MICROCONTROLLER BASED STEPPER MOTOR DRIVE FOR AN ELEVATOR SYSTEM

A

Thesis Report

Submitted in partial fulfillment of the requirement for the award of degree of

MASTER OF ENGINEERING

In

POWER SYSTEM & ELECTRIC DRIVES

Submitted By

Hirdaypal Singh

(Roll No. 800941013)

Under Guidance of

Mr. SOUVIK GANGULI

Assistant Professor, EIED
Thapar University, Patiala

DEPARTMENT OF ELECTRICAL AND INSTRUMENTATION ENGINEERING
THAPAR UNIVERSITY, PATIALA-147004

CERTIFICATE

I hereby certify that the work which is being presented in the Project entitled, **"Microcontroller based stepper motor drive for an elevator system"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Power System & Electric Drive submitted in Electrical & Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Mr. Souvik ganguly(Asst.Prof.)** and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Hirday pal singh

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

13-07-11

Mr. Souvik ganguli
Electrical & Instrumentation Engineering Department
Thapar University Patiala

Countersigned by

Dr.Smarajit Ghosh 14/7/11
Head
Electrical & Instrumentation Engineering Department
Thapar University (Patiala)

Dr. S.K. Mohapatra
Dean (Academic Affairs)
Thapar University,
Patiala

ACKNOWLEGEMENT

I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this project work. First of all, I thank the Almighty, who gave me the opportunity and strength to carry out this work. My greatest thanks are to my parents who bestowed ability and strength in me to complete this work.

I would like to express my gratitude to Mr. Souvik Ganguli, Asstt. Prof. EIED, Thapar University, Patiala for guidance and support throughout this project work . He has been a constant source of inspiration to me throughout the period of this work. I consider myself extremely fortunate for having the opportunity to learn and work under his supervision over the entire period.

I would also take this opportunity to express my gratitude and sincere thanks to Dr. Smarajit Ghosh, Professor and Head, EIED, Thapar University, Patiala for his valuable support. I am also thankful to the entire faculty and staff of EIED, Thapar University, Patiala for their help, inspiration and moral support.

My thanks to all my classmates for their encouragement and help.

HIRDAY PAL SINGH

Roll No: 800941013

ABSTRACT

The thesis addresses a microcontroller based unipolar stepper motor drive used for an elevator system. The elevator system developed can move up and down, along with the feature of floor display indicator incorporated in it. The opening and closing of elevator doors are indicated with green and red LED's respectively. The elevator system is divided into four floors of equal heights, so that the stepper motor took equal number of steps while moving from one floor to another. The elevator system is operated using micro switches, the control program is developed in C language and Kiel compiler is used to converts this control program into executable file or say in a HEX code.

TABLE OF CONTENTS

CHAPTER 2

LITERATURE REVIEW

CHAPTER 3

RESULTS AND DISCUSSION

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MCU……...…….…Microcontroller control unit
MICROC……...…….….Microcontroller
MICROP……...…….…..Microprocessor
CPU……...……...Central processing unit
RAM……..Random access memory
ROM…….……..Read only memory
ADC……..Analog-to-digital converter
CMOS……...…….Complementary metal-oxide semiconductor
CCR…….……Condition code register
I/O……...……...Input and output
SFR……...…….Special function register
SCI……...…….....Serial communication interference
CLK…….……....Clock
GND……..……...Ground
LCD……...……Liquid crystal display
PWM……...……...Pulse-width modulation
IC…….……Integrated chip
DC……...……...Direct current
AC……...……...Alternating current
IR……..……Infra-red
PC……...……Personal computer
LED…….Light emitting diode
CLK……Click left mouse button
PMSM……..Permanent magnet stepper motor
VRSM……...……..Variable reluctance stepper motor
TTL…….,.……...Transistor-transistor logic

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION

This chapter includes the brief discussion of microcontroller, stepper motor drives and the applications of microcontroller and stepper motor. Stepper motors with microcontrollers are used in numerous applications when there is need of controlling the motion. This is due to the robust structure of stepper motor, absence of brushes and better control capability when they are used with microcontrollers. These motors have excellent response to starting, stopping and reversing. The microcontroller is used for sending control pulses to the stepper motor drive. The microcontroller is used both to control the speed as well as position of stepper motor. The speed of stepper motor is directly proportional to the frequency of drive input pulses which is controllable with the help of microcontrollers and the rotation is proportional to the number of output pulses.

1.2 8051 MICROCONTROLLER

1.2.1 MICROPROCESSOR AND MICROCONTROLLER

A digital microcontroller typically consists of three major components: Central Processing Unit (CPU), program and data memory, and an Input/ Output (I/O) system. The CPU controls the flow of information among the components of the computer. It also processes the data by performing digital operations. Most of the processing is done in the Arithmetic-Logic Unit (ALU) within the CPU. When the CPU of a computer built on a single printed circuit board, the computer is called a minicomputer. A microprocessor is a CPU that is compacted into a single-chip semiconductor device. Microprocessors are general-purpose devices, suitable for many applications [1]. A computer built around a microprocessor is called a microcomputer. The choice of I/O and memory devices of a microcomputer depends on the specific application. For example, most personal computers contain a keyboard and monitor as standard input and output devices. A microcontroller is an entire computer manufactured on a single chip. Microcontrollers are usually dedicated devices embedded within an application. For example, microcontrollers are used as engine controllers in automobiles and as exposure and focus controllers in cameras. In order to serve these applications, they have a high concentration of on-chip facilities such as

serial ports, parallel input output ports, timers, counters, interrupt control, analog to-digital converters, random access memory, read only memory, etc. The I/O, memory, and on-chip peripherals of a microcontroller are selected depending on the specifics of the target application. Since microcontrollers are powerful digital processors, the degree of control and programmability they provide significantly enhances the effectiveness of the application [1].

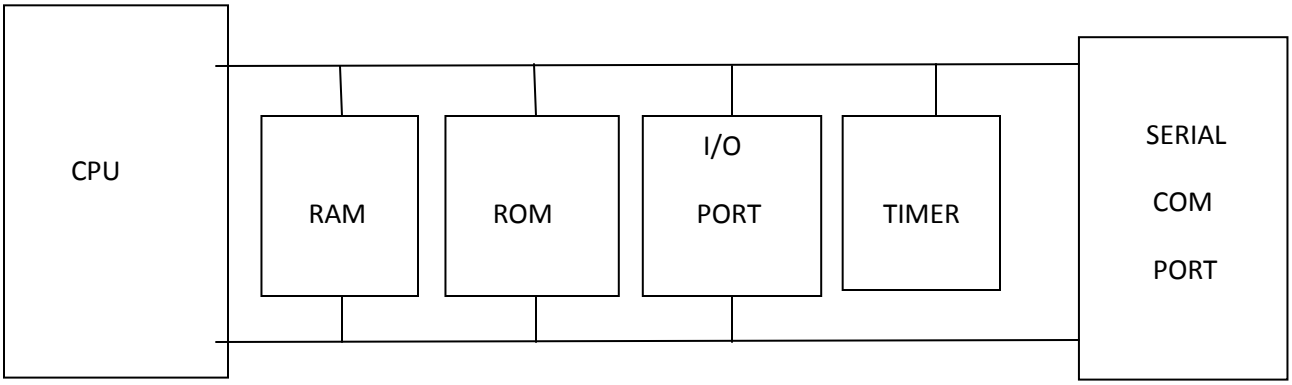

FIGURE 1.1A MICROPROCESSOR SYSTEM

CPU	TIMER	ROM
RAM	I/O PORT	SERIEL COM PORT

FIGURE 1.1B MICROCONTROLLER SYSTEM

1.2.2 MICROCONTROLLER STRUCTURE

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications [1]. A microcontroller has a CPU in addition to a fixed amount of RAM, ROM, I/O ports and a timer on a single chip. The fixed amount of RAM, ROM and I/O ports in microcontroller makes them ideal for many applications in which cost and space are critical. The microcontrollers have an 8-bit data bus. They are capable of addressing 64K of program memory and a separate 64K of data memory. The 8051 has 4K of code memory implemented as on-chip. Read Only Memory (ROM). The 8051 has 128 bytes of internal

Random Access Memory (RAM). The 8051 has two timer/counters, a serial port, 4 general purpose parallel input/output ports, and interrupt control logic with five sources of interrupts [1].

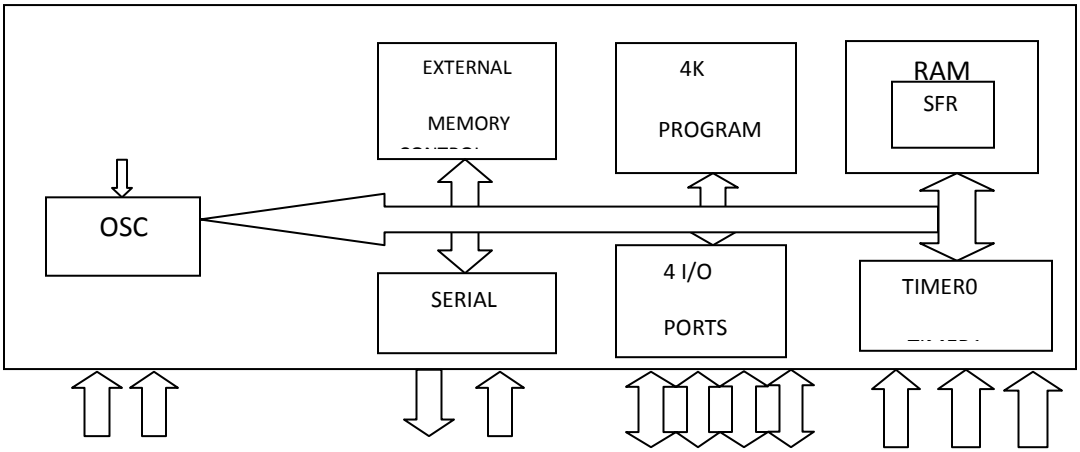

FIGURE 1.2 8051 MICROCONTROLLER STRUCTURE

1.2.3 PIN CONFIGURATION AND DESCRIPTION

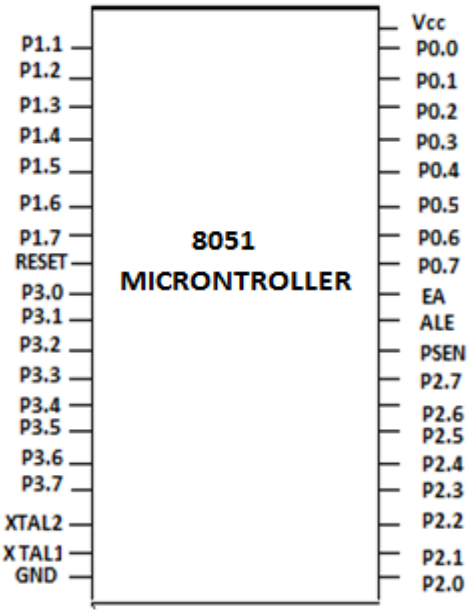

FIGURE 1.3 8051 MICROCONTROLLER

PINS 1-8(PORT 1): Each of these pins can be configured as an input or an output.

PIN 9(RS): A logic one on this pin disables the microcontroller and clears the contents of most registers. In other words, the positive voltage on this pin resets the microcontroller. By applying logic zero to this pin, the program starts execution from the beginning.

PINS 10-17(PORT 3): Similar to port 1, each of these pins can serve as general input or output. Besides, all of them have alternative functions:

PIN 10(RXD): Serial asynchronous communication input or Serial synchronous communication output.

PIN 11(TXD): Serial asynchronous communication output or Serial synchronous communication clock output.

PIN 12(INT0): Interrupt 0 input.

PIN 13(INT1): Interrupt 1 input [1].

PIN 14(T0): Counter 0 clock input.

PIN 15(T1): Counter 1 clock input.

PIN 16(WR): Write to external (additional) RAM.

PIN 17(RD): Read from external RAM.

PIN 18, 19 (X2, XI): Internal oscillator input and output. A quartz crystal which specifies operating frequency is usually connected to these pins. Instead of it, miniature ceramics resonators can also be used for frequency stability [1].

PIN 20(GND): Ground.

PIN 21-28(PORT 2): If there is no intention to use external memory then these port pins are configured as general inputs/outputs. In case external memory is used, the higher address byte, i.e. addresses A8-A15 will appear on this port.

PIN 29(PSEN): If external ROM is used for storing program then a logic zero (0) appears on it every time the microcontroller reads a byte from memory.

PIN 30(ALE): Prior to reading from external memory, the microcontroller puts the lower address byte (A0-A7) on P0 and activates the ALE output. After receiving signal from the ALE pin, the external register memorizes the state of P0 and uses it as a memory chip address. Immediately after that, the ALU pin is returned its previous logic state and P0 is now used as a Data Bus. As seen, port data multiplexing is performed by means of only one additional (and cheap) integrated circuit. In other words, this port is used for both data and address transmission.

PIN 31(EA): By applying logic zero to this pin, P2 and P3 are used for data and address transmission with no regard to whether there is internal memory or not. It means that even there is a program written to the microcontroller, it will not be executed. Instead, the program written to external ROM will be executed. .

PIN 32-39(PORT 0): Similar to P2, if external memory is not used, these pins can be used as general inputs/outputs. Otherwise, P0 is configured as address output (A0-A7) when the ALE pin is driven high (1) or as data output (Data Bus) when the ALE pin is driven low (0).

PIN 40(VCC): +5V power supply.

1.2.4 INPUT AND OUTPUT PORTS

All 8051 microcontrollers have 4 I/O ports each comprising 8 bits which can be configured as inputs or outputs. Accordingly, in total of 32 input/output pins enabling the microcontroller to be connected to peripheral devices are available for use [1].

Pin configuration, i.e. whether it is to be configured as an input (1) or an output (0), depends on its logic state. In order to configure a microcontroller pin as an input, it is necessary to apply logic zero (0) to appropriate I/O port bit. In this case, voltage level on appropriate pin will be 0.

Similarly, in order to configure a microcontroller pin as an input, it is necessary to apply a logic one (1) to appropriate port. In this case, voltage level on appropriate pin will be 5V (as is the case with any TTL input).

1.3 DESIGN CONSIDERATION OF STEPPER MOTOR

The electric compatibility between the motor and the driver are most critical factors in a stepper motor system design, some general guidelines in the selection of these components are [2]:

1.3.1 INDUCTANCE

Stepper motors are rated with a varying degree of inductance. A high inductance motor will provide a greater amount of torque at low speeds and lower torque at high speeds.

1.3.2 MOTOR STIFFNESS

Stepper motor runs stiffly due to the design. By reducing the current flow to the motor by small percentage will smoothes the operation and by increasing current will increase the stiffness but also provide more torque.

1.3.3 MOTOR HEAT

Stepper motors are designed to run hot between the ranges of 50-90 degree. But many times current may cause excessive heating and damage to the motor insulation and winding [2].

1.4 TYPES OF STEPPER MOTOR

On the bases of wire arrangement stepper motors are categorized into two types [3].

1. Unipolar stepper motor.
2. Bipolar stepper motor.

1.4.1 UNIPOLAR STEPPER MOTOR

A unipolar stepper motor has two windings per phase, one for each direction of magnetic field. Since in this arrangement a magnetic pole can be reversed without switching the direction of current, the commutation circuit can be made very simple (eg. a single transistor) for each winding. Typically, given a phase, one end of each winding is made common: giving three leads per phase and six leads for a typical two phase motor [4].

