Another type of hall buttons have one button just to inform the controller it must serve the specified floor without direction of motion [4, 8].

II. Car Operating Panel (COP)

It is a panel mounted in the car as shown in figure 2.17 containing the car operating controls, such as call register floor request (keys), door open , close, alarm emergency stop and whatever other buttons or key switches are required for operation [4].

Figure 2.17 Car Operating Panel (COP).

III. Floor Request Buttons (keys)

These buttons are located on a button panel on the interior of each elevator car in (cop). The controller interacts with these buttons by receiving pressed signals (input) indicating the desired floor number and elevator car which they were pressed from.

The numbers of floor requests (keys) depend to the number of floors in building, keys labeled arrangement as shown in figure 2.18 that passengers can use to direct the elevator cars to the floor that they would like to go to. It also sends light on/off signals to indicate the status of the buttons [4].

Figure 2.18 Floor Request Buttons (key).

There are other buttons with request keys, These buttons allow passengers to open the elevator car doors or keep pressing it to keep them open, only when the elevator is stopped at a floor, Door Close, Emergency Stop Button, Emergency Bell Button, Fireman button….etc. [8].

IV. Car Position Indicators (displays)

In elevator cars, a visual display car position indicator shall be provided above the car control panel and over the landing door to show the position of the elevator in the hoistway and direction arrow. As the car passes or stops at a floor served by the elevators, the corresponding numerals shall illuminate, and sometimes with an audible signal shall sound. The controller interacts with this display by sending (output for controller) a signal that tells it which floor number to display, the common types are used (a dot matrix ,segmented and LED that changes to indicate the floor level) as shown in figure 2.19 [4, 8].

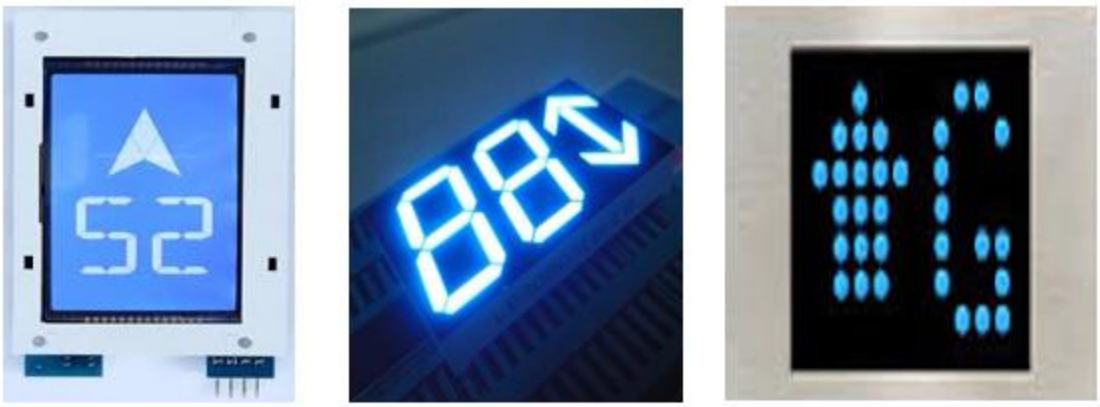


Figure 2.19 displays

V. Position Sensor and Read Sensor

The most common systems; a magnetic sensor or light sensor. A magnet is located on each floor and, as a sensor (Mono-stable as shown in figure 2.20.an attached to side of the elevator car passes a magnet, it increases or decreases the floor number depending on the direction of travel.

Figure 2.20 a) Mono-stable magnat sensor b) light sensor

In addition, light sensor attached the side of the car reads a series of holes on a long vertical tape in the shaft as shown in figure 2.20.b. 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 [8].

- **Types of Elevator Control Systems**

There are 3 main types for elevator control systems as follows [1]:

1. **Single Automatic operation**

First automated system and the simplest automatic lift control; single call button on each landing floor, Passengers has exclusive use of the car until trip is complete.

If a passenger in the car presses a call pushbutton corresponding to the required destination floor, the lift moves direct to this floor bypassing any intermediate floors at which landing hall buttons have been pressed. When a landing call pushbutton is pressed and the lift is free, the call is immediately answered. If the lift is in use, a landing signal indicates a “lift busy”, this type of control is only suitable for short travel passenger lifts, light traffic demand and very low carrying e.g. Small residential buildings [1].

2. **Selective Collective Operation**

Most common, this is a generic designation for those types of control where all landing and car calls made by pressing pushbuttons are registered and answered in other floor.

The lift automatically stops at landings for which calls have been registered. Collective control can be either of the single button, or of the two pushbutton types, and it can be:

Collective Single (Non-directional) pushbutton provides a single pushbutton at each landing. This pushbutton is pressed by passengers to register a hall call irrespective of the desired direction of travel. Thus, a lift will stop to answer the call [1].

Full collective (directional collective) The two pushbutton full collective control (also designated directional collective control) provides each landing with one UP and one DOWN pushbutton and passengers are requested to press only the Pushbutton for the intended direction of travel.

The momentary pressure on a car or landing push shall be stored in the system memory until answered and any number of calls so activated shall be stored [1].

The elevator shall answer the calls in the order in which the landings are reached and once the elevator has started travelling in one direction, it shall answer the car and landing calls for that direction only. The elevator shall not reverse until it has answered the highest or lowest outstanding call in its original direction.

When no more calls are registered in, the lift direction the elevator can reverses its direction of travel and answers the calls in the new direction. Used in this research [1].

3. **Group Automatic Operation**

For large buildings with many elevators which are controlled with programmable to respond. A single elevator will not always be able to cover with all the passenger traffic in a building. Thus, a number of elevators may be installed, often side-by-side, motivating the group elevators to answer the same call [1].

- **Special Operating Mode Of Control**

Many considerations give elevator its role to deal with some abnormal cases such as

  - Anti-crime protection.
  - Inspection service

- Fire service.
- Medical emergency/code-blue service.

All these cases will be served as high priority and this is one of most important feature in now a days modern life demands [1].

2.3 Hydraulic Elevators

It is the same to the rope traction Elevator excepting the ropes and counterweight as shown in figure 2.1.

Hydraulic Elevator consist mainly of piston at the bottom of the elevator that pushes the elevator up as an electric motor forces oil or another hydraulic fluid into the piston instead of ropes and counterweight in addition to hydraulic pump shown in figure 2.20.a. In addition, the machine room for Hydraulic elevators is located at the lowest level adjacent to the elevator shaft as in figure 2.21.a.

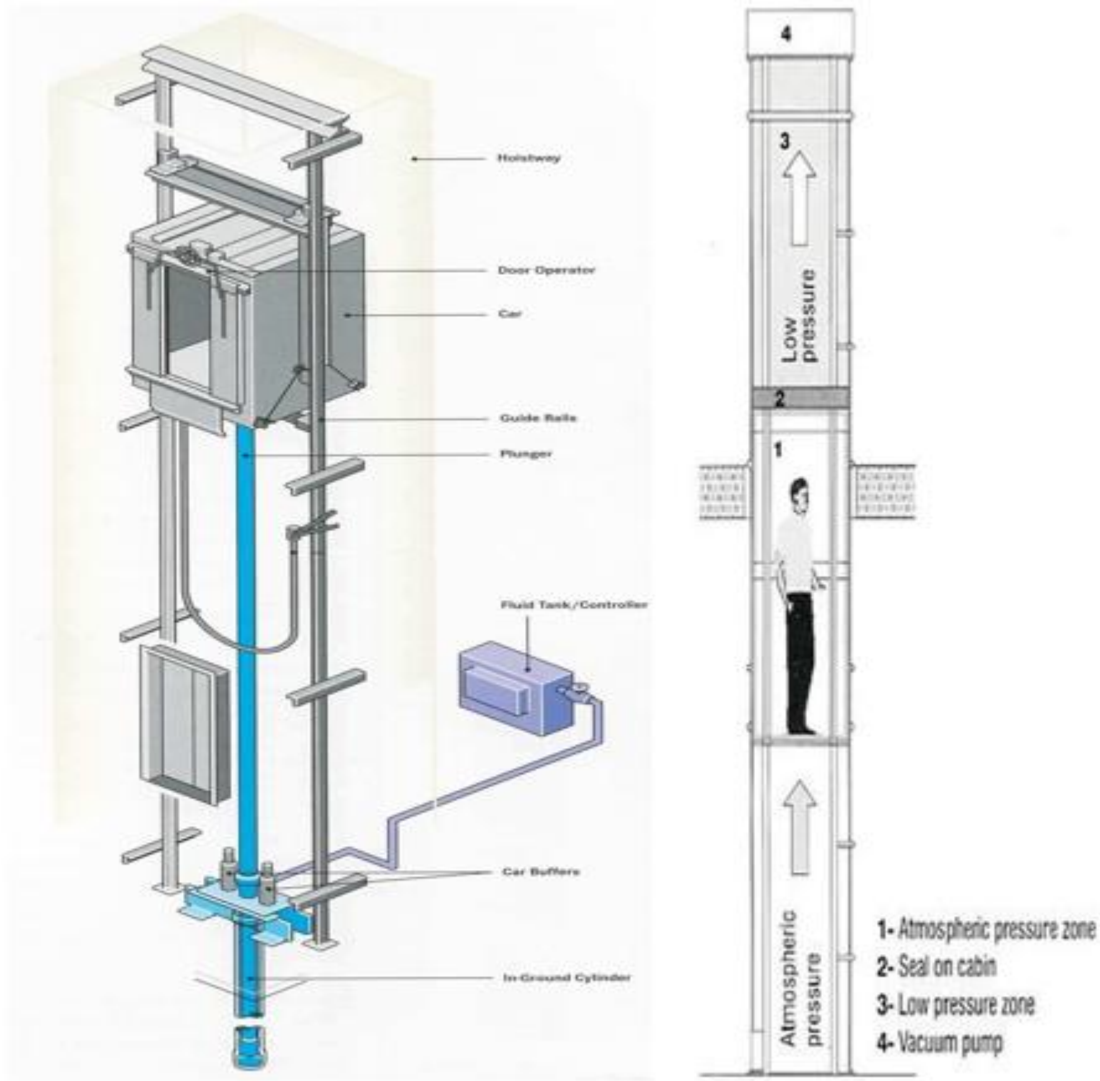
Hydraulics Elevator is very slow speed, limited height traveling; not recommended for more than 3 landings and/or travel exceeding 20m (due to strength and length of the hydraulic jacks) [9-11].

2.4 Pneumatic Elevators

Pneumatic elevators are raised and lowered by controlling air pressure in a chamber in which the elevator sits. By simple principles of physics, the difference in air pressure above and beneath the vacuum air. The vacuum pumps or turbines generates higher and lower atmospheric pressures that pull car up to the next floor and the slow release of air pressure that floats car down as in figure 2.21.b. They are especially ideal for existing homes due to their compact design because excavating a pit and hoist way are not required [12, 13]

This type of elevator cannot build enough pressure to push the car to more than 2- 3 storied high building.

[4].



a) Hydraulic elevator **b) Pneumatic elevator**

Figure 2.21 a) Hydraulic elevator b) pneumatic elevator.

2.5 History of Elevators

Archimedes in 312 C. invented the first reference elevator. From some literacy source, elevator were developed as cable on a hemp rope as shown in figure 2.22 and powered by hand or by through animals. This type of elevator was installed in the Sinai Monastery of Egypt. In the 17th century, the very small type of elevators were placed in the -building in England and France [4].

Figure 2.22 elevator on hemp and powered by hand.

In 1793, Lvan Kuliben created an elevator with the screw lifting mechanism for the winter place of Saint Petersburg. In 1816, an elevator was established in the main building of Sub-moscow village called Arkhamgel-skoye. In the middle 1800's, there were many type of curd elevators that carried freight [4].

Most of them ran hydraulically. The first hydraulic elevators used a plunger below the car to raise or lower the elevator. A pump applied water pressure to a plunger, or steel column, inside a vertical cylinder.

In 1852, Elisha Otis introduced the safety elevator, which prevented the fall of the cab, if the cable broke. In 1857 March 23rd, the first Otis passenger elevator was installed in New York City. The first electric elevator was built by Werner Von Siemens in 1880 [4].

In 1874, J.W. Meaker patented a method which permitted elevator doors to open and close safely. In 1882, when hydraulic power was a well-established technology, a company later named the London Hydraulic Power

Company was formed. In 1929, Clarence Conrad Crispen, with Inclinator Company of America, created the first residential elevator[4].

2.5.1 HISTORY OF ACOMPLISHMENT

- 1970 Elevator Industries successfully markets inexpensive, simple relay logic controllers
- 1975 Microprocessor-based controls are pioneered with multiple design patents issued.
- 1986 Ortiz brothers acquire patents and rights to manufacture control systems and replacement boards.
- 1986 elevator controls established.
- 1987 Design philosophy solidified.
- 1987 PC-based monitoring pioneered.
- 1988 Modular, backward compatible controller architecture developed.
- 1989 Core PC boards established as standardized components for all products.
- 1991 V800 product platform developed.
- 1990 has continued product refinement and reliability improvement.
- 2000, the first vacuum elevator was offered commercially in Argentina
- 2001 Ez-LINKTM serial communication system introduced.
- 2002 Advanced InteractTM central and remote monitoring System introduced.

- 2003 V900 next generation platform integrates A17.1-2000 / B44-00 code consolidates PC boards.

- 2007 Over 18,000 units shipped [8].

2.5.2 Literature Reviews

The primary function of the elevator controller is essentially to receive and process a variety of signals from several different components of a whole elevator system. It is able to send signals in response to the once it receives in order to operate all of the other components in the system. This exchange of signals is how the elevator controller is able to keep the elevators running smoothly on a day-to-day basis.

- **Relay based controller** (electromechanical switching)

A relay is a very dependable device consisting of an electromagnet that opens and closes contacts, routing the logic to various circuits. A simple elevator with a few stops and manual door operation can be served well by a relay controller. Relays can also be used for more complex elevators, and in fact were until the 1980's. However, the number of relays required can make it difficult to troubleshoot should there ever be a problem.

A basic relay logic as shown in figure 2.23 of a traction elevator with manual doors and an elevator panel consists of a power circuit, which supplies the motor, and a low voltage control circuit, this one supplies the relay coils, the motor brake and the doors locking mechanism [3].

At the head of the control circuit, there are the safety switches. The automatic switch cuts off the relay circuit if an insulation failure occurs (for example between the doors safety circuit and the ground); that would be dangerous because the elevator may move with open doors. The interlock allows the requests when the door is closed. However the motion of the elevator is allowed by a second switch, when the door is locked. The finals work if the elevator goes over an extreme floor (due to a motor brake failure, for example). The car safety brake is a mechanical device that prevents the elevator from falling; it works in combination with the governor.

Although relay logic has been replaced by modern electronic devices, it is still now used on old elevators [3].

Figure 2.23 Relay based controller.

The following applications may be recommended as suitable for controllers using electromagnetic relay technology:

- Single lifts only.
- Drive speed up to 1 m/s [3].

- **Solid-State Logic Technology**

It includes both discreet transistors circuits and integrated circuit boards. It gives improved reliability, lower power consumption and easy fault diagnosis than electromagnetic relay technology as in figure 2.24.

