



Inspiring Excellence

**Design and Development of a Stepper Motor Position Control System in Micro-stepping
Mode Using Atmega32 Microcontroller**

Mahmud Hasan

Nehal Hasnain Refath

Rifat Ferdus

Supervisor: Dr. Amina Hasan Abedin

**A thesis submitted in partial fulfillment for the degree of Bachelor of Science in the
Department of Electrical and Electronics Engineering, BRAC University**

April 2018

DECLARATION

We, hereby, declare that the thesis titled “Design and Development of a Stepper Motor Position Control System in Micro-stepping Mode Using Atmega32 Microcontroller” is submitted to the Department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the Bachelor of the Bachelor of Science degree. This is our original and was not submitted elsewhere for the award of any other degree or any other publication.

Signature of Supervisor

Dr. Amina Hasan Abedin

Assistant Professor

EEE Department

Brac University

Signature of Authors

Mahmud Hasan (EEE)

ID: 13121010

Nehal Hasnain Refath (EEE)

ID: 13221012

Rifat Ferdus (EEE)

ID: 14121052

ACKNOWLEDGEMENTS

This paper is the work of Mahmud Hasan, Nehal Hasnain Refath and Rifat Ferdus students of Electrical and Electronics Engineering (EEE) Department of BRAC University. The paper has been prepared as an effort to compile the knowledge of our years of study in the University and produce a final document, which addresses the Motor Control System, power electronics components and its various properties.

Firstly, we would like to thank our Creator, the Almighty, and The most merciful and most gracious for giving us the strength and ability to complete this research. We are sincerely thankful to our honorable thesis supervisor, Dr. Amina Hasan Abedin, for her immense support undying effort and guidance in completing of our research. We are also grateful, to not only our faculties of Electrical and Electronics Engineering (EEE) department but also each and every faculty who has been blessing in disguise for us during our entire study process here in BRAC University for educating and enhancing our knowledge. Finally, we would like to give a special thank you to our beloved parents, brothers and sisters for their love and support, and our friends for their constant support and encouragement.

Abstract

Stepper motor mainly operates on open loop controller which actually causes loss of steps and slip of steps and so closed loop position control is being developed to overcome these drawbacks. The controller is usually of around 32 bit microcontroller in order to give quick and reliable control operation. Such controller has a wide range of applicants in manufacturing sector for instance this controller is used for precision control applications as well as to control a robot having multiple degrees of freedom. Because of emergence of various low cost microcontroller with their ease of interfacing with digital components and also there is no need for analog to digital conversion circuitry usually required for AC and DC motors that makes stepper motor economical and easy to control. The objectives is to design and develop the controller for position control of stepper motor using microcontroller which makes the rotation of a stepper motor with maximum operating speed with maximum precision. This thesis describes the designing of control circuit for stepper motor by microcontroller program Atmega32 which can control the position of stepper motor properly by using micro-stepping method.

Table of Contents

Chapter1

Introduction.....	1
1.1 Introduction.....	1
1.2 Scope.....	2
1.3 Related Work.....	2
1.4 Outline.....	5

Chapter2

DC Motors Theory.....	6
2.1 DC Motor.....	6
2.2 Brushed DC Motor.....	6
2.3 Brushless DC Motor.....	7
2.4 Stepper Motor.....	8
2.5 Types of Stepper Motor.....	9
2.5.1 Variable Reluctance Motor (VRM)	9
2.5.2 Permanent Magnet Stepper Motor (PMSM)	11
2.5.3 Hybrid Stepper Motor (HSM).....	12
2.6 Advantages and Disadvantages of Stepper Motor	13
2.7 Unipolar Stepper Motor.....	14
2.8 Bipolar Stepper Motor.....	14

2.9 Stepper Motor Control	16
2.9.1 Full Step Sequence.....	16
2.9.2 Half Step Sequence	19
2.9.3 Micro Step Sequence	20

Chapter 3

Proposed System Description.....	22
3.1 Introduction.....	22
3.2 Proposed System Diagram.....	22
3.3 ATmega32 Key Features.....	23
3.4 Digital to Analog Converter.....	24
3.5 Types of DAC Circuits.....	24
3.5.1 Binary Weighted DAC	25
3.5.2 R/2R Ladder DAC.....	28
3.6 Specifications of DACs.....	30
3.7 DAC Interfacing with Microcontroller.....	32
3.8 Description of Relay Driver.....	35

Chapter 4

Proposed Circuit Implementation	38
4.1 Introduction	38
4.2 Angular Increment.....	39

4.3 Simulation Synopsis.....	40
------------------------------	----

Chapter 5

Conclusion.....	49
-----------------	----

5.1 Summary.....	49
------------------	----

5.2 Discussion.....	49
---------------------	----

5.3 Future Scope.....	50
-----------------------	----

References.....	51
------------------------	-----------

Appendix I.....	55
------------------------	-----------

List of Figures

Figure 2.1: Brushed DC Motor.....	7
Figure 2.2: Three Phase Brushless DC Motor Connections.....	7
Figure 2.3: Stator and Rotor design of Stepper Motors.....	9
Figure 2.4: Variable Reluctance Motor.....	10
Figure 2.5: Cross Sectional Diagram of a 4-phase Permanent Magnet Step Motor.....	11
Figure 2.6: Teeth And Magnetic Structure of The rotor of a Hybrid Step Motor.....	12
Figure 2.7: Schematic and Pin Diagram of the Unipolar Stepper Motor.....	14
Figure 2.8: Schematic and Pin diagram of the Bipolar Stepper Motor.....	15
Figure 2.9: Schematic Diagram of the Coil Connection of Stepper Motor.....	15
Figure 2.10: Step Sequence of One-phase - Full Step of the Stepper Motor.....	17
Figure 2.11: Step Sequence of Two-phase - Full Step of the Stepper Motor	18
Figure 2.12: Step Sequence of One and Two-phase - Half Step of the Stepper Motor.....	19
Figure 2.13: Micro-Steps Sequence of the Stepper Motor.....	21
Figure 3.1: Diagram of Proposed System Description.....	22
Figure 3.2: Digital to Analog Converter (DAC).....	24
Figure 3.3: N bit Binary Weighted DAC.....,,,	25
Figure 3.4: N bit R/2R Ladder DAC.....	28
Figure 3.5: Linearity Specifications of DAC.....	31
Figure 3.6: Settling Time Specifications	32
Figure 3.7: DAC interfacing with Microcontroller (ATMEGA 32).....	34
Figure 3.8: Internal Diagram of DAC.....	35

Figure 3.9: Internal Circuit Diagram of ULN 2003A (Darlington Array).....	36
Figure 4.1: Unipolar Stepper Motor in Proteus.....	38
Figure 4.2: Angular Increment for Each Steps.....	39
Figure 4.3: Design of the Proposed Circuit in Proteus.....	42

List of Tables

Table 2.1.....16

Table 2.2.....17

Table 2.3.....18

Table 2.4.....19

Table 4.1.....44

Table 4.246

Abbreviation

IBM	International Business Machine
DMos	Double-Diffused Metal-Oxide Semiconductor
IC	Integrated Circuit
PCB	Printed Circuit Board
PIC	Programmable Interface Controller
FPGA	Field Programmable Gate Array
RAM	Rapid Access Memory
DAC	Digital To Analog Converter

Chapter 1

Introduction

1.1 Introduction

A stepper motor is a brushless dc motor that generates rotation in a particular angular increment when its stator coils are energized in a programmed manner. Rotation happens due to the magnetic interaction between poles of the sequentially energized stator windings and rotor poles. The rotor has no electrical coils, but has salient and magnetized poles. A stepper motor moves in steps, rather than rotating smoothly as a conventional motor. Increment size is calculated in mechanical degrees and depending on the system application, it can vary.

A stepper motor internal structure permits a very simple control system, step motor has several special features that make it an ideal candidate for using as a positioning control device.

- i. Frequency are proportional to the Speed and the amount of rotation and number of the input pulses. So speed has a wide rotational range which can be achieved and accurate positioning is possible with small angular steps increment.
- ii. Precise operation can be provided with open loop control because positional error is non-cumulative and the accuracy of a stepper motor is very high.
- iii. Very reliable machine device.
- iv. Device starting, stopping and reversing process are easy.

Stepper motor has some disadvantages too. Some of them are:

- i. Resonances can be occurred at certain speed ranges.
- ii. Step accuracy and resolution are directly dependent to the device structure and can be a limiting factor for various applications where high resolution is required.

Micro-stepping ^[1, 2, 3] is a method for driving stepper motor where input pulses are converted to discrete estimations of the sinusoidal functions. These results in fractional input signals as opposed to zero, full positive or full negative input signals used in commonly employed full-step and half-step excitation methods. This method results in very smoother operation which has less vibration and highly enhanced angular resolution of motor control. This process namely minimizes the effects of the step motor's disadvantages which are mentioned above.

1.2 Scope

This thesis discussion presents a step motor position control system design with the aim of achieving micro-stepping capabilities. Step motors are critically analyzed and different types of driving methods including several possible control processes are discussed in the theory section. A proteus simulation circuit designed and implemented in order to control the position of the rotor with a smallest angular step possible by the micro controller.

1.3 Related Work

Stepper motors earlier studies are dated back to 1920's ^[4]. Although the stepper motor structure was known and during the 19th century used as an electromagnetic engine. A three-phase variable reluctance step motor was introduced as a position control device in the article named "The Application of Electricity in Warships" for very first time. ^[4, 5] In this article, a 3-phase variable reluctance stepper motor and a mechanical rotary switch that causes motor to move in steps of 15° were described.^[5] Inventions which enhanced step accuracy and torque capability of stepper motors proceeded gradually^[6, 7] and the use of permanent magnets in stepper motor internal structure was introduced in this period. Hybrid stepper motor, which is widely used today, was invented, patented ^[8] and manufactured by General Electric Company in 1952. First examples of this motor were used as synchronous motors running at low speeds. However, hybrid motors have been used increasingly as a position control device ever since. Especially, in 1960's and 1970's, evolving computer technology has also led an improvement in stepper motor technology. Computer manufacturers, such as IBM, has used stepper motors as actuators in terminal devices and promoted production of high performance stepper motors. In addition, as

microprocessors become widespread, the use of stepper motors became very easier and more preferable due to the particular discrete nature.

Since the roots of stepper motor applications are found nearly a century earlier from now, it is possible to observe almost all footsteps of the digital control history by reviewing stepper motor drivers used throughout time. As mentioned before, first examples utilized to excite stepper motors were manual mechanical switches. Later, with the invention of switching devices such as thyatron gas tubes, more complicated and autonomous driver options have become in use. A widespread implementation was an information storage application. Operation instructions for numerically controlled machines were stored in the form of perforations on tapes and a sensing device was used to convert the information on the tape to the control signals for the switches. These switches were slowly replaced by thyristors and transistors. The implementation of logic circuit of drivers became simpler with lower costs with the appearance of the integrated circuits.

Stepper motors are reliable and widely used in the industry due to their low price, simplicity and low cost driver solutions. There are many alternatives to implement the power converter block of a driver, partly or whole on-chip solutions are provided by various manufacturers. L297-L298 IC's from ST Microelectronics is among the classic and popular driver solutions. In 2006, ST announced a new 65W fully integrated step motor driver IC ^[9]. Ji et al. ^[10] proposes an application circuit utilizing DMOS full-bridge motor driver IC LMD18245 ^[11] from National Semiconductor.

There are also highly sophisticated single chip solutions for driving step motors in the market today. Allegro's A3981 ^[12] and A4980 ^[13] and AMIS-30523^[14, 15] from ON Semiconductor (formerly AMI Semiconductor) are some examples of this type of drivers. Although on chip solutions offer simple construction, a fully-functional drive circuitry with a few external components; discrete driver solutions are still widely used. One reason for this is that the maximum output power capability of the system is limited in on-chip solutions. Power converter block constructed with discrete components gives more flexibility in this case. Similarly, drive

control logic circuitry options are also limited and cannot be changed for most integrated driver IC's. However, a PCB level driver with driver logic being handled by a microcontroller device offers a flexible environment to improve various driver algorithms. The core component of a PCB level step motor driver is the processor where driver logic is generated. Besides microcontrollers, DSP's (Digital Signal Processor) are also widely used as processors in step motor drivers. An application report from Texas Instruments describes an implementation of a micro-stepping algorithm for TMS320F2808 DSPs.^[16] Bellini et al^[17] presents a micro-stepping implementation where a DSP is utilized as the processor of the driver where a new pulse width modulation (PWM) method, which is called as mixed-mode PWM, is proposed.

FPGA based step motor driver solutions were also implemented in recent years^[3, 17, 18] although not as very widespread as microcontrollers or DSP's. Le and Jeon^[17] propose an open-loop step motor driver based on FPGA. Although FPGA based solutions are relatively a harder to implement and more costly, their rationale is that FPGA based designs are compatible with Application Specific Integrated Circuit (ASIC) conversion and thus they may be preferred. Another example to FPGA based solutions is the work of A. Astarloa et al.^[3] where they propose a step motor control system which achieves the control of the micro-stepping resolution on-the-fly; which most of the step motor driver systems cannot offer, by dynamically reconfiguring block RAM in the FPGA used as the processor of the driver.

An application note from Microchip,^[19] introduces a step motor driver implementation realized with a new generation PIC family, called the dsPIC. The study covers different controlling schemes and micro-stepping options with several resolutions. Ran Zhang et al^[20] present a PIC18F2331 based step motor driver with closed-loop current control using micro-stepping method for a two-phase hybrid step motor. In this study, a high and low side driver IR2110^[21] is used to drive the discrete power block elements and cycle-by-cycle current limiting, a feature of the driver IC, is utilized to control current output of the driver.

Many other studies on step motor driving exist in the literature. Different driving approaches exist for different step motor structures, such as linear motors.^[2] A driving scheme for 5-phase step motors is proposed in.^[23] Step motor modeling is a major area where Morar^[24] discusses accurate simulation modeling of step motors. Closed-loop positioning methods form another major area.^[25] Brown et al^[2] introduce a damping circuit to solve oscillation problems.

1.4 Outline

This thesis has six chapters. In this thesis Chapter 1 is the introductory chapter where mainly a brief introduction to stepper motor, why it used and related work on stepper motors are given. Chapter 2 discusses on motors and stepper motor control methods which are the main component to be driven with the proposed circuitry and Chapter 3 provides a brief discussion on proposed open loop drive circuitry components of a unipolar stepper motor. The proposed driver circuit implementation is presented in Chapter 4. Lastly, Chapter 5 puts forward conclusion which has the brief summary, discussion and future scope.