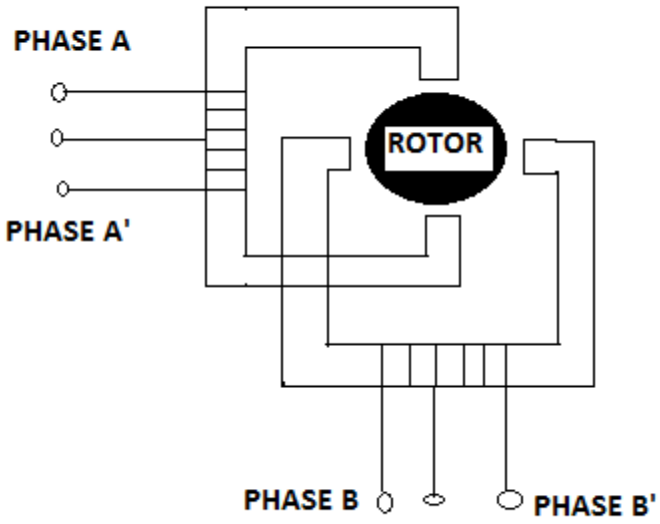

FIGURE 1.4 UNIPOLAR STEPPER MOTOR

1.4.2 BIPOLAR STEPPER MOTOR

Bipolar motors have a single winding per phase. The current in a winding needs to be reversed in order to reverse a magnetic pole, so the driving circuit must be more complicated; typically with an H-bridge arrangement (however there are several off the shelf driver chips available to make this a simple affair). There are two leads per phase, none are common. Static friction effects using an H-bridge have been observed with certain drive topologies.

Because windings are better utilized, they are more powerful than a unipolar motor of the same weight. This is due to the physical space occupied by the windings. A unipolar motor has twice the amount of wire in the same space, but only half used at any point in time, hence is 50% efficient (or approximately 70% of the torque output available). Though bipolar is more complicated to drive, the abundance of driver chip means this is much less difficult to achieve. This type of motor is more efficient than a unipolar stepper motor.

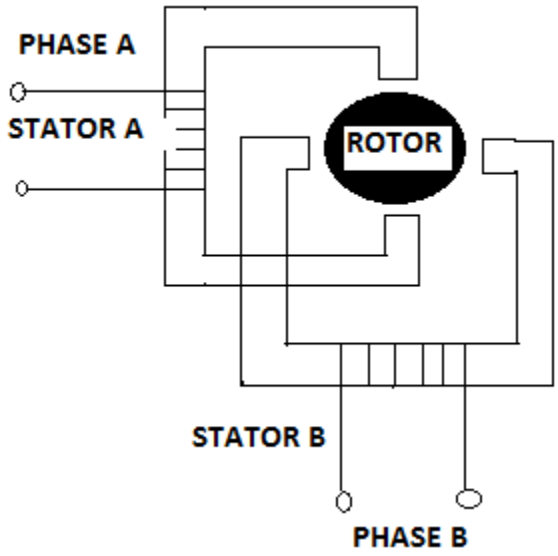

FIGURE 1.5 BIPOLAR STEPPER MOTOR

1.5 TYPES OF STEPPER MOTOR DRIVES

1.5.1 INTRODUCTION

The stepper motor drivers receive low level signals from the control system and convert the m into electrical pulses to run the motor. One step pulse I may be required s required for every step of the motor shaft. In full step mode with a standard 200 step motor, 200 step pulses required to complete one revolution. In micro stepping mode the driver may be required to generate 50,000 or more step pulses per revolution [3].

Speed and torque performance of stepper motor is based on the flow of current from the driver to the motor winding. The lower the inductance the faster the current gets to the winding and better the performance of the motor. To reduce the effect of inductance most type of driver circuits are designed to supply a voltage greater than the motor rated voltage.

There are three types of drives used in commercial and in industrial applications.

1. Unipolar drive.
2. R/L drive
3. Bipolar chopper drive.

A stepper motor is driven from a low voltage dc source. . The dc power supply to drive circuits should be as much free from harmonics as is possible. For optimum torque output from a stepper motor, the drive circuits should possess the following features [3]:

During turn on the current in the phase winding should rise from zero to desired level in short duration.

During turn off the current in the phase winding should be forced to zero in short duration.

1.5.2 UNIPOLAR DRIVE

The facts drive is named unipolar because the fact that the current flow is limited to one direction. The switch set of unipolar drive is fairly simple and expensive. The drawback to using unipolar drive is its limited capability to energize all the windings at any one time. In unipolar stepper motor the torque is proportional to the square of current, the direction of current in phase winding for operating this motor is unimportant. In this manner the four phase windings A, B, C and D are excited in appropriate manner. The phase windings have large inductance, there for in series in each phase winding a forcing resistance is inserted so to reduce the circuit time constant. A resistance is also placed in series with the freewheeling diode in order to dissipate the energy stored in the phase winding inductance [3].

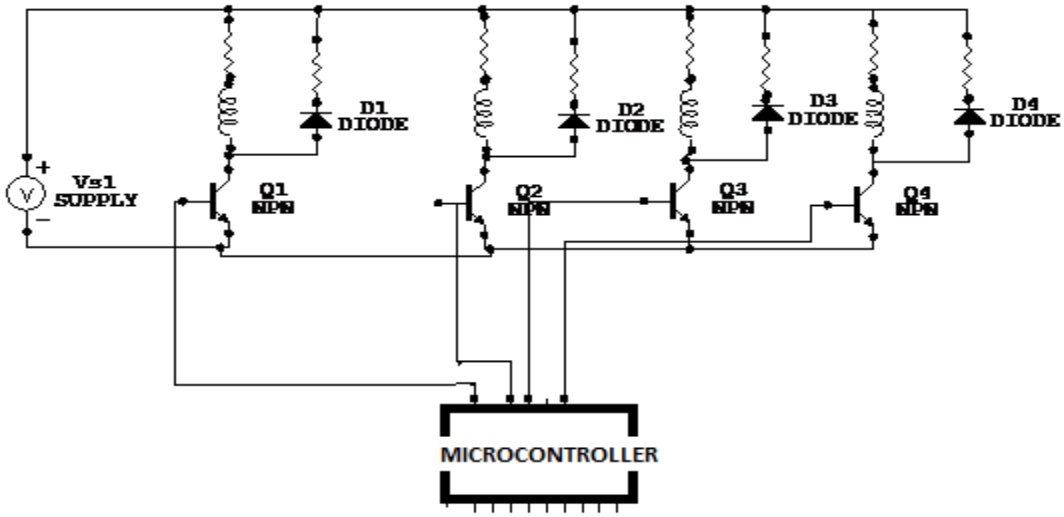

FIGURE 1.6 UNIPOLAR DRIVE CIRCUIT

1.5.3 RESISTANCE/LIMITED DRIVE

These drives are too old as compared to the today driver's technology. They only exist in some applications because they are simple and inexpensive. The drawback to using these drives is that these drives rely on "dropping resistor" to get almost 10 times the amount of motor current rating necessary to maintain a useful increase in speed. This process produces an excessive amount of heat and must rely on Dc power supply for its current source [3].

1.5.4 BIPOLAR DRIVE

These drives are mostly used in the industrial applications. These drives are more expensive but they offer high performance and efficiency. In bipolar stepper motors the torque is proportional to phase current. The direction of rotation in the motors is dependent on the direction of current in phase windings. For rotation in one direction therefore bi-polar drive circuit for each phase is required. Here T1 to T4 are controllable switches. These switches may be any semiconductor switch depends on the current to be handled by them. When the switches are closed, current flows in the direction of arrows associated with them. The diodes in parallel with each of the semiconductor controllable switch protect these controllable switches from voltage spikes produced from the switching of the inductor [3].

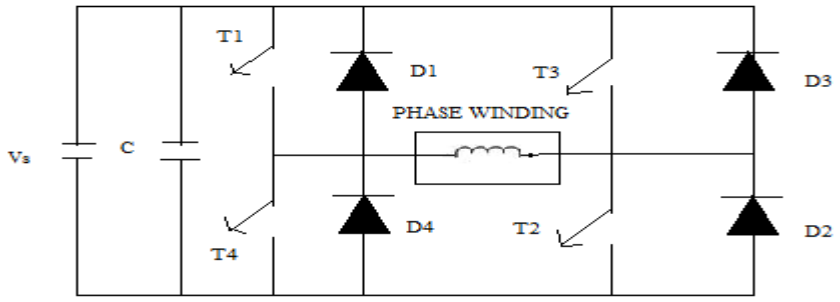

FIGURE 1.7 BIPOLAR DRIVE CIRCUIT

1.6 SELECTION CRITERIA FOR STEPPER MOTOR

When a stepper motor is selected for a specific task eight different things must be considered. Operating speed in steps/second according to the task of need of the task, Torque produce by the motor in N-m, Load inertia of the motor, Required step angle resolution, Time to accelerate in ms, Time to decelerate in ms, Type of drive to be used and size and weight consideration. These considerations are important for selecting a stepper motor for a required task [5].

1.7 APPLICATIONS OF STEPPER MOTOR AND MICROCONTROLLER

The stepper motors with microcontrollers used in many applications due to the easy control and robust structure. Stepper motor used in mainly position, speed and measurement applications. The microcontrollers are used in elevators in commercial sector [6]. Stepper motor and microcontroller is used in cattle feed mixing with the aim of optimizing the efficiency and consistency of mixing in order to improve the quality of feed for producing best quality and quantity of milk [7]. Peristaltic pumps and stepper motors control the quantities of liquid nutrient additives and actuators vary the feed rate of the mix and extension of the feed conveyor. To The stepper motors are used for storing the kinetic energy of the vehicles by minimizing the fuel consumption and emissions by the regeneration of braking energy and by minimizing the inefficient part of load operation of the engine [8]. The stepper motor is used in colour non impact printers and in computer disk drivers [9]. Microcontrollers are used in many sectors due to programmable capability. The microcontrollers are used to control the remotely located stepper motors [10]. Stepper motor drive is used as on load tapchanger in electric locomotive. Microcontroller with stepper motor is used for pipelining control. Stepper motors are used in CNC mills and robotics. Stepper motors are used in space satellites for the purposes of solar array drive [11]. Stepper motors are used in automatic camera steering [12] and in spacecraft applications. Steppers motors are used in thrust control in marines and in aviation applications. Microcontrollers are used in induction motors drives and with stepper motor it is used in automotive fuel pumps [13]. Microcontroller and stepper motors are used in surveillance system to protect forests from fire. The surveillance system has a capacity for detecting fires and other ecological crimes [14]. The system uses a digital camera that is controlled remotely by a microcontroller. In positioning the camera, the system uses a stepper motor for the vertical movement of the camera and a DC motor for the horizontal movement. The system also captures information about the temperature, humidity, wind speed and direction. A stepper motor and microcontroller is used for measuring water level. A microcontroller and stepper motor are used in solar tracing applications for automatically movement of solar panel towards the sun for increasing the efficiency of photovoltaic cells. Stepper motors and microcontrollers are used in making turn tables and antenna positioning system for testing the electromagnetic radiation emitted by the electronic and electrical component in open area test site [15].

1.8 CONCLUSION

Stepper motor are provides good control over the position and speed. These days in the elevators stepper motors are used because of the robust design, absence of brushes and good control over speed and position. The microcontroller is used to send control pulses to stepper motor driver because of the programmable property. These control pulses are not sufficient to drive the motor so we are using stepper motor drives. These drivers find good features over the semiconductor switches which require a firing circuit. Control over the firing circuit is not easy and losses are also more as compared to the drivers.

CHAPTER 2 LITERATURE REVIEW

2.1 INTRODUCTION

Stepper motors and microcontrollers are used in many applications like position control, speed control and measurement applications. Different applications and controlling techniques are described by different authors in this literature review. Microcontrollers and stepper motors are widely used in elevator system. Different types of elevator systems based on stepper motor and microcontroller are also described in this literature review. An elevator is a transport device used to move goods or people vertically. In British English and other Commonwealth English, elevators are known more commonly as lifts, although the word elevator is familiar from American movies and television shows. In the 1800s, with the advent of electricity, the electric motor was integrated into elevator technology by German inventor Werner von Siemens [24]. With the motor mounted at the bottom of the cab, this design employed a gearing scheme to climb shaft walls fitted with racks. By 1903, this design had evolved into the gearless traction electric elevator, allowing hundred-plus story buildings to become possible and forever changing the urban landscape. Multi-speed motors replaced the original single-speed models to help with landing-leveling and smoother overall operation. Electromagnet technology replaced manual rope-driven switching and braking. Besides, Push-button controls and various complex signal systems modernized the elevator even further. Safety improvements have been continual, including a notable development by Charles Otis.

T.S. Weerakoon described a novel drive topology for a five phase stepper motor in detail. This paper presented that a low cost, standard stepper motor drive IC are used to derive a novel drive topology for five phase stepper motor which enables closed loop speed and position control powered by inner current control loop. The designed driver have full-step, half-step, clockwise and counter clockwise drive modes with the speed control and current control [2].

S.G. Abeyratne gave simplification about how we can transfer the stored energy in the inductance winding of uni-polar stepper motor when semiconductor switches are off. Author designed a circuit having no zener suppressors and condensers which are used mostly in uni-

polar drive circuit in industries for the protection of driving circuit from over voltages due this stored energy. Author implemented fast recovery diodes which are capable of transferring this stored energy back to the system and also enhances the torque capability of motor [4].

Zhang Yagun et al presented a design of elevator system. 89S52 microcontroller is used to control the position of elevator system. LCD is used to display the real time information of elevator. IR sensors are connected to every floor which detects the position of the elevator. The L298 driver is used to operate the stepper motor. MCU is used to controls the speed as well as position of the stepper motor [6].

G.S Barlow et al designed a microcontroller based automatic feed mixer with the aim of optimizing the efficiency and consistency of the mixer. Peristaltic pumps and stepper motor control the quantity of liquid nutrient additives and microcontroller is employed for controlling the speed of stepper motor [7].

Pullen et al proposed a controlling system to reduce the fuel consumption and emissions by the regeneration of braking energy and minimizing inefficient part load operation of engine by using microcontroller and stepper motor [8].

J.Barrow et al developed an integrated chip for colour non impact printer and motor control system. The two IC's include three switch mode PWM controllers, two unipolar stepper motor drives, a DC motor H bridge, a serial control post and a host of peripheral system functions. The controller is used to control the speed as well as position of the stepper motor [9].

Mountinho et al described a surveillance system for forest environment and protects the natural area from fire. The stepper motor is used for moving the digital camera vertically and PIC 16F877 microcontroller is used to control the motion of stepper motor. This system also captures the humidity, wind speed and direction [14].

Zarar Bin Mohammad Jenu designed a stepper motor based position controlling system for a antenna and a turntable for measuring the electromagnetic field of a electrical or electronics component located at open area for testing. The stepper motor and a DC motor are used to drive the turntable and antenna positioning device. Microcontroller is used to control these motors [15].

Ross Bannatyne et al introduced 89S51 microcontroller based applications, overview, pin configuration and working. This paper also introduced about the software tools used in programming [16].

Chin Ming Hsu et al constructed a set of 89S51 microcontroller teaching tool for students to learn the basic knowledge of microcontroller system designing, software simulation tools and flash downloading tools [17].

Xiahua Zhangx et al presented a stepper motor control system is which adopted high-performance AVR microcontroller Atmega128 and serial communication. The system mainly consists of host computer, console and a stepper motor. The data is interacted by the serial port between PC and console. Ultimately, the direct control of stepper motor by the PC is achieved. After debugging of the designed control system, the results demonstrate that the control circuit and program are simple and practical [18].

Archan Patel et al proposed the design and hardware implementation of bipolar micro-stepping drive for disc rotor type stepper motor. This type of motor has advantages like high torque at high speed, very low moment of inertia and low power consumption. The micro-stepping control improves the positioning accuracy; eliminate low speed ripples and resonance effects [19].

V Marceno et al presented a novel fully integrated 65W stepper motor driver IC. It is capable of control up-to two stepper motors and four DC motors. This IC is fully protected and programmable and reduced the need of external components for protection due to in built protection facility with high flexibility [22].