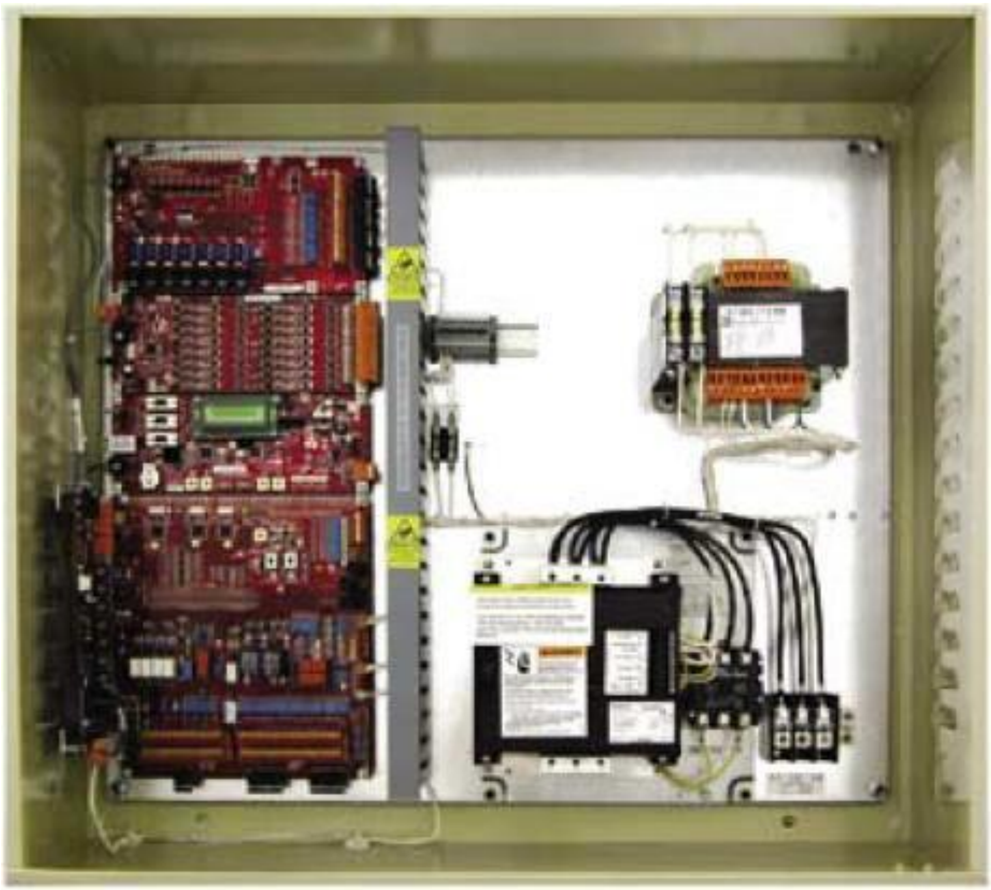


Figure 2.24 Solid-State Logic.

The following applications are recommended as suitable for controllers using solid-state logic technology [3]:

- Single lifts and duplex groups.
- Drive speed up to 2 m/s.
- Passenger lifts in low traffic situations in medium-rise buildings, i.e. up to 12 stories [3].

- **PLC controller** (computer-based technology)

Figure 2.25 PLC controller (computer-based technology).

In the late 1960's Programmable Logic Controllers (PLCs) were first introduced. The primary reason for designing such a device was eliminating the large cost involved in replacing the complicated relay based machine control systems [14].

When production requirements changed so did the control system. This becomes very expensive when the change is frequent. Since relays are mechanical devices, they also have a limited lifetime, which required strict adhesion to maintenance schedules. Troubleshooting was also quite tedious when so many relays are involved. Now picture a machine control panel that included many, possibly hundreds or thousands, of individual relays. The size could be mind-boggling. How about the complicated initial wiring of so many individual devices! These relays would be individually wired together in a manner that would yield the desired outcome. As can be seen, there were many problems with this relay based design [14].

To use a programming technique most people were already familiar with and replace mechanical parts with solid-state ones.

In the mid70's the dominant PLC technologies were sequencer state-machines and the bit-slice based CPU.

- PLC based controllers are suitable for the following:
    - All lifts types.
    - All drive speeds (i.e. 0.5 m/s to 10 m/s).
    - Lift groups of all sizes.

Chapter Three

CIRCUIT DESIGN

CHAPTER THREE

Circuit Design

3.1 Component Description

The main components used in this dissertation are:

1. Microcontroller (the brain of control system).
2. Dc motors (As actuator).
3. Motor drivers (for interfacing purpose).
4. 7segment displays and LEDs (indicators).
5. Lamp and fan (humans comfort).
6. Pushbutton and Switches (playing an input role).
7. Latches (input signals stacking).

3.1.1 Microcontroller

Basically a microcontroller is a computing device, and is a single integrated circuit ("Silicon chip" or IC) used to form part of a product that incorporates some software Program control. As a microcontroller is basically part of a computing system it can be used in applications requiring control, operator and user display generation, simple sequencing and many other mundane tasks.

A microcontroller device is not simple, but in general, a microcontroller unit may be considered as a computing device offering internal memory and a high level of input and output (I/O) device options. Ideally the use of a microcontroller device minimizes the number of external devices used in the system, and integrates as much of the external interfacing to switches, motors or other input / output devices as is practically possible [15].

Microcontrollers offer a low cost computing solution. Alternative computing solutions come in many forms, and microprocessor devices can be considered a ‘cousin’ of the microcontroller device, but are optimized to manipulate high volumes of data and to provide the facilities for several tasks or windows to operate at any one time [1, 15].

Difference between microcontroller and microprocessor

A microprocessor (abbreviated as μP or MPU) is a computer electronic component made from miniaturized transistors and other circuit elements on a single semiconductor integrated circuit (IC) (microchip or just chip). The central processing unit (CPU) is the most well-known microprocessor.

Microcontroller is basically a computer on a chip. (Abbreviated as μc or MCU).

Some of the primary differences between MCU and MPU, typically, MCU uses on-chip embedded Flash memory in which to store and execute its program as shown in figure 3.1. Storing the program in this way means that the MCU has a very short start-up period and can be executing code very quickly. The only practical limitation to using embedded memory is that the total available memory space is finite. Most Flash MCU devices available on the market have a maximum of 2 Mbytes of Program memory and, depending on the application, this may prove to be a limiting factor. MPUs do not have memory constraints in the same way. They use external memory to provide program and data storage. The program is typically stored in non-volatile memory, such as NAND or serial Flash, and at start-up is loaded into an external DRAM and then commences execution. This means the MPU will not be up and running as quickly as an MCU but the amount of DRAM and NVM you can connect to the processor is in the range of hundreds of Mbytes and even Gigabytes for NAND [15, 16].

Another difference is power, by embedding its own power supply; an MCU needs just one single voltage power rail. By comparison, an MPU requires several difference voltage rails for core, DDR etc. The developer needs to cater for this with additional power ICs / converters on-board. From the application perspective, some aspects of the design specification might drive device selection in particular ways. For example, is the number of peripheral interface channels required more than can be catered for by an MCU Or, does the marketing specification stipulate a user interface capability that will not be possible with an MCU because it does not contain enough memory on-chip or has the required performance.

Microprocessor devices are relatively high cost items when compared to microcontrollers because of this high performance capability. In comparison, microcontrollers, as their name suggests, are in general optimized for control applications and not data manipulation. However, the principles and jargon often encountered in PCs are often replicated when considering microcontroller units [17].

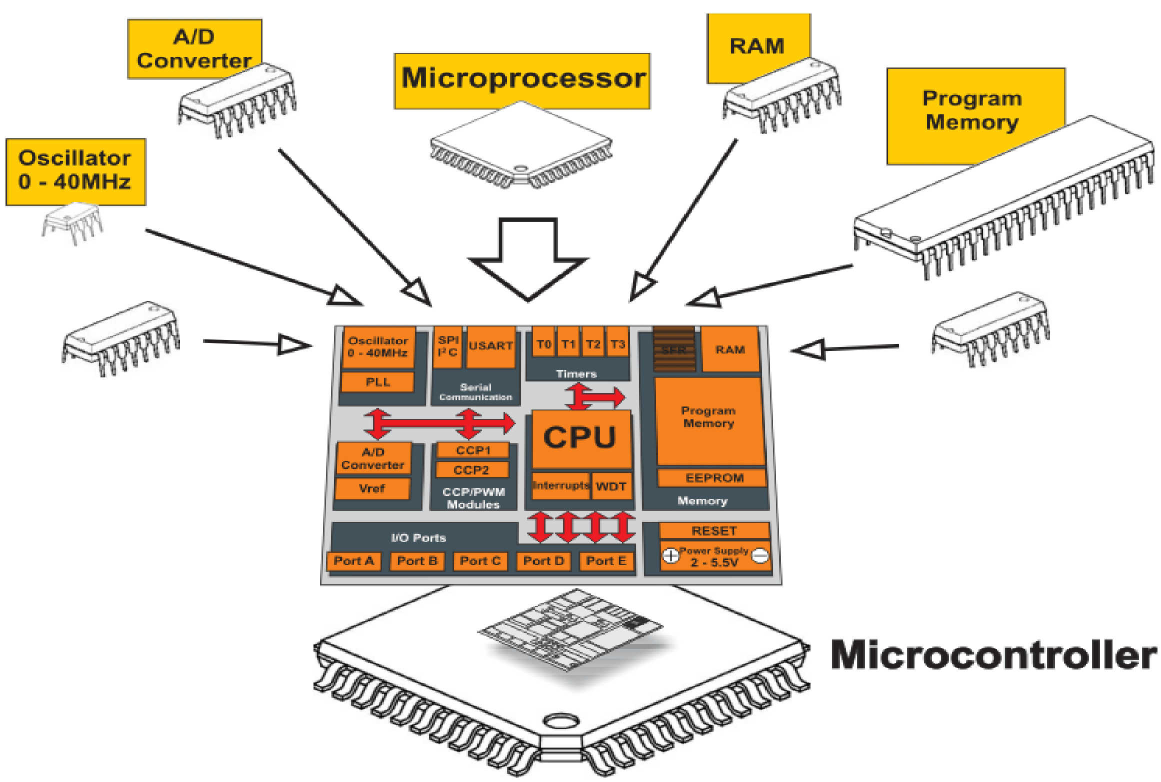


Figure 3.1 the different between microcontroller and microprocessor.

When embarking on the first design and knowing that, it is highly likely there will be many product variations. In that case, it is very possible a platform-based design approach will be preferred.

Another key aspect that will drive the selection between an MCU and an MPU is the need for a real-time/deterministic behavior of the application. Because of the processor core used in an MCU, as well as the embedded flash and considering the software used that is either an RTOS or bare metal The MCU will definitely take the lead on this aspect and will address perfectly the most time critical and deterministic applications.

Final point to consider is power consumption. While MPUs do have low power modes there are not as many or as low as the ones you would find on a typical MCU. With the external hardware supporting an MPU has an added factor, putting an MPU into a low power mode might also be slightly more complex. Also, the actual consumption of an MCU is magnitudes lower than an MPU, in low power mode for example with SRAM and register retention, you can consider a factor 10 to 100.

Many manufactures produce a various types of microcontrollers some of them are used widely because of fast execution, simplicity of programming and other considerations so AVR is used in this project to play the microcontroller role[18].

AVR microcontroller

AVR was developed in the year 1996 by ATMEL Corporation. The architecture of AVR was developed by Alf-Egil Bogen and Vegard Wollan. AVR derives its name from its developers and stands for Alf-Egil Bogen Vegard Wollan RISC microcontroller, also known as **A**dvanced **V**irtual **R**ISC [15].

AVR microcontrollers are available in three categories:

I. Tiny-AVR – Less memory, small size, suitable only for simpler applications

II. Mega-AVR – These are the most popular ones having good amount of memory (up to 256 KB), higher number of inbuilt peripherals and suitable for moderate to complex applications.

III. Xmega-AVR – Used commercially for complex applications, which require large program memory and high speed.

Comparison features of ATMEGA 16 Family Members

As shown in table 3.1 below the technical feature of different type of ATMEGA, for this project ATMEGA 16A will be selected because it suitable memory size and I/O ports is satisfy our application.

Table 3.1 Compare ATMEGA16 with Family

Device	ATMEGA16A	ATMEGA32A	ATMEGA164P	ATMEGA324P	ATMEGA644P	ATMEGA1284P
Flash	16K	32K	16K	32K	64K	128K
SRAM	1K	2K	1K	2K	4K	16K
EEPROM	0.5K	1K	0.5K	1K	2K	4K
Max Freq (MHz)	16	16	20	20	20	20
Touch Chnls	12	12	16	16	16	16
Ext Interrupts	3	3	32	32	32	32
USART	1	1	2	2	2	2
SPI	1	1	3	3	3	3
Pico Power	N	N	Y	Y	Y	Y
Vcc	2.7 to 5.5	2.7 to 5.5	1.8 to 5.5	1.8 to 5.5	1.8 to 5.5	1.8 to 5.5
PWM	4	4	6	6	6	6

The ATMEGA16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC Architecture. By executing powerful instructions in a single clock cycle, the ATMEGA16 achieves Throughputs approaching 1 MIPS per MHz allowing the system to designer to optimize power consumption versus processing speed [15].

The AVR core combines a rich instruction set with 32 general-purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

There are number of popular families of microcontrollers, which are used in different applications as per their capability and feasibility to perform the desired task, most common of these are 8051, AVR and PIC microcontrollers. In this chapter, we will introduce you with AVR family of microcontrollers [15].

- **The main features of ATMEGA 16**

i. Naming Convention

- The AT refers to ATMEL the manufacturer.
- Mega means that the microcontroller belong to Mega-AVR category.
- 16 signifies the memory of the controller, which is 16KB
- 16 Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbyte SRAM, 32 General purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare Modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, An SPI serial port, and six software selectable power saving modes.

- The Idle mode stops The CPU while allowing the USART, two-wire interface, A/D Converter, SRAM; Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping.

- The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run [15].

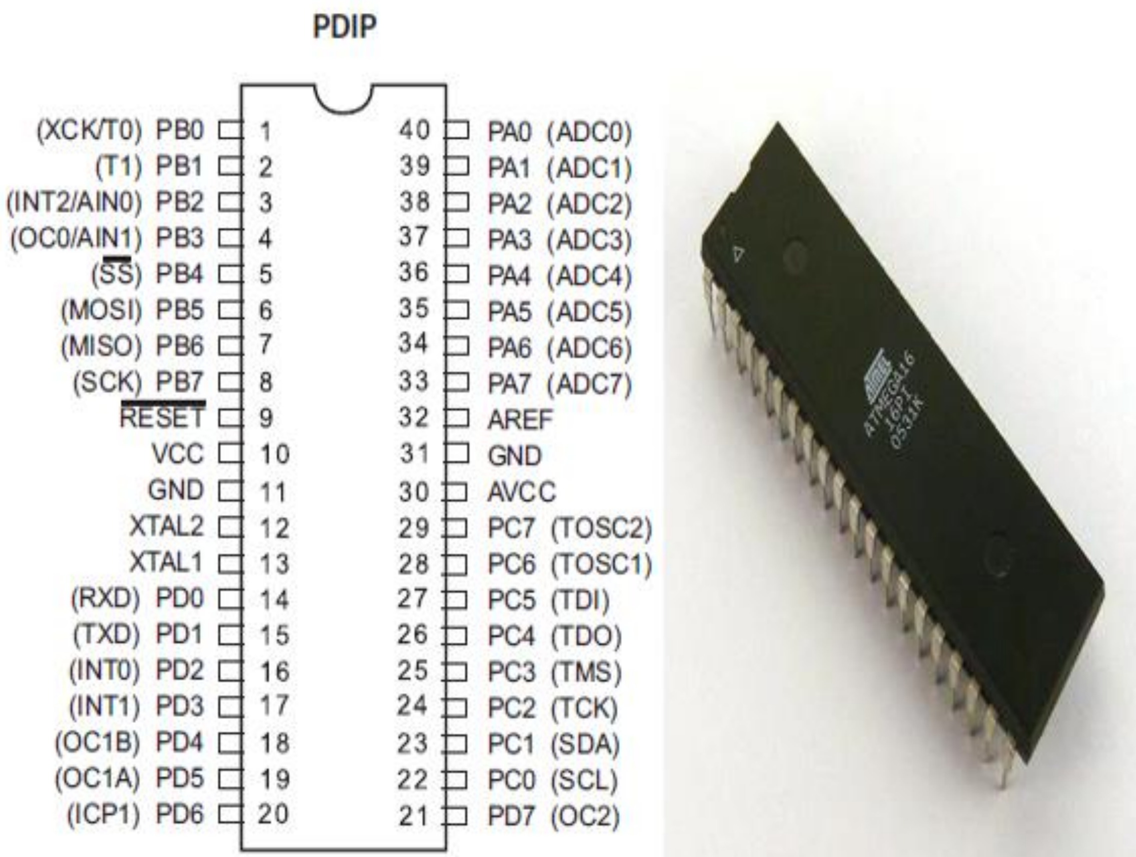


Figure 3.2 Pin out of ATMEGA16

ii. Pin Descriptions

VC: supply voltage.