Chapter 2

DC Motors Theory

2.1 DC Motors

Direct current supply is always fed to DC motors. The magnetic fields are generated either by supplying direct current to electromagnetic windings or with permanent magnets. The rotor rotates on its axis as a result of electromagnetic interactions with the stator. During motor operation, the magnetic fields repel and attract each other, and the rotor continues to rotate. Based on the construction, DC motors are categorized into three types: brushless DC motors, brushed DC motors, and stepper motors

2.2 Brushed DC motor

A brushed DC motor-which is one of the types of DC motors, is mechanically commutated to switch the current so that it does not require an external switch from the controller. The stator is made up of either electromagnetic windings or of a permanent magnet, sometimes called field windings or field pole. The rotor consists of windings, called armature, that is attached to a commutator-brush assembly. An armature is an electric component of the DC motor that is linked to the rotating shaft. The brush-commutator assembly supplies current to the armature. The commutator is forged from copper and surrounds the armature in two or more segments (Fig. 2.1). The brushes are of carbon and skim over the commutator. As the motor turns, the brushes slide on various sections of the commutator. Different windings of the armature are attached to these sections of commutator, causing current in each winding to switch direction at the proper time for the individual winding. The commutator also contributes the motor in reversing its direction of motion, thus changing the polarity of the voltage. This fluctuation of the voltage, causes a fluctuation in the magnetic field around the armature, resulting in the forces on the armature to be reversed.

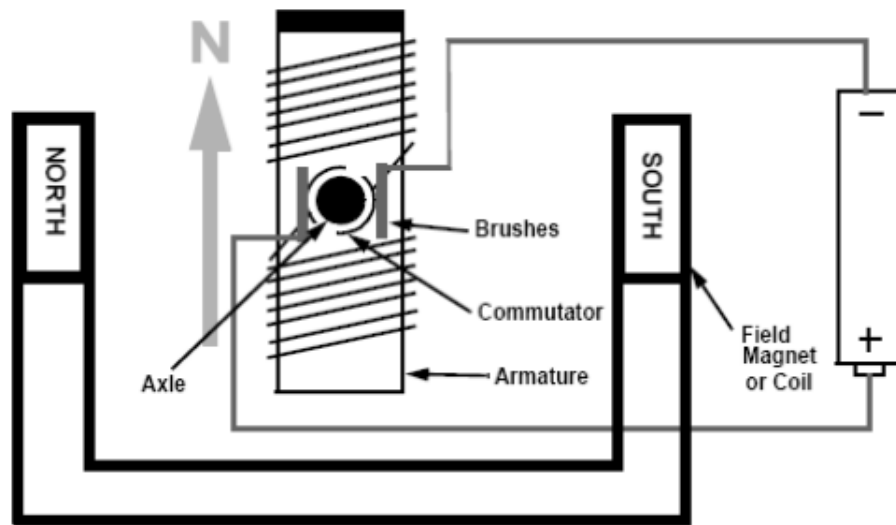


Figure 2.1: Brushed DC Motor^[27]

2.3 Brushless DC Motor

Brushes of DC motor wear out over time and may cause sparking. For this reason concept brushless DC motor used to run expensive equipment which needs long term and reliable running without changing anything. Brushless DC motor has same output as brush DC motor except the construction procedure.

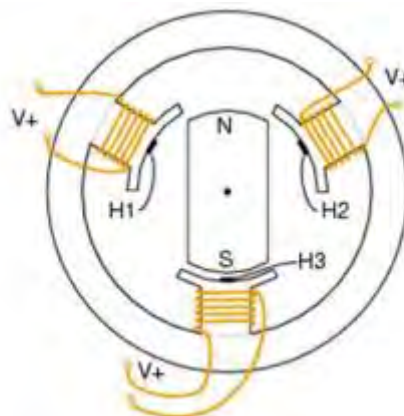


Figure 2.2: Three Phase Brushless DC Motor Connections^[28]

The rotor is of permanent magnet where stator has a coil arrangement as shown in the picture. Feeding dc power to each set of coil will energize each of them and become an electromagnet. Simple force interaction between rotor's permanent magnet and stator's electromagnet rotates the rotor. Stator coil creates same polarity of rotor permanent magnet. So opposite pole of rotor and stator are attracted to each other. When attracted poles come near, next set of coils are energized. This process goes on and rotor rotates.

There is a drawback of brushless DC motor. It is that only one set coils are energized so there is a power loss in the system. It can be fixed by energizing such way that when first set of coils rotating the rotor, the next coils are pushing the rotor. This is done by passing same polarity current through second coil. Electronic controller and sensor are used to control coil energization process.

2.4 Stepper Motor

A stepper motor is generally a brushless dc motor. The rotor of a stepper motor is composed of permanent magnets with poles and a stator with windings. The rotor is formed using a single magnet mounted in alignment with the rotor axis and two pole pieces with many teeth. The teeth are staggered to produce many salient poles. It is easy to use stepper motor for positioning and moves in step by step based on pulses which was supplied to the stator windings part. The direction of rotation can be changed by reversing the pulse sequence and speed is controlled by the frequency of pulses or pulse rate. As the rotor aligns with one of the stator poles, the second phase is energized. The two phases alternate on and off and also reverse polarity. There are mainly four steps. One phase lags the other phase by one step. This is equivalent to one fourth of an electrical cycle or 90° . The angle of each rotational movement obviously depends on the rotor teeth and the number of stator of the stepper motor.

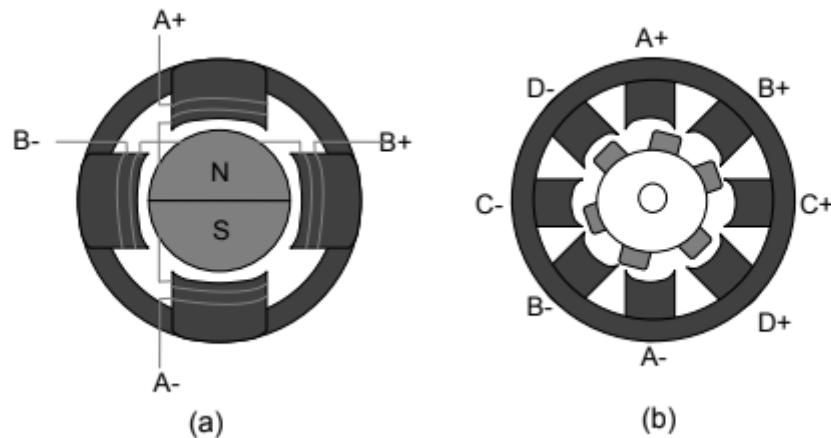


Figure 2.3: Stator and rotor design of stepper motors

(a) Two-phase Permanent-magnet (PM) stepper motor, and

(b) Four-phase Variable reluctance (VR) stepper motor (adapted from ²⁹)

2.5 Types of Stepper Motor

The operating principle of stepper motor is very simple as the rotor rest position adjusts with a fixed angle rotation by using excitation switches from one section to any other. This is the basic concept for all the types of stepper motor. However based on the machine structure and operation principle stepper motor has three types.

They are:

- i. Variable Reluctance Motor (VRM)
- ii. Permanent Magnet Stepper Motor (PMSM)
- iii. Hybrid Stepper Motor (HSM)

2.5.1 Variable Reluctance Motor (VRM)

Wound stators and wired iron rotors makes up the Variable reluctance step motors . The basic operation principle is easier to visualize from a constructional point of view. The poles become

magnetized by energizing the stator windings using DC current. Rotation will occur due the attraction between rotor teeth and the energized stator poles. Both stator and rotor high permeability allows high magnetic flux to pass through even if a low magneto motive force is applied. When the rotor teeth are directly lined up with the stator poles, the rotor is in a position of minimum reluctance to the magnetic flux. Thus rotor keeps rotating when present step is de-energized and the next step of the phase sequence is energized. This phase angle can be changed by changing stator poles and rotors teeth.

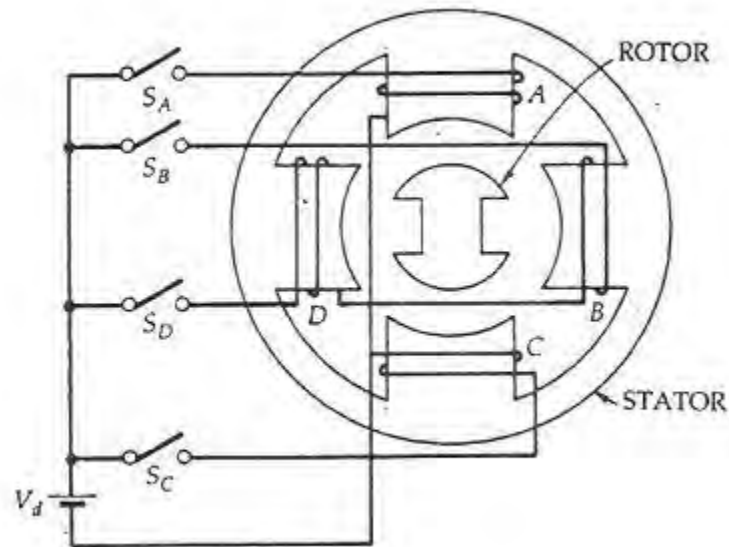


Figure 2.4: Variable Reluctance Motor^[30]

Step motors basically have lower step angles-generally less than 30° . Increasing the number of phases and rotor teeth can make even much lower step angle achievable. A single step in this motor conjures up in bringing into existence, the rotation of 15 degree. This motor gives up the ease of control with an additional electrical phase to obtain a higher step number.

A substitute to this approach for increasing the number of steps is using two or more teeth per pole. This motor attains 42 number of steps, thus producing step angle of 8.57° .

2.5.2 Permanent Magnet Stepper Motor (PMSM)

A permanent magnet in rotor is used this type of motor. Rotor has no teeth like VRM. Rotor is magnetized with alternating south and north poles located in a straight line which is parallel to the rotor shaft. Due to this increased magnetic flux intensity occurs. This is the reason why PM motor shows improved torque characteristics in comparison with that of the VRM type.

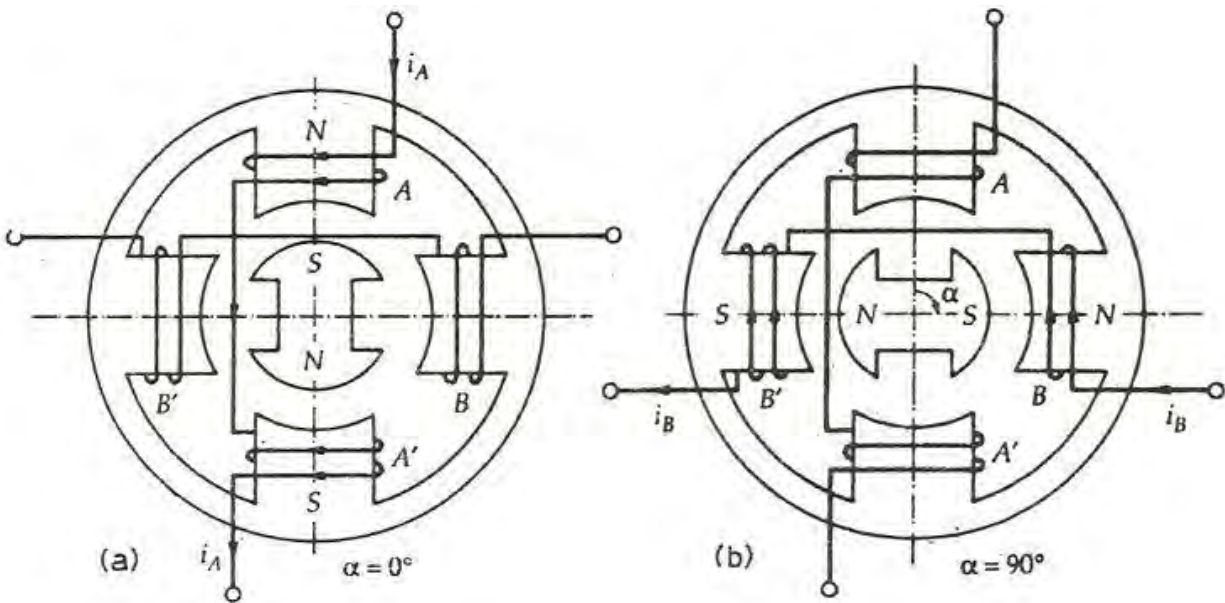


Figure 2.5: Cross sectional diagram of a 4-phase permanent magnet step motor. ^[31] (a) Stator teeth and cylindrical magnet rotor and (b) winding arrangement.

The rotor stays at a detent position forced upon by the magnetic structure of the motor when there is zero excitation occurring. The position may vary from the rotor rest position when excitation is in place. Rotation in a permanent magnet step motor is controlled by alternating excitation between successive phases. Motor can be rotate in forward or backward step. Step angle of Fig 2.5 motor is 90° . Step angle of permanent magnet motors can be decreased by putting into existence, more stator teeth and phases.

2.5.3 Hybrid Stepper Motor (HSM)

Hybrid stepper motor operation is the combination of Permanent Magnet Stepper Motor (PMSM) and Variable Reluctance Motor (VRM). The term “hybrid” came due to this reason. Hybrid stepper motor is more expensive than PMSM and VRM however it provides better step resolution, torque and speed.

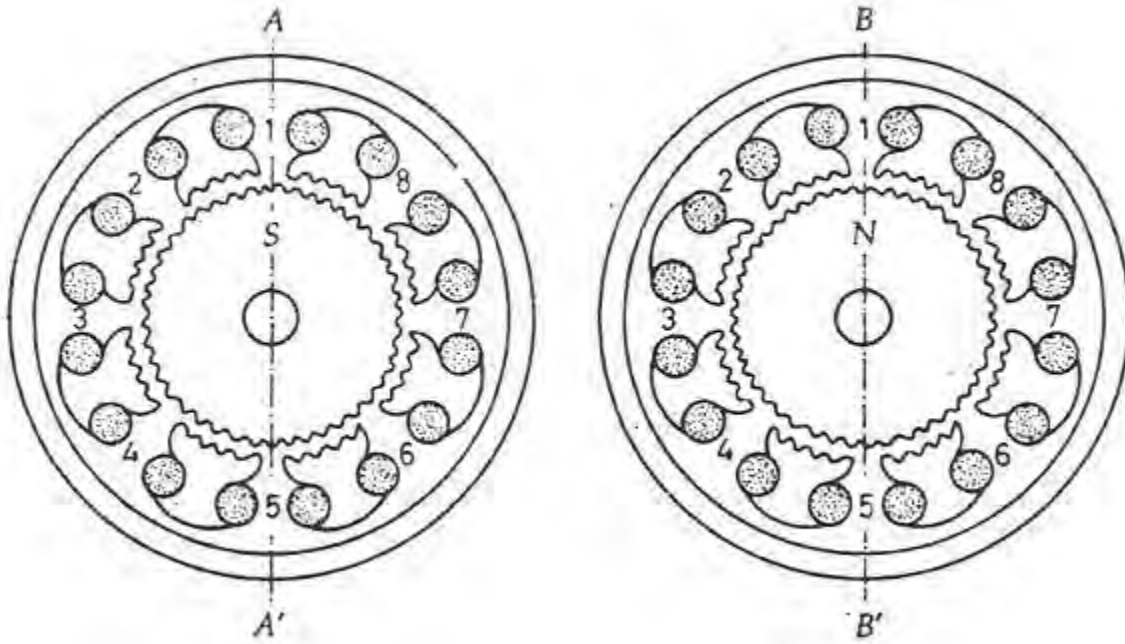


Figure 2.6: (a) Teeth and magnetic structure of the rotor of a hybrid step motor. (b) Bifilar windings of two phases of a 4-phase stator of a hybrid motor^[32]

In a typical hybrid step motor, the rotor structure is cylindrically stacked into two more sections as shown in Figure 2.6 (a). Interior of the rotor is a cylindrical permanent magnet with different poles in consecutive stacks and the teeth of the rotor are laminated steel. Different stacks have a fractional tooth pitch difference.