Hiufenq Jiao et al developed a stepper motor based tracking system for increasing the efficiency of photovoltaic panel. The stepper motor is so controlled by microcontroller that it is always incident to sunlight at a angle of 90 degree. Position sensors are used to control the position of stepper motor [25].

Hilni Fadzil et al designed a mechanical structure for solar tracker based on stepper motor. The circuit consists of two stepper motors for driving the solar panel, a microcontroller for controlling the stepper motors and a pyranometer for measuring the intensity of solar radiations. The solar panel moves in the direction where the pyranometer receives maximum radiations [26].

Xiwei et al designed a microcontroller and stepper motor based system which automatically measure and dynamically track the water level in the rivers. For this purpose AT89S52 microcontroller, position detection sensor, display module and a stepper motor is used for measuring and displaying the water level [27].

2.2 OBJECTIVE OF THE THESIS

The aim of this research is to choose a new sensor less technology for controlling the position of stepper motor for an elevator system. There are different techniques for controlling the stepper motor such as fuzzy logy, neural network and microcontroller. Microcontroller finds a best method for controlling the position of stepper motor. Different types of drives are used to operate the stepper motor because the microcontroller output is not sufficient to operate the stepper motor. Out of all drives ULN2003 is a less costly drive IC for stepper motor. The real time information of the elevator moving through the floors is displayed with the LCD. After each stop of stepper motor the opening and closing of the door of elevator is represent by two LED's. The control program is written in C language and Kiel compiler software is used to change this high level program into HEX code. By using DScope we can download this HEX code to the 89S51 microcontroller for position control purposes.

2.3 ORGANIZATION OF THESIS

The work carried out has been summarized in four chapters.

Chapter 1 highlights the brief introduction, operation; working types of stepper motor drives are also explained. The 8051 Microcontroller pin configuration and complete description is explained in this chapter

Chapter 2 highlights the brief introduction summary of work carried out by various researchers. The objective of the work is also identified and the outline of the thesis is also given in this chapter.

Chapter 3 highlights the components description like full bridge rectifier. LM7805 voltage regulator IC, LCD pin configuration and features, ULN2003 stepper motor driver, hardware circuit and hardware layout is completely explained in this chapter.

Chapter 4 highlights the software used for changing high level programs into executable programs. In this chapter explanation of Kiel compiler, How to use this software and procedure of changing a C code into HEX code is completely explained.

2.4 CONCLUSION

Different controlling techniques are used for controlling position and speed of stepper motor. Different microcontroller techniques used for controlling the speed and position of a stepper motors. Out of all the techniques Microcontroller is one of the easiest technique for controlling the speed and position of stepper motor. The speed of the stepper motor is control by changing the delay time of pulses and control over the position is done by control the number of pulses. Also unipolar stepper motor is better than a bipolar motor because of the reduced step angle and there is no need of reversal the current in the unipolar drive for changing the rotation of stepper motor. As in microcontroller we can change the control program according to the need with serial port programming by a personal computer.

CHAPTER 3 RESULTS AND DISCUSSION

3.1 INTRODUCTION

The elevator is connected with the unipolar six wire stepper motor. 89S51 microcontroller is used to send controlled pulses to ULN2003 stepper motor drive to control the position of stepper motor. The microcontroller, Stepper motor and ULN2003 driver IC required different Dc voltages for operation. This DC supply is provided either by battery on a single phase 220/12/5v step down transformer and this AC supply is converted into DC by a full bridge rectifier. LM7805 IC is used for regulation purposes and also provides dc 5V at its output.

3.2 LIST OF COMPONENTS AND DESCRIPTION

NUMBER	COMPONENT DESCRIPTION
1	FULL WAVE BRIDGE RECTIFIER WITH 4 IN 4001 DIODES, 220/12V TRANSFORMER (SINGLE PHASE) AND 1000UF CAPACITOR.
2	LM7805 VOLTAGE REGULATOR.
3	89S51 MICROCONTROLLER.
4	LCD (16*2).
5	ULN2003 STEPPER MOTOR DRIVER.
6	TWO ORDINARY LED (GREEN AND RED).

TABLE A LIST OF COMPONENTS

3.2.1 FULL WAVE BRIDGE RECTIFIER

3.2.1.1 INTRODUCTION

Full wave bridge rectifier is used to convert the ac supply into dc supply. For this purpose four IN4001 general purpose diodes are used in bridge configuration. An electrolytic capacitor of 1000 microfarad is connected parallel with these diodes to filter-out the harmonics present in the DC output supply. For step down purposes at input side a single phase transformer 220/13.5 volts is connected as shown in the figure 3.1 below.

3.2.1.2 BRIDGE RECTIFIER CIRCUIT DIAGRAM

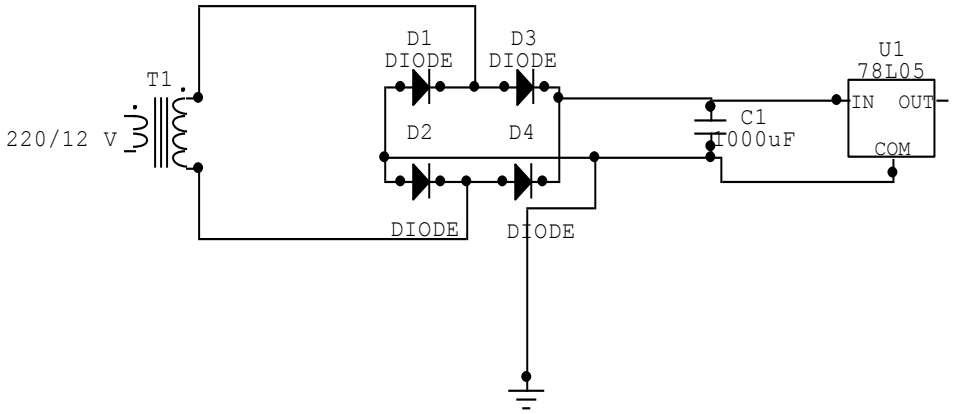

FIGURE 3.1 FULL BRIDGE RECTIFIER

3.2.2 LM7805 VOLTAGE REGULATOR

3.2.2.1 INTODUCTION

The LM7805 voltage regulator is used for regulation purposes. The LM78XX series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents [31].

3.2.2.2 FEATURES

There are many features of LM7805 voltage regulator. LM7805 voltage regulator is capable of generating output current upto 1A and output voltages from 5V-24V. It provides thermal overload protection and short circuit protection to the connected device. LM7805 have lower power consumption and provides better voltage regulation to the connected system. A heat sinker for heat dissipation is also connected at the top of the IC. Many times fixed output voltage makes it useful for many applications. In the present work LM7805 is used for regulation purposes and for changing 12V DC to 5V DC [31].

3.2.2.3 BLOCK DIAGRAM

In the block 3.2 given below the voltage regulator provides fixed or adjustable voltages and protections like short circuit protection, over temperature and also filters the harmonics due to the presence of series pass element at the output terminal.

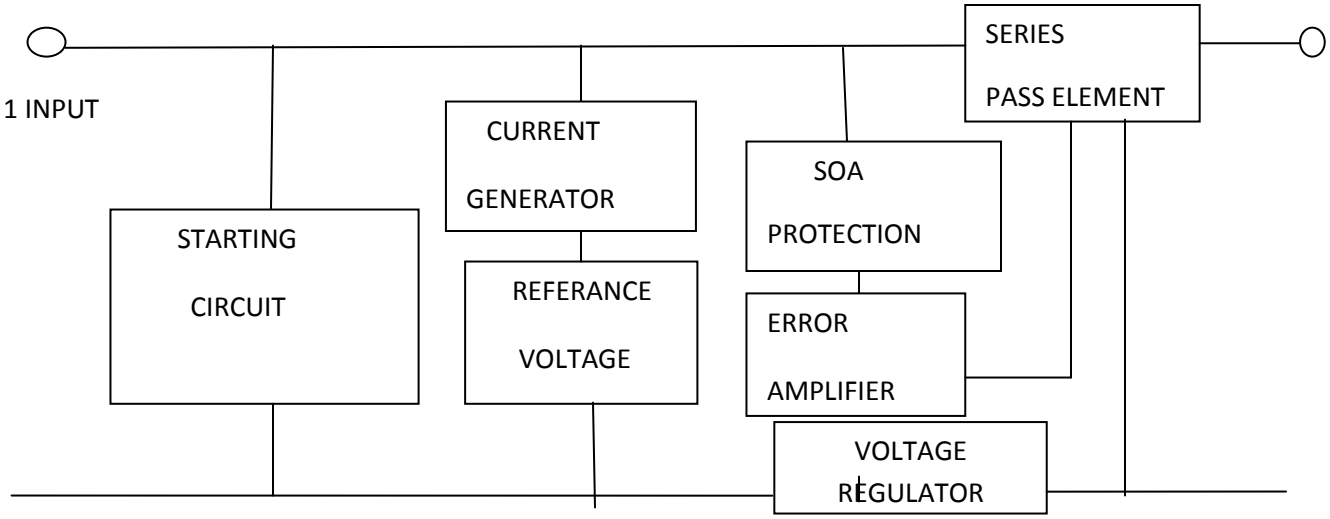

FIGURE 3.2 BLOCK DIAGRAM OF LM7805

3.2.2.4 PIN DESCRIPTION

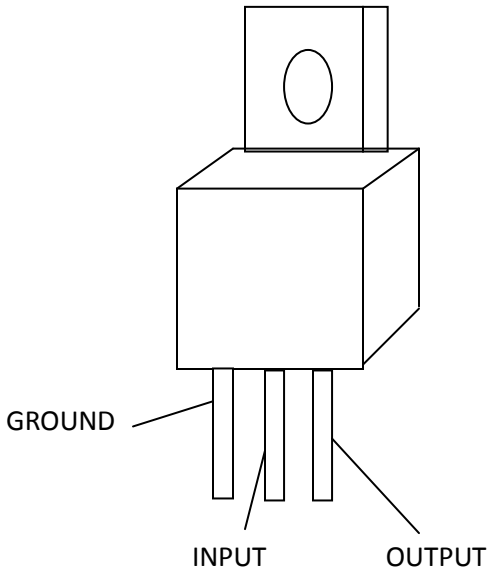

FIGURE 3.3 LM7805

3.2.3 89S51 MICROCONTROLLER

3.2.3.1 INTRODUCTION

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of In-System Programmable Flash memory [16]. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80S51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry.

3.2.3.2 FEATURES

The ATMEL 89S51 microcontroller has many features like it is able to operate between 4V to5.5 voltage ranges. AT89S51 has 128x8 bit internal RAM and 32 programmable I/O pins. It has two 16 bit timers and counters. Due to these features this microcontroller is used in many applications. Due to the capability of serial port programming it is used for controlling the speeds of Dc motor connected with a host system and in remote sensing applications [16].

3.2.3.3 PIN DESCRIPTION

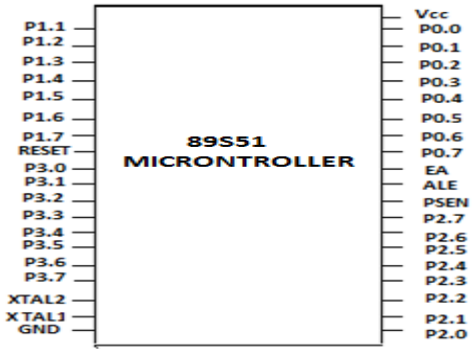

FIGURE 3.4 89S51 MICROCONTROLLER

VCC: Supply voltage.

GND: Ground.

PORT 0: Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification [16].

P0RT 1: Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current because of the internal pull-ups. Port 1 also receives the low-order address bytes during Flash programming and verification.

PORT PIN	ALTERNATE FUNCTION
P1.5	MOSI(USED FOR IN SYSTEM PROGRAMMING)
P1.6	MISO(USED FOR IN SYSTEM PROGRAMMING)
P1.7	SCK(USED FOR IN SYSTEM PROGRAMMING)

TABLE B PORT I

PORT 2: Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses (MOVX @ DPTR).

PORT 3: Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being

pulled low will source current because of the pull-ups. Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

PORT PIN	ALTERNATE FUNCTION
P3.0	RXD(SERIAL INPUT PORT)
P3.1	TXD(SERIAL OUTPUT PORT)
P3.2	INT0
P3.3	INT1
P3.4	T0
P3.5	T1
P3.6	WR
P3.7	RD

TABLE C PORT 3.

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DIS-RTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/PROG: Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN: Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP: External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming [17].

XTAL1: Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2: Output from the inverting oscillator amplifier.

3.2.4 ULN2003 DRIVER FOR STEPPER MOTOR

3.2.4.1 INTRODUCTION

The ULN2003 is a unipolar stepper motor driver. These drivers are inexpensive and with high efficiency. This driver has 16 pins and more efficient than the semiconductor switches based stepper motor drives. These drivers are also lowers the effect of inductance. The microcontroller output pulse is not sufficient to drive the stepper motor. So, we used driver to operate the stepper motor. These drivers are more efficient than the drives have semiconductor switches like Transistor, MOSFET and IGBT because their control is simplest than the semiconductor switches. The probability of failure is less and the problem like missing of pulse for triggering is absent in the uln2003 drives. The ULN2001A, ULN2002A, ULN2003 and ULN2004Aare high voltage, high current darling-ton arrays each containing seven open collector darling-ton pairs with common emitters [31]. Each channel rated at 500mAand can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout. These versatile devices are useful for driving a wide range of loads including solenoids, relays DC Motor’s and LED displays filament lamps.

3.2.4.2 FEATURES

ULN2003 stepper motor driver has many features like it has seven darling tons per package and output current of 500mA per driver. Its diodes can be paralleled for higher current applications and suppression diodes are present for inductive loads output. ULN has simplified layout and low power consumption. ULN is easy to control by employing a microcontroller and able to

drive wide range of loads including stepper and DC motors. Outputs are pinned opposite to inputs for simplified layout [31].

3.2.4.3 PIN DIAGRAM

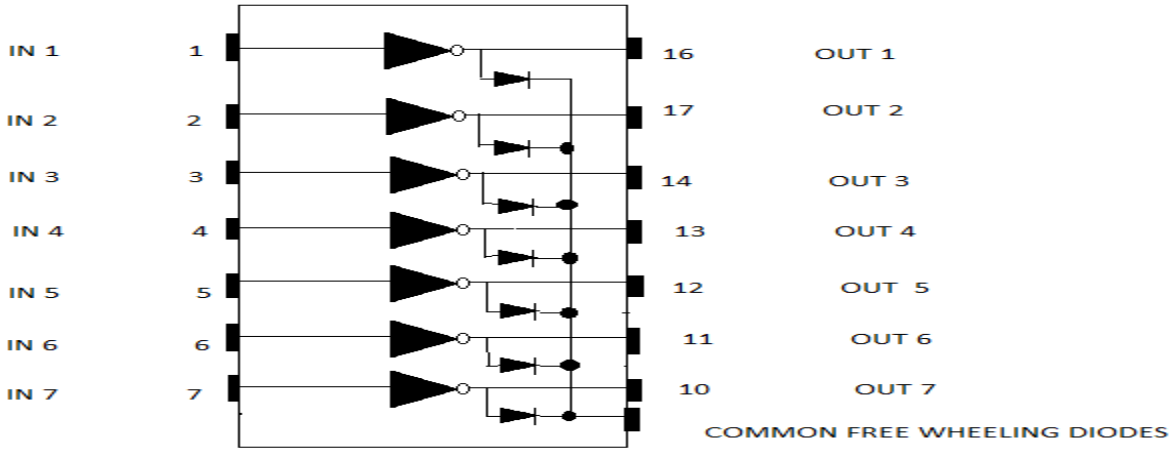

FIGURE 3.5 PIN DIAGRAM OF ULN 2003

3.2.4.4 ABSOLUTE MAXIMUM RATING

SYMBOL	PARAMETER	VALUE	UNIT
Vo	OUTPUT VOLTAGE	50	V
Vin	INPUT VOLTAGE	30	V
I c	CONTINIOUS BASE CURRENT	500	mA
I b	CONTINIOUS COLLECTOR CURRENT	25	mA
T am	OPERATING AMBIENT TEMPERATURE RANGE	20-85	C
T j	JUNCTION TEMPERATURE	150	C

TABLE D ABSOLUTE MAXIMUM RATING OF ULN 2003

3.2.5 LIQUID CRYSTAL DISPLAY (LCD)

3.2.5.1 INTRODUCTION

The LCD's are used for displaying purposes. In recent years the LCD is finding widespread use replacing LED's due to many reasons. The declining prices of LCD as compared to LED increase the demand of LCD in many applications. The LCD has ability to display numbers, characters and graphics. The LCD has ease to programming for characters, numbers and graphics. LCD has a incorporated refreshing controller. In the present thesis LCD is used for displaying real time information of elevator moving through the floors [31].