GND: Ground.

Port A (PA7...PA0)

Port A serves as the analog inputs to the A/D Converter, also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port A pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability.

When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PORT B (PB7...PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability.

As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATMEGA16 as listed in appendix A table 6.2.

PORT C (PC7...PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5 (TDI), PC3 (TMS) and PC2 (TCK) will be activated even if a reset occurs. Port c also serves the functions of various special features of the ATMEGA16 as listed in appendix A table 6.3.

PORT D (PD7...PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).

As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATMEGA16 as listed in appendix A table 6.4.

RESET:

Reset Input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running.

XTAL1:

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2:

Output from the inverting Oscillator amplifier.

AVCC:

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF

AREF is the analog reference pin for the A/D Converter [15].

3.1.2 The Motor

Motor is converting electrical energy into mechanical energy and the power supply can be either DC type or AC type and the drive a variable voltage, variable frequency unit.

In elevators, motors of all types are used in various hoisting machine applications to raise and lower the load being transported, to open and close car and landing doors, raise and lower freight elevator landing doors and car gates, in fans that provide ventilation of car enclosures. To operate hydraulic elevator pumps and where required to keep the bottom of the hoistway dry and free of moisture in pit sump pumps as well [3, 19].

The power developed by main motor in traction lift shall be transmitted directly to the driving sheave which is to be located on the same shaft as the motor. The main shaft shall be supported on two large bearings that may be of the sleeve.

- **Dc motor**

A motor is define as a device that converts electrical energy into mechanical work using rotary motion.

There are always two main options available for designer whether to use a DC motor or an AC motor. When it is about speed, weight, size, cost... DC motors are always preferred than AC motors but the availability of AC source give it more advantages with complexity of speed controllability using inverter which produce variable frequency . Although the DC motor is easy to control i.e. speed via change the applied voltage and inverting the rotation via change the polarity as see in 3.1.3 but additional circuits is needed to produce the DC current

In this application DC motor was used to demonstrate the concept of motion in real elevator AC motors are used to perform the conversion of electrical energy in to mechanical motion [3].

DC Motor uses for up elevator and down and use to open door and close and finally uses as fan for ventilation

3.1.3 Motor driver (H Bridge)

To control the direction of DC motor L293D IC, which is an H bridge, is an electronic circuit that enables a voltage to be applied across a load in either direction. These circuits are used to allow DC motors to run clockwise and counter clockwise direction.

The name "H-Bridge" is derived from the actual shape as shown in figure 3.3 ,As you can see in the figure 3.3 there are four switching elements named as "High side left", "High side right", "Low side right", "Low side left". When these switches are turned on in pairs motor changes its direction accordingly. Like, if we switch on High side left and Low side right then motor rotate in forward direction, as current flows from power supply through the motor coil goes to ground via switch low side right. The concept of this is shown in the figure 3.3

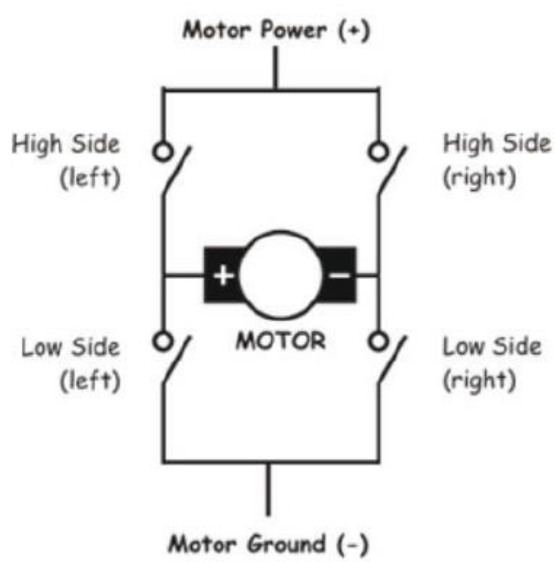


Figure 3.3 concept of H Bridge

Moreover, for protection of circuit from back EMF output diodes are building in the IC. The output supply (VCC2) has a wide range from 4.5V to 36V, which has made L293D a best choice for DC motor driver.

L293D Dual H-Bridge Motor Driver

L293D is a dual H-Bridge motor driver, so with one IC we can interface two DC motors, which can be controlled in both clockwise and counter clockwise direction, and if you have motor with fix direction of motion. You can make use of all the four I/Os to connect up to control four DC motors. Moreover. The output supply (VCC) has a wide range from 4.5V to 36V, which has made L293D a best choice for DC motor driver [20].

The L293 comes in a standard 16-pin, dual-in line integrated circuit package as in figure 3.4. There is an L293 and an L293D part number. Pick the "D" version because it has built in freewheel diodes to minimize inductive voltage spikes.

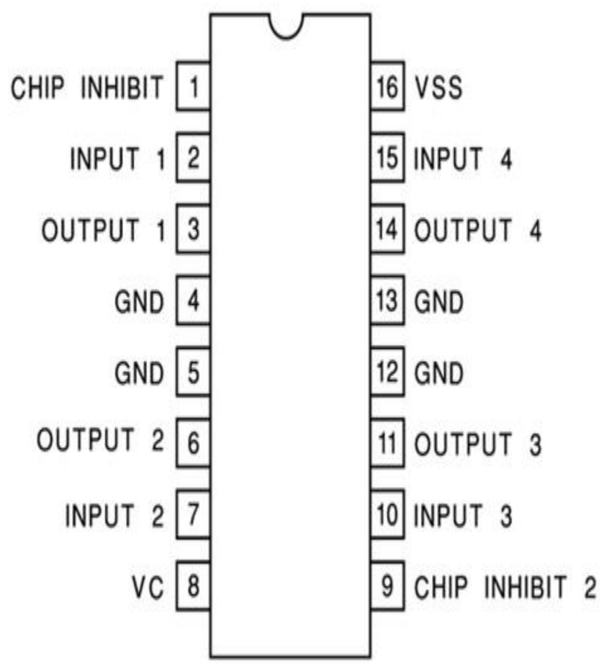


Figure 3.4 Pin of L293D.

Input logic 00 or 11 will stop the corresponding motor, Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively, Enable pins 1 and 9. The motor action according to the input pins shown in table 3.4. In addition, the figure 3.5 shown how to be interface with two motor.

Table 3.2 Motor action based on status of input

Input(1A,4A)	Input(2A,3A)	Motor action
Logic 0	Logic 0	Stop
Logic 0	Logic 1	Moves clockwise
Logic 1	Logic 0	Moves anticlockwise
Logic 1	Logic 1	Stop

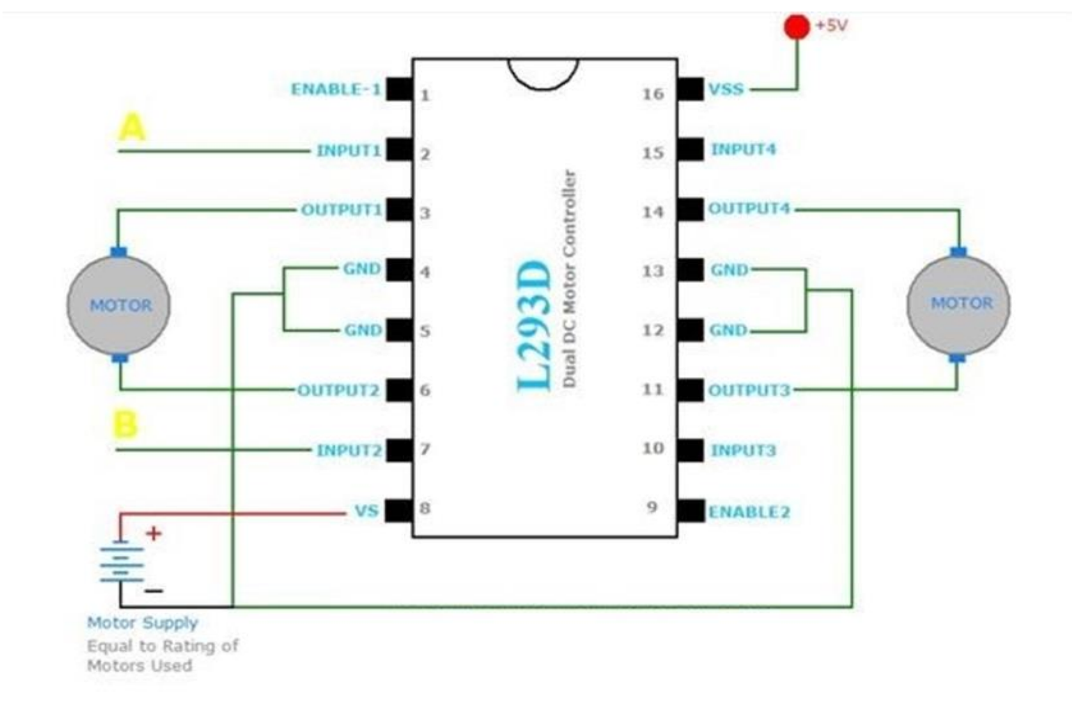


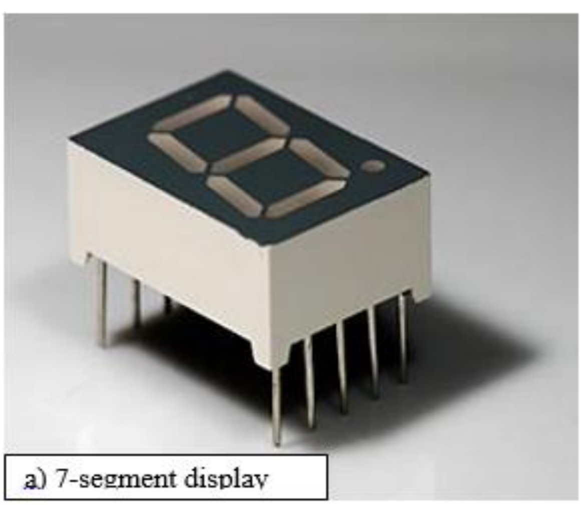
Figure 3.5 L293D interfacing with DC motor.

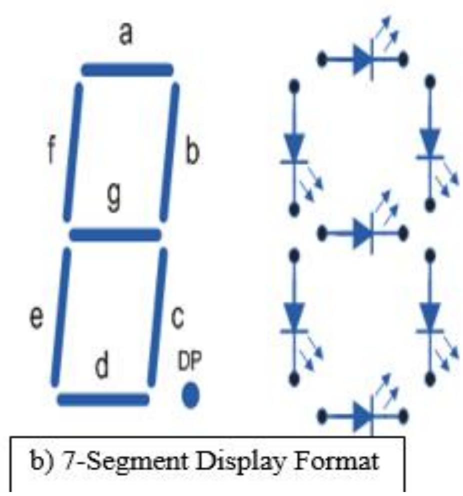
3.1.4 Seven segment (7 segment) display

A Seven-Segment Display (SSD) shown in figure 3.6.a, or seven-segment indicator, is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot-matrix displays.

Seven-segment displays are widely used in digital clocks, electronic meters, and other electronic devices for displaying numerical information. A seven-segment display, as its name indicates, is composed of seven elements. Individually on or off, they can be combined to produce simplified representations of the Arabic numerals. The seven segments are arranged as a rectangle of two vertical segments on each side with one horizontal segment on the top, middle, and bottom. Typically 7-segment displays consist of seven individual colored LED’s (called the segments), within one single display package. In order to produce the required numbers or HEX characters from 0 to 9 and A to F respectively and some letters (g , H , J , L , o , P , q and U), on the display the correct combination of LED segments need to be illuminated and BCD to 7-segment Display Decoders such as the 74LS47 do just that [18].

A standard 7-segment LED display generally has 8 input connections as shown in figure 3.6.b.


a) 7-segment display



b) 7-Segment Display Format

Figure 3.6 a) 7-segment display b) 7-Segment Display Format.

BCD to 7-Segment Display Decoders

A Binary Coded Decimal (BCD) to 7-segment display decoder such as the TTL 74LS47 or 74LS48, have 4 BCD inputs and 7 output lines as shown in figure 3.7, one for each LED segment. This allows a smaller 4-bit binary number (half a byte) to be used to display all the decimal numbers from 0 to 9 and by adding two displays together; a full range of numbers from 00 to 99 can be displayed with just a single byte of 8 data bits.

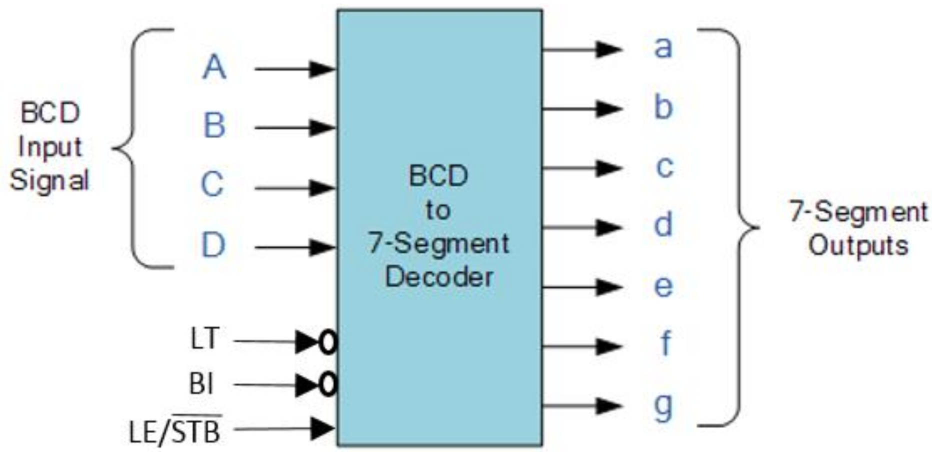


Figure 3.7 BCD to 7-Segment Decoder

Uses: five form 7 segment as display four putted above the landing door at every floors to indicator for passengers (out the car) where the elevator position and direction [18].

In addition, one inside the car to inside passengers. -there are four for Avery floor outside the hositway and other one inside car to indicator there are car elevator position

3.1.5 Light Emitting Diodes

Commonly called LEDs, are real unsung heroes in the electronics world. They do dozens of different jobs and are found in all kinds of devices. Among other things, they form numbers on digital clocks, transmit information from remote controls, light up watches and tell you when your appliances are turned on.

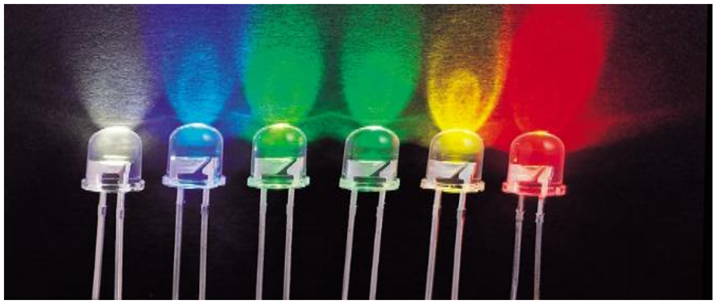


Figure 3.8 LEDs

Can be color of the light (corresponding to the energy of the photon) as shown in figure 3.8.