The stator structure of hybrid motors with misaligned rotor teeth are not different from stator structures of other motor types discussed so far. However, the windings are arranged in bifilar scheme that allows induction of magnetic flux in both directions. Bifilar filing in a two phase hybrid motor is shown in Figure 2.6 (b). In a hybrid step motor, during excitation of phase A, of

the rotor become attracted and align to a specific position in hybrid stepper motor. While excitation changes into phase-B, the rotor attracted in the opposing direction and movement is achieved in an indicated direction designed by proper teeth alignment. Variations types of hybrid motors can be obtained either by using a permanent magnet as the stator instead of using the rotor or by enforcing teeth misalignment to stator stacks instead of the rotor.

2.6 Advantages and Disadvantages of Stepper Motor

Advantages of Stepper Motor:

- i. The input pulse is proportional to rotation angle of the motor.
- ii. The motor has full torque at standstill.
- iii. Precise positioning and repeatability of movement since good stepper motors have an accuracy of 3 – 5% of a step and this error is non-cumulative from one step to the next.
- iv. Excellent response to starting, stopping and reversing.
- v. Very reliable since there are no contact brushes in the motor. Therefore the life of the motor is simply dependent on the life of the bearing.
- vi. The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.
- vii. It is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft.
- viii. A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses.

Disadvantages of Stepper Motor:

- i. Resonances can occur if not properly controlled.
- ii. Not easy to operate at extremely high speeds.
- iii. It has limited ability to handle large inertia load.
- iv. Use more current than D.C. motors.

2.7 Unipolar Stepper Motor

The unipolar stepper motor has five or six wires and four coils (actually two coils divided by center connections on each coil). The center connections of the coils are tied together and used as the power connection. They are called unipolar steppers because power always comes in on this one pole.

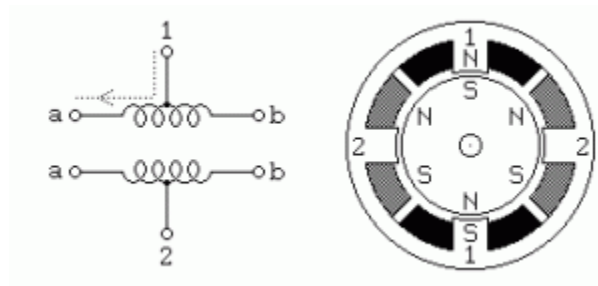


Figure 2.7: Schematic and Pin diagram of the unipolar stepper motor^[33]

2.8 Bipolar Stepper Motor

Bipolar steppers have no common center connection. The motor usually has four wires coming out of it. They have two other independent sets of coils. This two coil mainly distinguish them from unipolar steppers which are cable of measuring the resistance between the wires and two pairs of wires are of equal resistance. If you've got the leads of your meter connected to two wires that are not connected (i.e. not attached to the same coil), you should see infinite resistance (or no continuity).

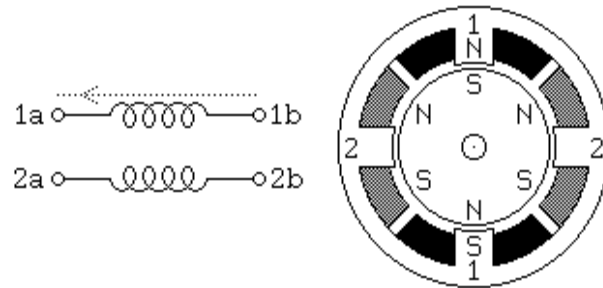


Figure 2.8: Schematic and Pin diagram of the bipolar stepper motor^[33]

We will talk mostly on "Unipolar stepper motors" which is most common type of stepper motor available in the market. A simple example of 6 lead step motor is given below and in 5 lead step motor wire 5 and 6 are joined together to make 1 wire as common.

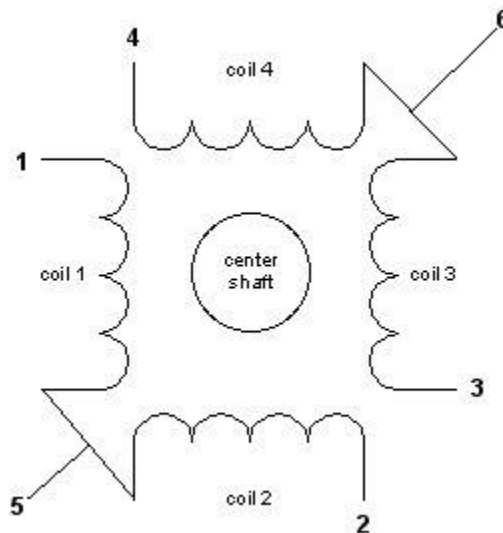


Figure 2.9: Schematic diagram of the coil connection of stepper motor

2.9 Stepper Motor Control

Stepper drives control how a stepper motor operates; there are three commonly used excitation modes for stepper motors, full step, half step and micro stepping. These excitation modes have an effect on both the running properties and torque the motor delivers. Now we give a short description about half step and full step and later a details description on micro stepping is given .

Stepper motors can be driven in two different patterns or sequences namely:

- i. Full Step Sequence
- ii. Half Step Sequence
- iii. Micro Step Sequence

2.9.1 Full Step Sequence

In the full step sequence, two coils are energized at the same time and motor shaft rotates. The order in which coils has to be energized is given in the table below.

Table 2.1: Coil Energized Sequence for Full Step

STEP	A	B	A'	B'
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

There are two types of full step excitation modes. In one-phase on - full step, the motor is operated with only one phase energized at a time. This mode requires the least amount of power from the driver of any of the excitation modes.

Table 2.2: Coil Energized Sequence for Full Step (One Phase at a Time)

STEP	A	B	A'	B'
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

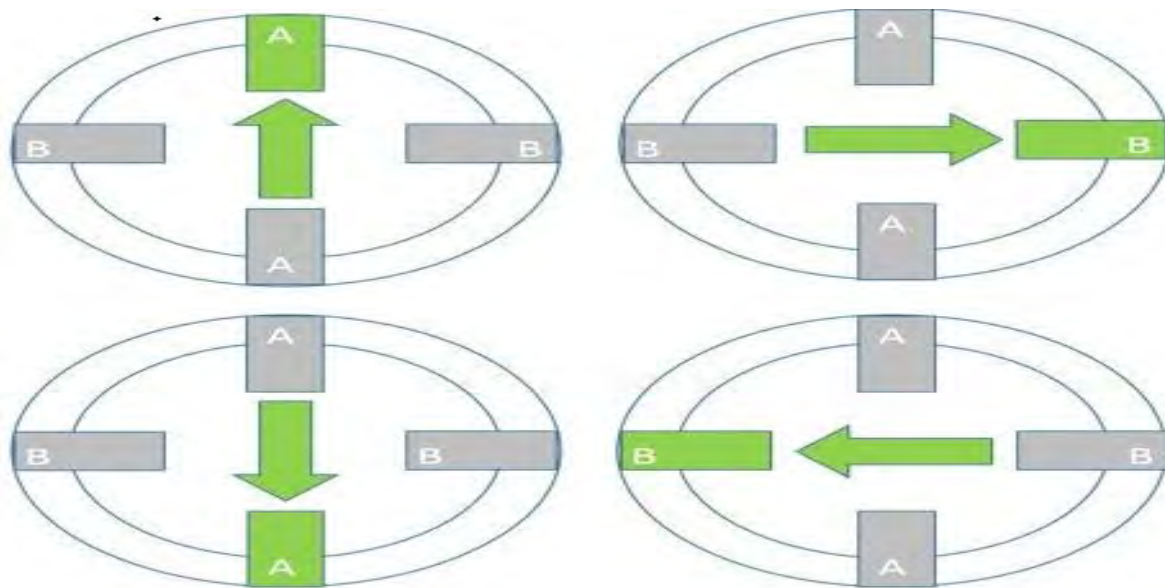


Figure 2.10: Step sequence of one-phase - full step of the stepper motor^[34]

In two-phase on - full step, the motor is operated with both phases energized at the same time. This mode provides improved torque and speed performance. Two-phase on provides about 30% to 40% more torque than one phase on, however it requires twice as much power from the driver.

Table 2.3: Coil Energized Sequence for Full Step (Two Phase at a Time)

STEP	A	B	A'	B'
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

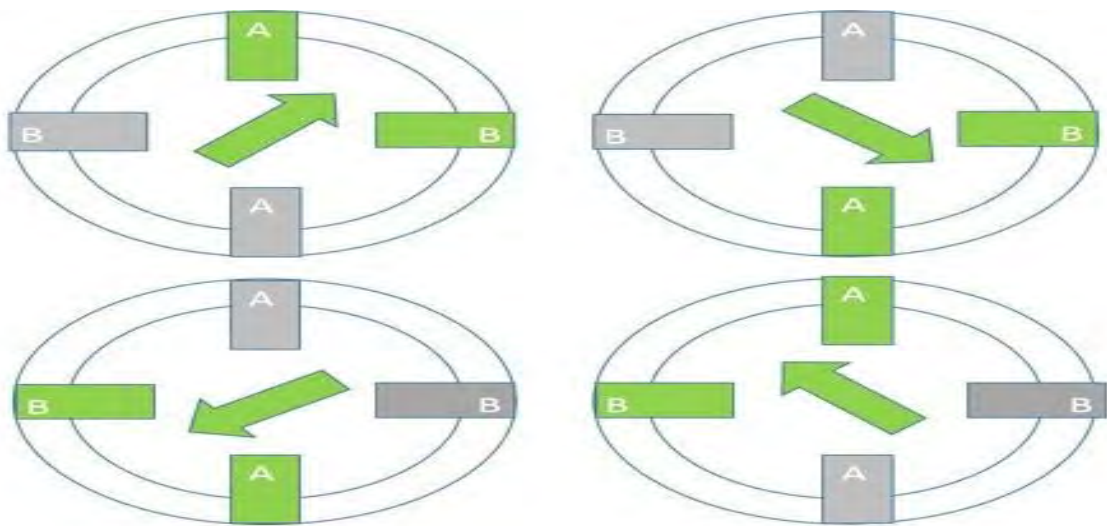


Figure 2.11: Step sequence of two-phase - full step of the stepper motor^[34]

2.9.2 Half Step Sequence

In Half mode step sequence, motor step angle reduces to half the angle in full mode. So the angular resolution is also increased i.e. it becomes double the angular resolution in full mode. Also in half mode sequence the number of steps gets doubled as that of full mode. Half mode is usually preferred over full mode. Table below shows the pattern of energizing the coils.

Table 2.4: Coil Energized Sequence for Half Step

STEP	A	B	A'	B'
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

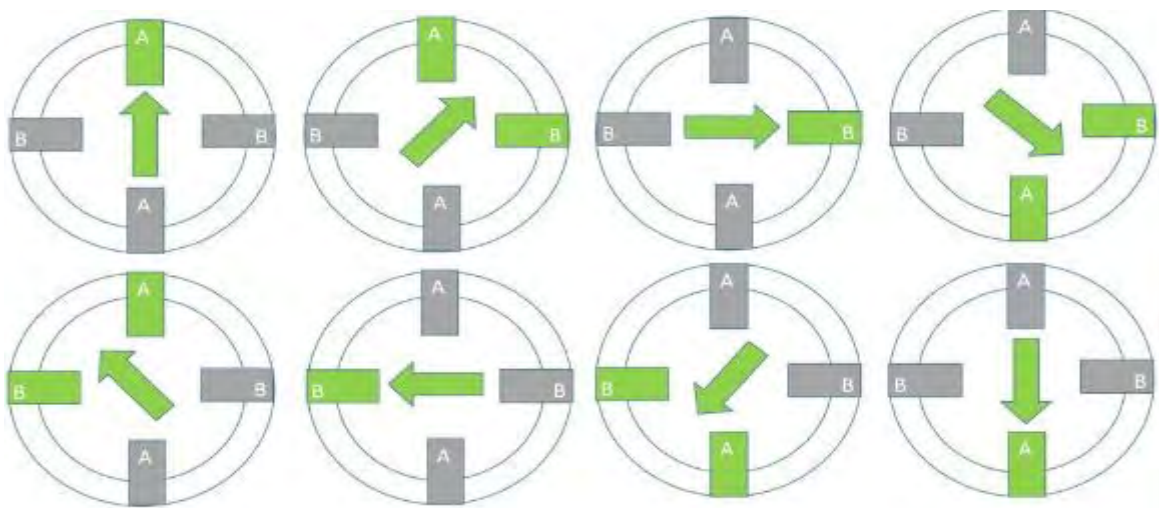


Figure 2.12: Step sequence of one and two-phase - half step of the stepper motor^[34]

2.9.3 Micro Step Sequence

Micro-stepping increases system performance at lower cost and system complexity in contrast to full- and half-step driving modes. Micro-stepping reduces noise and resonance problems and raises step accuracy and resolution. The way to make the stator flux of a stepper motor move more steadily compared to full or half step drive modes is the micro-stepping, with less jerking off and reduced noises to 0Hz. It also has a smaller step angles and better positioning making.

When a stepper is driven in full-step and half-step modes the stator flux is rotated 90 and 45 electrical degrees, respectively every step of the motor. There are varieties of micro-stepping modes possible from 1/3 the step size of a full step to 1/32 or further less. Although it is possible to use non integer fractions of a full step, but in reality it does not occur.

A stepper motor is a synchronous motor, where the rotor's ceasing position is in synchronous with the stator flux. The stator flux rotates the rotor in synchronous with itself, hence, making the rotor to move to a new firm and ceasing position. The torque (T) generated by the motor is a function of the distance between the stator flux (f_s) and the rotor position (f_r) and the holding torque(T_H).

$$T = T_H \sin(f_s - f_r)$$

Where f_s and f_r are given in electrical degrees.

The relationship between electrical and mechanical angles is given by the formula:

$$F_{electrical} = \frac{n}{4} \cdot f_{mech}$$

Where n is the number of full-steps per revolution.