3.2.5.2 FEATURES

LCD has many features make it useful for many applications like 5x8 dots with curser and built in controller to control the LCD. LCD required only 5V DC supply for working which is easy driven from a battery or from a rectifier as in the present thesis. It has 16x2 ability for character, number and graphics display [31]. The LCD with more features are also in the market but 16x2 LCD find more useful in many applications due to the low prices and compact structure .

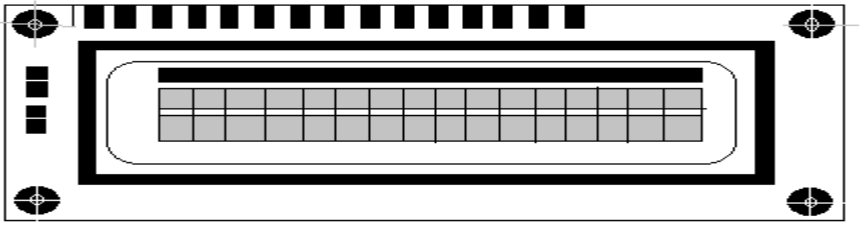

FIGURE 3.6 LCD (LIQUID CRYSTAL DISPLAY)

3.2.5.3 MECHANICAL DATA FOR LCD

MEHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
MODULE DIMENSION	80X30	Mm
VIEWING AREA	66X16	Mm
DOT SIZE	.56X.66	Mm
CHARACTER SIZE	2.96X5.56	Mm

TABLE E MECHANICAL DATA FOR LCD

3.2.5.4 PIN DESCRIPTION

PIN	SYMBOL	I/O	DESCRIPTION
1	Vss	_	G**round**
2	Vcc	_	+ 5V supply
3	Vee	_	Power supply to control contrast
4	RS	I	RS=0 to select command register RS=1 to select data register
5	R/W	I	R/W=0 for write R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8 bit data bus
8	DB1	I/O	The 8 bit data bus
9	DB2	I/O	The 8 bit data bus
10	DB3	I/O	The 8 bit data bus
11	DB4	I/O	The 8 bit data bus
12	DB5	I/O	The 8 bit data bus
13	DB6	I/O	The 8 bit data bus
14	DB7	I/O	The 8 bit data bus

TABLE F PIN DESCRIPTION OF LCD

3.2.6 SIX WIRE UNIPOLAR STEPPER MOTOR

3.2.6.1 INTRODUCTION

The unipolar stepper motor has advantages like no need to reverse the current in the winding to change the rotation of the motor. These motor comes in the market with a reduced step angle of 7.5 degree. The control of these motors is easy when we are applying microcontroller to generate control pulses to operate these motors through a unipolar stepper motor driver [4]. The rotational angle and direction of each step is determined by the construction of the motor as well as the step

pattern input. The motor used in the present thesis is PN #27964 with four phase unipolar stepper motor that is easily controlled with the application of the microcontroller.

3.2.6.2 TECHNICAL SPECIFICATIONS

TERMS	SPECIFICATION
RATED VOLTAGE	12 VDC
RATED CURRENT/PHASE	259Ma
NUMBER OF PHASE	4
DC COIL RESISTANCE	50 OHM/PHASE
STEP ANGLE	7.5 DEGREE/PHASE
EXCITATION METHOD	2-2 PHASE (UNIPOLAR)

TABLE G TECHNICAL SPECIFICATIONS OF UNIPOLAR STEPPER MOTOR

3.2.6.3 MECHANICAL SPECIFICATIONS

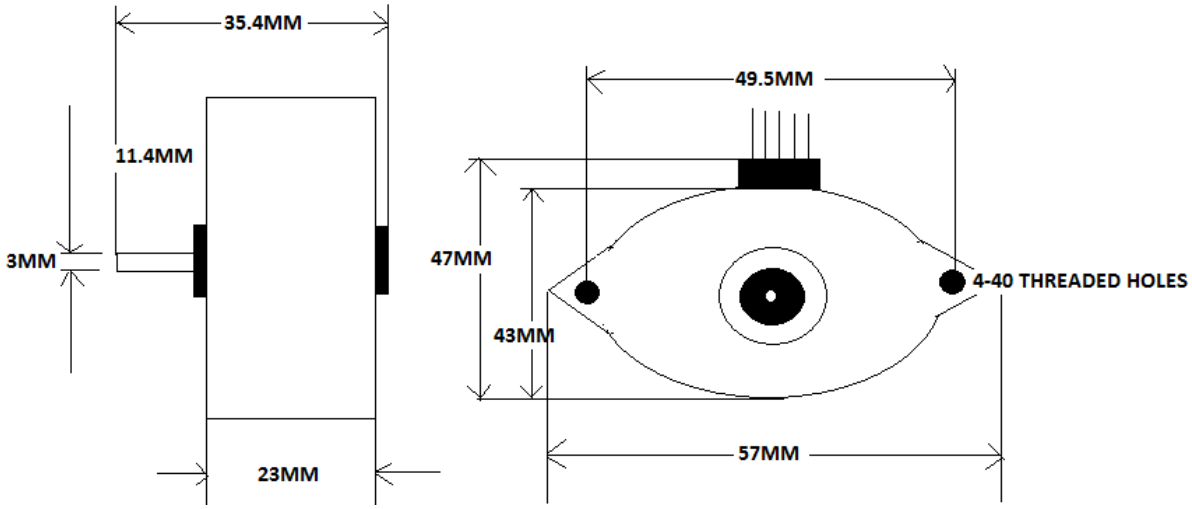

FIGURE 3.7 MECHANICAL SPECIFICATIONS

3.2.6.4 MOTOR CONNECTIONS WITH ULN2003 DRIVER

Connect the six wires of the unipolar stepper motor in order to control its position. The connection diagram is given in the figure and the wire colour represents which phase it is cleared from the table.

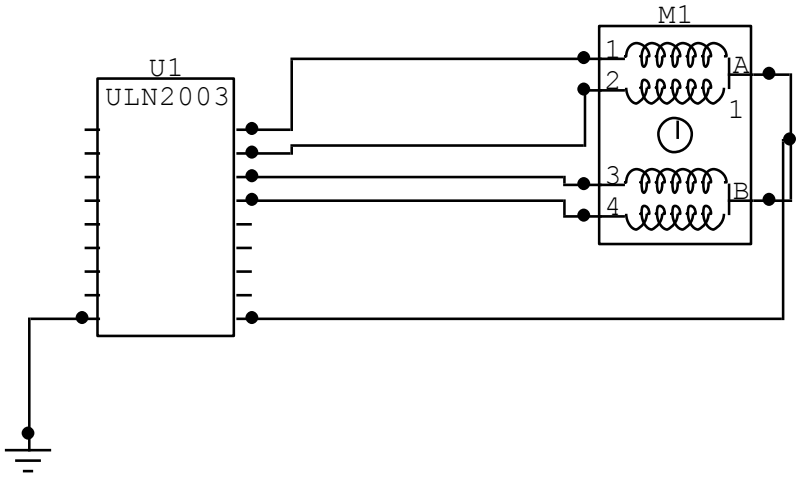

FIGURE 3.8 STEPPER MOTOR CONNECTIONS WITH ULN2003

3.2.6.5 WIRES SPECIFICATIONS

DEGREES PER STEP	7.5
STEPS PER REVOLUTION	48
PHASE1	BLACK
PHASE2	ORANGE
PHASE3	BROWN
PHASE4	YELLOW
COMMON	RED

TABLE H WIRES SPECIFIATIONS

3.3 OVERVIEW

The design project is divided into four floors of equal distance ground floor, Ist floor, second floor and third floor. The stepper motor is connected to the elevator lifting pulley with belt as shown in the figure. The stepper motor pulley will took 40 full rotations to lift the elevator to cover the distance from one floor to another. The LCD displays the following real time information about the elevator for a example UPWARDS while moving upwards , DOWNWARDS while moving downwards, STILL when elevator is at stop position and also

giving the information about the floor in which the elevator moving and stop. The Micro-switch's are used to select the desired location for elevator. After reaching at desired floor after some delay green LED glow which represents the opening of elevator door and after five sec delay green LED off and red LED glow which represents the closing of elevator door. After some delay this red LED off. The floors are made from iron sheets and the elevator is made up of steel sheet of low weight

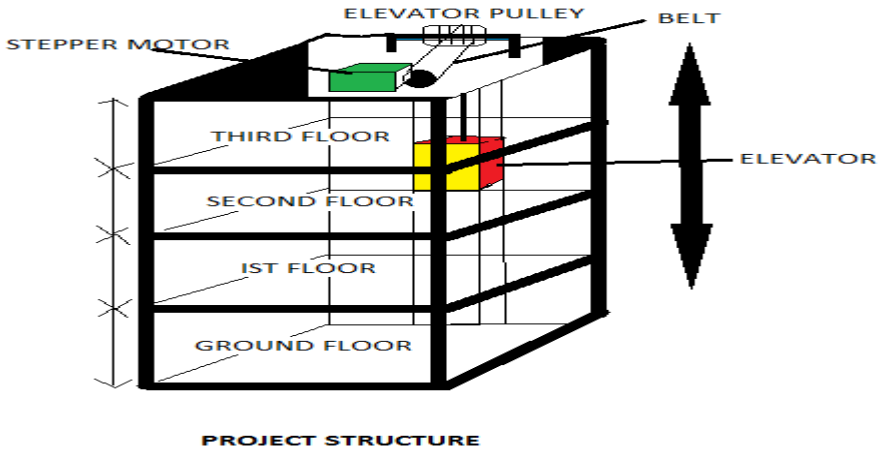

FIGURE 3.9 PROJECT STRUCTURE

.

