NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

STEP MOTOR CONTROL

Graduation Project
EE – 400

Student : Boran Tolga Demir (980568)

Supervisor : Mr. Özgür Cemal Özerdem

Lefkoşa - 2002

ACKNOWLEDGEMENT

First of all I would like to thank to Özgür Cemal Özerdem who was supervisor of my project and also all other teachers With their endless knowledge I easily overcome many difficulties and learn a lot of things about Step Motor Driving systems preparing this project is anise experience of my life

Also I would like to thank to all my family, my friends, my housemate, and all my instructors because they never leave me alone and try to help me during my education without their encouragements I would not be where I am now.

Finally I want to thank my family, especially my parents. Without their endless support and love for me, I would never achieve my current position

ABSTRACT

Graduation project is devoted to the investigation of stepper motor control driven by a peripheral device. The basics of stepper motor control, type of stepper motors, advantages, disadvantages, problems, drive cricuits, type of drive cricuits and PIC circuit are considered. The structure of stepper motors are given.The types of the stepper motors is given. However the characteristics and the parameters are considered in first chapter. In the second chapter we investigate the drive technologies of the stepper motors. And the charesteristics and its arguments are trying to be given. In the third chapter PIC control unit is considered.

The characteristics of stepper motor is investigated and drive technologies of stepper motor is also investigated. Therefore PIC cricuit is selected to control the stepper motor unit. As the PIC circuit is easy to find, cheap and stable for the control applications. And also easy to program the running code. In the last chapter an application of a stepper motor control by a PIC circuit could be seen.

INTRODUCTION

This project is aimed to show the principle techniques of controlling driving a stepper motor by a Peripheral Interface Controller called shortly PIC. It is well common circuit manufactured by Microchip. We are facing with stepper motors usually in control field and automation. It has advantages better than brushes motors as the controlling of stepper motor is more flexible. We have some parameters to manage the behaviors of this kind of motors. The structure of the stepper motor generally consists of four coils and one shaft. Therefore it can be runned step bye step driven by a controller circuit. The brush motors can be used but to fully control of these motors, highly complex mechanical or electronically driver equipments are necessary to use. So that using stepper motors is more intelligent as u can freely and fully control of its behaviors. There are kinds of stepper motors depends on your needs. There are also different kinds of motor drive techniques.

The PIC is circuit has general application areas. You can use it as an adder, a complex input output device or a processing unit. It has a kernel, registers, data memory block, SFR (Special Functional Registers), Input and Output bi directional ports, Serial Interface. A/D converter. Huge programmable area. It can be run from 4MHz to 20MHz. It can be serially programmable. It needs a basic circuit and a software to program. There are so many resources about it. In this project we used a PIC16F84A-04P (4-10Mhz) and a stepper motor NMB 055L-048 (Permanent Magnet) . I want to show how simple and cheap a stepper motor control. It is very useable, flexible and adaptable to the needs of the designer.