Uses with any hall button or keys to indictor it is pressed and shutoff when the answer, in addition, when overloading happen the LED is flashy until release the overload [18].

Led used to indicator two things, firstly to indicate for passenger the push button is pressed with green color and the secondly when overload sensors is active the led flashy with red color.

3.1.6 Lamp of elevator

Lamps designed especially for lighting elevator cars, it's always work when elevator in active mode. To operate the elevator with power saving sleep mode is used to turn the lamp of when the elevator is not occupied.

3.1.7 Pushbutton Switches

Several terms are used to describe switch contact

- **Definition**

  - **Pole**: number of switch contact sets.
  - **Throw**: number of conducting positions, single or double.
  - **Way**: number of conducting positions, three or more.
  - **Momentary**: switch returns to its normal position when released.
  - **Open**: off position, contacts not conducting.
  - **Closed**: on position, contacts conducting, there may be several on positions [21].

- **Types of switches**

I. **Push-to-make Single Pole, Single Throw = SPST Momentary**

A push-to-make switch returns to its normally open (off) position when you release the button, Shown in figure 3.9.a.

II. **Push-to-make Single Pole, Double Throw = SPDT**

This switch can be on in both positions, switching on a separate device in each case. It is often called a changeover switch. For example, a SPDT switch can be used to switch to do something at one position and do something else at other position as shown in figure 3.9.b.

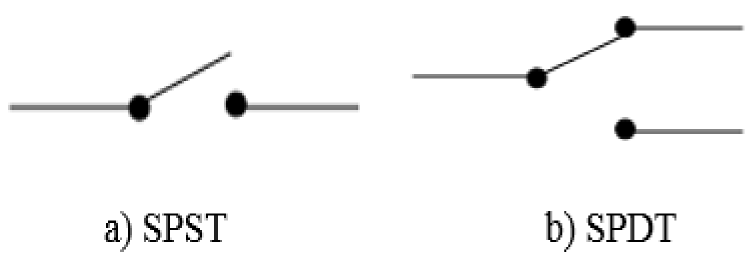


Figure 3.9 Types of switches.

- **Uses of switches**

I. **hall push button and Floor Request Buttons (keys):**

There are 3 floor in addition to the ground (lobby) used two hall push button placed in any floor one of them indicate to passenger if they want to move up floor from their location and the second if they want to move down. Except the lobby and the last floor, they have only one push button because there is no any floors upper the last or any under the lobby floor. The Floor Request Buttons (keys) four keys placed inside the elevator car for inside request to specify the travel destination.

In addition, the push button to open door, which if passenger form inside the elevator car decided to go out when open closing. Replace for all push button and keys by switch [22]:

II. **Limit switches:**

There are two limit uses in this application:

a. Ending limit putted at button of hoistway as safety.

b. For opening door to indicate that door has opened

III. **Sensors**

There are 2 sensors putted as switches:

A. IR sensor: Represented when any crossing for IR sensor like open the switch.

B. Overload sensor: Represented as when overload existing as close the switch.

3.1.8 Latches

- **Uses D flip flop**

The D flip-flop tracks the input, making transitions with match those of the input D. The D stands for "data"; this flip-flop stores the value that is on the data line. It can be thought of as a basic memory cell. A D-flip-flop can be made from a set/reset flip-flop by tying the set to the reset through an inverter as shown in figure 3.10. The result may be clocked. And Output depends on clock when [23, 24]:

- Clock high: Input passes to output.
- Clock low: Latch holds its output.

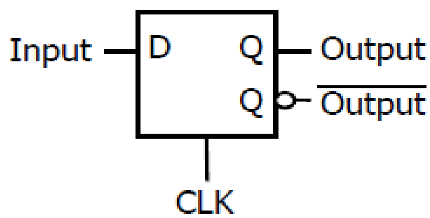


Figure 3.10 D flip flop

Uses to store the request of keys and hall call until to serve (show in LED of requests) then reset the flip-flop to new requests [23, 24].

3.2 The Circuits

The project consist of three circuits work together to perform the control process for elevator

3.2.1 Circuit for inside elevator car

Figure 3.11 shown Circuit inside elevator car, shown the lamp and fan for ventilation, addition to the keys with their LEDs and one from 7-SEG inside the elevator car, and DC motor to open the doors with their limit switch to indicate when open.

In addition, two switch for IR and overload sensors are working as a safety elements.

3.2.2 Circuit for (hoistway) the hall buttons and motor

Figure 3.12 shown Circuit for hoistway shown the hall push button and DC motor and final bottom Limit switch.

3.2.3 Circuit for control

Figure 3.13 shown Circuit for the microcontroller and the L293D

- ❖ And the Figure 3.14 shown all **the Circuits**

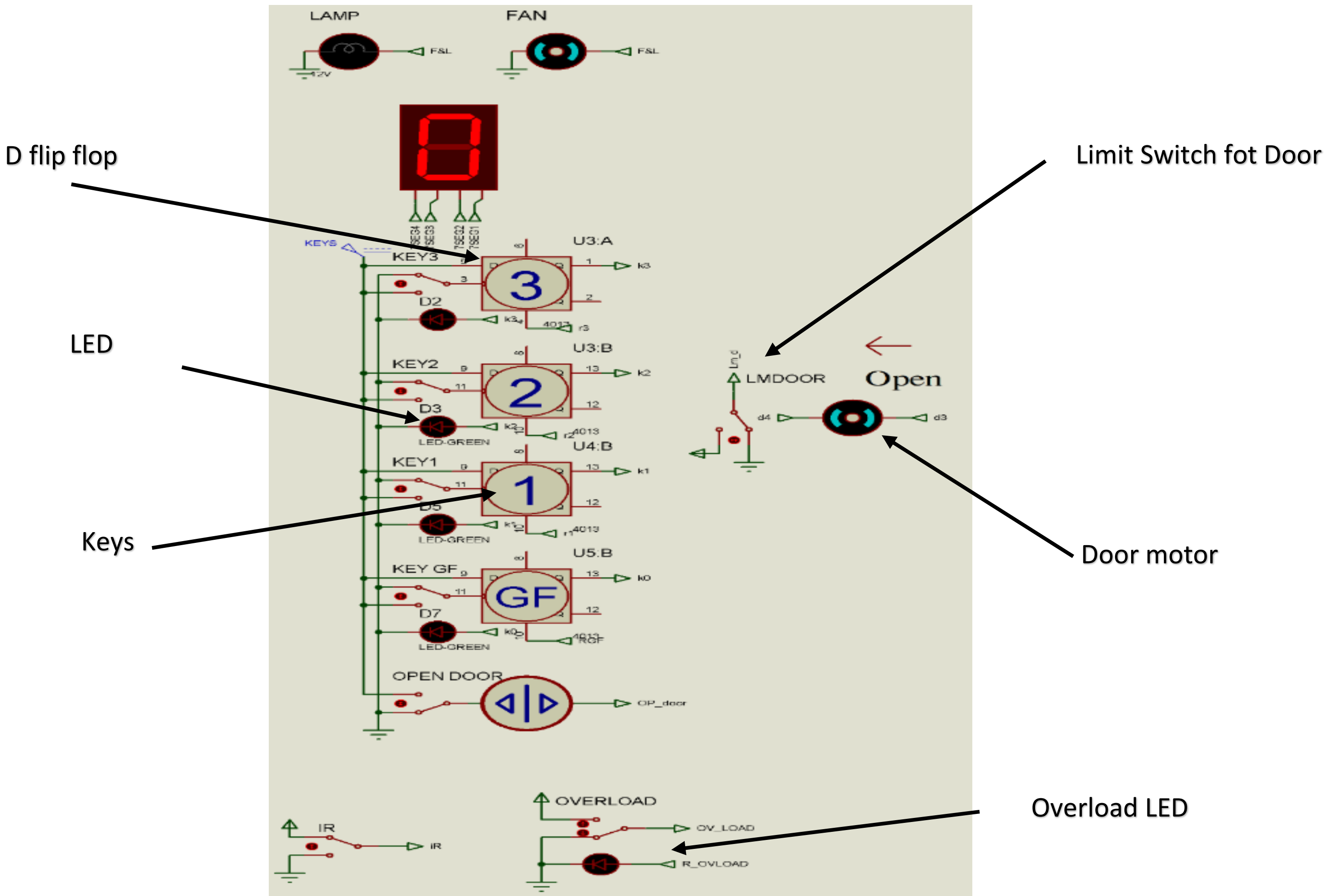


Figure 3.11 Circuit in side elevator car

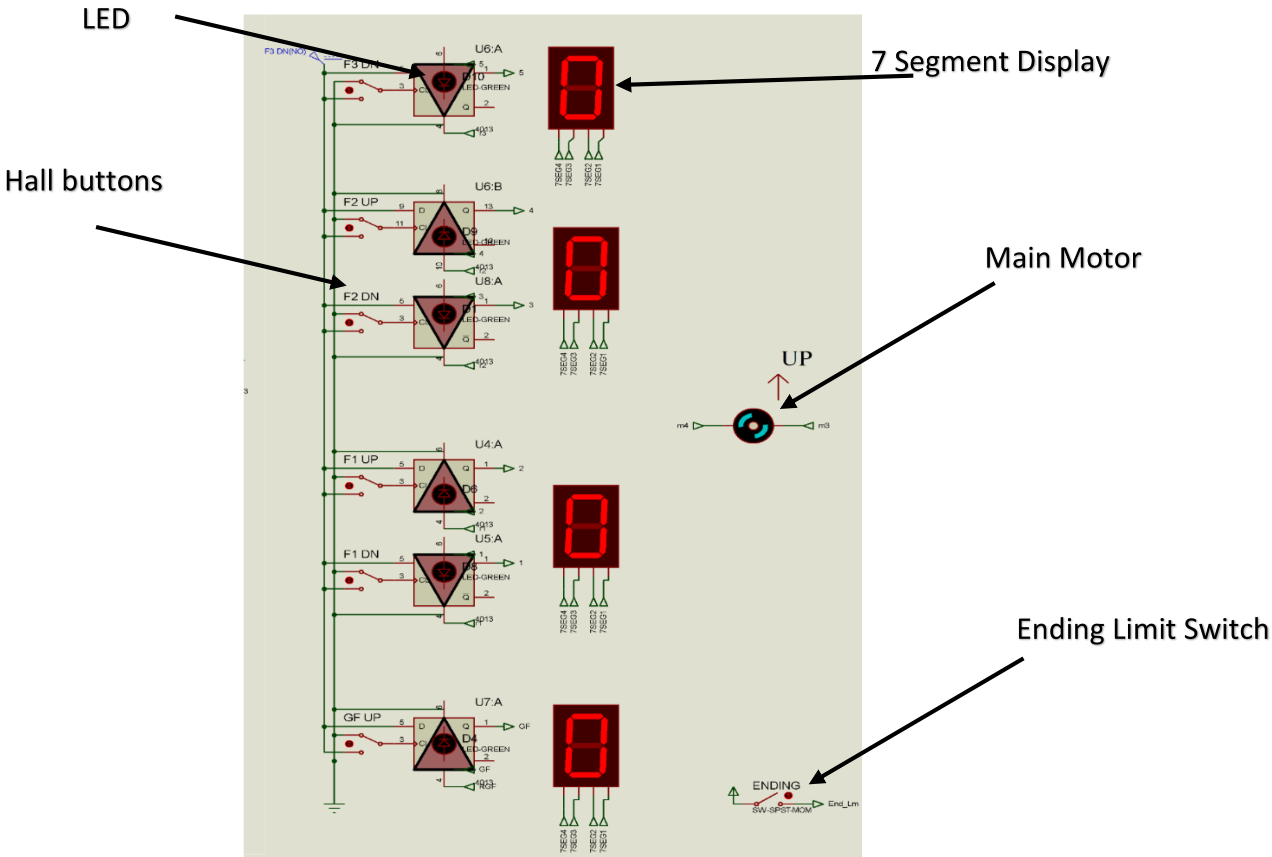


Figure 3.12 Circuit for hoistway

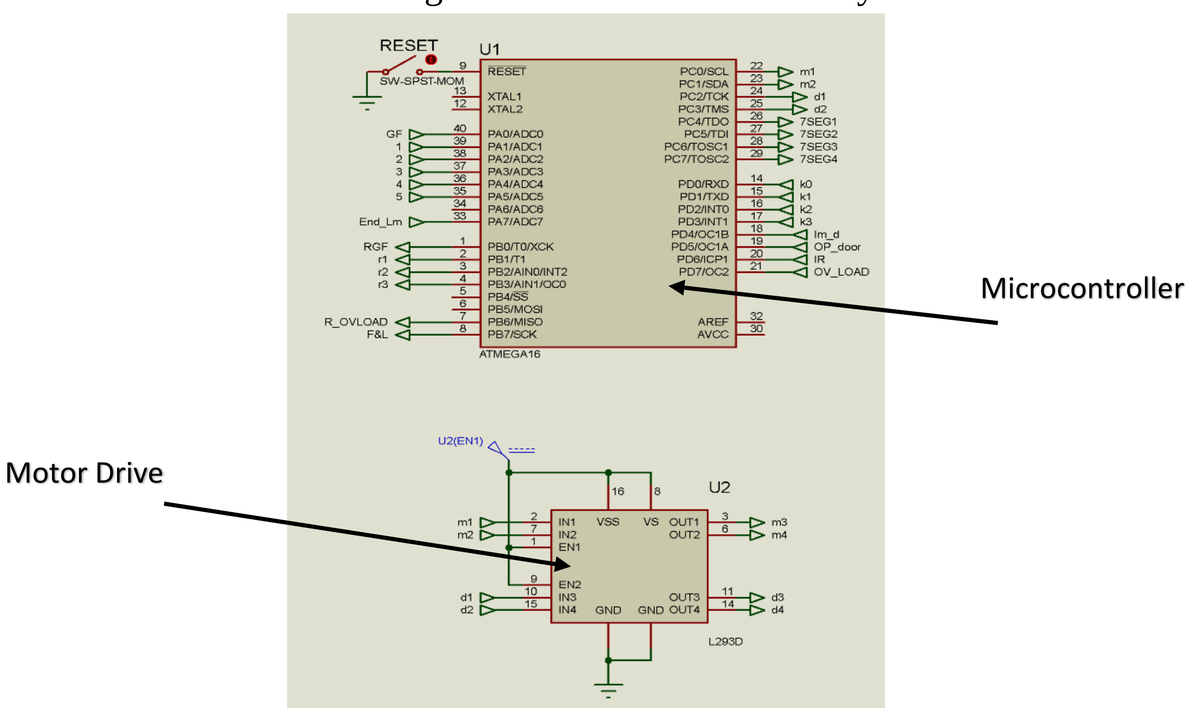


Figure 3.13 Circuit for control.