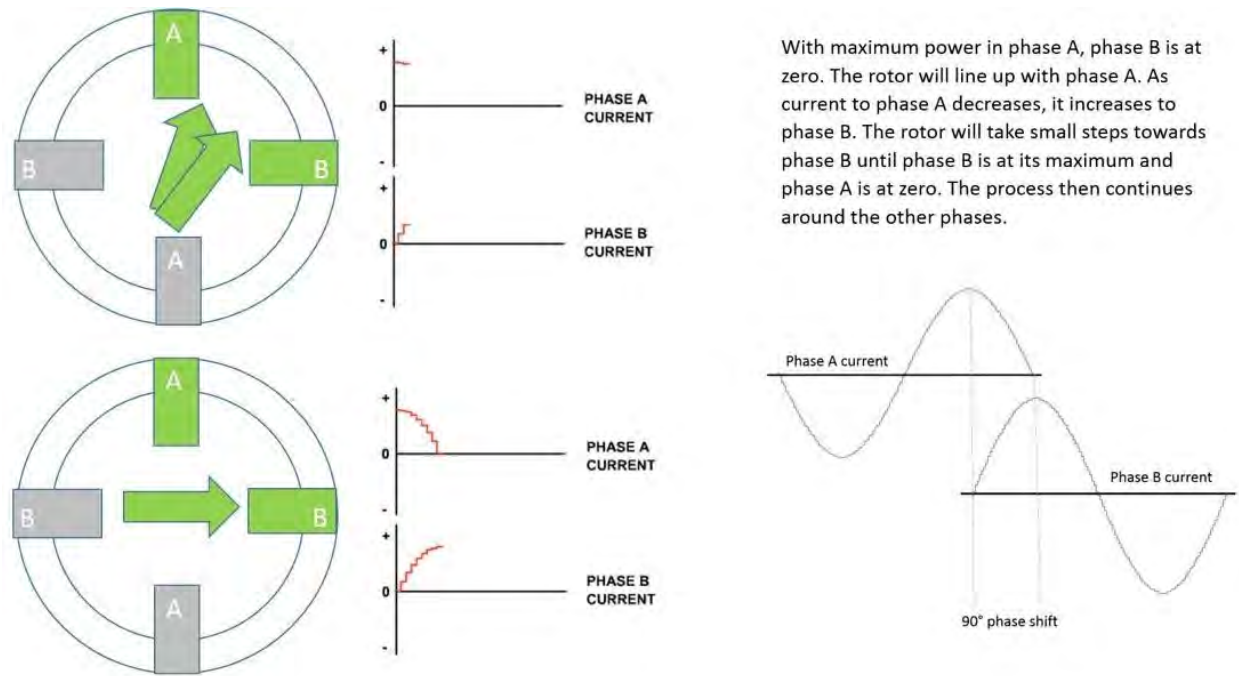


Figure 2.13: Micro-Steps Sequence of the stepper motor ^[34]

This chapter discusses about the basic theory of DC motor and Stepper motor control theory which is essential for readers to read before proceeding to next chapter. Next chapter will discuss about the proposed control system design for running the stepper motor in micro-stepping mode.

Chapter 3

Proposed System Description

3.1 Introduction

To control the position of the unipolar stepper motor, we use the coil energized concept. For that we used DAC for binary conversion of voltage, Atmega32 for providing binary value to the DAC and Relay for switching purpose.

3.2 Proposed System Diagram

For our proposed design and development of a stepper motor position control system in micro-stepping mode using Atmega32 micro-controller, we follow the below diagram:

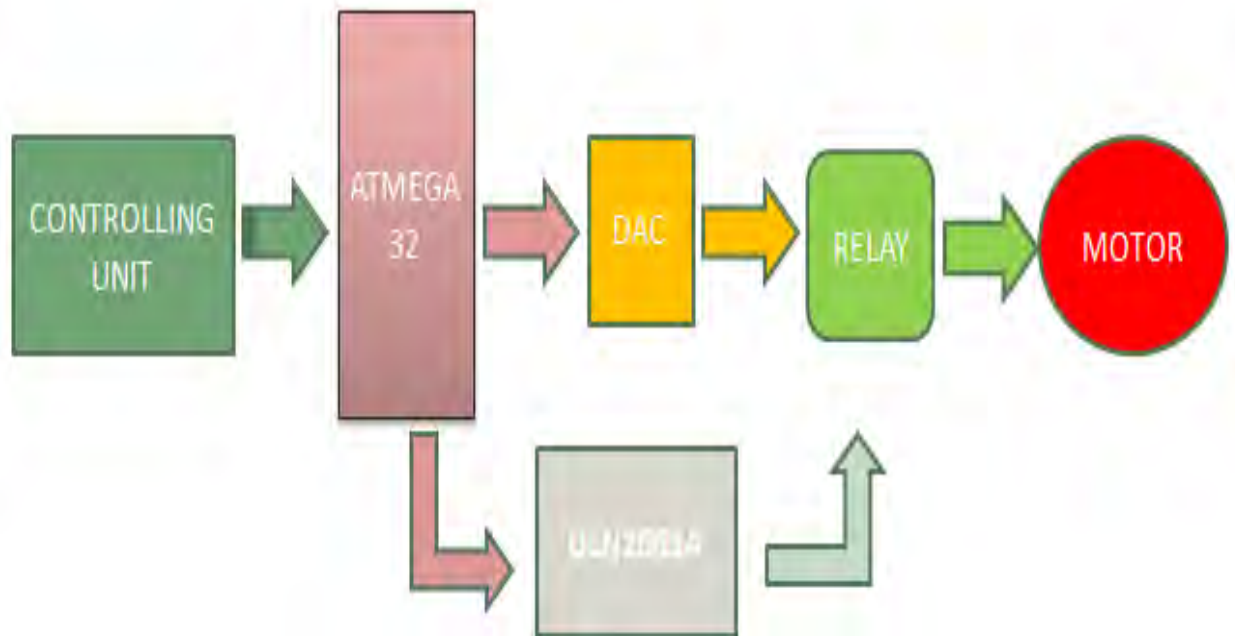


Figure 3.1: Diagram of Proposed system Description

3.3 ATmega32 Key Features

I/O pins: Normally ATmega32^[35] is capable of dealing with analogue inputs. Port A can be used as digital I/O lines. Even each individual pin can be used as a single input channel to the internal ADC, plus a pair of pins AREF, AVCC & GND. Sequentially Atmega32 has Port B, Port C and Port D to use as I/O pins depending on the programmer's choice.

One pins cannot be used for two purposes (for an example: At the same time Port B pins cannot work as a Digital I/O pin while the Internal ADC is activated) . This is mainly depend upon programmer to solve the problem in the circuitry and the program. In this case programmer advised to use the priority tables and the datasheet for the internal configuration.

Special Microcontroller Features:

- i. Six sleep modes: Idle, ADC noise reduction, power-save, power-down, standby and extended standby.
- ii. Internal calibrated RC oscillator
- iii. External and internal interrupt sources
- iv. Power on reset and programmable brown-out detection

3.4 Digital to Analog Converter

Digital to analog conversation is a process in which digital inputs are converted to analog inputs.

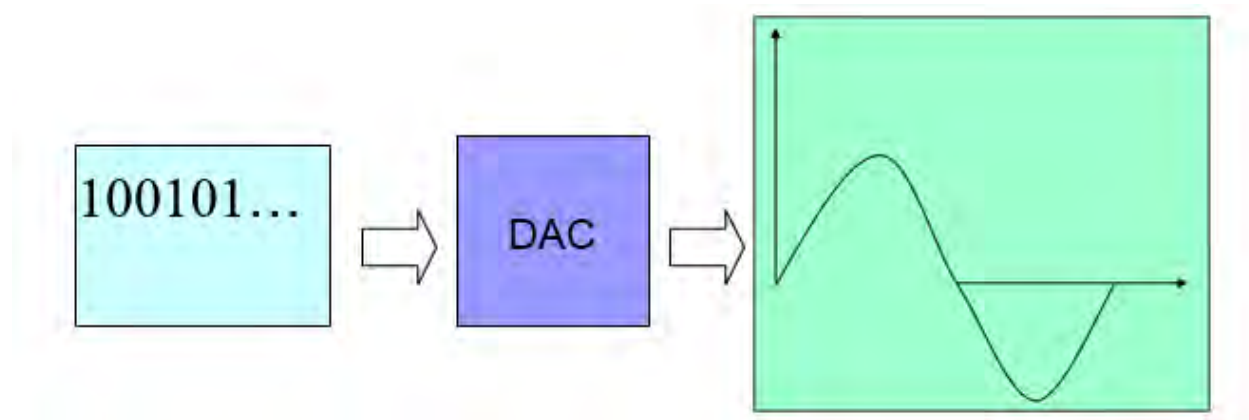


Figure 3.2: Digital to Analog Converter (DAC)

Normally microcontroller generates digital form of output but the controlling system don't accept digital data whenever they need analog signal as they so here for getting analog signal, DAC is used which converts digital data into equivalent analog voltage.

3.5 Types of DAC Circuits:

- i. Binary weighted
- ii. R-2R ladder

3.5.1 Binary Weighted DAC

The binary-weighted-resistor DAC works according to the characteristics of inverting Op Amp circuit. So the value of the output voltage is equal to the inverted sum of all input voltages. If the values of input resistor are considered to multiples of two: $1R$, $2R$ and $4R$ and the value of output voltage would be equal to the sum of V_1 , $V_2/2$ and $V_3/4$. Here V_1 corresponds to the most significant bit (MSB) while V_3 is the least significant bit (LSB).

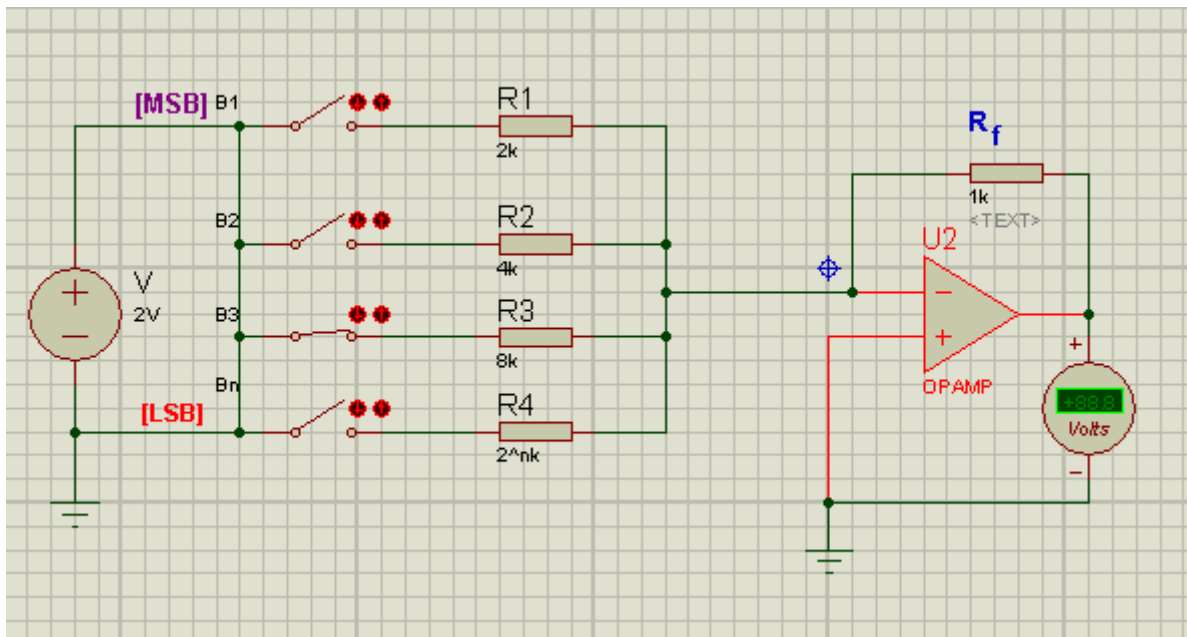


Figure 3.3: N bit Binary weighted DAC

It has feedback resistor R_f . The binary word (i.e. $B_1 B_2 B_3 \dots B_n$) is indicated by the switch position. Here for converting current to voltage an op-amp is used in the circuit.

While the switches became closed the respective currents are flowing through resistors as shown in the circuit diagram above.

Since we know op-amp input current is zero, the addition current flows through feedback resistor.

$$\therefore I = I_1 + I_2 + I_3 + \dots + I_n$$

Voltages V_1 for B_1 through V_n for B_n are either V_{ref} if corresponding bit is high or ground if corresponding bit is low.

V_1 is MSB

V_n is LSB

If $R_f = R/2$

Formula to find out current

$$I_0 = V_R \sum_{n=1}^N B_n / 2^{(n-1)} R$$

I_0 = sum of currents leaving junction

R = resistance corresponding to MSB

N = number of input bits

B_1 = MSB

$$V_0 = -R_f I_0$$

V_0 = voltage output from amplifier

R_f = feedback resistance Resolution = $V_R / 2^N$

$$V_0 = -R_f [I_1 + I_2 + I_3 + \dots + I_n]$$

$$V_0 = -R_f \left[B_1 \frac{V_R}{R2^1} + B_2 \frac{V_R}{R2^2} + B_3 \frac{V_R}{R2^3} + \dots + B_n \frac{V_R}{R2^n} \right]$$

$$V_0 = -\frac{R_f}{R} \left[B_1 \frac{V_R}{R2^1} + B_2 \frac{V_R}{R2^2} + B_3 \frac{V_R}{R2^3} + \dots + B_n \frac{V_R}{R2^n} \right]$$

$$V_0 = -V_R [B_1 2^{-1} + B_2 2^{-2} + B_3 2^{-3} + \dots + B_n 2^{-n}]$$

As an example consider an 8 bit DAC.

When input binary sequence B1B2 B3 B4 B5 B6 B7 B8=11010000

$$V_0 = -V_R [B_1 2^{-1} + B_2 2^{-2} + B_3 2^{-3} + B_4 2^{-4} + B_5 2^{-5} + B_6 2^{-6} + B_7 2^{-7} + B_8 2^{-8}]$$

$$V_0 = -V_R [1.2^{-1} + 1.2^{-2} + 0.2^{-3} + 1.2^{-4} + 0.2^{-5} + 0.2^{-6} + 0.2^{-7} + 0.2^{-8}]$$

$$V_0 = \frac{V_R 13}{16}$$

Advantages of Binary weighted DAC

- i. Simple Analysis
- ii. Fast Conversion

Disadvantages of Binary weighted DAC

- i. This setup requires a wide range of accurate values of resistors.
- ii. A 10 bit DAC needs resistors ranging from R to R/1024.
- iii. Requires low switch resistances in transistors
- iv. Usually limited to 8-bit resolution.

3.5.2 R/2R Ladder DAC

DAC where only the resistor of R and 2R is used the so-called R/2R DAC. A disadvantage of the former DAC design was several different precise input resistor values: one unique value per binary input bit is needed .