3.4 HARDWARE CIRCUIT DIAGRAM

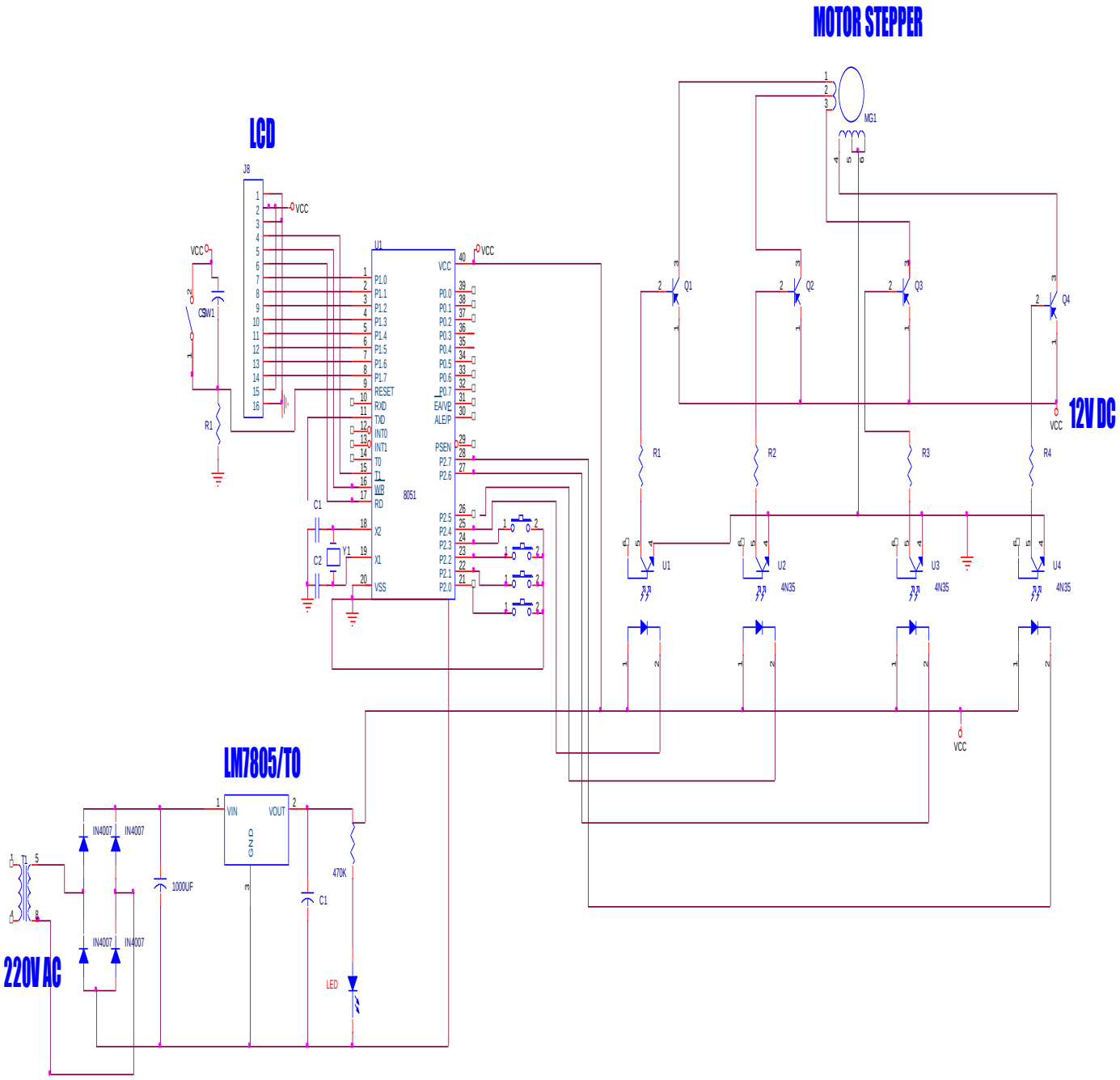

FIGURE 3.10 CIRCUIT DIAGRAM

3.5 HARDWARE LAYOUT

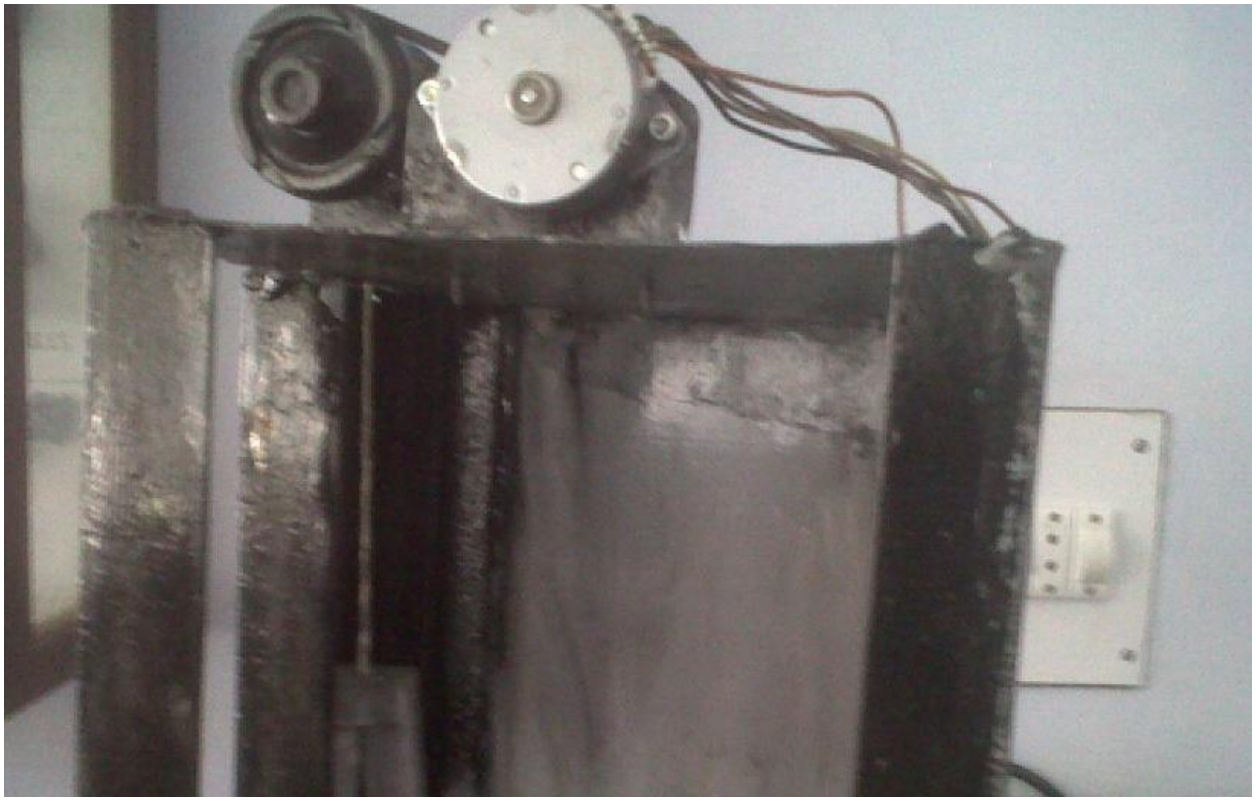

FIGURE 3.11 LIFTING PULLEY CONNECTED TO THE ELEVATOR AND STEPPER MOTOR

FIGURE 3.12 STEPPER MOTOR CONNECTED WITH ELEVATOR LIFTING PULLEY

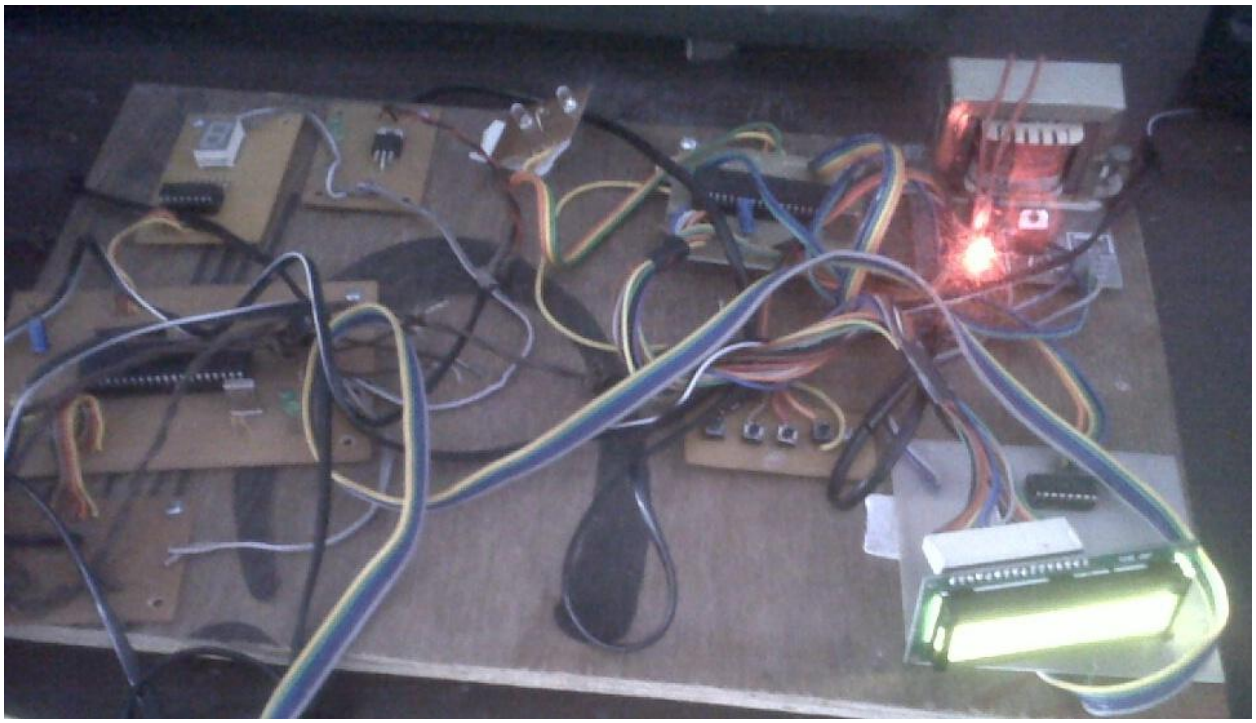

FIGURE 3.13 CONTROL PANEL FOT THE ELEVATOR

3.5 CONCLUSION

All the components of the hardware except transformer required DC supply for working such as LCD, ULN2003, 89S51 Microcontroller and stepper motor. For fulfill this need diode rectifier is used in bridge connected configuration for changing AC to DC for all the components. The Microcontroller pins XTAL1 and XTAL2 are connected to a frequency oscillator for frequency stability purposes. The non electrolytic capacitors are connected with this crystal oscillator for changing the triangular pulse to rectangular pulse. The ULN2003 IC connected with six wire unipolar stepper motor required control pulses on its input pins for operation which he receives from the Microcontroller. We use drivers because the microcontroller output is not sufficient to drive the stepper motor.

CHAPTER 4 SOFTWARE IMPLEMENTATION

4.1 KIEL COMPILER

4.1.1 INTRODUCTION

Kiel compiler is a Microsoft windows based software development tool used to change high level language's (Assembly and C) into executable file or HEX code used to burn the microcontroller. It provides a easy environment for the user for written C code and assembly used for microcontrollers. The software size is code limited it is 2KB for 8051 microcontroller. Micro Vision is a Windows based front end for the C Compiler and Assembler. It was developed in the USA as was the printed manual set. Compiler, Assembler and Linker options are set with simple mouse clicks. Micro Vision runs on Windows 3.1, 95 and NT. The Compiler, Assembler and Linker are DOS executables. This provides maximum flexibility. This Integrated Development Environment (IDE) has been expressly designed with the user in mind. A full function editor is included. All IDE functions are intuitive via pull down menus with prompted selections. An extensive Help utility is included. External executables can be run from within Micro-Vision [17].

4.1.2 START UP WITH MICRO VISION

Double-click on the Micro-Vision icon to start the user interface. As shown in the figure 4.1.

FIGURE 4.1 ICON OF MICRO VISION

After you invoke Micro Vision, The window shown in Figure 4.2 appears. From this window, you can create projects, edit files, configure the tool, assemble, link, and invoke the debugger. This window allows the user to start a new project. The project is written in both C and Assembly language.

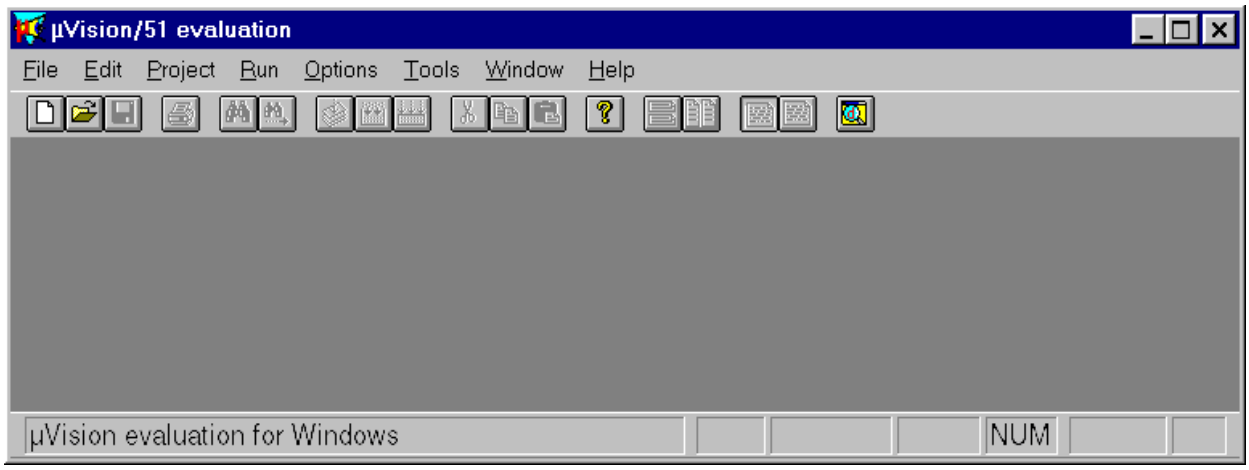

FIGURE 4.2 MICRO VISION WINDOW

4.1.3 CREATING A NEW PROJECT

1. Open the project menu and choose New Project. The window shown in the figure 4.3 appears after starting a new project.
2. Enter the name of project. Enter the name usb.prj and press OK usb.prj will be entered under File name. Figure will open after we entered the name [17].

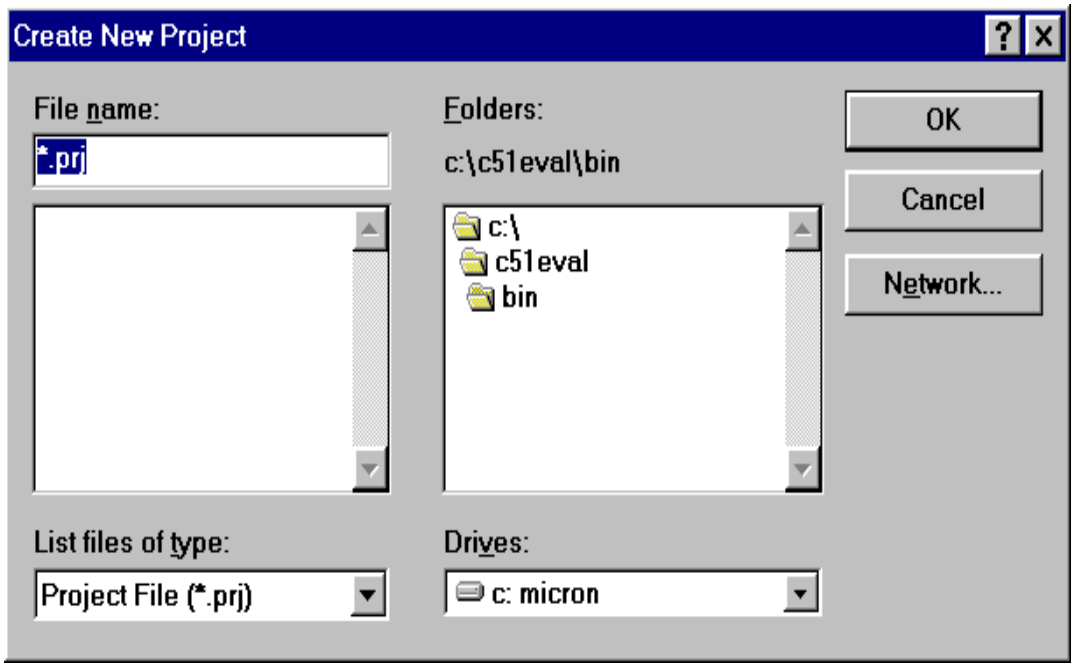

FIGURE 4.3 WINDOW FOR CREATING A NEW PROJECT

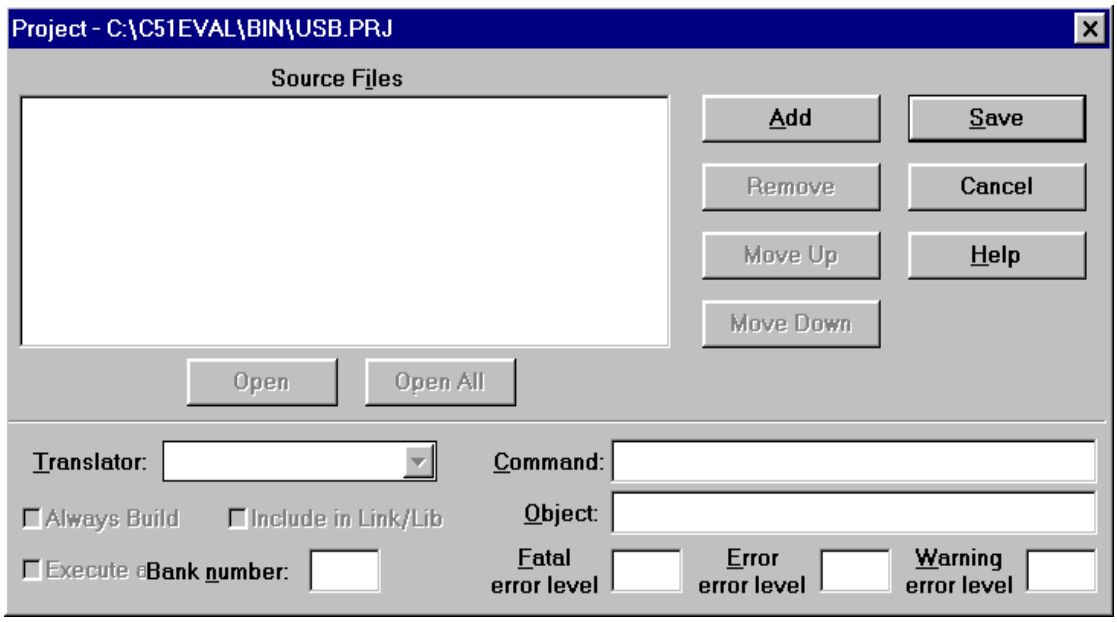

FIGURE 4.4 WINDOW FOR ENTER THE NAME OF PROJECT

This window is used to add various files to your project. These include ASCII files, C and assembly source, and macros. The list is quite extensive and is found in the Translator window if any files are present in the project. Note this window is blanked out at this time. This window is accessible at any time by selecting Project/Edit Project and you may easily edit your file list. Note that any assembler files must be last in this list. If they are not, changes made to them may not be reflected in the final object file. When creating a new project in this manner no source files are yet available. Therefore, select save to close this window. The project named usb.prj will be active. Click on Project and confirm usb.prj is visible at the bottom of this pull down menu.

4.1.4 CREATING A NEW SOURCE CODE FILE

An example of C program is given in order to explain how to use Kiel compiler for written codes. For written a C code open the file menu and choose NEW to go to Micro vision integrated editor. For creating a C source file use the editor to type Example.c as shown in the figure. When you have entered the file, do a save as to the c:\c51eval\bin directory as in Figure. The filename should be Example.c. If you have the source code as a file, choose Open and get Example.c in the usual fashion. Your window should be similar to Figure 4.4. Note the color syntax allowing

increased readability of your source code. This sample program uses a number of simple C source lines to demonstrate the Kiel tool set.