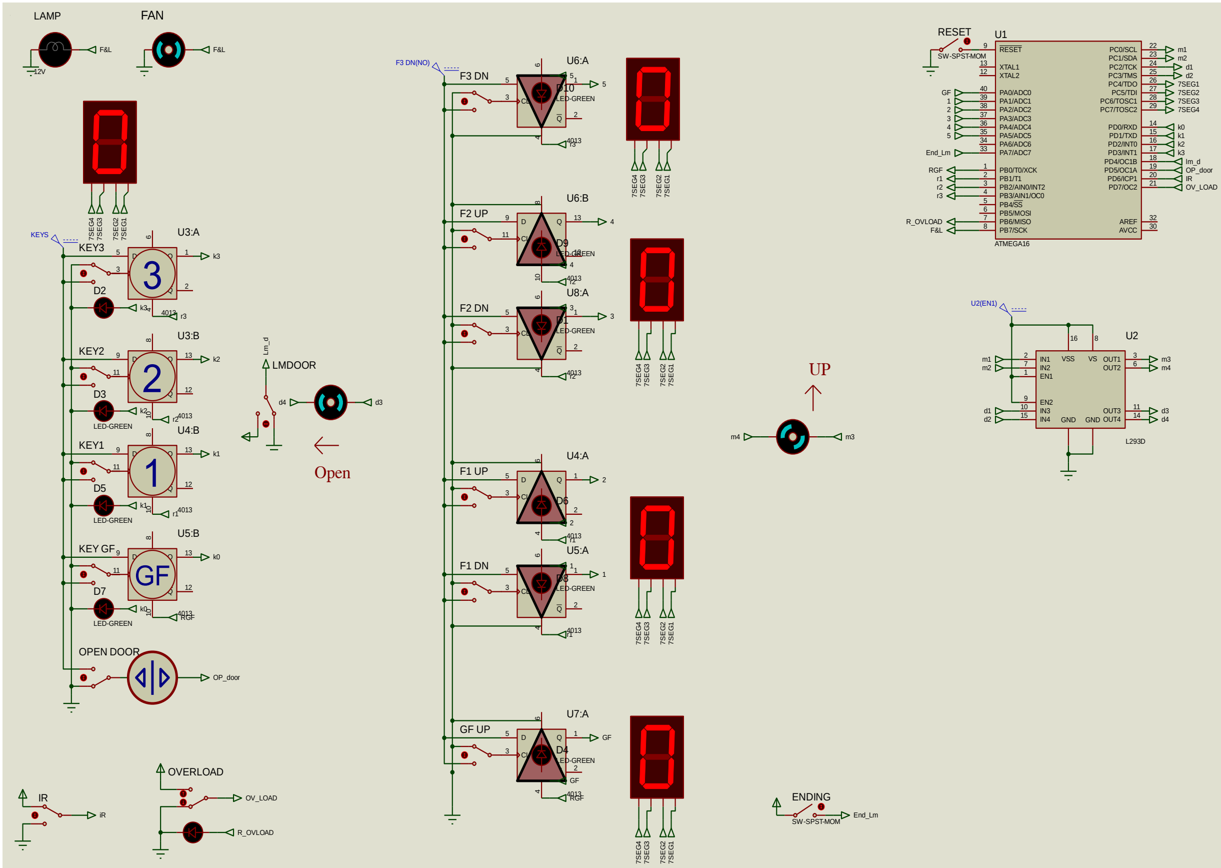


Figure 3.14 All the circuit.

3.3 System Design

When keys hall push button is pressed the latch D flip flop save and latch the presses and fed to microcontroller , the whole operation of system was controlled by microcontroller (ATMEGA16) ,and microcontroller receiver signals from sensors and limit switch, all these input for microcontroller .

When signals provided to microcontroller the microcontroller treated with signals ,after treatment the microcontroller decided(decision making) to send for move or not, if decided to move its send signal to motor drive chip was between microcontroller and motor for which direction of motor upper or lower or right or left.

The connection block diagram for the whole system is in Figure 3.15.

3.3.1 Block diagram

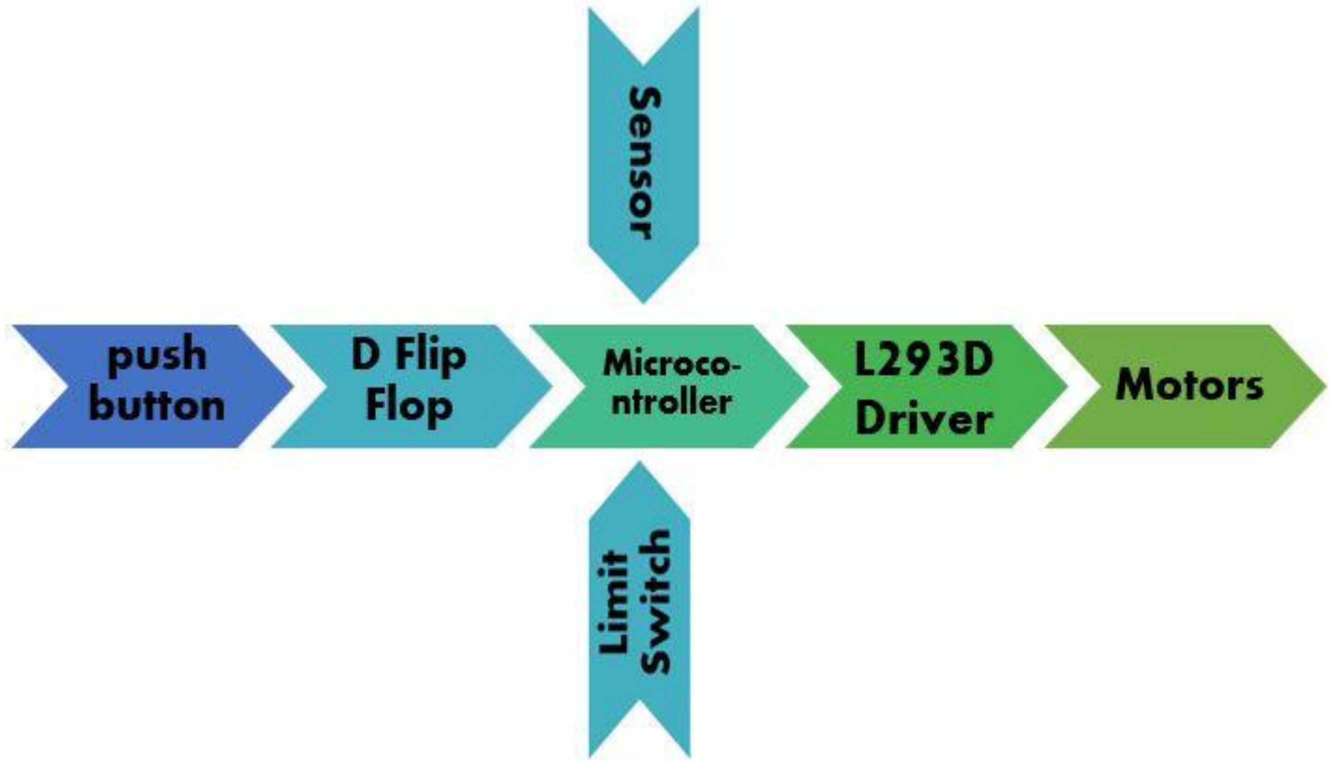
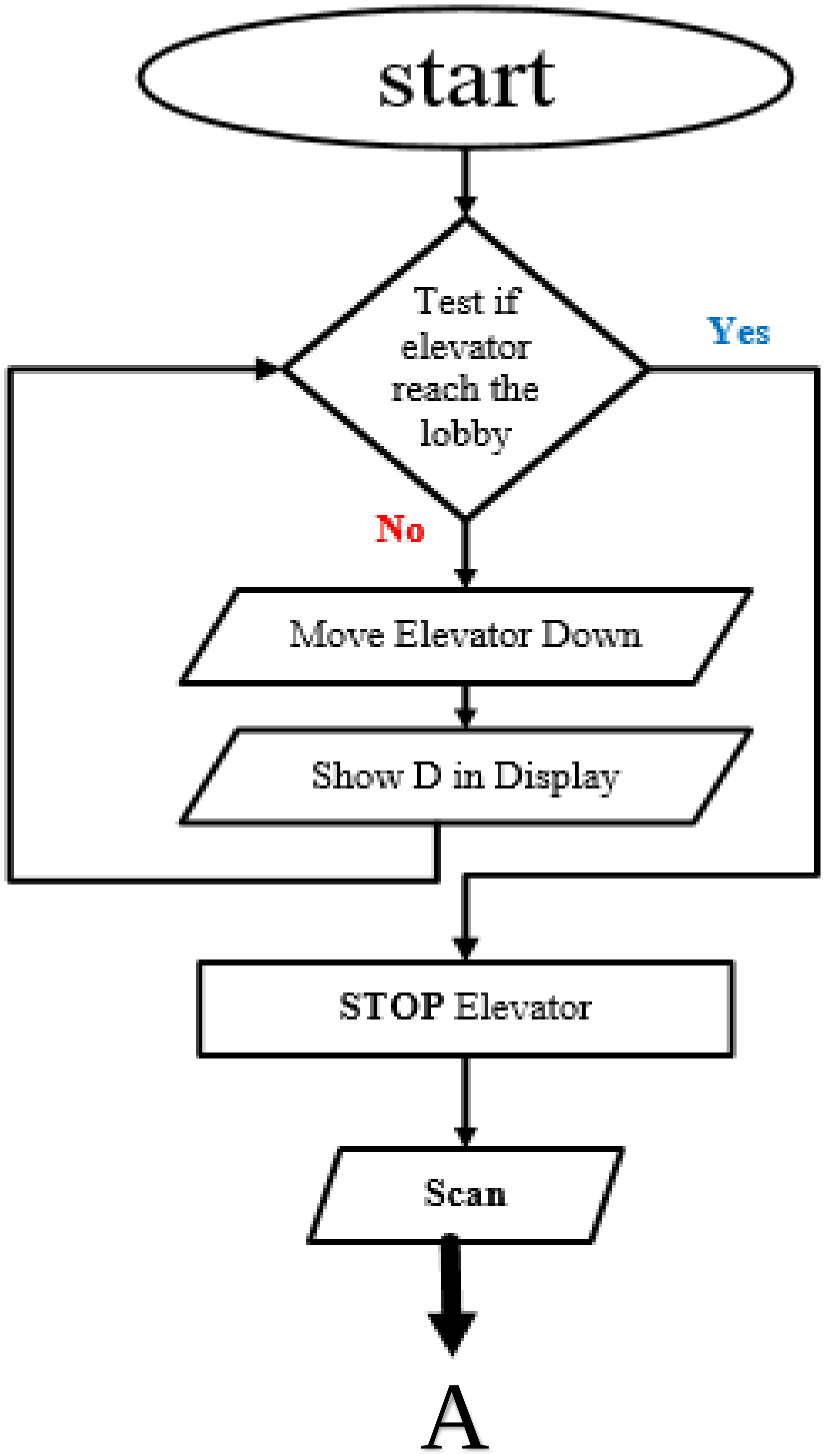


Figure 3.15 Block diagram of elevator control system.

3.3.2 Flow Chart

The flow chart shown in Figure 3.16 describes the flow chart of the proposed software.



Flow chart of Reveling (at begin)

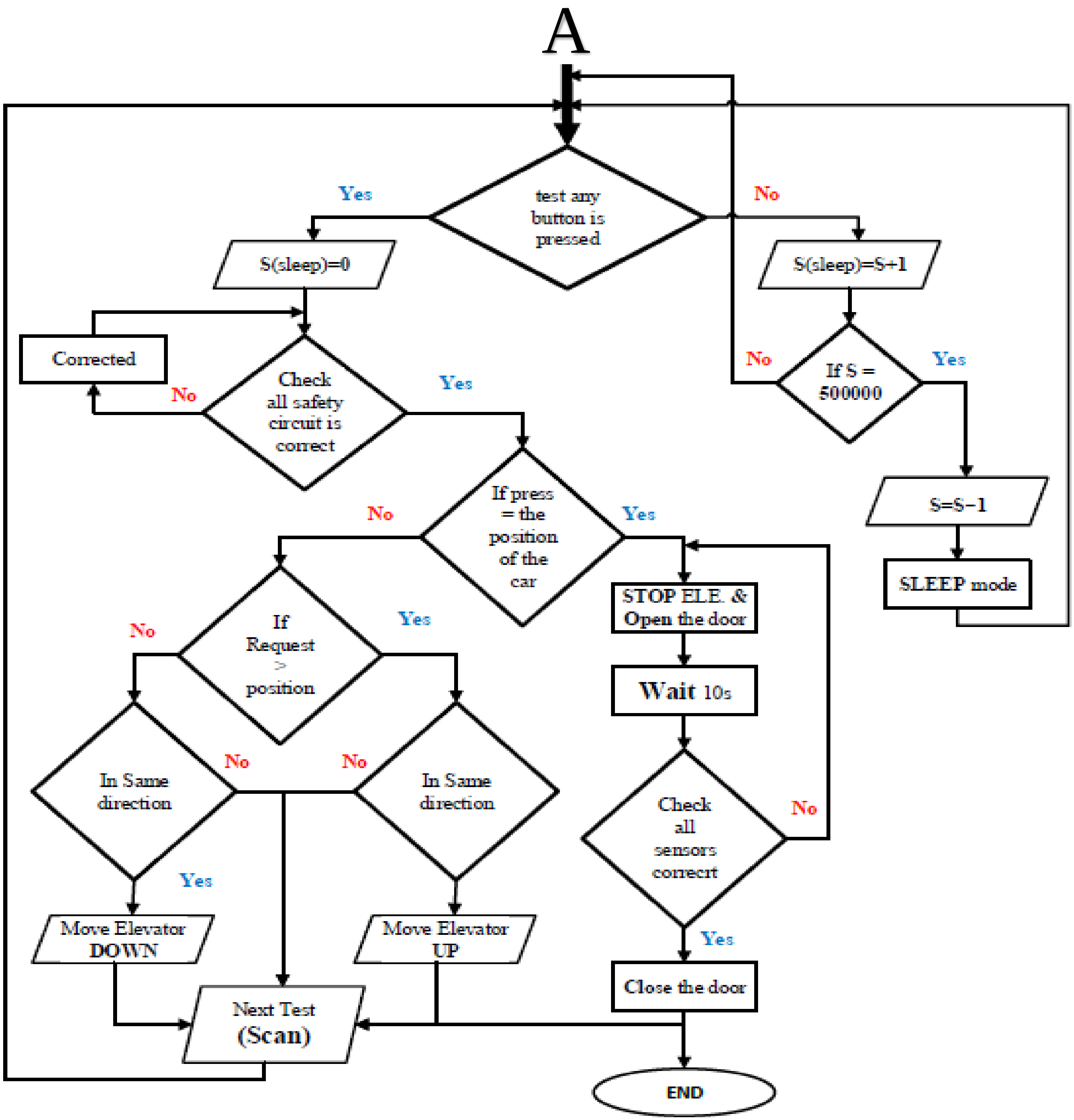


Figure 3.16 Flow char

Chapter Four

SIMULATION AND RESULT

CHAPTER FOUR

Simulation and Results

4.1 Simulation

System was simulated using Proteus 8.1 to verify the software made by Bascom 2.0.5 this will save the time and give the designer an opportunity to make any modification before test the embedded system in real life.

4.1.1 BASCOM - AVR

BASCOM-AVR is four programs in one package, it is known as an IDE (Integrated Development Environment); it includes the Program Editor, the Compiler, the Programmer and the Simulator all together.

For more details Appendix B

Construction of Bascom-AVR code

i. **Start by these comment**

```
$regfile = "atmega16.dat"     'Bascom needs to know the-micro (Atmega16)
$crystal = 4000000            'Bascom needs to know how fast it is going
                              'Frequency

Config PortX = Output/Input   'make these micro pins outputs or inputs
```

ii. **DO ___ LOOP**

```
      Do                      'start of a loop
Program
      Loop                    'return to do and start again
```

iii. **Then SUBs program:**

```
Label:
        Sub program
   Return                          'return to main program
```

4.1.2 Proteus virtual system modelling (VSM)

Proteus Virtual System Modelling (VSM) software offers the ability to co-simulate both high and low-level micro-controller code in the context mixed mode SPICE circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller based designs. For the first time ever, it is possible to develop and test such designs before a physical prototype is constructed.

With this Virtual System Modelling facility, you can transform your product design cycle, reaping huge rewards in terms of reduced time to market and lower costs of development.