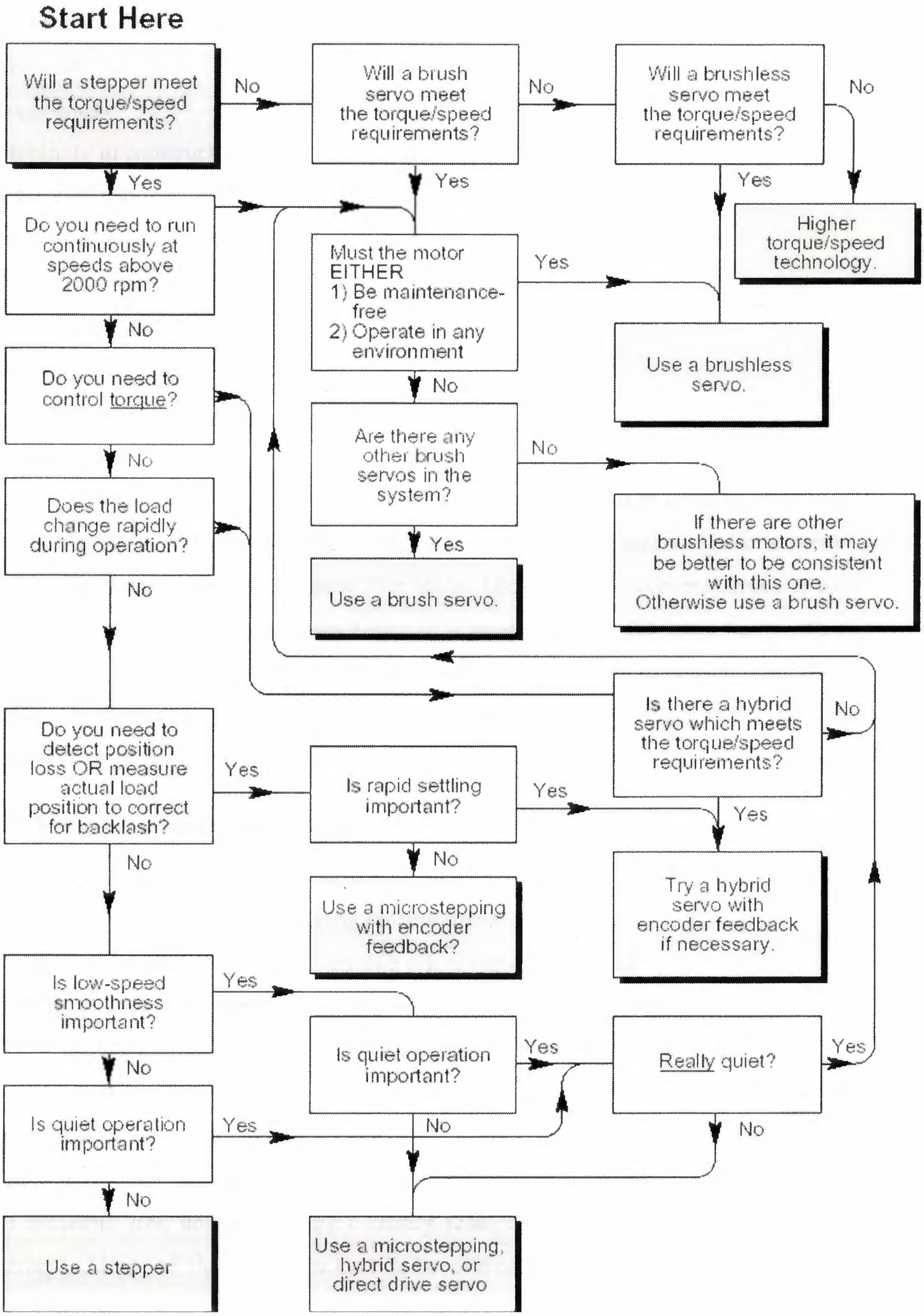
For the stepper motor I wrote the PIC program and i compiled it by using MPLAB from Microchip. It generated the HEX file. Then i upload the binary code to the eeprom of the chip by an anonymous program called PICPROG2. I placed the chip onto my test board and it is executed.

1. STEPPER MOTOR CONTROL

1.1. Step Motor Technologies

1.1.1. Application Areas of Motor Types

The following section gives some idea of the applications that are particularly appropriate for each motor type, together with certain applications that are best avoided. It should be stressed that there is a wide range of applications that can be equally well met by more than one motor type, and the choice will tend to be dictated by customer preference, previous experience or compatibility with existing equipment. Cost-conscious applications will always be worth attempting with a stepper, as it will generally be hard to beat the stepper's price. This is particularly true when the dynamic requirements are not severe, such as "setting" type applications like positioning a guillotine back-stop or a print roller. High-torque, low-speed, continuous-duty applications are also appropriate for step motors. At low speeds, it is very efficient in terms of torque output relative to both size and input power. Microstepping can improve lowspeed applications such as a metering pump drive for very accurate flow control. High-torque, high-speed, continuous-duty applications suit the servo motor, and in fact, a step motor should be avoided in such applications because the high-speed losses can cause excessive motor heating. A DC motor can deliver greater continuous shaft power at high speeds than a stepper of the same frame size. Short, rapid, repetitive moves are the natural domain of steppers or hybrid servos due to their high torque at low speeds, good torque-toinertia ratio and lack of commutation problems. The brushes of the DC motor can limit its potential for frequent starts, stops and direction changes. Low-friction, mainly inertial loads can be efficiently handled by the DC servo provided the start/stop duty requirements are not excessive. This type of load requires a high ratio of peak to continuous torque and in this respect the servo motor excels. Very arduous applications with a high dynamic duty cycle or requiring very high speeds may require a brushless motor. This solution may also be dictated when maintenance-free operation is necessary. Low-speed, high-smoothness applications are appropriate for microstepping or direct drive servos. Applications in hazardous environments or in a vacuum may not be able to use a brush motor. Either a stepper or a brushless motor is called for, depending on the demands of the load. Bear in mind that heat dissipation may be a problem in a vacuum when the loads are excessive.