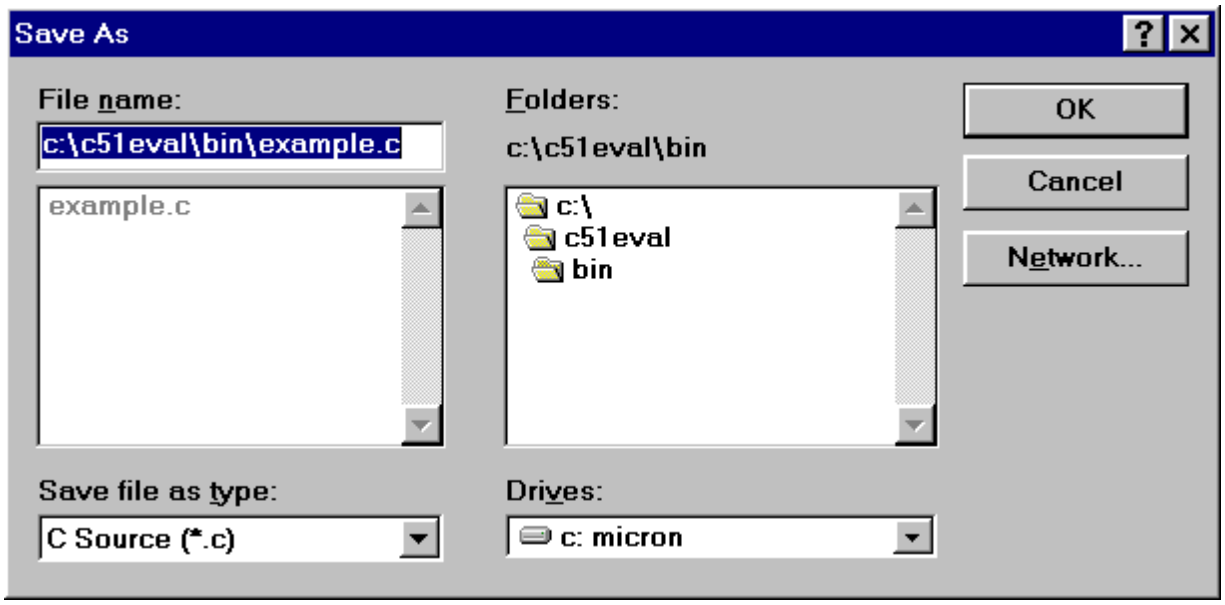

FIGURE 4.5 WINDOW FOR SAVE A C SOURCE FILE

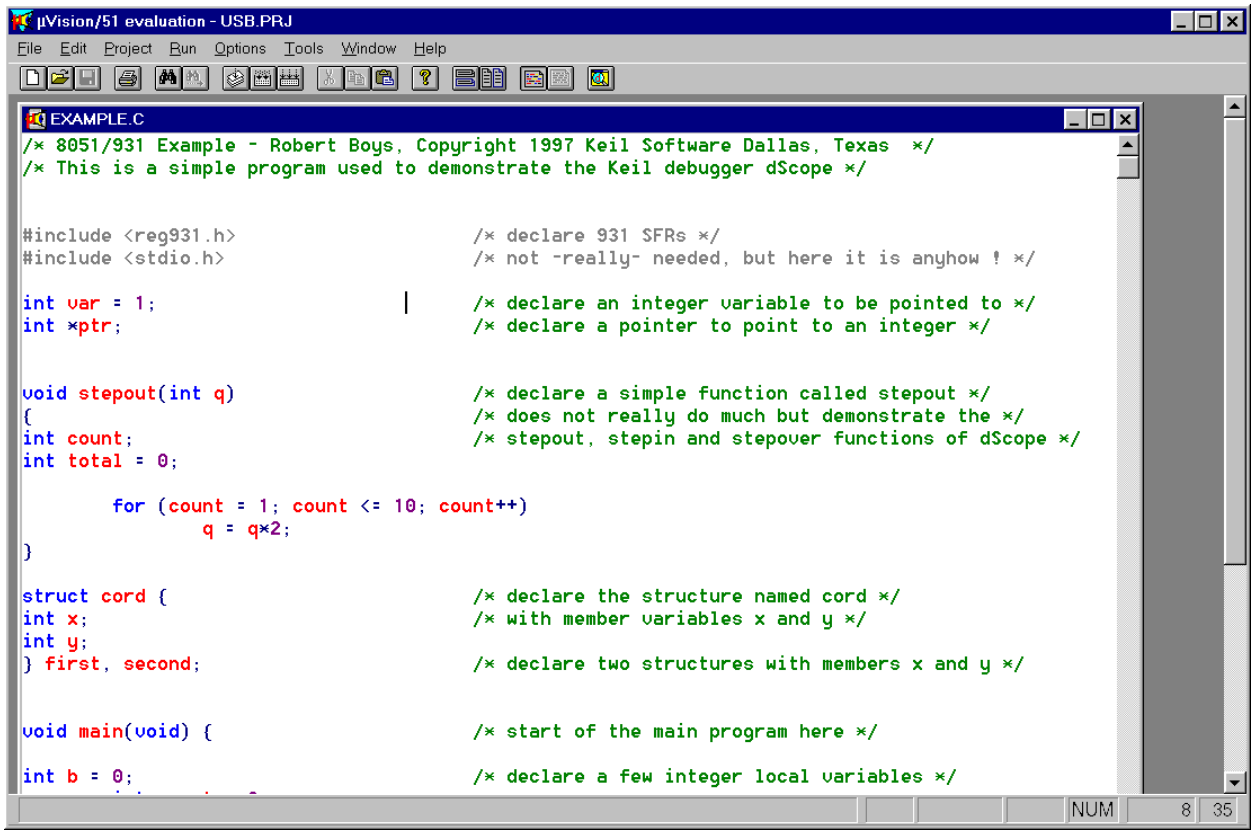

FIGURE 4.6 WINDOW FOR A EXAMPLE OF C PROGRAM

4.1.5 BUILDING A C CODE PROJECT FOR MICROCONTROLLER

1. Open the Project menu and choose Edit project. The window shown in figure 4.7 appears. This is where you add various files to your project [17].

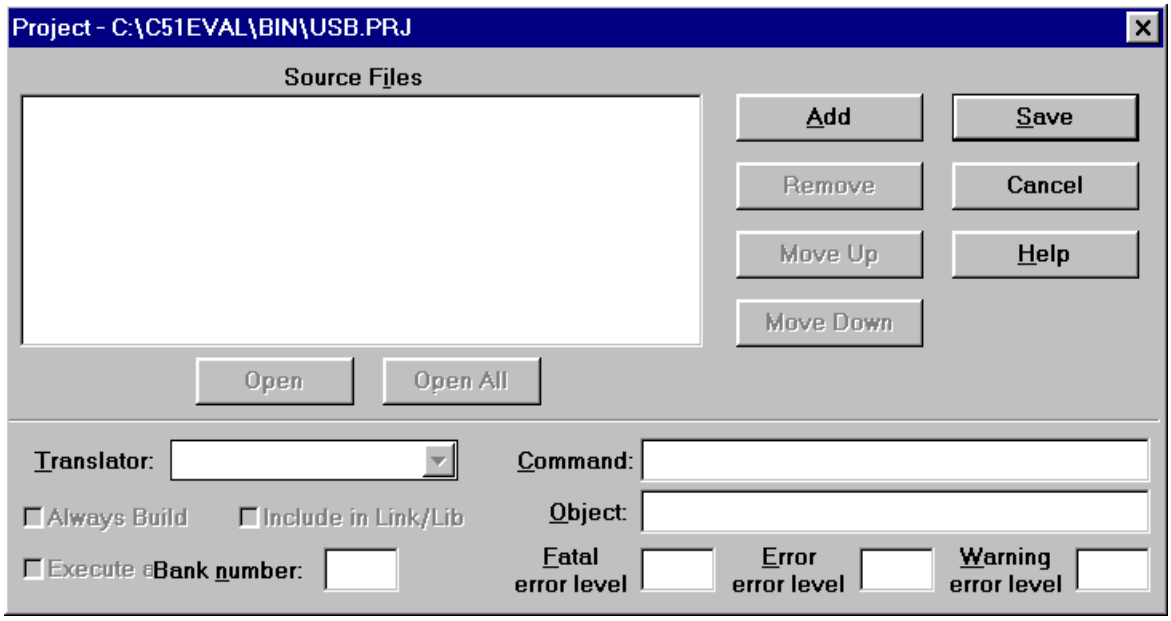

FIGURE 4.7 WINDOW FOR EDITING A NEW PROJECT.

2. Choose ADD. The window shown in figure 4.8 appears. This is where we add the files to our project.

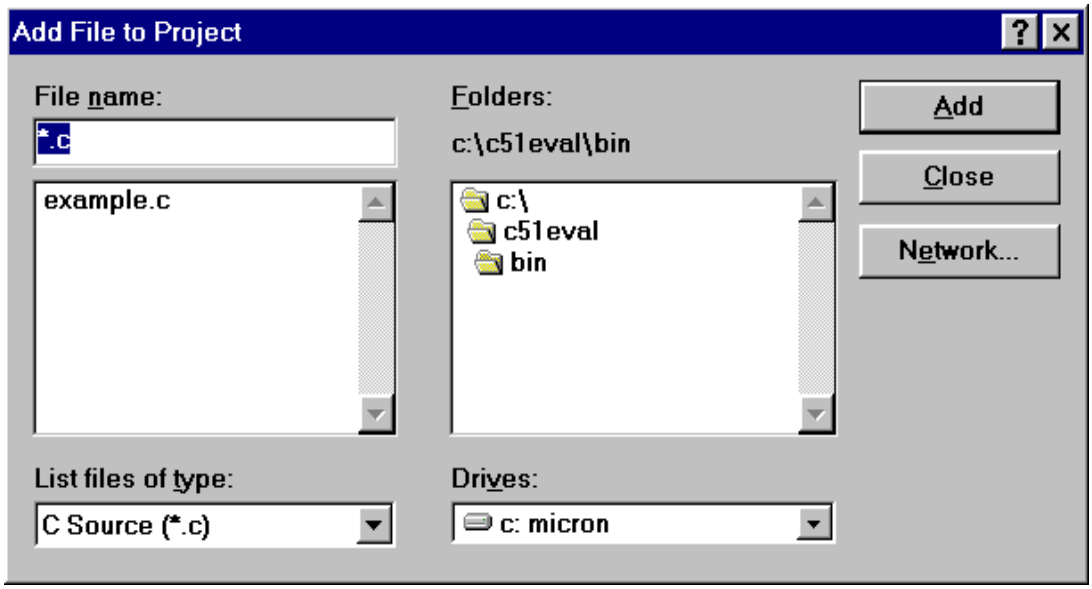

FIGURE 4.8 WINDOW FOR ADD FILES TO PROJECT

3. Select the file Example.c and press enter.
4. Choose ADD then CLOSE. The file Example.c will be listed in the source file window.
5. Make sure include in Link/Lib and checked.
6. Choose SAVE.

4.1.6 CONFIGURING THE MAKE UTILITY, ASSEMBLER AND LINKER

Kiel Software has a Make utility that can compile and link C and/or assembly coded files. It also has other configuration and default setting features. Before using the Make utility, configure the make options. Make the changes as indicated, and leave all other options set to their default values.

1. Configure the make utility.
    - Open the options menu and choose make.
    - Configure the options in the window as shown in figure 4.9. Press OK.

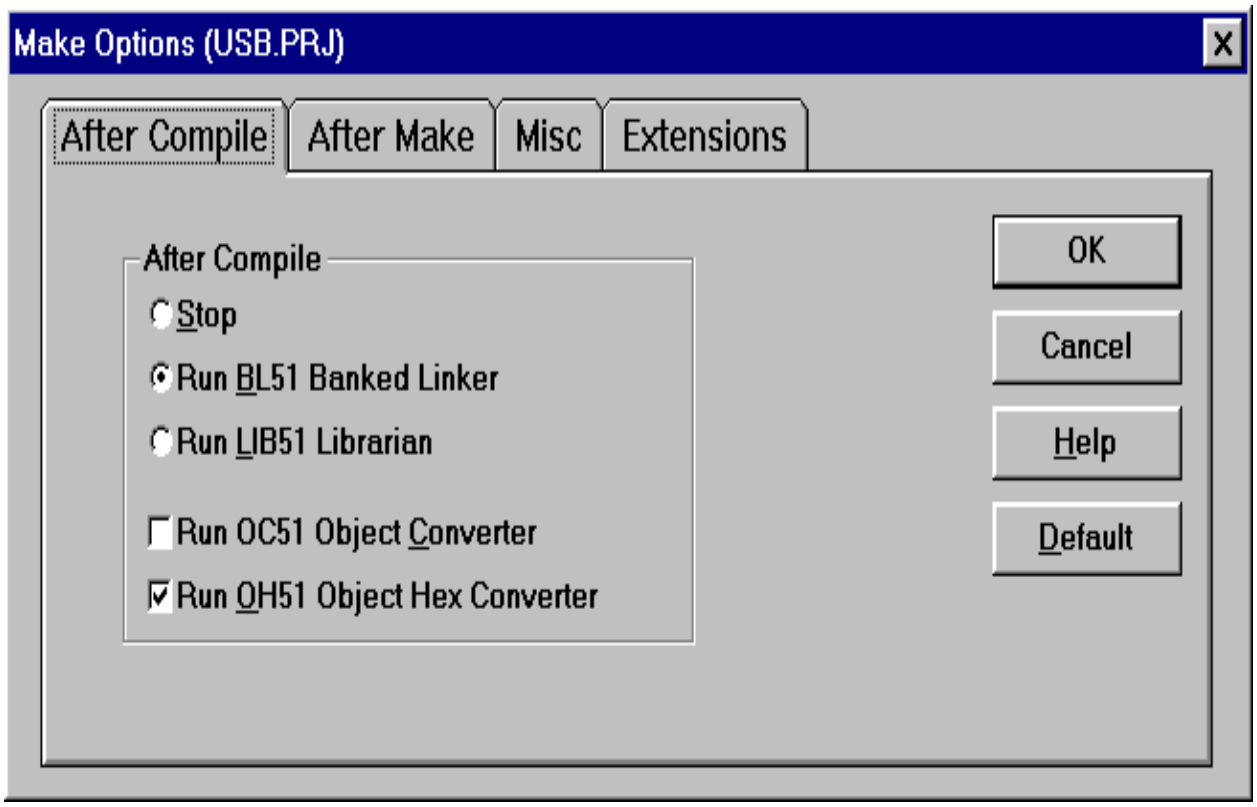

FIGURE 4.9 WINDOW FOR MAKE OPTIONS

2. Configure the compiler.
    - Open the Options menu and choose C51 Compiler. Click on the Object tab.
    - Note the DOS command line options are visible at the bottom of the window. We will see what actions are taken with your selections.
    - Configure the options in the window under Object as shown in Figure and press OK. It is important to turn on the Include debug information. This will allow us to see the source code in the debug window in the simulator and the monitor.
    - The assembler is not used in this project. Assembler options are set in a similar way as the other components of the tool chain.