The designer can interact with the design using on screen indicators such as LED and LCD displays and actuators such, as switches and buttons. The simulation takes place in real time.

The most important feature of Proteus VSM is its ability to simulate the interaction between software running on a micro-controller and any analog or digital electronics connected to it.

The micro-controller model sits on the schematic along with the other elements of pro-duct design. It simulates the execution of designer object code (machine code), just like a real chip. If the program code writes to a port, the logic levels in circuit change accordingly, and if the circuit changes the state of the processor's pins, this will be seen by the program code, just as in real systems [17].

In short, Proteus VSM improves efficiency, quality and flexibility throughout the design process [17].

For more details Appendix C

As shown in chapter 3 the circuit designed by proteus 8 software (figure 3.11 to figure 3.14).

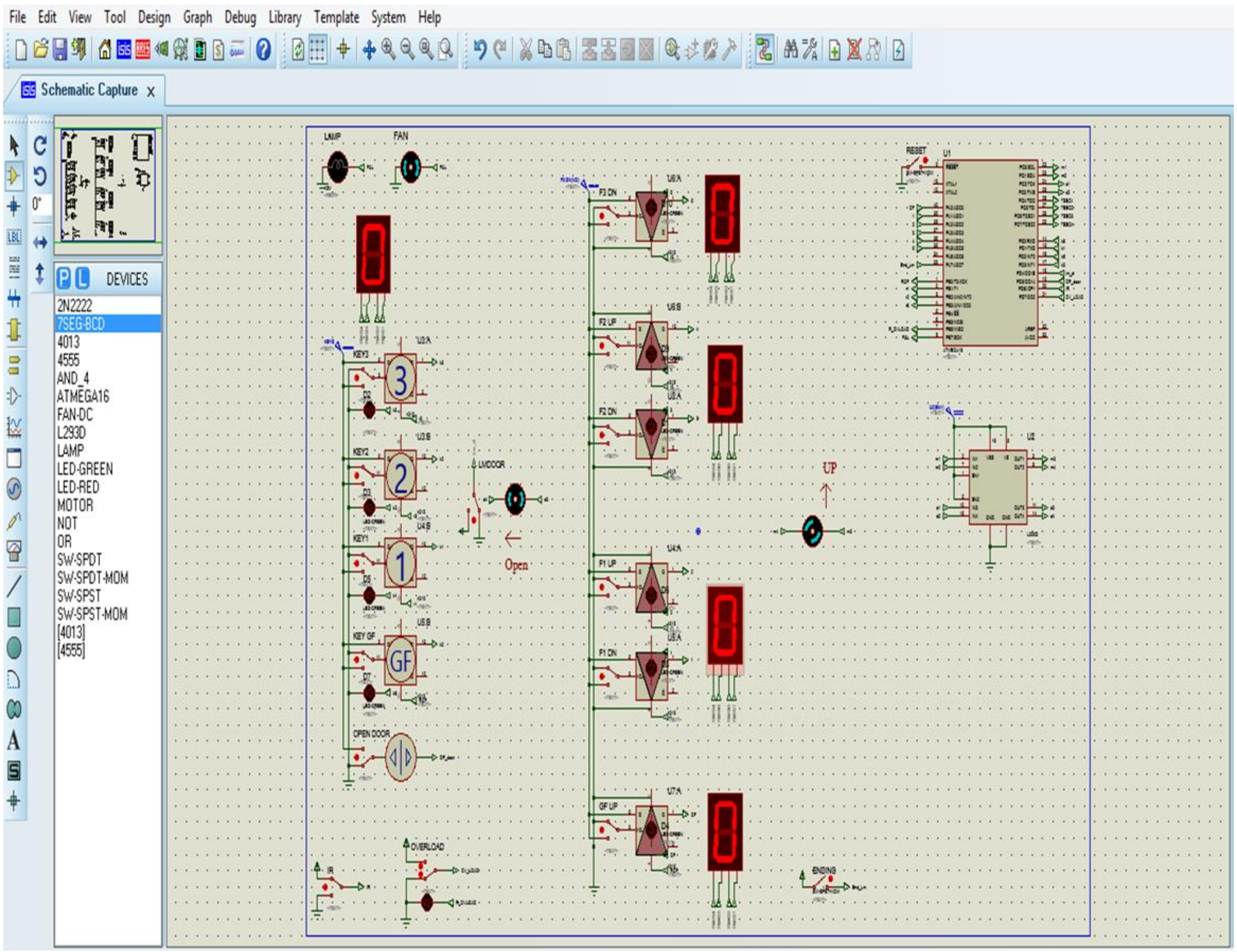


Figure 4.1 Proteus 8 software.

4.2 Result

- To show the result use the proteus to show, take one case of implementation, before start the lamp and fan inside the elevator all are turn off, when start the elevator downing to reach the lobby and touch the final limit as shown in figure 4.3 and in 7SEG shows d for downing.

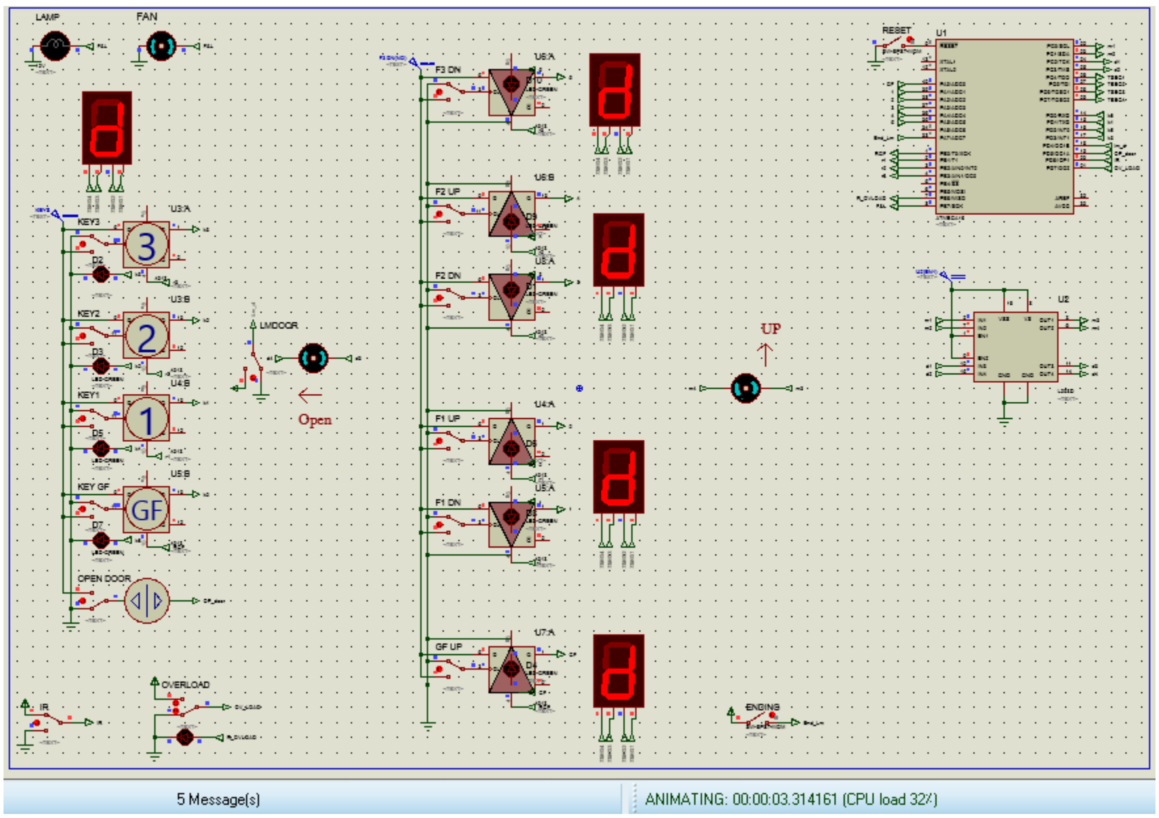


Figure 4.2 elevator goes down.

- When the final limit it pressed that indicate the elevator in the lobby floor now in floor 2 the hall push button up is pressed as shown in figure 4.4, now the lamp and fan is turn on.

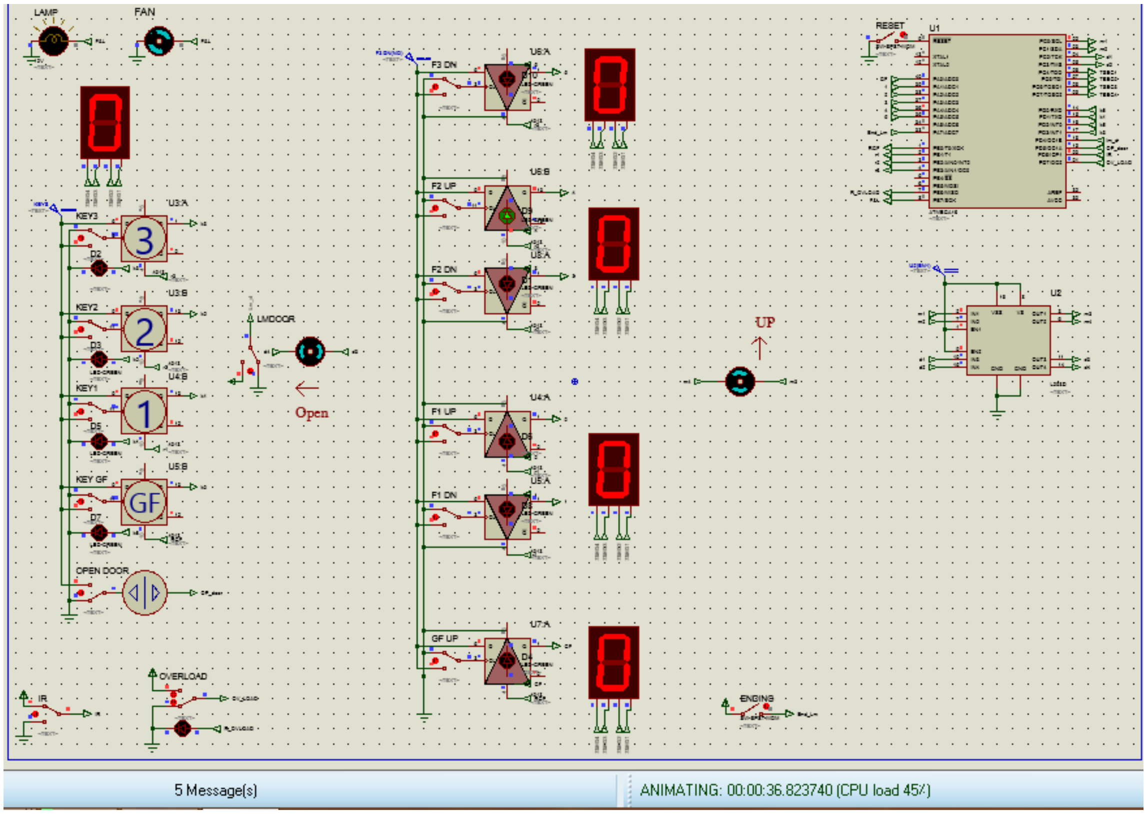


Figure 4.3 Elevator reach the lobby

- The figure 4.5 shows the elevator reach and press key to go to 3rd floor.

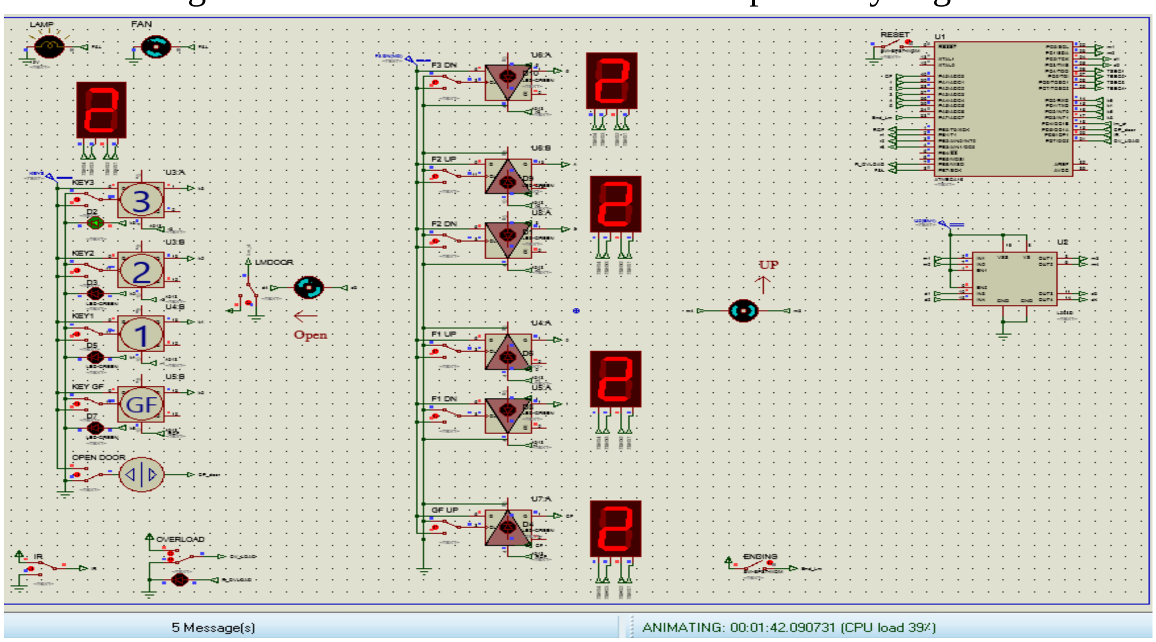


Figure 4.4 Shows the elevator reach and press key to go to 3rd floor.

- The figure 4.6 shows the elevator reach to 3rd floor

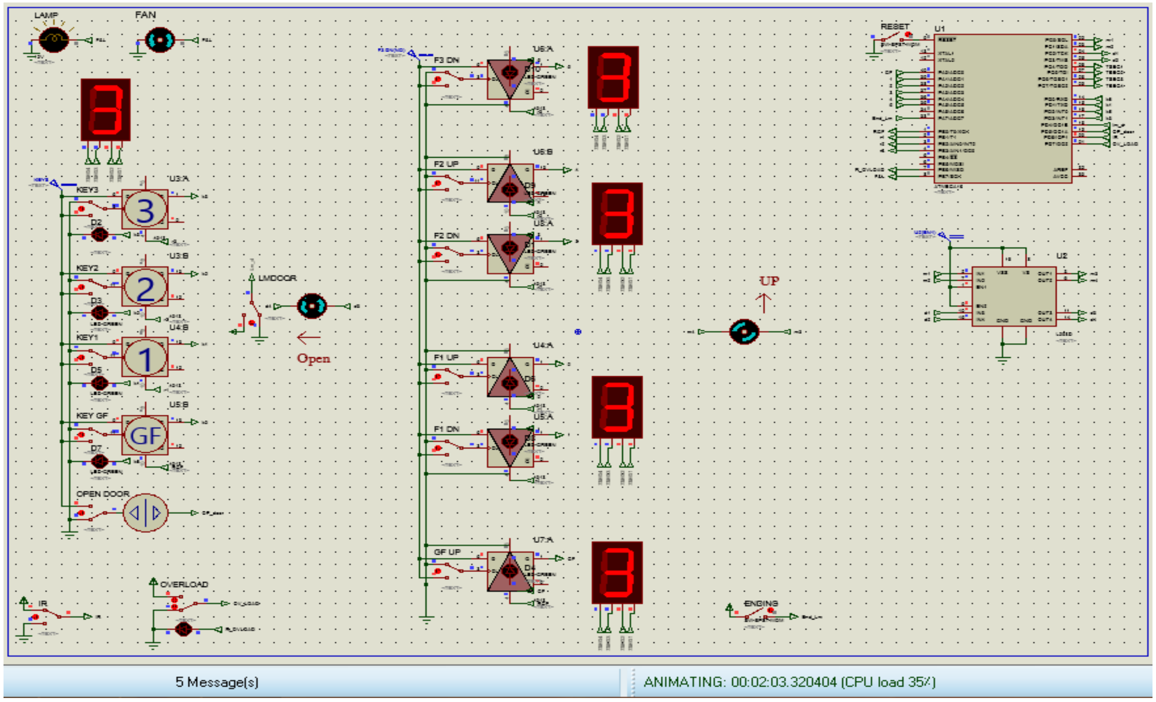


Figure 4.5 Shows the elevator reach to 3rd floor

- The figure 4.7 shows in case two passenger one them in floor two pressed the upper in hall button and the one in first floor it is gone down.