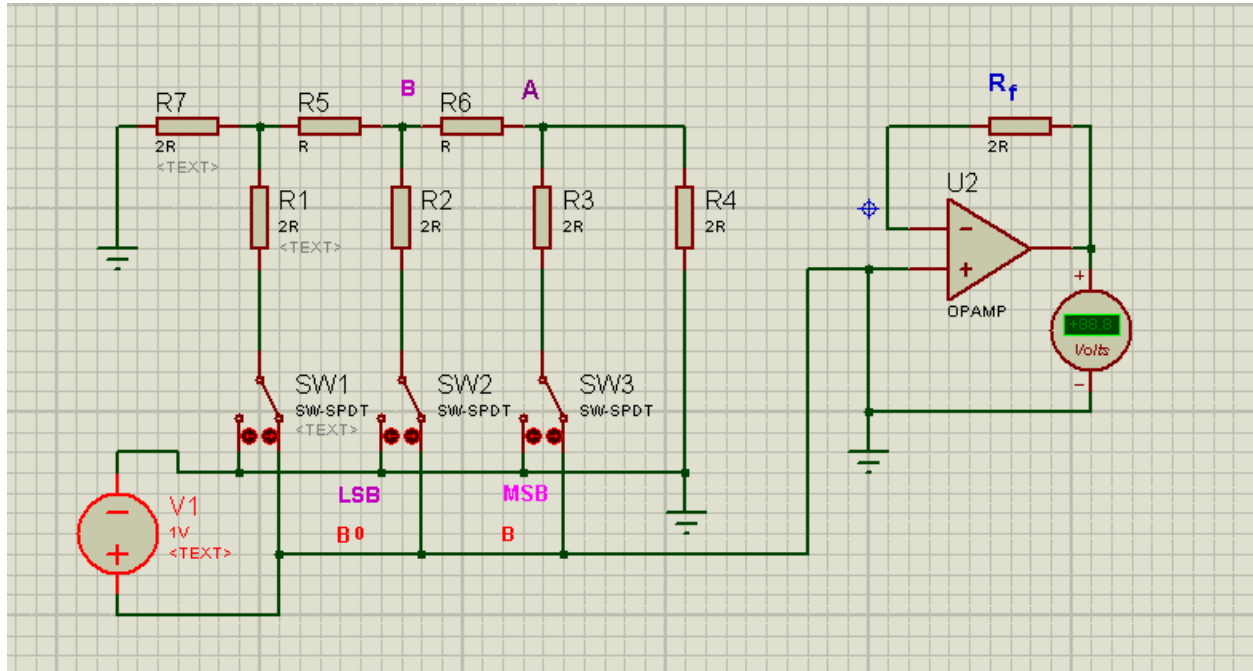


Figure 3.4: N bit R/2R Ladder DAC

$$V_0 = -V_R \sum_{n=1}^N B_n / 2^{(n-1)} R \quad \text{where } B_1 \text{ is MSB, ; where considering } N \text{ is number of bits}$$

$$\sum I_n = V_R / 2R \sum_{n=1}^N b_n / 2^{(n-1)} \quad \text{where } B_1 \text{ is MSB}$$

$$V_0 = -R_f I_0$$

$$V_0 = -R_f [I_1 + I_2 + I_3 + \dots + I_n]$$

$$V_0 = -R_f [B_1 \frac{V_R}{R2^1} + B_2 \frac{V_R}{R2^2} + B_3 \frac{V_R}{R2^3} + \dots + B_n \frac{V_R}{R2^n}]$$

$$V_0 = -V_R * \frac{R_f}{R * 2^n} [B_1 2^{n-1} + B_2 2^{n-2} + B_3 2^{n-3} + \dots + B_n 2^{n-n}]$$

As an example, consider a 2 bit DAC

When input binary sequence B1B2 = 01

$$V_0 = -V_R [B_1 2^{-1} + B_2 2^{-2} + B_3 2^{-3} + B_3 2^{-3} + B_3 2^{-3} + B_3 2^{-3} + B_3 2^{-3} + B_n 2^{-n}]$$

Applying the node analysis at A

$$0 = \frac{V_A + \frac{2}{3} * R}{\frac{2}{3} * R} + \frac{V_A - V_B}{R}$$

$$0 = \frac{V_A + \frac{2}{3} * R}{\frac{2}{3} * R} + \frac{V_A - V_B}{R}$$

$$0 = \frac{V_A + \frac{2}{3} * R + V_A - V_B}{\frac{2}{3} * R}$$

$$V_B = \frac{V_A + \frac{2}{3} * R}{2}$$

Applying the node analysis at B

$$0 = \frac{V_B + \frac{V_B + V_R}{R}}{\frac{2}{3} * R} + \frac{V_B - V_A}{R}$$

$$0 = \frac{V_B + \frac{V_B + V_R}{R} + V_B - V_A}{\frac{2}{3} * R}$$

$$V_A = V_R / 2 + 2V_B$$

Substituting the equation of V_B the above equation, we get

$$V_A = \frac{V_R}{2} + 2V_B$$

$$V_A = \frac{V_R}{4} + 5V_B$$

$$V_A = -\frac{V_R}{8}$$

The output voltage of the complete set up

$$V_0 = -(2R/R) V_A$$

$$V_0 = -(2R/R) (-V_R/8)$$

$$V_0 = V_R/8$$

$$V_{0(fs)} = V_R/2^N; \text{ where } N \text{ is number of bits}$$

$$\text{Resolution} = -V_R(1 - 1/2^N); \text{ where } N \text{ is number of bits}$$

Advantages of R/2R Ladder DAC

- i. Only two resistor values (R and 2R)
- ii. Does not require high precision resistors

Disadvantages of R/2R Ladder DAC

- i. R/2R ladder Dac has a lower conversion speed .

3.6 Specifications of DACs

- i. Resolution
- ii. Speed
- iii. Linearity
- iv. Settling Time
- v. Reference Voltages
- vi. Errors

Resolution

Resolution is the smallest analog increment corresponding to 1 LSB change. An N-bit resolution can provide distinct analog value of 2^N times.

Normally common DAC has a resolution of 8-16 bit.

Resolution = $V_{LSB} = \frac{V_{ref}}{2^N}$: where N= number of bits

Speed

Rate of conversion of a single digital input to its analog equivalent

Conversion rate mainly depends on:

- i. clock speed of input signal
- ii. settling time of converter

While the input changes rapidly, DAC speed must be high.

Linearity

The difference between the actual output and the desired analog over the full range of expected values

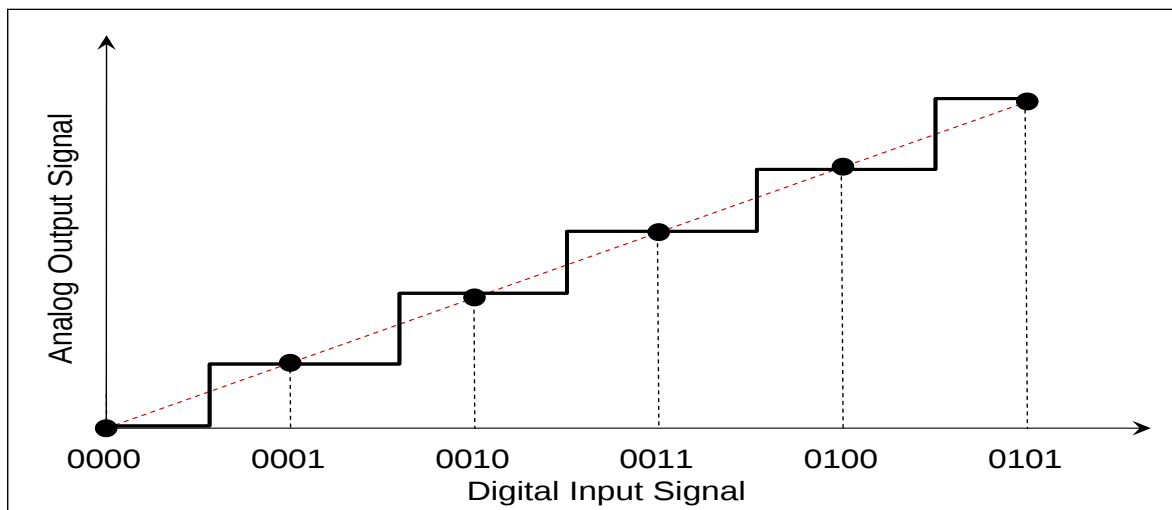


Figure 3.5: Linearity Specifications

Settling Time

Time is needed to settle the output signal within $\pm 1/2$ LSB of its final value after a given change in input scale limited by slew rate of output amplifier. Normally, a change in analog voltage would occur while adding a new binary word into DAC.

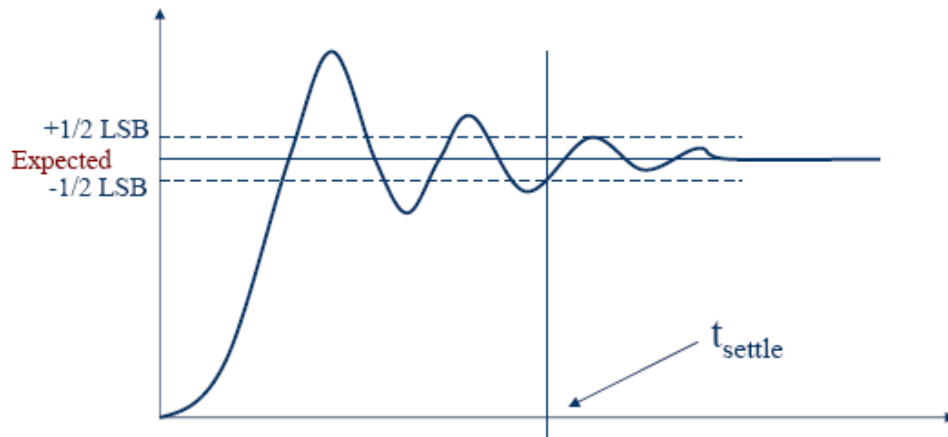


Figure 3.6: Settling Time Specifications of DAC

Reference Voltages

Used for finding out how each digital input will be assigned to each voltage division

Types:

- i. Non-multiplier DAC: fixed V_{ref}
- ii. Multiplier DAC: external source provide V_{ref} .

3.7 DAC Interfacing With Microcontroller

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

In these systems microcontroller generates output which is in digital form but the controlling system requires analog signal.

In the figure shown, we use 8-bit DAC 0808 which converts digital data into equivalent analog current. For converting digital pulses to analog digital to analog converter is as device used widely. For creating DAC two methods are followed as binary weighted and R-2R ladder. DAC 0808 uses the R-2R method since it can achieve a high degree of precision. The first criterion for judging a DAC is its resolution, which is the function of the number of binary inputs. The common ones are 8, 10 and 12 bits. The resolution is decided by the number of data bit inputs since the number of analog output levels is equal to 2^n , where n is the number of data inputs. DAC 0808 provides 256 discrete voltage or current levels of output. In DAC 0808, the digital inputs are converted into current, I_{OUT} and by connecting a resistor to I_{OUT} pin, we convert the result to voltage. The total current provided by I_{OUT} pin is a function of binary numbers at the A1-A6 pins inputs of DAC 0808, where A1 is the LSB, A6 is the MSB for the inputs. The inputs pins of DAC are connected to microcontroller pin of C0-C7 from where user selected the binary inputs. Hence for converting the current into equivalent voltage I to V converter is needed and so we use an op-amp. We connect I_{OUT} to inverting terminal of op-amp. The diagram of microcontroller to DAC connection, followed by connection of op-amp to that of the DAC is as follows:

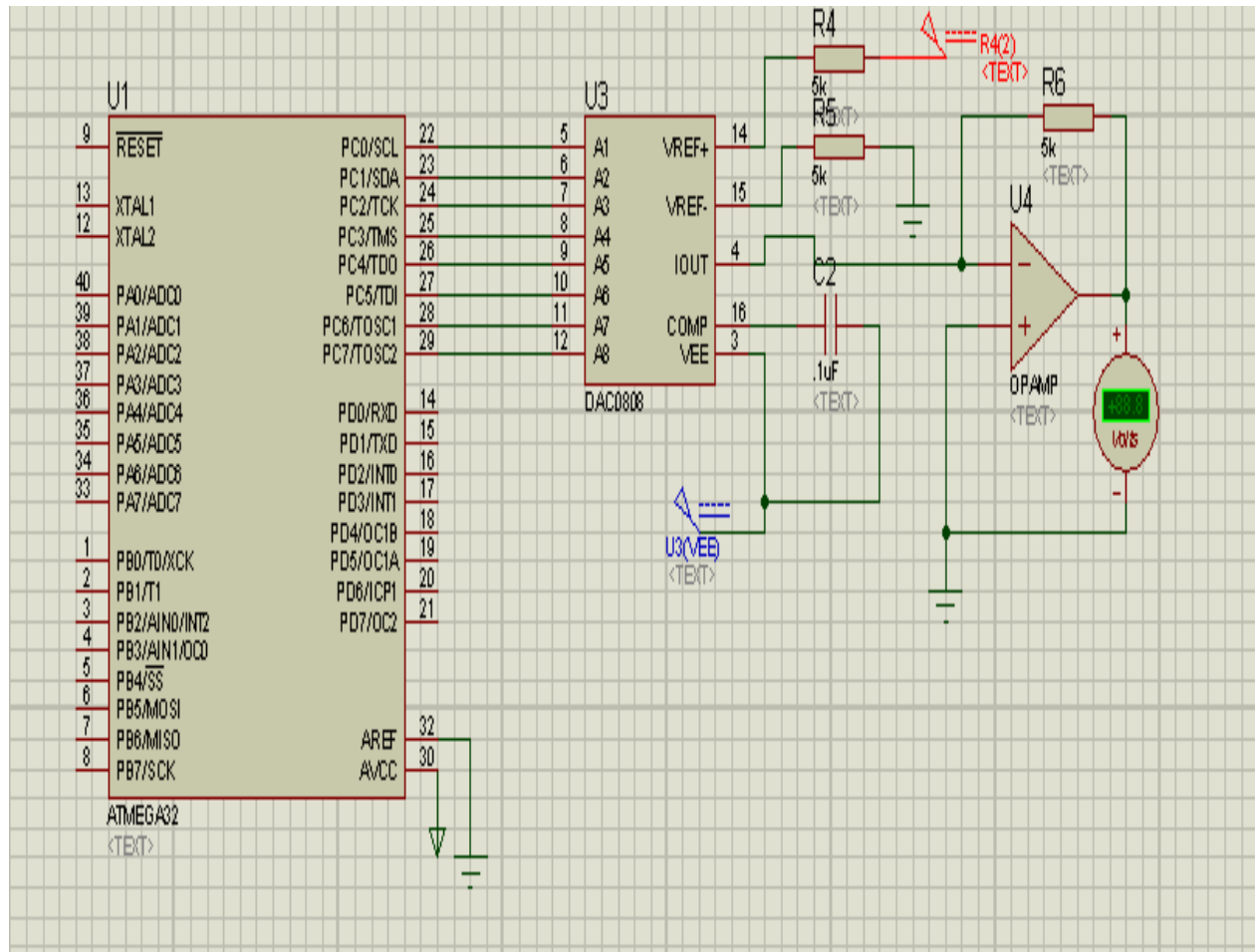


Figure 3.7: DAC interfacing with microcontroller (ATMEGA 32)

According to theory of DAC Equivalent analog output is given as:

$$V_o = V_{ref} \left[\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right]$$

4. 1 COM pin (Pin # 9)

The ULN2003A is a Darlington transistor array which is made of seven NPN Darlington pairs that has a high-voltage outputs with internal suppression diodes which insure from problems associated with inductive loads. The high breakdown voltage and internal suppression diodes capable of inductive loads switching. A single Darlington pair collector-current rating is 500 mA.