1.1.2. Stepper Motor Benefits

Stepper motors have the following benefits:

- Low cost
- Ruggedness
- Simplicity in construction
- High reliability
- No maintenance
- Wide acceptance
- No tweaking to stabilize
- No feedback components are needed
- They work in just about any environment
- Inherently more failsafe than servo motors.

There is virtually no conceivable failure within the stepper drive module that could cause the motor to run away. Stepper motors are simple to drive and control in an open-loop configuration. They only require four leads. They provide excellent torque at low speeds, up to 5 times the continuous torque of a brush motor of the same frame size or double the torque of the equivalent brushless motor. This often eliminates the need for a gearbox. A stepper-driven system is inherently stiff, with known limits to the dynamic position error .

1.1.3. Stepper Motor Disadvantages

Stepper motors have the following disadvantages:

- Resonance effects and relatively long settling times
- Rough performance at low speed unless a microstep drive is used
- Liability to undetected position loss as a result of operating open-loop
- They consume current regardless of load conditions and therefore tend to run hot
- Losses at speed are relatively high and can cause excessive heating, and they are frequently noisy (especially at high speeds).
- They can exhibit lag-lead oscillation, which is difficult to damp. There is a limit to their available size, and positioning accuracy relies on the mechanics (e.g., ballscrew accuracy). Many of these drawbacks can be overcome by the use of a closed-loop control scheme.

1.2. There are three main stepper motor types:

- Permanent Magnet (P.M.) Motors
- Variable Reluctance (V.R.) Motors
- Hybrid Motors

1.2.1. Permanent Magnet (P.M.) Motors.

The tin-can or "canstack" motor shown in Fig. 1.1 is perhaps the most widely-used type in non-industrial applications. It is essentially a low-cost, low-torque, low-speed device ideally suited to applications in fields such as computer peripherals. The motor construction results in relatively large step angles, but their overall simplicity lends itself to economic high-volume production at very low cost. The axialair gap or disc motor is a variant of the permanent magnet design which achieves higher performance, largely because of its very low rotor inertia. However this does restrict the applications of the motor to those involving little inertia. (e.g., positioning the print wheel in a daisy-wheel printer).

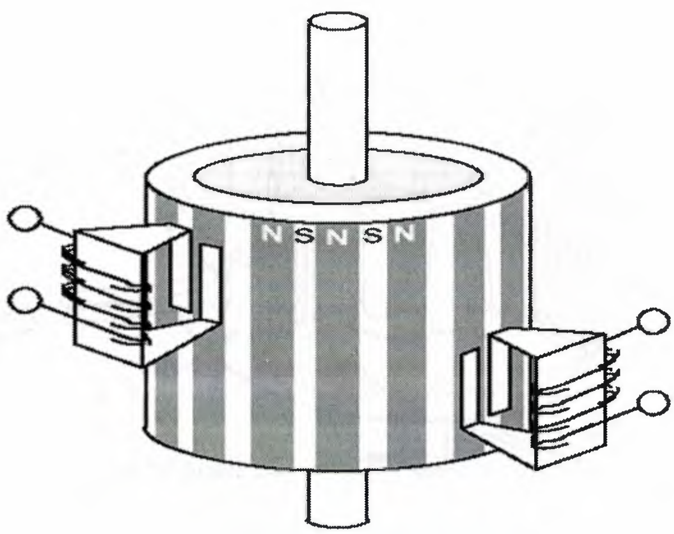


Fig. 1.1 "Canstack" or permanent magnet motor

1.2.2. Variable Reluctance (V.R.) Motors.

There is no permanent magnet in a V.R. motor, so the rotor spins freely without "detent" torque. Torque output for a given frame size is restricted, although the torque-to-inertia ratio is good, and this type of motor is frequently used in small sizes for applications such as micro-positioning tables. V.R. motors are seldom used in industrial applications (having no permanent magnet). They are not sensitive to current polarity and require a different driving arrangement than the other motor types.