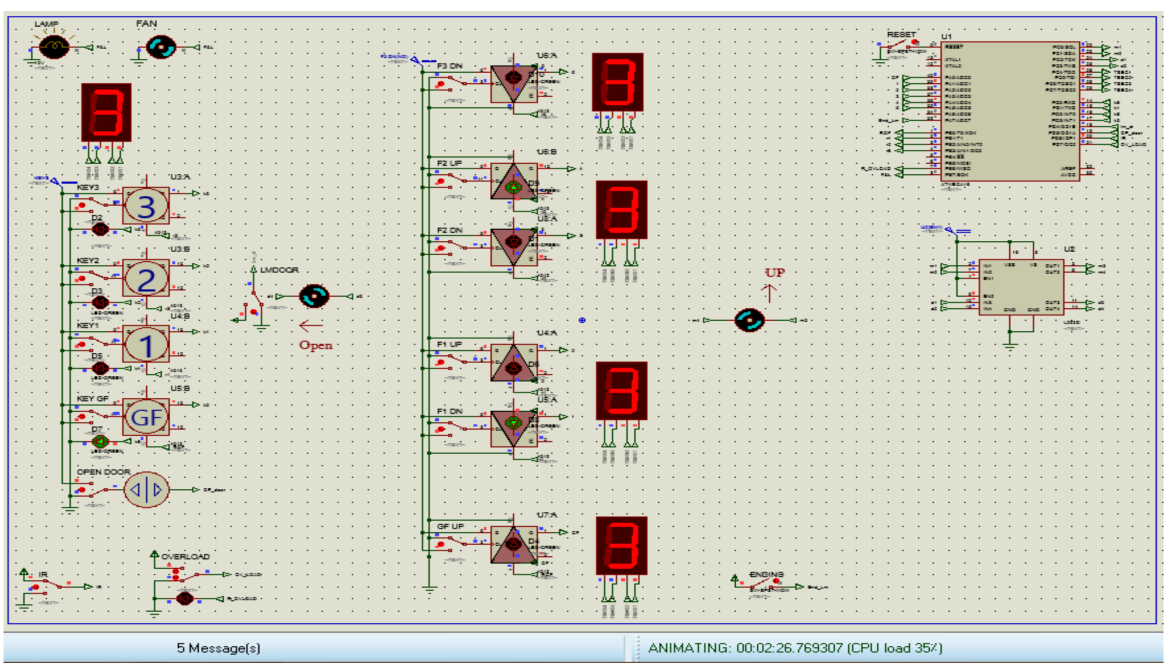


Figure 4.6 shows multi pressed.

- When elevator goes down must be ignore and opposite direction that is shown in figure 4.8, Ignore the upper from 2nd floor and goes to the first floor , then when the car reach first floor pressed from inside to ground floor.

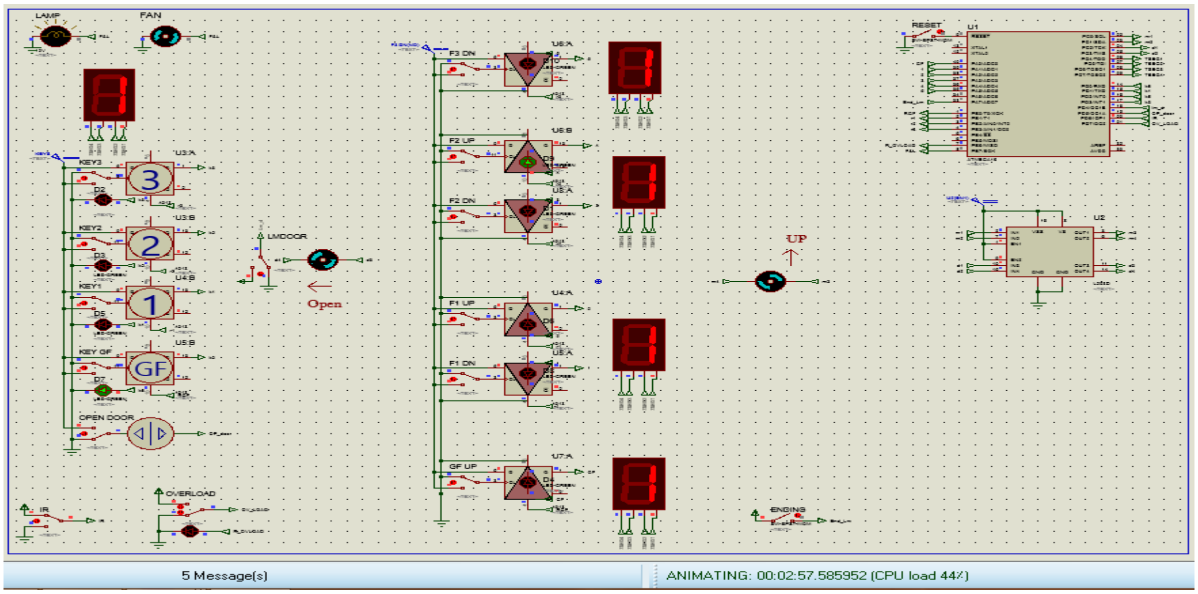


Figure 4.7 Shows ignoring case.

- The elevator goes to ground floor to keep in direction as is shown in figure 4.9.

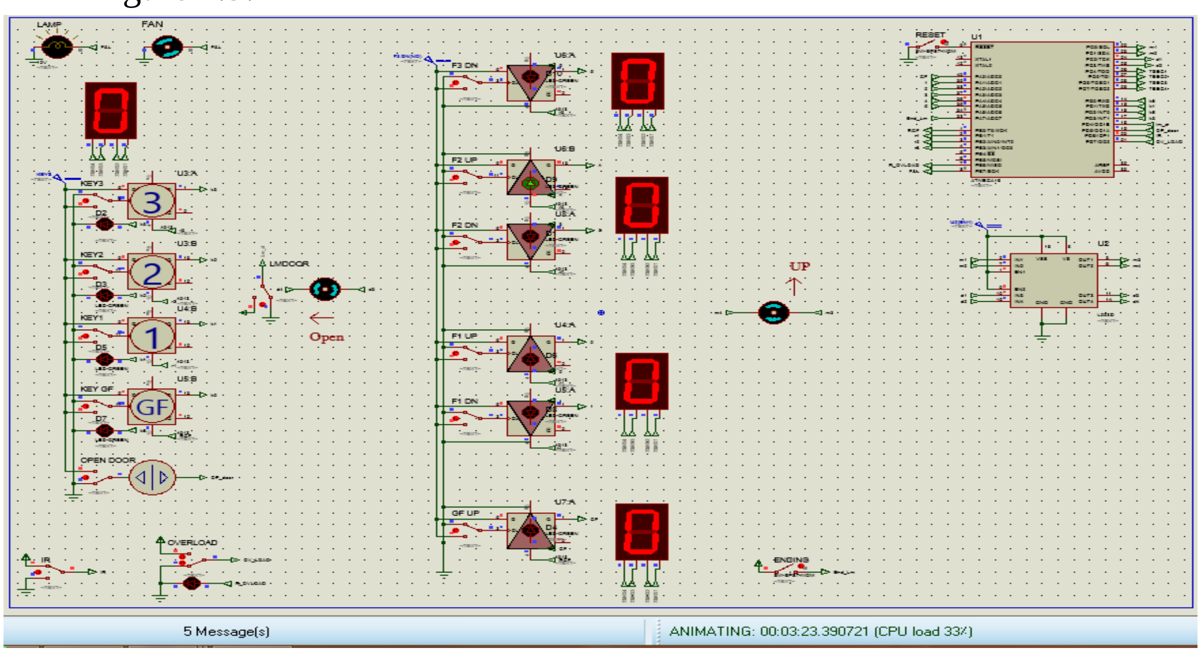


Figure 4.8 Keep in direction.

- Then the elevator reverse the direction to go the 2nd floor to answer the request as shown in figure 4.10.

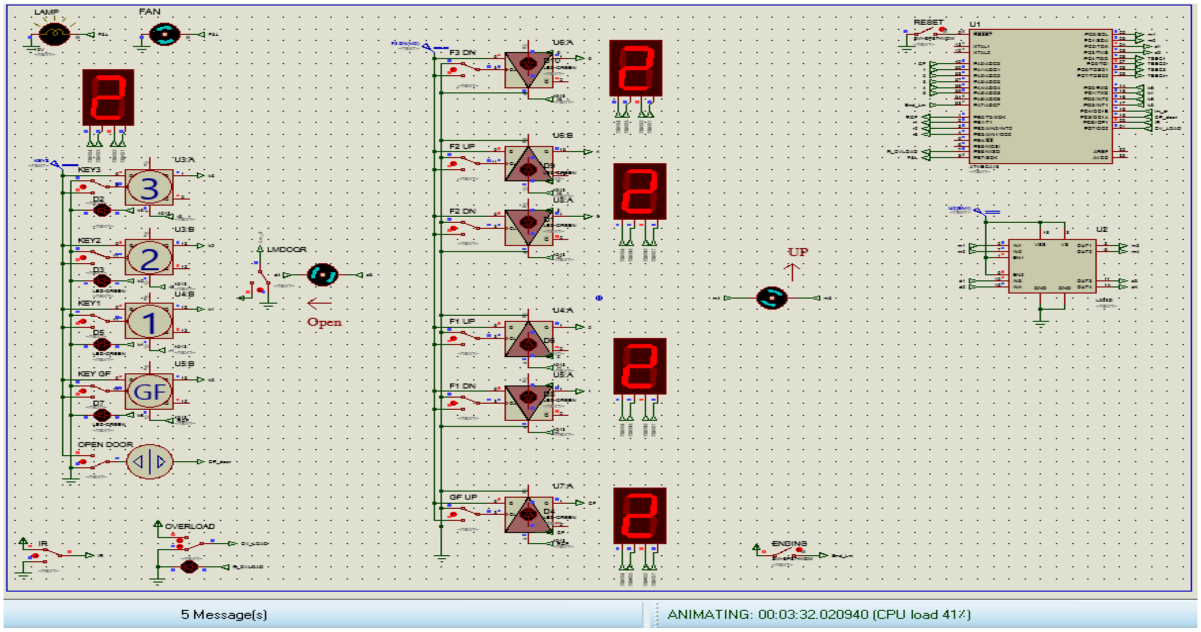


Figure 4.9 answer the 2nd floor.

- Now assume founded overload case (connect overload switch to high) the door is not open and the red led flashy to indicate here is overload case as shown in figure 4.11.

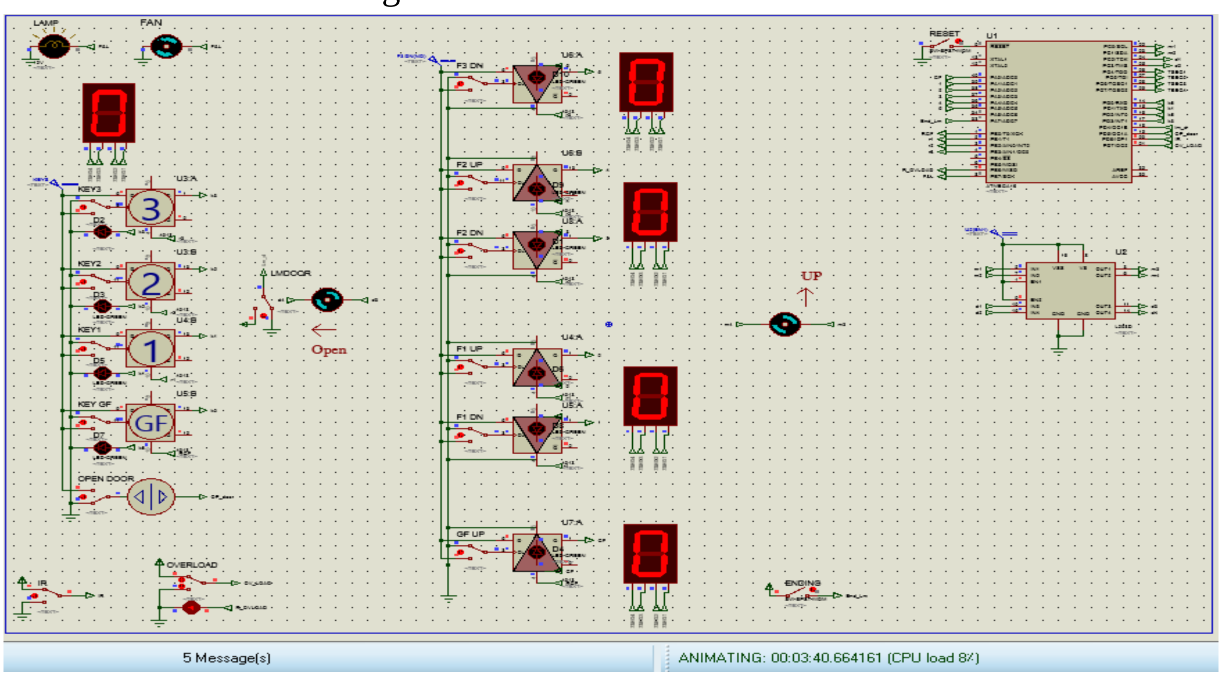


Figure 4.10 Overload case.

- After that stay time to inter in sleep mode that is turn off the lamp and the fan inside the elevator as shown in figure 4.12.

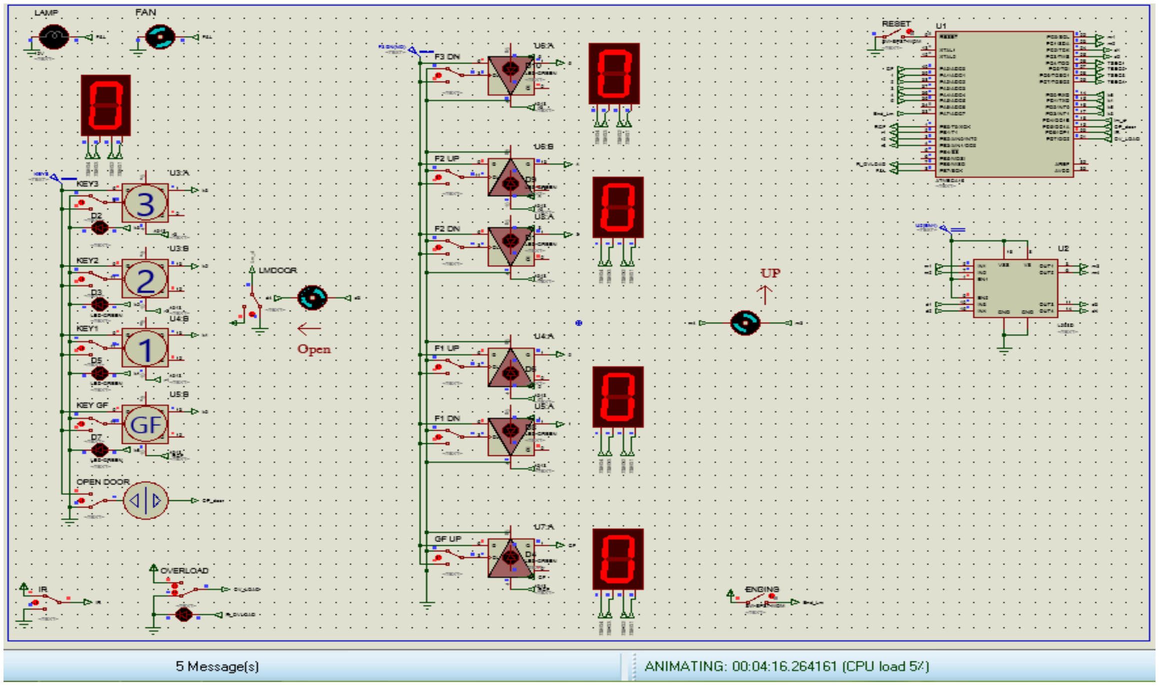


Figure 4.11 Sleep mode

- If any push button the elevator wake up from sleep mode as shown in figure 4.13.