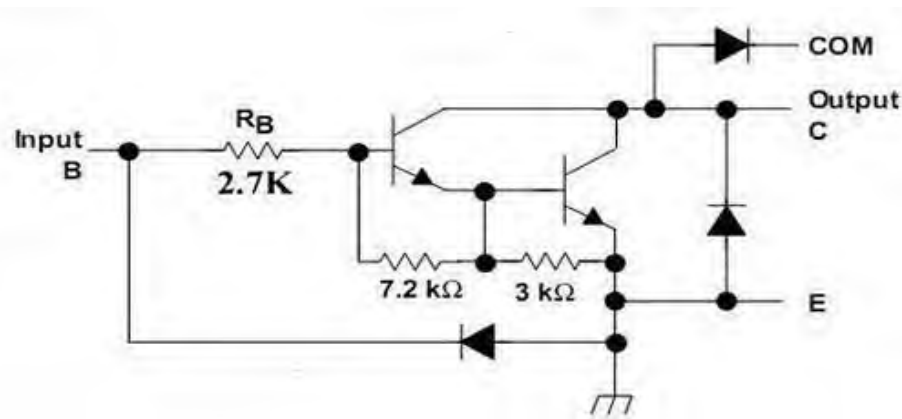


Figure 3.9: Internal circuit diagram of ULN 2003A (Darlington Array)^[37]

In real life, the relay circuit driver has a wide range of applications in real life. Major applications which are associated with ULN-2003A are given below.

- i. Logic buffers.
- ii. Line drivers.
- iii. Relay drivers (for driving different loads).
- iv. Lamp drivers.
- v. LED display drivers (display devices).
- vi. Motor (stepper and DC brushed motor) drivers.

This chapter discusses about the contents of proposed position control system of the unipolar stepper motor. This will give a brief view about the each contents basic concept so that readers can understand the next chapter where implementation of the proposed circuit will be discussed.

Chapter 4

Proposed Circuit Implementation

4.1 Introduction

In order to rotate the unipolar stepper motor to a desired angle, assigned voltages must be applied to one and two appropriate coils out of four coils of the stepper motor. For example, to make the stepper motor to rotate from 0 degree to -56.25 degree for example, then Voltage A1 must be 0.9754V and Voltage A2 must be 4.9V and voltages at coil B1 and coil B2 must be 0 degree.

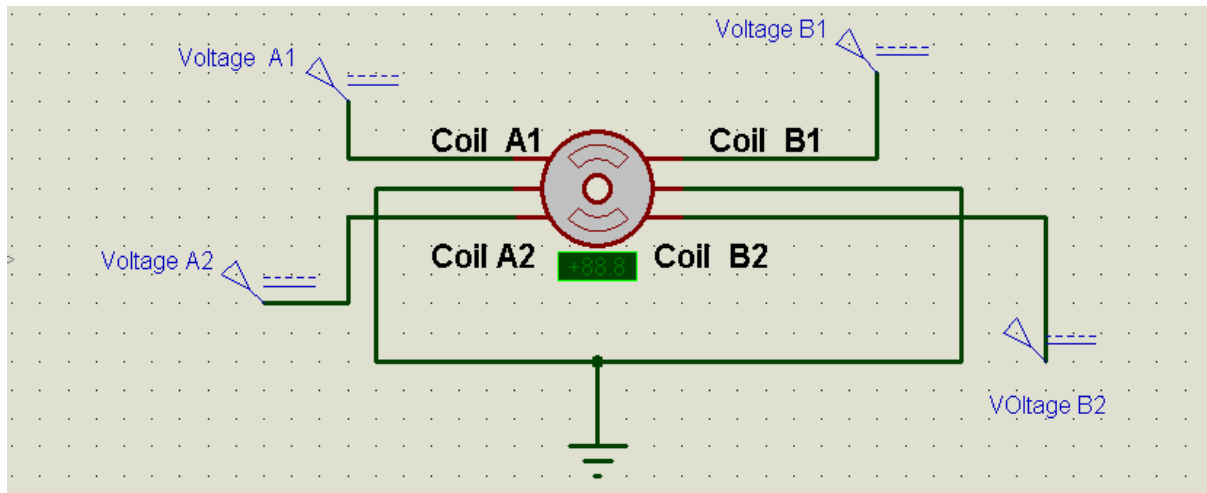


Figure 4.1: Unipolar Stepper Motor in Proteus

The above diagram and arrangement is left redundant if the stepper motor is inclined to move sequentially and automatically in micro-steps. Making the stepper motor to move automatically in micro-steps in clockwise and anticlockwise directions requires clever integration of electrical components with that of the unipolar stepper motor and the code in the Code Vision AVR that are to be hexed.

4.2 Angular Increment

In order to move the stepper motor rotor to a particular angle we follow figure 4.2 co-ordinate calculation. The motor coil pin terminals are divide four parts as A1, A2, B1, and B2. To move the rotor to a particular angular position we energized each coil. The co-ordinates are fixed in such a way so that for every angular position only two coils will be energized and will not be in same coil connection. This process simplified our circuit switching mechanism as now with only four relay we can switch each terminal according to a particular angle.

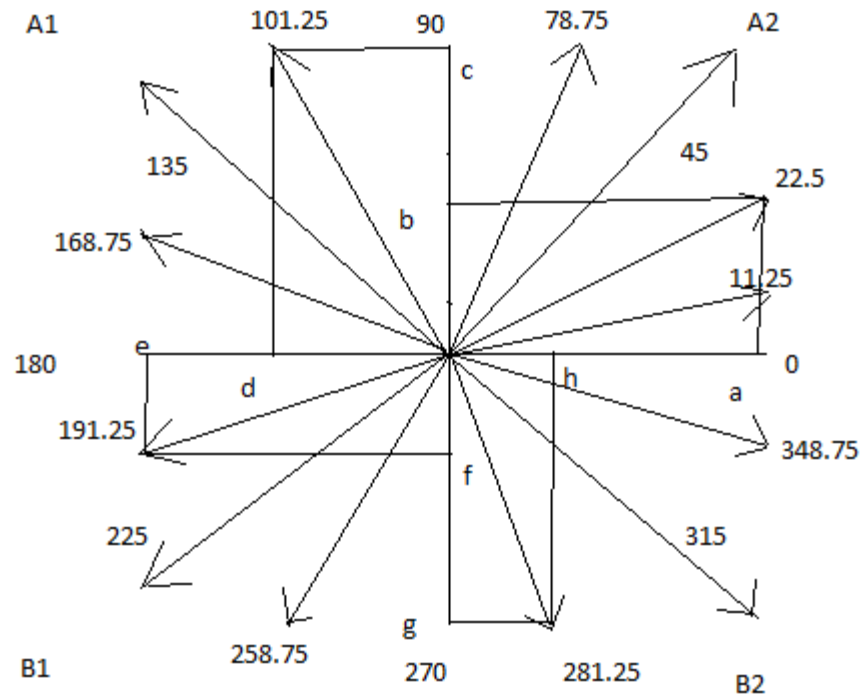


Figure 4.2: Angular Increment for Each Steps

For angular position calculation we follow the sequence from 90° , 45° , 22.5° , 11.25° and so on. Suppose we need to move the rotor to 90° . In order to do that we need to energized fully A1 and A2 to 5v. This creates a magnetic force in two coils such that rotor will rotate to opposite polarity of the polarity generated by this two coils. In proteus simulation the angular movement for both fully energized A1 and A2 coils is shown 90° . For 45° , A2 voltage will be 5v and A1 voltage will be 0v to move. Now when we moved to 45° to 22.5° step calculation, we can see that the priority of A2 and B2 are the highest. So these coils are needed to be energized in order

to get our desired steps. For that we connected a straight line on adjacent which create a triangle and intersect position is a and b.

$$B2 = 5 \cos(90 - \text{desired angle}) \quad \text{Real axis}$$

$$B2 = 5 \sin(90 - \text{desired angle}) \quad \text{Imaginary axis}$$

Following this two formula each angle is calculated accordingly. The detailed calculated values are given in Table 4.1 and Table 4.2. For 11.25° angle rotation, we can see that the calculated angle is not same for a terminal. This happens when rotor initially start voltage requirement 4.9v and 0.9754v do not create enough magnetic force to move the rotor. To do that 3.28v is used replacing 0.9754 only initially not for all value. This phenomenon will happen, when you want to move the rotor from each 0, 90, 180, 270 and 360 degree.

For our proposed system, we showed $\frac{1}{16}$ steps per full rotation of the rotor.

4.2 Simulation Synopsis

In figure 4.3 shows circuit implementation in proteus to make the stepper motor move automatically and sequentially in micro-steps. There are four coils in unipolar stepper motor. Necessary and suitable voltages of assigned magnitudes must be applied to the appropriate and suitable coils of the stepper motor one at a time so as to make the stepper motor move sequentially and automatically in small steps.

This automatic and sequential rotation of stepper motor in micro-steps requires continuous and automatic conversion of each and every digital binary inputs in DAC (Digital to Analog Converter) one at a time to generate the correct and the right voltages one step at a time, which are to be fed to the suitable coils of the stepper motor vie relays so that it rotates in micro-step. That is why, the microcontroller, ATmega32 is connected to DAC, where automatic binary

inputs are fed from PORTC and PORTB of the microcontroller one at a time to the DAC to be converted to analog ones. The whole integration is shown in figure 4.3.

Since, there are four coils in stepper motor, four analog voltage channels are required to feed the coils with assigned voltages one at a time throughout the rotation of the stepper motor. This calls for the need of four DAC. But for the sake of reducing the number of pins in ATmega32 microcontroller, yet serving the same purpose of micro-stepping, two DAC are used that comes with the addition of four extra relays. But how does the relay work and manage the operations?

Two DAC generates two separate voltage channels but the four coils of stepper motor must be fed by four separate voltage channels (one for each). This is where the concept of relay interferes out of the blue. Each DAC has got two relays, where each of the relay is connected to one corresponding coils, making each DAC to produce two separate voltage channels for every two coils via relays as shown in figure 4.3. Relays act as a simple switching mechanism between the DAC and the stepper motor. Relays are programmed to operate through coding by making use of the connection between the relay and the relay driver, ULN2003A and ATmega32 to such an extent that, when a motor wants to rotate from one angle to the other, necessary digital binary inputs are passed into the DAC and the correct analog voltages get produced, which are passed to the appropriate corresponding relays and relays will only be switched on the specific coils where the required analog voltages are needed to energize those coils to make the stepper motor move in micro-steps throughout its clockwise and/or anticlockwise rotation. Once again, figure 4.3 shows the entire connections of ATmega32, two DACs, relay driver, four relays and one unipolar stepper motor, where PORTC of ATmega32 is cascaded with DAC1 which is connected to Relay A1 and Relay B2, which further joins Coil A1 and Coil B2 respectively. PORTA of ATmega32 is cascaded with DAC2 which is connected to Relay A2 and Relay B1, which further joins Coil A2 and Coil B1 respectively.

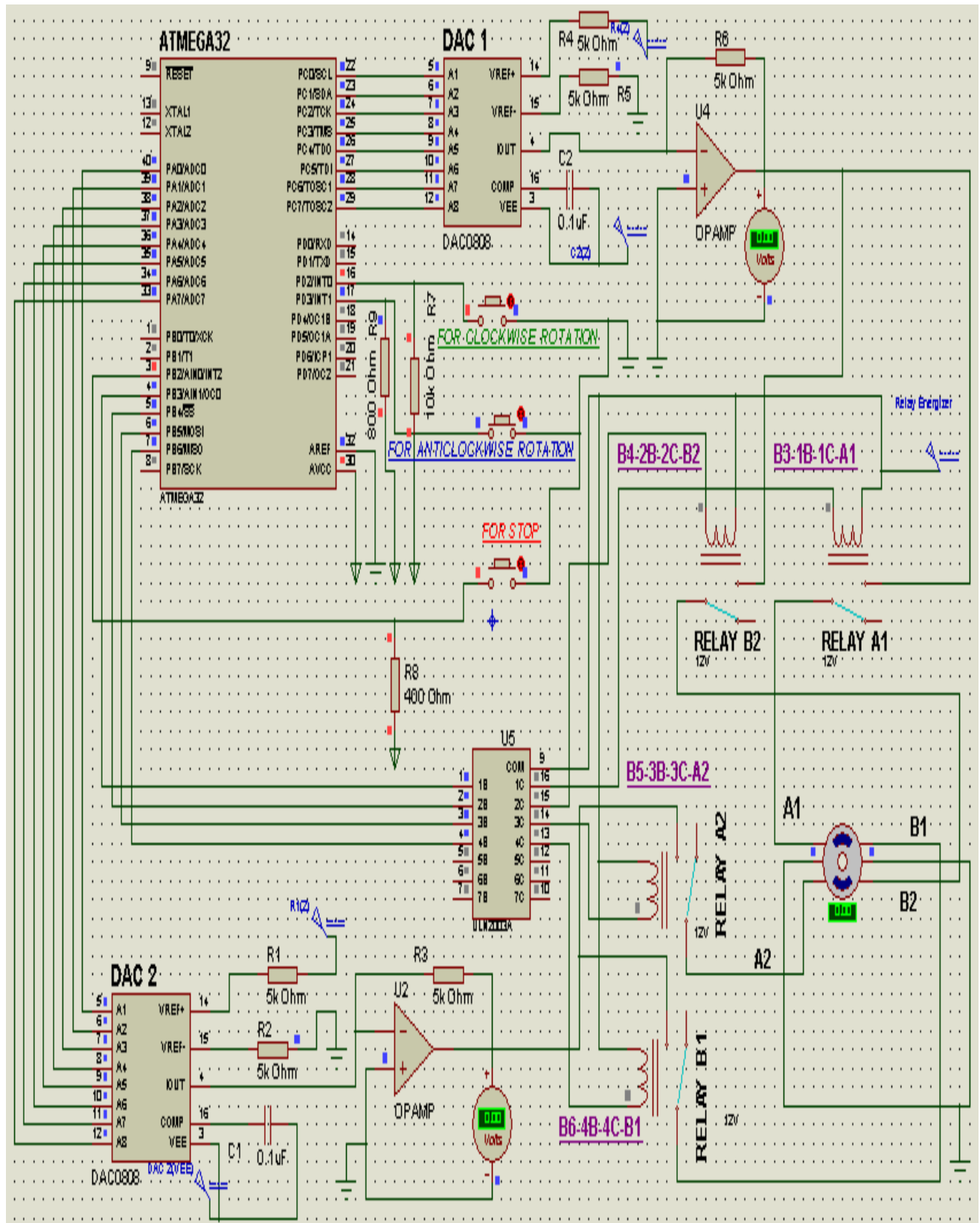


Figure 4.3: Design of the proposed circuit in proteus

For example, if the stepper motor is bound to rotate to -315 degree from any angle, then binary inputs of 11111111 must be applied in DAC1 and respectively, the analog voltage from DAC1 becomes 5V and to pass this voltage to appropriate and suitable coil such as Coil B2 for its -315th degree rotation, Relay B2 must be switched on and Relay A1 and Relay A2 are switched off. If the stepper motor is bound to rotate to +11.25 degree from any angle, then binary inputs in DAC2 must be 11111101 and binary inputs in DAC1 must be 10101000 and the analog voltage from DAC2 and DAC1 becomes 4.9V and 3.28V respectively and to pass these voltages to appropriate and suitable coils such as Coil A2 and Coil B2 for its +11th degree rotation, Relay A2 and Relay B2 must be switched on respectively and other relays are switched off. The same conception applies throughout the motion of the stepper motor.