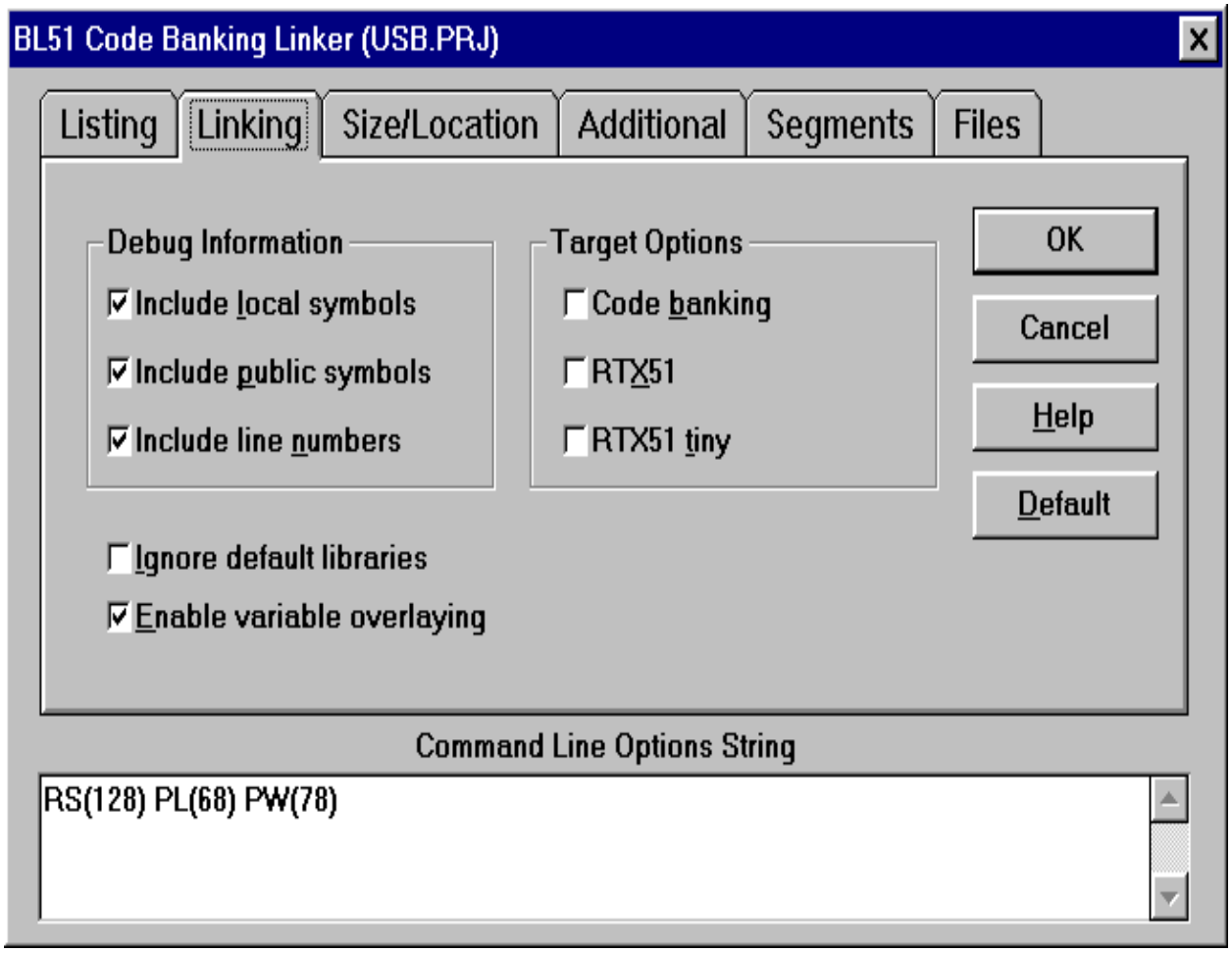

FIGURE 4.10 WINDOW SHOWING BL51 CODE BANKING LINKER

3 Configure the Linker.

- Open the Options menu and choose BL51 Code Banking Linker and click the Linking tab.
- Ensure the default conditions as shown are properly set. Note this is where the Banking, Real-time Operating system and variable overlaying are controlled. Figure showing this.

4.1.7 COMPILING THE CODE AND CREATING AN EXECUTABLE FILE

With the tool configured, we are ready to run the compiler and linker using the Make utility.

1. Click on the "Build All" icon (it has three arrows pointing downwards) or open Options and select Make.

- If the program specified (Example.c) has any errors, they will appear on the screen. Use the editor to correct the error(s) in the source code and save the file. We can double-click on the error of interest and Micro Vision will take you to the offending line in the source code. You can edit this line and rebuild the project by repeating this section: beginning at step 1.
- If there are no errors, the code is assembled and linked with the executable code ready to be downloaded to the board. The Project Status window will state "Make Successful - HEX File Created" if everything is working properly.

Continue to the next section.

The following files in the directory \bin are associated with this project.

- Example.c original source file - needed for debugging purposes.
- Example.bak a backup file produced by Micro Vision.
- Example.lst a listing file of the source example.
- Example.obj a reloadable object file. Needs to be linked.
- USB.prj the Project File. Note that the output code assumes the name of the project file.
- USB.m51 map file.
- USB.hex Intel Hex File Created by the Object to Hex Converter oh51.exe.
- USB absolute object file with debugging information (if so set in the compiler option). This file is the input for the Kiel simulator DScope and emulators. This file is created by the linker [17].

4.2 DSCOPE

4.2.1 INTRODUCTION

DScope software is used burn microcontroller. DScope is a software debugger and simulator. DScope is the debugger connected to a hardware system. Programs are run on the target hardware rather than in the virtual microcontroller in your PC. The Kiel debugger DScope will communicate with the Intel RISM monitor contained in the EPROM on the USB board. The COM port will be dependent on your configuration and the speed is 9600 baud. The upper left hand corner of Figure 4.11 shows that DScope is active in this case. This will change to DScope if the appropriate dll (i.e. rism51.dll) is chosen. If you select 8051.dll, we will get DScope [34].

1. To establish communication with the evaluation board, choose the correct driver using the pull down list on the left portion of the debugger toolbar. In this example, rism51.dll is chosen.
2. Rism51.dll: When the serial link is connected and the rism51.dll is selected for the first time, the led's will blink. The reset button in the DScope Toolbox window will also cause the lights to blink. If we get this far, things are working properly. DScope should be displayed in the upper left hand corner.
3. This will happen only if communication is established. If the wrong COM port or speed is selected, a series of communication I/O timeouts will be indicated in the Command window. Should this happen; follow the instructions under Communications Setup.

4.2.2 COMMUNICATION SETUP

We need to follow these instructions if proper communication was not established. After a series of attempts DScope will error out as indicated in the command window and a "Target System not found "window will open. You can now change the COM port and baud rate settings.

Select the appropriate COM port for your system. The speed must be 9600 for the external UART and the Serial Break must not be enabled for this exercise. Make sure your serial cable is plugged into the UART!. After we set the COM port and baud rate to 9600, click on Try Again**.**

4.2.3 LOADING THE EXECUTABLE FILE IN THE DEBUGGER ENVIROMENT

Executable file means hex code file which is obtained from the C code and assembly high level languages. There are many software's available in the market which changes high level languages to executable files.

1. Open the File menu and choose Load Object file.
2. In the Window provided, select the file: usb. The file is downloaded to the board and the debugger. This file has no extension. Press OK.
3. To see the Command window if not already visible, select View/Command Window. To see the source code, select the Debug window in the same fashion if necessary. Module: example will appear at the top of debug screen [34].

We should see the source in the Debug window as in Figure 4.11 depending on how you have set your windows. We can single step using the step-into icon. You are now ready to use the DScope window (really DScope) to step through code, set breakpoints, and issue the Go command to start program execution. We can examine special function registers, memory locations, and register values, etc. breakpoints are inserted for providing time delay. This window also displays the number of register are used to store the program.

4.2.4 COMPILER OPTIMIZATION TECHNIQUES

In order to generate code which is efficient both in terms of memory size and execution speed, the compiler must employ many optimization techniques. Data flow analysis and procedural analysis. Constant propagation, folding and dead code elimination. Loop invariant code motion and strength reduction. Register allocation and instruction scheduling. By employing these techniques we can increase the speed of control program. Try to eliminate the expression and commands in the control program which are not used practically to the task. Try to written a short control program which is sufficient for controlling [34].

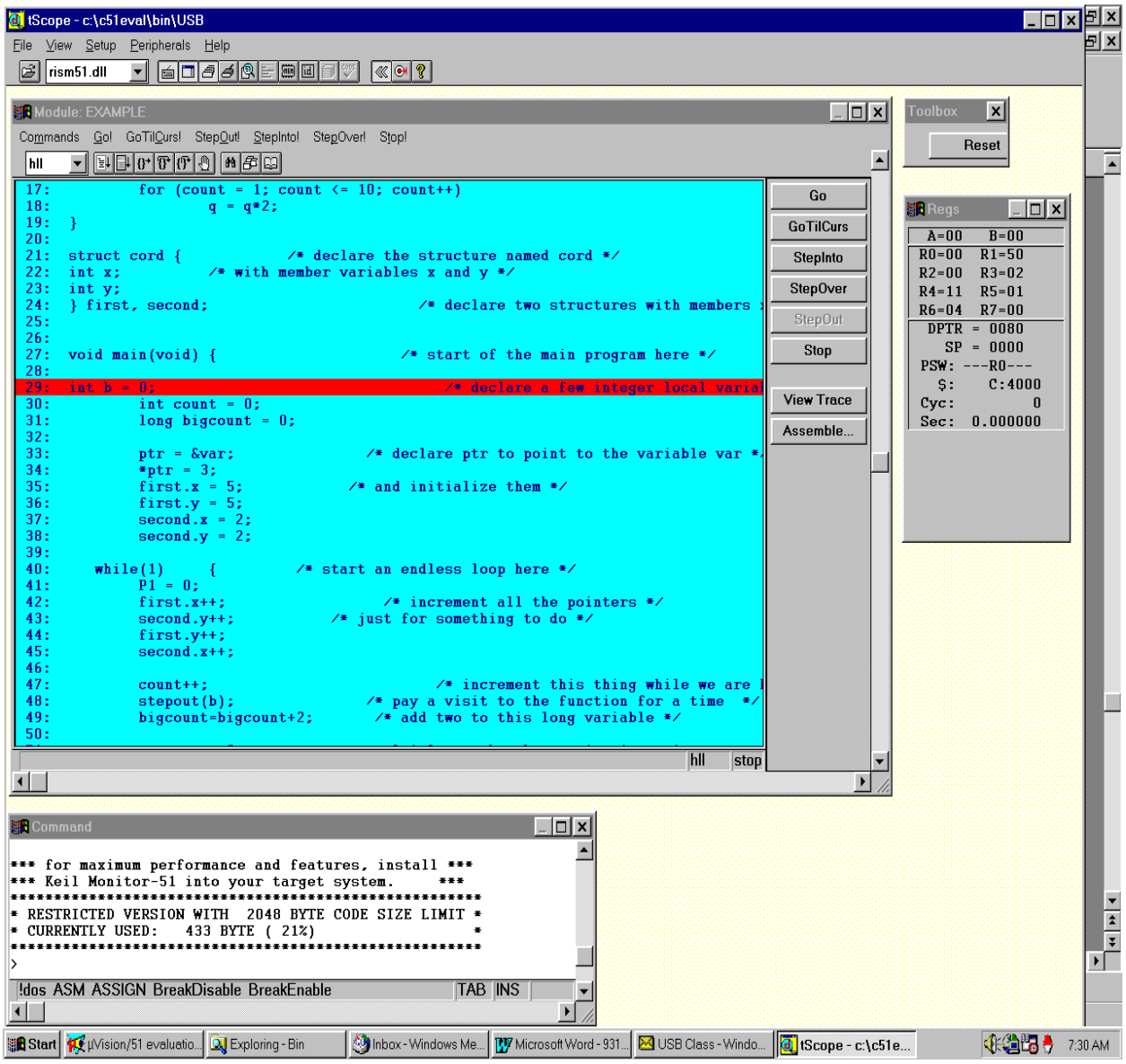

FIGURE 4.11 DEBUG WINDOW

4.2.5 HIGH LEVEL AND ASSEMBLY LANGUAGE LISTING

1. Note the screen displays High Level language source code information. This is because the Debug window is set to view HLL as indicated in the box just beneath “Commands”.
2. Open this box and change it to Mixed and the screen will display both HLL and Assembly listings. Also under commands/view/mixed.
3. Change it back to HLL.

4.2.6 SINGLE STEPPING

1. DScope uses “Step Into” to single step one instruction at a time. “Step Into” is also used to enter a function in the same fashion.
2. “Step Over” means to skip over a function that you are not interested in.
3. “Step Out” is used to exit a function you are currently in. “Step Out” is very useful if we find our self in a function you are not interested in and need to return quickly to your intended function.
4. If DScope gets stuck - do a reset using the RESET box on the Toolbox or click on the button next to the Help “?”, reload the file and/or the DLL. We can also try the 8051.dll to isolate any problems from the evaluation board. We can also use the hardware RESET button on the board.
5. Put a breakpoint on Line 49 and press Go. Click on Step Into until we enter the function Step-out.
6. Repeat the process using Step Over and you will not enter the function although it will be executed. This is useful to skip function calls we are not interested in debugging.
7. Note that the Step Out button is grayed out and not available in DScope. Step Out is available in the simulator DScope and provides a quick escape from a function by executing the next return instruction [34].

4.2.7 BREAKPOINTS

1. Reset using the RESET box on the Toolbox or click on the button with the red pointer at top of screen. The lights on the board should change.
2. Click on line 41 P1 = 0 and a colored bar appears marking this position.
3. We could click on Go TilCurs! to reach this point or we could double-click and a breakpoint is set. Set a breakpoint here with the double click. The [BR0] indicates the first breakpoint.
4. Click on GO and the program will run and stop at the breakpoint. - 6 LEDs will be on.
5. Click on Step Into and the instruction will be executed extinguishing the LEDs.
6. Double click on the breakpoint to remove it.

7. Click on Line 52 P1 = 0xff and press Go TilCurs!.
8. Click on Step Into to execute the instruction and 6 of the LEDs will light up.
9. Click on RESET.

4.2.8 INLINE ASSEMBLERS

1. Open the Commands option and click on Inline Assembler or if Show Dialog bar is active, click on the Assemble button in the Debug window 4.11.
2. The Inline Assembler window opens as in Figure 4.12.
3. Note that you could enter mnemonics in the window titled enter MCS-51 instruction: Do not do this at this time. Just note the ability to make small changes in the program without recompiling the program.
4. Click on Close.

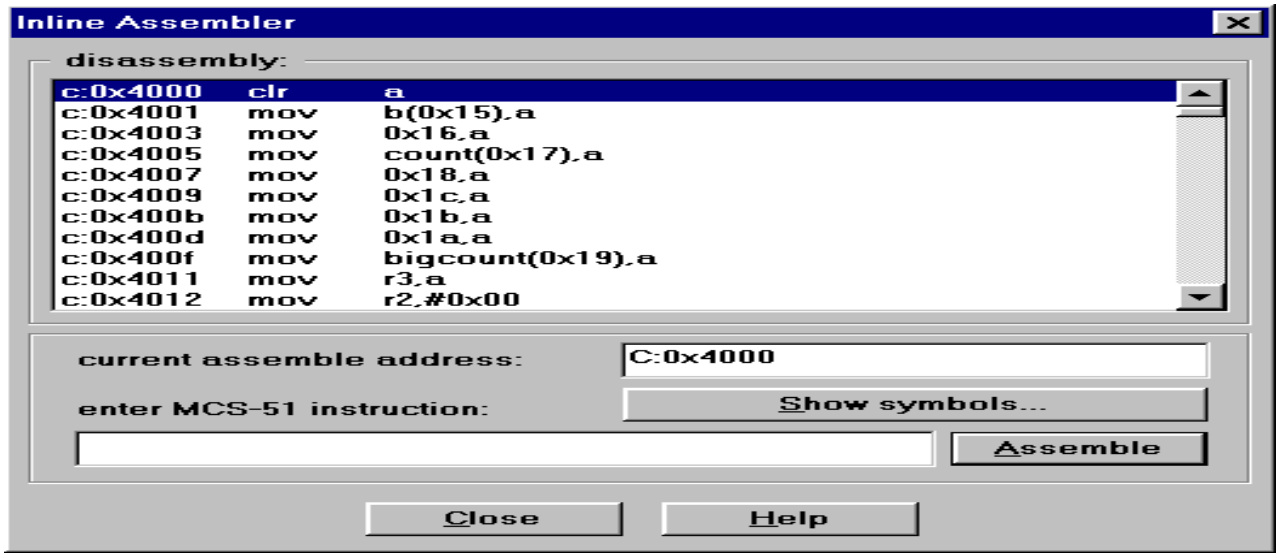

FIGURE 4.12 ASSEMBLER WINDOW

4.2.9 MEMORY WINDOW

1. The Memory Window shows as the default the data space starting at D: 0x0000. This is the area where data variables are kept in this example.
2. Open the View menu and select Memory Window. Position this window off the Debug window.
3. Note that as you step through the program that the contents of the memory changes as the variable values are adjusted.