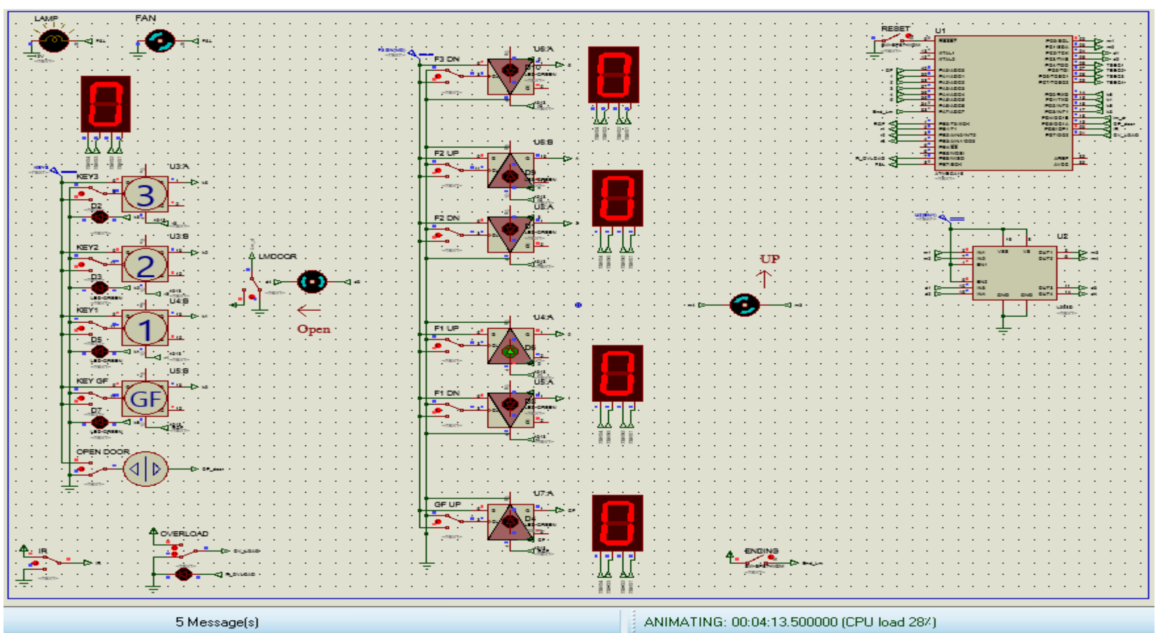


Figure 4.12 Wake up again from sleep mode.

The result in table taken examples of motion and simulation of control the elevator by microcontroller as shown in table 4.1.

Table 4.1 the result of simulation.

Current position of the Elevator	**Actions done by passenger/passengers go to**	**Simulated response of the Elevator**	**Comments on the response**
Ground floor	First floor	Move UP	Reach first floor
First floor	Ground floor , 2nd floor	Move UP to 2nd floor	The direction Before goes up
2nd floor	Ground floor, first floor down	Move down to first then G.Floor	The direction move down
Ground floor	3rd floor , 2nd down	Goes to 3rd floor	Ignore opposite direction
Ground floor	2nd down	Goes to 2nd floor	No any other request
2nd floor	Case Overload	Flashy Red LED	Door doesn't closed
2nd floor	Case IR	Reopen the door	Reopen door
2nd floor	NO any request (period)	Off lamp and fan	Sleep mode
…..			

Chapter Five

CONCLUSION AND RECOMMENDATIONS

CHAPTER FIVE

Conclusion and Recommendation

5.1 Conclusion

In this approach, the ATMEGA 16 microcontroller is chosen as the core control component for elevator control system and DC motor as the implementation component. Based on the key pressed the elevator moves either in upward or downward direction and 7SEG are used for detecting the location of the elevator, thus acquiring time information for opening and closing of the door of elevator. To make the elevator more comfortable for passenger, Selective collective operation is implemented.

This ATMEGA16 can be service four stop (3 floor) floor, when every floors has two switches outside, and can detect IR and overload Sensors addition to final limit switch ,this microcontroller can server two motor one of machine (upward - downward) and the second for door (open – close)

In addition, can be service a Sleep mode is used to perform power saving for efficient operation.

To create program for three floors (4 stops) because it should have many cases at same time, while was overcome by creating one case then one and adding to the program code.

5.2 Recommendations

- Use keypad in side elevator for (keys) instate of switch to increase the floors and minimum the pins of microcontroller.
- Use encoder for hall push button to increase hall push button and reduce the pins of microcontroller.

- Trying to complete to group elevators.
- Connecting the system with other system such as firefighting alarm.
- Provide the system with self-diagnoses to help the end user in troubleshooting.

5.3 The future of elevators

Audio Announcement activated

The elevators may be able to audibly with boarding passengers and recognize by their voiceprints whether they are original population of the building or strangers. If they are strangers they may have to say a password that the car computer is programmed to accept before they can be taken to their intended floor. In addition, if they are intruders with the wrong password, the elevator will refuse to close its doors.

REFFERENCES

1. PATJOSHI, R.K., Design and implementation of embedded based elevator control system. 2010, National Institute of Technology Rourkela.

2. Barney, G.C., Elevator traffic handbook: theory and practice. 2003: Routledge.

3. n.p. www.electricalknowhow.com n.d. [cited 2014 14/October/2014].

4. Luo, L., A UML Documentation for an Elevator System. 2000, December.

5. Patrao, C., J. Fong, and L. Rivet, Energy efficient elevators and escalators. 2009.

6. n.p. " Energy-Efficient Elevator Machines". www.boma.org/research/Documents n.d. [cited 2014 1/October/2014].

7. Strobl, F. and A. Wisspeintner, Specification of an Elevator Control System.

8. n.p. Specifications for Microprocessor Elevator Controls". www.elevatorcontrols.com/pdfs/manuals/specs/EC_Specifications.pdf. n.d. [cited 2014 13/October/2014].

9. Canley, A.M., HYDRAULIC ELEVATOR DESCENT. 1954, Google Patents.

10. McMurray, D.C., Hydraulic elevator drive system. 1976, Google Patents.

11. Nakamura, I., et al., HYDRAULIC ELEVATOR. 1974, US Patent 3,842,943.

12. Ronning, A.G., Pneumatic elevator. 1924, Google Patents.

13. Sors, C.A., Pneumatic elevator by depressure. 1995, Google Patents.

14. Jie, Z., Application of PLC in elevator control system. Journal of Liaoning Normal University (Natural Science Edition), 2009. 3: p. 013.

15. n.p. ATMEGA16. www.atmel.com n.d. [cited 2014 14/October/2014].

16. Cheah, S.H., Microcontroller based lift control system. 2006, Universiti Malaysia Pahang.

17. Su, B. and L. Wang. Application of Proteus virtual system modelling (VSM) in teaching of microcontroller. in E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on. 2010. IEEE.

18. Rafiquzzaman, M., Microprocessors and microcomputer-based system design. 1995: CRC press.

19. Hirota, J. and H. Koh, Lift control system. 1985, Google Patents.

20. Instruments, T., Quadruple Half-H Drivers. L293 (D) datasheet, 1986.

21. Jain, R.P., Modern digital electronics. 2003: Tata McGraw-Hill Education.

22. Horner, J., Automotive electrical handbook. 1986: Penguin.

23. Hanna, J.E., " D" Flip-flop. 1981, Google Patents.

24. Kok, C.W. and Y.C. Tam, D flip-flop. 2008, Google Patents.

25. Kuhnel, C., BASCOM Programming of microcontrollers with ease: An introduction by program examples. 2001: Universal-Publishers.

Appendices

APPENDIX A

Appendix (A) data sheets ATMEGA 16

The ATMEGA16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATMEGA16 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed.

Features

- High-performance, Low-power ATMEL® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
    - 131 Powerful Instructions – Most Single-clock Cycle Execution
    - 32 x 8 General Purpose Working Registers
    - Fully Static Operation
    - Up to 16 MIPS Throughput at 16 MHz
    - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
    - 16 Kbytes of In-System Self-programmable Flash program memory
    - 512 Bytes EEPROM
    - 1 Kbyte Internal SRAM

- Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
- Data retention: 20 years at 85°C/100 years at 25°C(1)
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
- Programming Lock for Software Security

- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
  - Single-ended Channels
  - Differential Channels in TQFP Package Only

- 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

* Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
* I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
* Operating Voltages
  - 2.7V - 5.5V for ATMEGA16L
  - 4.5V - 5.5V for ATMEGA16
* Speed Grades
  - 0 - 8 MHz for ATMEGA16L

- 0 - 16 MHz for ATMEGA16

- Power Consumption @ 1 MHz, 3V, and 25°C for ATMEGA16L.

AVR CPU Core

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Architectural Overview

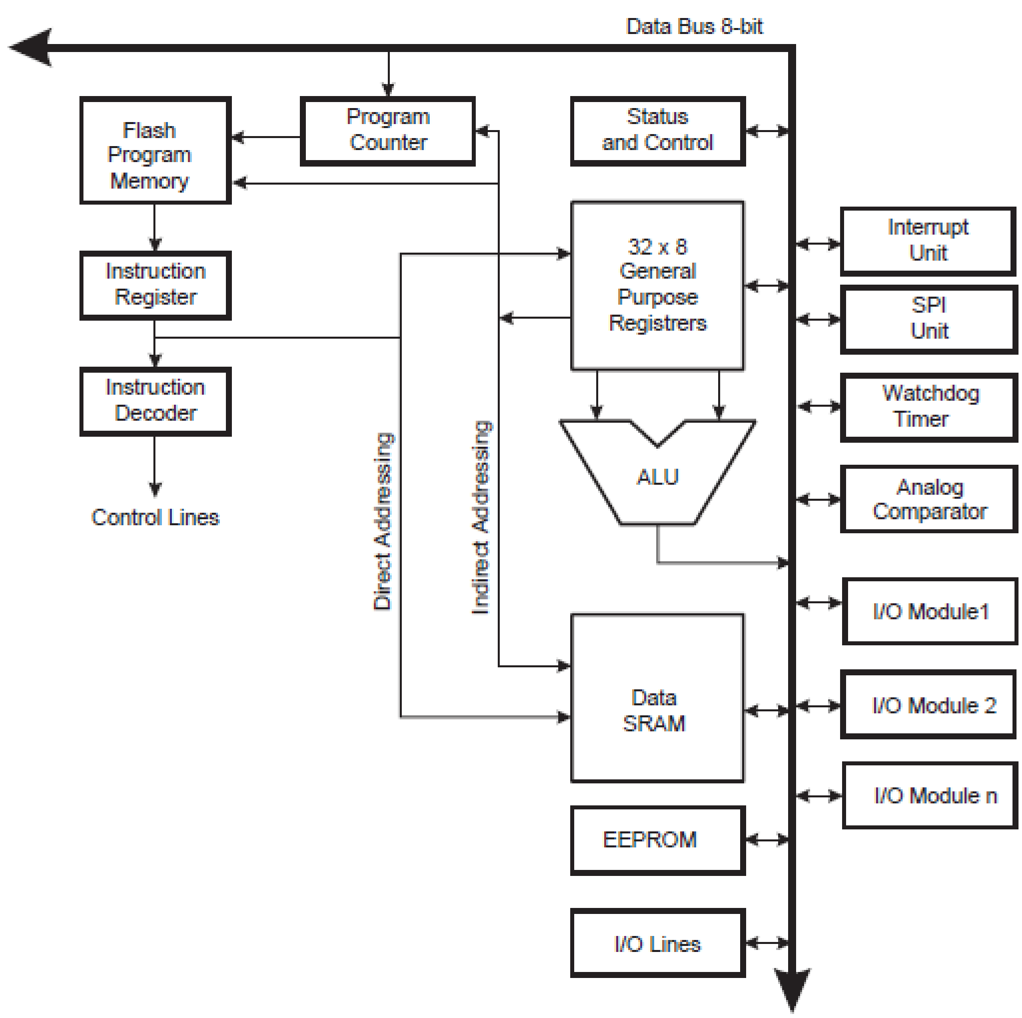


Figure 6.1 Show the Block Diagram of the AVR MCU Architecture

In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 × 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the address pointers can also be used as an address pointer for look up tables in Flash Program memory. These Added function registers are the 16-bit X-register, Y-register, and Z-register.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format.

Every program memory address contains a 16-bit or 32-bit instruction. Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps. A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position.

The lower the interrupt vector address, the higher the priority. The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, $20 - $5F.

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general-purpose working registers. Within a single clock cycle, arithmetic operations between general-purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

PORT B Pins Alternate Functions

Table 6.1 Port B Pins Alternate Functions

Port Pin	**Alternate Functions**
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB4	SS (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/) XCK (USART External Clock Input/Output)

PORT C Pins Alternate Functions

Table 6.2 Port C Pins Alternate Functions

Port Pin	**Alternate Functions**
PC7	TOSC2 (Timer Oscillator Pin 2)
PC6	TOSC1 (Timer Oscillator Pin 1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

PORT D Pins Alternate Functions

Table 6.3 Port D Pins Alternate Functions

Port Pin	**Alternate Functions**
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

APPENDIX B

Appendix (B) BASCOM - AVR

BASCOM was invented in 1995. Many users gave feedback and helped with tips; the software improved a lot during the last 10 years and will so during the next decade.

After writing, the command using Bascom-AVR now to start the compiler that will change your high-level BASIC program into low-level machine code.

If your program is in error then a compilation will not complete and an error box will appear to get to correct the problem.

After successfully compile the program starts the programmer to loaded on the ATMEGA 16 microcontroller, then the BASCOM completes the programming process and automatically resets your microcontroller to start execution of your program [25].

APPENDIX C

Appendix (C) PROTEUS 8.1

Causes to Using Proteus VSM

Proteus VSM was the first product to bridge the gap between schematic and PCB for embedded design, offering system level simulation of microcontroller based designs inside the schematic package itself.

Over ten years later, Proteus VSM is still leading the field with more microcontroller variants and peripherals than any competing product, better debugging tools and instruments and a consistent focus on innovation.

If one person designs both the hardware and the software then that person benefits as the hardware design may be changed just as easily as the software design. In larger organizations where the two roles are separated, the software designers can begin work as soon as the schematic is completed; there is no need for them to wait until a physical prototype exists.

This page outlines some of the benefits to using Proteus VSM as an embedded prototyping tool in the workflow [17].

Proteus VSM for ATMEL® AVR®

Proteus VSM for AVR® contains everything we need to develop, test and virtually prototype embedded system designs based around the ATMEL® AVR® series of microcontrollers. The unique nature of schematic based microcontroller simulation with Proteus facilitates rapid, flexible and parallel development of both the system hardware and the system firmware. This design synergy allows engineers to evolve their projects more quickly, empowering them with the flexibility to make hardware or firmware changes at will and reducing the time to market [17].