In figure 4.3, Int0 is used for pending current task of any microcontroller and start clockwise rotation of unipolar stepper motor in micro-step angle of 11.25 degree. Int1 is used for pending current task of any microcontroller and start anticlockwise rotation of unipolar stepper motor in micro-step angle of 11.25 degree. Int2 is used for pending current task of any microcontroller and stop the rotation of unipolar stepper motor.

As shown in the diagram, each of the buttons, connecting to Int0, int1 and Int2 are connected to pull-up resistors of 400 Ohm. If pull-up resistors are not connected to those input pins, then it would be difficult to tell whether the state of the input is high or low. The state of the pin would be referred to as the floating state. In order to prevent this state, pull up resistors are connected to the input pins.

Table 4.1: Binary Inputs that are required to generate an Analog Voltage in DAC and thus, eventually generates theoretical rotational angles in degrees of unipolar stepper motor for micro-stepping for **clockwise rotation**

Inputs DAC1	Inputs DAC2	Outputs DAC1	Outputs DAC2	Voltages				Angle (degree)
				A1	A2	B1	B2	
10101000	11111011	3.28V	4.9V	0V	4.9V	0V	3.28V	+11.25
01100001	11101100	1.91V	4.61V	0V	4.61V	0V	1.91V	+22.5
00110010	11111011	0.9754V	4.9V	0V	4.9V	0V	0.9754V	+33.75
00000000	11111111	0V	5V	0V	5V	0V	0V	+45
00110001	11111011	0.9754V	4.9V	0.9754V	4.9V	0V	0V	+56.25
01100001	11101100	1.91V	4.61V	1.91V	4.61V	0V	0V	+67.5
10101000	11111011	3.28V	4.9V	3.28V	4.9V	0V	0V	+78.75
11111111	11111111	5V	5V	5V	5V	0V	0V	+90
11111011	10101000	4.9V	3.28V	4.9V	3.28V	0V	0V	+101.25
11101100	01100001	4.61V	1.91V	4.61V	1.91V	0V	0V	+112.5
11111011	00110010	4.9V	0.9754V	4.9V	0.9754V	0V	0V	+123.75
11111111	00000000	5V	0V	5V	0V	0V	0V	+135
11111011	00110010	4.9V	0.9754V	4.9V	0V	0.9754V	0V	+146.25
11101100	01100001	4.61V	1.91V	4.61V	0V	1.91V	0V	+157.5
11111011	10101000	4.9V	3.28V	4.9V	0	3.28V	0V	+168.75
11111011	10101000	4.9V	3.28V	4.9V	0	3.28V	0V	+168.75
11111111	11111111	5V	5V	5V	0V	5V	0V	+180

Inputs DAC1	Inputs DAC2	Outputs DAC1	Outputs DAC2	Voltages				Angle (degree)
				A1	A2	B1	B2	
10101000	11111011	3.28V	4.9V	3.28V	0V	4.9V	0V	+191.25
01100001	11101100	1.91V	4.61V	1.91V	0V	4.61V	0V	+202.5
00110010	11111011	0.9754V	4.9V	0.9754V	0V	4.9V	0V	+213.75
00000000	11111111	0V	5V	0V	0V	5V	0V	+225
00110010	11111011	0.9754V	4.9V	0V	0V	4.91V	0.9754V	+236.25
01100001	11101100	1.91V	4.61V	0V	0V	4.61V	1.91V	+247.5
10101000	11111011	3.28V	4.9V	0V	0V	4.9V	3.28V	+258.75
11111111	11111111	5V	5V	0V	0V	5V	5V	+270
11111011	10101000	4.9V	3.28V	0V	0V	3.28V	4.9V	+281.25
11101100	01100001	4.61V	1.91V	0V	0V	1.91V	4.61V	+292.5
11111011	00110010	4.9V	0.9754V	0V	0V	0.9754V	4.9V	+303.75
11111111	00000000	5V	0V	0V	0V	0V	5V	+315
11111011	00110010	4.9V	0.9754V	0V	0.9754V	0V	4.9V	+326.25
11101100	01100001	4.61V	1.91V	0V	1.91V	0V	4.61V	+337.5
11111011	10101000	4.9V	3.28V	0V	3.28V	0V	4.9V	+348.75
11111111	11111111	5V	5V	0V	5V	0V	5V	+360/0

Table 4.2: Binary Inputs that are required to generate an Analog Voltage in DAC and thus, eventually generates rotational angles of unipolar stepper motor for micro-stepping in degrees for **anticlockwise rotation**

Inputs DAC1	Inputs DAC2	Outputs DAC1	Outputs DAC2	Voltages				Angle (degree)
				A1	A2	B1	B2	
11111011	10101000	4.9V	3.28V	0V	3.28V	0V	4.9V	-11.25
11101100	01100001	4.61V	1.91V	0V	1.9V	0V	4.61V	-22.5
11111011	00110010	4.9V	0.9754V	0V	0.9754V	0V	4.9V	-33.75
11111111	00000000	5V	0V	0V	0V	0V	5V	-45
11111011	00110010	4.9V	0.9754V	0V	0V	0.9754V	4.9V	-56.25
11101100	01100001	4.61V	1.91V	0V	0V	1.91V	4.61V	-67.5
11111011	10101000	4.9V	3.28V	0V	0V	3.28V	4.9V	-78.75
11111111	11111111	5V	5V	0V	0V	5V	5V	-90
10101000	11111011	3.28V	4.9V	0V	0V	4.9V	3.28V	-101.25
01100001	11101100	1.91V	4.61V	0V	0V	4.61V	1.91V	-112.5
00110010	11111011	0.9754V	4.9V	0V	0V	4.9V	0.9754V	-123.75
00000000	11111111	0V	5V	5V	0V	5V	0V	-135
00110010	11111011	0.9754V	4.9V	0.9754V	0V	4.9V	0V	-146.25
01100001	11101100	1.91V	4.61V	1.9V	0V	4.61V	0V	-157.5
10101000	11111011	3.28V	4.9V	3.28V	0V	4.9V	0V	-168.75
11111111	11111111	5V	5V	5V	0V	5V	0V	-180
11111011	10101000	4.9V	3.28V	4.9V	0V	3.28V	0V	-191.25

Inputs DAC1	Inputs DAC2	Outputs DAC1	Outputs DAC2	Voltages				Angle (degree)
				A1	A2	B1	B2	
11101100	01100001	4.61V	1.91V	4.61V	0V	1.91V	0V	-202.5
11111011	00110010	4.9V	0.9754V	4.9V	0V	0.9754V	0V	-213.75
11111111	00000000	5V	0V	5V	0V	0V	0V	-225
11111011	00110010	4.9V	0.9754V	4.9V	0.9754V	0V	0V	-236.25
11101100	01100001	4.61V	1.91V	4.61V	1.91V	0V	0V	-247.5
11111011	11101000	4.9V	3.28V	4.9V	3.28V	0V	0V	-258.75
11111111	11111111	5V	5V	5V	5V	0V	0V	-270
10101000	11111011	3.28V	4.9V	3.28V	4.9V	0V	0V	-281.25
01100001	11101100	1.91V	4.61V	1.91V	4.61V	0V	0V	-292.5
00110010	11111011	0.9754V	4.9V	0.9754V	4.9V	0V	0V	-303.75
00000000	11111111	0V	5V	0V	5V	0V	0V	-315
00110010	11111011	0.9754V	4.9V	0V	4.9V	0V	0.9754V	-326.25
01100001	11101100	1.91V	4.61V	0V	4.61V	0V	1.9V	-337.5
10101000	11111011	3.28V	4.9V	0V	4.91V	0V	3.28V	-348.75
11111111	11111111	5V	5V	0V	5V	0V	5V	-360/0

Taking into account of the efficiency of the unipolar stepper motor in stimulation, maximum angular diversions occur when the stepper motor is theoretically supposed to rotate to 202.5th degree when theoretically 4.61 V and 1.91V are applied to Coil A1 and Coil B1 respectively, but in stimulation on Proteus, the stepper motor rotates to 203rd degree. Since, the theoretical step angle of the stepper motor is 11.25 degree, hence the maximum percentage error of stepper motor rotation in software is as follows:-

$$\frac{203-202.5}{11.25} * 100\% = 4.44\%$$

Efficiency=95.56%

In this chapter, we briefly discussed about the use of Atmega32, interrupt and input output function. From microcontroller, we give command to the DAC and through relay we energize different terminal of the motor for a fixed angle. From DAC 1 terminal A1 and B2 is energized and from DAC 2 terminal A2 and B1 is energized. This chapter will give a brief idea about the implementation process of our proposed system circuitry. The code for Atmega32 microcontroller is given to the Appendix I.

Chapter 5

Conclusion

This chapter starts with a brief summary of the work presented in chapter 3 and chapter 4. After that section, a brief discussion on the experiment is given. The final section provides a short view of possible improvements to the method presented in this thesis paper.

5.1 Summary

In our thesis, we mainly analyze on micro-stepping mode of stepper motor. We start with a brief discussion on various types of stepper motor and stepper motor control process. After that we design a circuit for position controlling of a stepper motor in micro-stepping mood by using ATMEGA32, DAC, Relay, Switch and Relay Drive (ULN003A) and later give a short explanation of system component. A brief explanation is given on working circuit mechanism of proposed position control system.

5.2 Discussion

In DAC method, we do not get exact 11 degree variations between the angles. The reason behind that for 4.9v and 5v we use the same hexadecimal code and for value which are more than 3 bit after the decimal, we have used 3 bit approximation. We designed the system in such a way so that after the command using the push button the system can move either clockwise 360 or anti-clockwise 360 degree. For voltage level manipulation, we have used R2R ladder DAC technique.

In PWM method, it is possible to load a certain degree in the motor. When we need to continuously rotate the motor at a certain degree gap like 11 degree, we face a problem. This problem relates to the time period of the generated signal and fixed number of interrupt routine available in the micro controller. After generation of the PWM wave, we can call the ISR for delay and initialization of timers for the next operation. The problem is that in this process we need 32 interrupt routine for 11 degree step up to 360 degree. Another approach can be changing

the value of OCR0 and OCR2A pin directly to generate our desired signal. This this process as we cannot control the time period of the two timers spontaneously our motor toggles and gives us a wrong output.

5.3 Future Scope

Future scope for proposed design:

1. We have conducted all of our simulations considering the ideal environmental peripherals. If some peripherals like temperature, humidity or the source change our calculation and output also changes. A system can be designed considering the changes to tackle the problems and always have our desired output.
2. In industrial condition ,all the line are high voltage lines .while for the technological advancement ,all the systems are modernized like uses of programmable CNC machine is one the rise. Using our concept and if we can reduce the error of the motor rotation, this concept can be used for assembling industrial produce. The two main plus point of this system is this system consumes a very low power and it is easily and continuously programmable. This system also has a great scalability factor which can also reduce the economic cost of establishing different CNC machine for different manufacturing purposes.
3. Adding a keypad and creating a close loop, we can control the quality and precision of our output.

References

- [1] G. Baluta, "Microstepping mode for stepper motor control," in International Symposium on Signals, Circuits and Systems , Iasi, Romania, 2007.
- [2] Sheng-Ming Yang, Feng-Chieh Lin, and Ming-Tsung Chen, "Micro-Stepping Control of a Two-Phase Linear Stepping Motor With Three-Phase VSI Inverter for High-Speed Applications," IEEE transactions on industrial applications, vol. 40, no. 5, September 2004.
- [3] A. Astarloa, U. Bidarte, A. Zuloaga, and I.M. de Alegria, "Reconfigurable microstepping control of stepper motors using fpga embedded ram," in Proceedings of the 29th Annual Conference of the IEEE of the Industrial Electronics Society, Roanoke, VA, USA, 2003, pp. 2221-2226.
- [4] W. McClelland, "The application of the electricity in warships," Journal of the Institution of Electrical Engineers, vol. 65, no. 369, pp. 866 - 869, September 1927.
- [5] Kenjo Takashi, Stepping motors and their microprocessor controls. Kanagawa, Japan: Clarendon Press, 1984.
- [6] C. L. Walker, "Improvements in and connected with electromagnetic step by step signaling and synchronous rotation," 137,150, 1919.
- [7] C. B. and Thain, J. H. Chicken, "Electrical Signaling Apparatus," 1,353,025, 1920.
- [8] K. M. Feiertag and J.T. Donahoo, "Dynamoelectric machine," 2589999, 1952.
- [9] V. Marano and D. Arrigo, "Novel fully-integrated 65W stepper motor driver IC," in International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2006.
- [10] Pu Ji, Xiuhe Wang, Feng Lin, Rong Zhang, and Shiyong Liu, "Application of LMD18245 in two phase hybrid stepping motor driver," in Proceedings of the Eighth International Conference on Electrical Machines and Systems, Nanjing, China, 2005, pp. 706-709.
- [11] National Semiconductor, "LMD18245 3A, 55V DMOS Full-Bridge Motor Driver ," Data Sheet 2006.

- [12] Allegro, "A3981, Automotive, Programmable Stepper Driver," Data Sheet 2010.
- [13] Allegro, "A4980, Automotive, Programmable Stepper Driver," Data Sheet 2010.
- [14] ON Semiconductor, "AMIS-30523 CAN micro-stepping motor driver," Data Sheet.
- [15] Peter Cox and Guido Remmerie, "Driver ICs elevate design of stepper-motor control," Power Electronics Technology, August 2007.
- [16] Mongkol Konghirun and Chawakorn Yodsanti, "Microstepping Bipolar Drive of Two-Phase Hybrid Stepping Motor on TMS320F2808 DSC," King Mongkut's University of Technology, Thonburi, Application Report 2008.
- [17] Ngoc Quy Le and Jae Wook Jeon, "An open-loop stepper motor driver based on fpga," in Proceedings of the International Conference on Control, Automation and Systems, Seoul, Korea, 2007, pp. 1322-1326.
- [18] Xiaodong Zhang, Junjun He, and Chunlei Sheng, "An approach of micro-stepping control for the step motors based on fpga," in Proceedings of the IEEE International Conference on Industrial Technology, Hong Kong, PRC, 2005, pp. 125-130.
- [19] Sorin Manea, "Stepper Motor Control with dsPIC® DSCs," Microchip Technology Inc., Application Note 2009.
- [20] Ran Zhang, Xiuhe Wang, Yubo Yang, and Dongwei Qiao, "Design of two-phase hybrid stepping motor driver with current closed-loop control based on PIC18F2331," in Proceedings of the International Conference on Electrical Machines and Systems, Wuhan, PRC, 2009, pp. 2998-3001.
- [21] International Rectifier, "IR2110 High and low side driver," Data Sheet 2005.
- [22] Microchip Technology Inc, "PIC18F2331/2431/4331/4431 Data Sheet," 2003.
- [23] T.S. Weerakoon and L. Samaranayake, "Development of a Novel Drive Topology for a Five Phase Stepper Motor," in IEEE Region 10 Colloquium and the Third ICIS, 2008.
- [24] Alexandru Morar, "Stepper motor model for dynamic simulation," Achta Electronica, vol. 44, no. 2, pp. 117-122, 2003.