4. We can change the memory area with the DISPLAY command in the Command window. Enter dc: 0x00 and the code area will be displayed.
5. Change back to the data area with d d:0x00

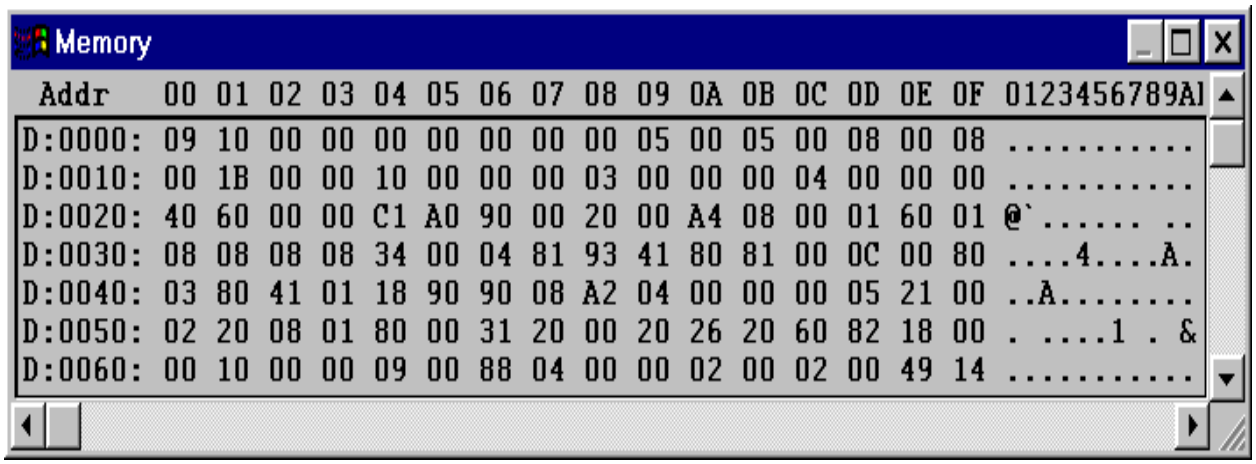

FIGURE 4.13 MEMORY WINDOW

4.2.10 WATCH WINDOW

1. The Watch Window displays memory contents as specified by their name. Structures can also be displayed.
2. Open the Setup menu and select Watch points. In the Expression: window, enter the following symbols one at a time followed by clicking on Define watch. Big count, first, second and var. Note that if you also select 0xnn the values will be displayed in decimal rather than hexadecimal notation. Try this for one or two of the variables.
3. These variables will now be displayed in the Watch point window. Click on Close [34].
4. Click on Step Over and the variable values will changes as appropriate.
5. Note: when are using DScope and if in the Update Memory Window and Update Watch Window (in the Setup menu) are activated - the variable values will change as the program is running. This feature is not possible with DScope running on a hardware board.

6. Open the Peripherals menu and select Configuration. Activate enable serial break and click on Apply then Close.
7. Do a RESET and start over as before. Press GO and then STOP and note that the values change in both the Watch and Memory windows when the processor is stopped.

4.2.11 USB ENUMERATION

USB devices are hot-pluggable. When a device is connected to a PC, the peripheral is detected and certain communication occurs between it and the host. The appropriate drivers will be selected and the peripheral will be installed on the host (the PC). This is the enumeration process and this can be demonstrated with the Intel Evaluation board.

4.2.12 DEMONSTRATING THE ENUMERATION PROCESS

1. Power up the 931 board without the serial cable connected. RISM need not be running.
2. Our USB equipped PC must have OSR 2.1 installed as mentioned above.
3. Connect the USB cable between the Intel board and the PC. The mouse arrow will turn into a hourglass, then a window indicating detection will appear.
4. If no valid USB device was detected If the hourglass returns to the mouse pointer; this means no valid USB device was detected [34].

4.2.13 CONCLUSION

Kiel compiler is a Microsoft window software tool used for changing a high level program into executable code or say in HEX code. Kiel compiler is used to change assembly and C code into HEX code. It is one of the easiest software with DScope software used to loading HEX code into the microcontroller according to the need of user

4.3 CONCLUSION

The applications of permanent magnet stepper motors have grown significantly in recent years in the appliance industry and the automotive industry, among others. These motors are used in a variety of industries, including high and low propulsion technology, computer peripherals, machine tools, robotics, etc. Sensor-less permanent magnet motors are preferable to encoder-based systems because of compactness, low cost, low maintenance, and high reliability. This project shows by making small changes during programming we can reduce the overall cost of elevator because need of sensors is totally removed. Moreover programming is also helps to display the running status of elevator through LCD and also display the floors through which the elevator is moving and stops.

4.4 FUTURE SCOPE

As for the future work, first of all we can implement the sensor less technique using a faster microcontroller. We can also control the speed of stepper motor to save the time when the elevator moving from ground floor to third floor or higher floors or from higher floors to lower floors because in this project the stepper motor speed is constant it takes more time for moving from higher floors to lower floors or vice-versa. For saving the time stepper motor moves faster from one building to another just before reaching to the desired stopping position its speed will be slower automatically this requires some change in the control program. This is achievable by changing the time delay of the control pulses. We can also show the interference of microcontroller with temperature sensors for displaying the temperature or provide safety from accident like fire to save human beings and elevator. If any accident like fire occurs an alarm attached to the elevator starts. We can also connect the RTC module with the microcontroller to display 24 hours display in the elevator and a small DC motor represent fan in the elevator for elevator facility purposes. By connecting the DC motor to the elevator doors by increasing the size of elevator we can implement the opening and closing of the doors same in the present model closing and opening of elevator doors are represented by LEDs (green and red).

APPENDIX A

C CODE PROGRAM

```
#include<lcdrout.h>

#include<stepper.h>

#define l1 P02   / Define port 2.

#define l2 P03   / Define port 3.

#define l3 P04   / Define port 4.

#define s1 P31   / Pin 3.1 connected to ULN.

#define s2 P32   / Pin 3.2 connected to ULN.

#define s3 P33   / Pin 3.3 connected to ULN.

#define s4 P34   / Pin 3.4 connected to ULN.

#define red P26   / pin 2.6 connected to red LED.

#define green P27   / Pin 2.7 connected to green LED.

void main()

{

unsigned char pxdata,ldata,floor,i;   / Unsigned character pxdata, ldata, floor and I.

unsigned int steps;   / Unsigned integer steps.

lcd_initialize();  / Initialize the LCD.

ACC=0x81;  / Displaying the characters in upper row of LCD.

lcd_cmd();  / Clear the contents of LCD.

lcd_display("AUTOMATED LIFT ",14);  / LCD displaying automated lift.
```

```
ms_delay(50);  / Inserting a time delay of 50 ms.

ACC=0xc0;

lcd_cmd();   / Clear the contents of LCD.

lcd_display("CONTROL   SYSTEM",16);   / LCD displaying control system.

secdelay(3);    / inserting a 3 seconds delay.

ACC=0x01;

lcd_cmd();

ACC=0x80;

lcd_cmd();

pxdata=ldata=0;  / Set pxdata and idata equal to zero.

s1=s2=s3=s4=1; / Set the pins 3.1 to 3.4 equal to one.

while(1) / Inserting a while loop.

{

ACC=0x80;

lcd_cmd();

lcd_display("    STILL         ",16);  / LCD displaying still  when pins 3.1 to 3.4 at high level signal.

ACC=0xc0;  / Displaying the characters in the second row  of LCD.

lcd_cmd();

lcd_display("FL-NO       ",12); / LCD displaying floor number.

if(s1==0)   /  If pin3.1 is at high level signal (1).
```

```
{
l1=l2=l3=1;
l1=0;
pxdata=0;   / If 11=0 then pxdata equal to zero.
}
else if(s2==0) / If pin3.2 is at high level signal(1)
{
l1=l2=l3=1; /
l2=0;
pxdata=1; / If pin 3.2 at high level then 12=0 and pxdata=1.
}
else if(s3==0)  / If pin 3.3 is at high level.
{
l1=l2=l3=1;
l3=0;
pxdata=2; / If pin 3.3 is at high level the 13=0 and pxdata equals to 2.
}
else if(s4==0) / If pin 3.4 is at high level.
{
l1=l2=l3=1;
l3=0;
```

```
pxdata=3; / If pin 3.4 is at high level pxdata=3.

}

if(pxdata==0 || pxdata==1 || pxdata==2 || pxdata==3)

{


if(pxdata > ldata)

{

floor= pxdata-ldata;

if(floor>0)

{

ACC=0x80;

lcd_cmd();

lcd_display("Moving Upwards    ",16);

for(i=ldata;i<pxdata;i++)

{

lcd_cmd1(0xc9);

lcd_data(48+i);

movbwd(450);

}

lcd_cmd1(0xc9);

lcd_data(48+i);
```

```
green=0;
red=1;
secdelay(5);
green=1;
red=0;
secdelay(5);
}
}
else  if(pxdata < ldata)
{
floor= ldata-pxdata;
ACC=0x80;
lcd_cmd();
lcd_display("Moving Downwards    ",16);
if(floor>0)
for(i=ldata;i>pxdata;i--)
{
lcd_cmd1(0xc9);
lcd_data(48+i);
movfwd(340);
```

```
}
lcd_cmd1(0xc9);
lcd_data(48+i);
green=0;
red=1;
secdelay(5);
green=1;
red=0;
secdelay(5);
}
ldata = pxdata;
lcd_cmd1(0xc9);
lcd_data(48+i);
red=1;
green=1;
}
}
}
```

END OF THE PROGRAM

REFERENCES

[1] MAZIDI MUHAMMAD ALI "The 8051 Microcontroller and Embedded Systems", Prentice Hall of India, New Delhi, India, pp. 492-507, Chapter 17.

[2] T.S. WEERAKOON AND L. SAMARANAYAKE "Development of a Novel Drive Topology for Five Phase Stepper Motor". IEEE region 10 colloquium and third ICIIS kharagpur, pp. 35-47, 2010.

[3] GOPAL K.DUBEY "Fundamentals of Electrical Drives", Narosa Publishing House, New Delhi, India, page no.. 280-286, Chapter 8, 14 Edition.

[4] S.G.ABEYRATNE AND U.I.DAYARATNE "ANew Power Conversion Strategy for a Uni-Polar Stepper Motor Drive". IEEE transaction on industrial electronics, pp. 213-217, 2010.

[5] KHALAF AZZEDDINE FERRAH AND JEHAD AL BANI YOUNES "Sensorless speed and position estimation in a stepper motor". IEEE transaction on industrial electronics, pp. 66-78, 2010.

[6] ZHANG YAGUN AND CHEN LONG "A design of elevator positioning control system model". IEEE international conference on neural networks and signal processing, pp. 535-538, 2008.

[7] GUPTA AND G.S BARLOW "Automatic cattle feed mixer". Sixth IEEE symposium on electronic design, pp 99, 2011.

[8] PULLEN AND ELLIS "Kinetic energy storage for vehicles". Hybrid vehicle conference, pp 91, 2005.

[9] MASSON AND J.BARROW 'Non impact printer power and motor control system on a chip". International conference on power electronics and drive system, pp 98, 1995.

[10] JASMIN VELAGIC, NEDIM OSMIC "Remote Control of Stepper Motor via Web Server". Conference on control and fault tolerance system, pp. 910-915, 2010.

[11] K. R. RAJAGOPAL, M. KRISHNASWAMY AND BHIM SINGH "An Improved High-Resolution Hybrid Stepper Motor for Solar-Array Drive of Indian Remote-Sensing Satellite". IEEE transaction on industry applications, Vol 33, No.4, pp. 906-913, 1997.

[12] MANOJ KUMAR MUKUL "Steering of camera by stepper motor towards active speaker using Microphone array". SICE annual conference 2008, pp. 19-23, 2008.

[13] JIANWEN SHO "A novel microcontroller based sensor less brushless DC motor drive for a automotive fuel pumps". IEEE transactions on industrial engineering, pp 1734- 1741, 2008.

[14] MOUNTINHO AND MARTINS "Progress on the design of surveillance system to protect forests from fire". IEEE conference on emerging technologies and factory automation, pp 191, 2003.

[15] WIBOWO AND ZARROR BIN MOHAMMAD JENU "Control system of a turntable and antenna positioning device in an open area test site'. International conference on intelligent and advanced system, pp 1, 2010.

[16] RASS BANNATYNE "Introduction to microcontrollers". IEEE transaction on education, pp 250-254, 1998.

[17] CHIN MING HSU AND HUI MEI CHAO "Constructing microcontroller teaching tool with the integration of hardware and software technology". International conference on new trends in information and service science, pp 270-276, 2009.

[18] XIAOHUA ZHANG, BINGJI XU "Research on stepper motor based control based on single chip and serial communication". World congress on intelligent control and automation, pp 3020-3032, 1998.

[19] ARCHAN PATEL AND K.S DENPIYA "Novel micro stepping technique for disc rotor type motor". International power electronics conference, pp. 968-972, 2010.

[20] JIEBIN ZHU "Design of multi axis control system for stepping motor". World international conference on information engineering, pp. 34-37, 2010.

[21] PETER CRNOSIJA “Microcontroller implement of optional algorithms for closed loop control of hybrid stepper motor driver”. IEEE transaction on industrial electronics, Vol. 47, No. 6, pp. 1319-1324, 2000.

[22] V MARCENO AND D. ARRIGO “ Novel fully integrated 65W stepper motor driver IC”. SPEEDA M2006, ISPE, electric driver, automation and motion, pp. 12-16, 2006.

[23] MOHAN UNDELAND AND RIOBBINS “Power electronics” chapter 7. Page no. 17,79-100, 2007.

[24] JIEBIN ZHU AND GAOHUA LIAO “Design of multi-axis motion control system for Stepping Motor”. WASE international conference on information engineering, pp. 23-26, 2010.

[25] HUIFENQ JIAO AND JIANZHONG “Design of automatic two axis sun tracking system”. International conference on mechanical automation and control engineering, pp 2323, 2011.

[26] ABAS AND HILMI FADZIL “Improved structure of solar tracker with microcontroller based control”. Second international conference on advances in computer and telecommunication, pp 55, 2010.

[27] XIWEI YANG AND MINYIKE “Water measurement network design and implementation’. Second international conference on information engineering and computer science, pp 1, 2010.

[28] MARSHAL AND ROBERTSON “An embedded microcontroller for space craft applications”. IEEE aerospace conference, pp 9, 2006.

[29] RUI HONG AND CHU LU “Project based lab teaching for power electronics and drives”. IEEE transactions on education, pp 108, 2008.

[30] EDWARD Y. YHO “A microcontroller based induction motor drive”. IEEE transaction on industrial electronics, pp 227-235, 1990.

[31] ANDRE BIACHI AND MASSIMOI VALIANI “DSP based versus microcontroller based variable frequency drives for domestic washing machines”. IEEE transactions on power electronics, pp 1047-1055, 2010.

[32] DAGMAI MAYER AND RALF ROSKAN "Discrete time compensating control of stepper motor for an active control leveler" Eighteenth Mediterranean conference on control and automation, pp 719-726, 2010.

WEB-LINKS USED

[31] www.datasheetCatalog.com

[32] www.Fairchildsemi.com

[33] www.8051projects.com

[34] www.Keil.com

[35] www.raisonance.com

[36] www.steppingmotors.com