- [25] M. Zribi and J. Chiasson, "Position Control of a PM Stepper Motor by Exact Linearization," IEEE transactions on automatic control, vol. 36, no. 5, pp. 620-625, May 1991.
- [26] R.H. Brown and K. Srinivas, "A damping circuit for chopper driven bifilar hybrid stepper motors," in Proceedings of the Power Electronics Specialists Conference, Milwaukee, WI , USA, 1989, pp. 446-451.
- [27] Condit R, Microchip Technology Inc.: Brushed DC motor fundamentals.
<http://www.mne.psu.edu/sommer/me445/AN905a.pdf> (accessed 6/11/2008).
- [28] National Instruments Corporation: Configuring a motion control system.
<http://zone.ni.com/devzone/cda/tut/p/id/3126#toc0>. (accessed 3/2/2008).
- [29] Christopher KT: Modern control technology - components and systems, 2nd ed, Delmar Thomson Learning, Novato, CA USA, 2001.
- [30] Variable Reluctance Stepper Motor. (n.d.). Retrieved April 15, 2018, from <https://circuitglobe.com/variable-reluctance-stepper-motor.html>
- [31] Permanent Magnet Stepper Motor. (n.d.). Retrieved April 15, 2018, from <https://circuitglobe.com/permanent-magnet-stepper-motor.html>
- [32] Hybrid Stepper Motor. (n.d.). Retrieved April 15, 2018, from <https://circuitglobe.com/hybrid-stepper-motor.html>
- [33] M., Eidiani. (n.d.). What is the difference between Bipolar motor and Unipolar motor? Retrieved April 15, 2018, from https://www.researchgate.net/post/What_is_the_difference_between_Bipolar_motor_and_Unipolar_motor
- [34] G. (2013, September 11). Stepper motors and drives, what is full step, half step and microstepping? Retrieved April 15, 2018, from <https://www.rs-online.com/designspark/stepper-motors-and-drives-what-is-full-step-half-step-and-microstepping>
- [35] ATmega32-8 Bit AVR MicroController. (n.d.). Retrieved April 15, 2018, from <https://www.javatpoint.com/atmega32-avr-microcontroller>

[36] Digital to Analog Converter (DAC). (n.d.). Retrieved April 15, 2018, from <http://microchipdeveloper.com/mcu1101:project-11>

[37] L., Loflin. (n.d.). How to Use the ULN2003A Darlington transistor Array with Examples. Retrieved April 15, 2018, from <http://www.bristolwatch.com/ele/uln2003a.html>

Appendix I

The following program represents the code in code vision AVR

```
#include <mega32.h>
#include <delay.h>
#include <stdlib.h>

interrupt [EXT_INT0] void ext_int0_isr(void)
{
    while(1) //For infinite microstepping clockwise rotation of stepper motor using INT0 unless
    INT0 is interrupted by INT2 to make the stepper motor stop or by INT1 for microstepping
    anticlockwise rotation of the stepper motor
    {
        #asm("sei")
        //for 0 degree
        PORTB.3=0;
        PORTB.6=0;
        PORTB.4=1;
        PORTB.5=1;
        PORTA=0xFF;
        PORTC=0xFF;
        delay_ms(2000);
        //for +11.25 degree
        PORTB.4=1;
        PORTB.5=1;
        PORTA=0xDF;
        PORTC=0x15;
        delay_ms(2000);
        //for +22.50 degree
        PORTB.4=1;
```

```

PORTB.5=1;
PORTA=0x37;
PORTC=0x86;
delay_ms(2000);
//for +33.75 degree
PORTB.4=1;
PORTB.5=1;
PORTA=0xDF;
PORTC=0x4C;
delay_ms(2000);
//for +45.00 degree
PORTB.4=1;
PORTB.5=1;
PORTA=0xFF;
PORTC=0x0;
delay_ms(2000);
//for +56.25 degree
PORTB.4=0;
PORTB.3=1;
PORTB.5=1;
PORTA=0xDF;
PORTC=0x4C;
delay_ms(2000);
//for +67.50 degree
PORTB.3=1;
PORTB.5=1;
PORTA=0x37;
PORTC=0x86;
delay_ms(2000);
//for +78.75 degree
PORTB.3=1;

```

```

PORTB.5=1;
PORTA=0xDF;
PORTC=0x15;
delay_ms(2000);
//for +90 degree
PORTB.3=1;
PORTB.5=1;
PORTA=0xFF;
PORTC=0xFF;
delay_ms(2000);
//for +101.25 degree
PORTB.3=1;
PORTB.5=1;
PORTA=0x15;
PORTC=0xDF;
delay_ms(2000);
//for +112.5 degree
PORTB.3=1;
PORTB.5=1;
PORTA=0x86;
PORTC=0x37;
delay_ms(2000);
//for +123.75 degree
PORTB.3=1;
PORTB.5=1;
PORTA=0x4C;
PORTC=0xDF;
delay_ms(2000);
//for +135 degree
PORTB.5=1;
PORTB.3=1;

```

```

PORTA=0x00;
PORTC=0xFF;
delay_ms(2000);
//for +146.25 degree
PORTB.5=0;
PORTB.4=0;
PORTB.3=1;
PORTB.6=1;
PORTA=0x4C;
PORTC=0xDF;
delay_ms(2000);
//for +157.5 degree
PORTB.3=1;
PORTB.6=1;
PORTA=0x86;
PORTC=0x37;
delay_ms(2000);
//for +168.75 degree
PORTB.3=1;
PORTB.6=1;
PORTA=0x15;
PORTC=0xDF;
delay_ms(2000);
//for +180 degree
PORTB.3=1;
PORTB.6=1;
PORTA=0xFF;
PORTC=0xFF;
delay_ms(2000);
//for +191.25 degree
PORTB.3=1;

```

```

PORTB.6=1;
PORTA=0xDF;
PORTC=0x15;
delay_ms(2000);
//for +202.5 degree
PORTB.3=1;
PORTB.6=1;
PORTA=0x37;
PORTC=0x86;
delay_ms(2000);
//for +213.75 degree
PORTB.3=1;
PORTB.6=1;
PORTA=0xDF;
PORTC=0x4C;
delay_ms(2000);
//for +225 degree
PORTB.3=0;
PORTB.4=1;
PORTB.6=1;
PORTA=0xFF;
PORTC=0x00;
delay_ms(2000);
//for +236.25 degree
PORTB.4=1;
PORTB.6=1;
PORTA=0xDF;
PORTC=0x4C;
delay_ms(2000);
//for +247.5 degree
PORTB.4=1;

```

```

PORTB.6=1;
PORTA=0x37;
PORTC=0x86;
delay_ms(2000);
//for +258.75 degree
PORTB.4=1;
PORTB.6=1;
PORTA=0xDF;
PORTC=0x15;
delay_ms(2000);
//for +270 degree
PORTB.4=1;
PORTB.6=1;
PORTA=0xFF;
PORTC=0xFF;
delay_ms(2000);
//for +281.25 degree
PORTB.4=1;
PORTB.6=1;
PORTA=0x15;
PORTC=0xDF;
delay_ms(2000);
//for +292.5 degree
PORTB.4=1;
PORTB.6=1;
PORTA=0x86;
PORTC=0x37;
delay_ms(2000);
//for +303.75 degree
PORTB.4=1;
PORTB.6=1;

```

```

    PORTA=0x4C;
    PORTC=0xDF;
    delay_ms(2000);
    //for +315 degree
    PORTB.4=1;
    PORTB.6=1;
    PORTA=0x00;
    PORTC=0xFF;
    delay_ms(2000);
    //for +326.25 degree
    PORTB.6=0;
    PORTB.4=1;
    PORTB.5=1;
    PORTA=0x4C;
    PORTC=0xDF;
    delay_ms(2000);
    //for +337.5 degree
    PORTB.4=1;
    PORTB.5=1;
    PORTA=0x86;
    PORTC=0x37;
    delay_ms(2000);
    //for +348.75 degree
    PORTB.4=1;
    PORTB.5=1;
    PORTA=0x15;
    PORTC=0xDF;
    delay_ms(2000);
    continue;
}
}

```

```

interrupt [EXT_INT1] void ext_int1_isr(void)
{
    while(1)//For infinite microstepping anticlockwise rotation of stepper motor using INT1
    unless INT1 is interrupted by INT2 to make it stop or by INT0 for microstepping clockwise
    rotation of the stepper motor
    {
        #asm("sei")
        //for 0 degree
        PORTB.3=0;
        PORTB.6=0;
        PORTB.4=1;
        PORTB.5=1;
        PORTC=0xFF;
        PORTA=0xFF;
        delay_ms(2000);
        //for -11.25 degree
        PORTB.4=1;
        PORTB.5=1;
        PORTC=0xDF;
        PORTA=0x15;
        delay_ms(2000);
        //for -22.50 degree
        PORTB.4=1;
        PORTB.5=1;
        PORTC=0x37;
        PORTA=0x86;
        delay_ms(2000);
        //for -33.75 degree
        PORTB.4=1;
        PORTB.5=1;
    }
}

```

```

PORTC=0xDF;
PORTA=0x4C;
delay_ms(2000);
//for -45.00 degree
PORTB.4=1;
PORTB.5=1;
PORTC=0xFF;
PORTA=0x0;
delay_ms(2000);
//for -56.25 degree
PORTB.5=0;
PORTB.4=1;
PORTB.6=1;
PORTC=0xDF;
PORTA=0x4C;
delay_ms(2000);
//for -67.50 degree
PORTB.4=1;
PORTB.6=1;
PORTC=0x37;
PORTA=0x86;
delay_ms(2000);
//for -78.75 degree
PORTB.4=1;
PORTB.6=1;
PORTC=0xDF;
PORTA=0x15;
delay_ms(2000);
//for -90 degree
PORTB.4=1;
PORTB.6=1;

```

```

PORTC=0xFF;
PORTA=0xFF;
delay_ms(2000);
//for -101.25 degree
PORTB.4=1;
PORTB.6=1;
PORTC=0x15;
PORTA=0xDF;
delay_ms(2000);
//for -112.5 degree
PORTB.4=1;
PORTB.6=1;
PORTC=0x86;
PORTA=0x37;
delay_ms(2000);
//for -123.75 degree
PORTB.4=1;
PORTB.6=1;
PORTC=0x4C;
PORTA=0xDF;
delay_ms(2000);
//for -135 degree
PORTB.4=1;
PORTB.6=1;
PORTC=0x00;
PORTA=0xFF;
delay_ms(2000);
//for -146.25 degree
PORTB.4=0;
PORTB.3=1;
PORTB.6=1;

```

```

PORTC=0x4C;
PORTA=0xDF;
delay_ms(2000);
//for -157.5 degree
PORTB.3=1;
PORTB.6=1;
PORTC=0x86;
PORTA=0x37;
delay_ms(2000);
//for -168.75 degree
PORTB.3=1;
PORTB.6=1;
PORTC=0x15;
PORTA=0xDF;
delay_ms(2000);
//for -180 degree
PORTB.3=1;
PORTB.6=1;
PORTC=0xFF;
PORTA=0xFF;
delay_ms(2000);
//for -191.25 degree
PORTB.3=1;
PORTB.6=1;
PORTC=0xDF;
PORTA=0x15;
delay_ms(2000);
//for -202.5 degree
PORTB.3=1;
PORTB.6=1;
PORTC=0x37;

```

```

PORTA=0x86;
delay_ms(2000);
//for -213.75 degree
PORTB.3=1;
PORTB.6=1;
PORTC=0xDF;
PORTA=0x4C;
delay_ms(2000);
//for -225 degree
PORTB.3=1;
PORTB.6=1;
PORTC=0xFF;
PORTA=0x00;
delay_ms(2000);
//for -236.25 degree
PORTB.6=0;
PORTB.3=1;
PORTB.5=1;
PORTC=0xDF;
PORTA=0x4C;
delay_ms(2000);
//for -247.5 degree
PORTB.3=1;
PORTB.5=1;
PORTC=0x37;
PORTA=0x86;
delay_ms(2000);
//for -258.75 degree
PORTB.3=1;
PORTB.5=1;
PORTC=0xDF;

```

```
PORTA=0x15;
delay_ms(2000);
//for -270 degree
PORTB.3=1;
PORTB.5=1;
PORTC=0xFF;
PORTA=0xFF;
delay_ms(2000);
//for -281.25 degree
PORTB.3=1;
PORTB.5=1;
PORTC=0x15;
PORTA=0xDF;
delay_ms(2000);
//for -292.5 degree
PORTB.3=1;
PORTB.5=1;
PORTC=0x86;
PORTA=0x37;
delay_ms(2000);
//for -303.75 degree
PORTB.3=1;
PORTB.5=1;
PORTC=0x4C;
PORTA=0xDF;
delay_ms(2000);
//for -315 degree
PORTB.3=1;
PORTB.5=1;
PORTC=0x00;
PORTA=0xFF;
```

```

    delay_ms(2000);
    //for -326.25 degree
    PORTB.3=0;
    PORTB.4=1;
    PORTB.5=1;
    PORTC=0x4C;
    PORTA=0xDF;
    delay_ms(2000);
    //for -337.5 degree
    PORTB.4=1;
    PORTB.5=1;
    PORTC=0x86;
    PORTA=0x37;
    delay_ms(2000);
    //for -348.75 degree
    PORTB.4=1;
    PORTB.5=1;
    PORTC=0x15;
    PORTA=0xDF;
    delay_ms(2000);
    continue;
}
}

```

```

interrupt [EXT_INT2] void ext_int2_isr(void) // For making the stepper stop from roating from
that postion where INT2 is applied unles INT2 is interrupted by INT1 and INT2
{
while(1)
{
#asm("sei")
    PORTB.3=0;

```

```

        PORTB.4=0;
        PORTB.5=0;
        PORTB.6=0;
    }
}
void main(void)
{
    PORTA=0x00;
    DDRA=0xFF;
    PORTB=0x00;
    DDRB=0x78;
    PORTC=0x00;
    DDRC=0xFF;
    PORTD=0x00;
    DDRD=0x00;
    // External Interrupt(s) initialization
    // INT0: On
    // INT0 Mode: Falling Edge
    // INT1: On
    // INT1 Mode: Falling Edge
    // INT2: On
    // INT2 Mode: Falling Edge
    GICR|=0xE0;
    MCUCR=0x0A;
    MCUCSR=0x00;
    GIFR=0xE0;
    #asm("sei")
    while (1)
    {
    }
}

```