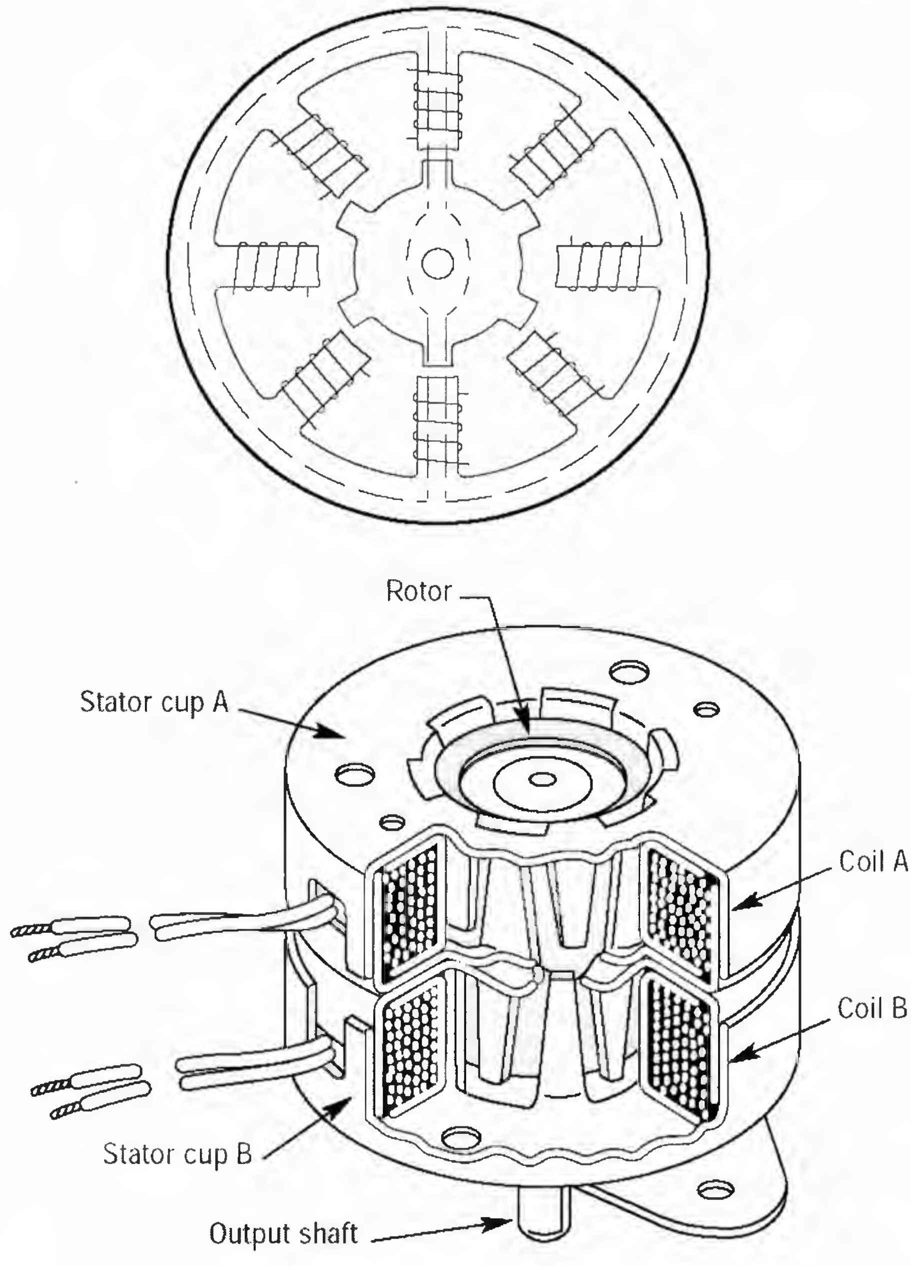


Fig. 1.2 Variable reluctance motor

1.2.3. Hybrid Motors.

The hybrid motor shown in Fig. 1.3 is by far the most widely-used stepper motor in industrial applications. The name is derived from the fact that it combines the operating principles of the other two motor types (P.M. & V.R.). Most hybrid motors are 2-phase, although 5-phase versions are available. A recent development is the "enhanced hybrid" motor, which uses flux-focusing magnets to give a significant improvement in performance, albeit at extra cost. The rotor of this machine consists of two pole pieces with three teeth on each. In between the pole pieces is a permanent magnet that is magnetized along the axis of the rotor, making one end a north pole and the other a south pole. The teeth are offset at the north and south ends as shown in the diagram. The stator consists of a shell having four teeth that run the full length of the rotor. Coils are wound on the stator teeth and are connected together in pairs. With no current flowing in any of the motor windings, the rotor will take one of the positions shown in the diagrams. This is because the permanent magnet in the rotor is trying to minimize the reluctance (or "magnetic resistance") of the flux path from one end to the other. This will occur when a pair of north and south pole rotor teeth are aligned with two of the stator poles. The torque tending to hold the rotor in one of these positions is usually small and is called the "detent torque". The motor shown will have 12 possible detent positions. If current is now passed through one pair of stator windings, as shown in Fig. 1.5(a), the resulting north and south stator poles will attract teeth of the opposite polarity on each end of the rotor. There are now only three stable positions for the rotor, the same as the number of rotor teeth.

The torque required to deflect the rotor from its stable position is now much greater, and is referred to as the " holding torque "

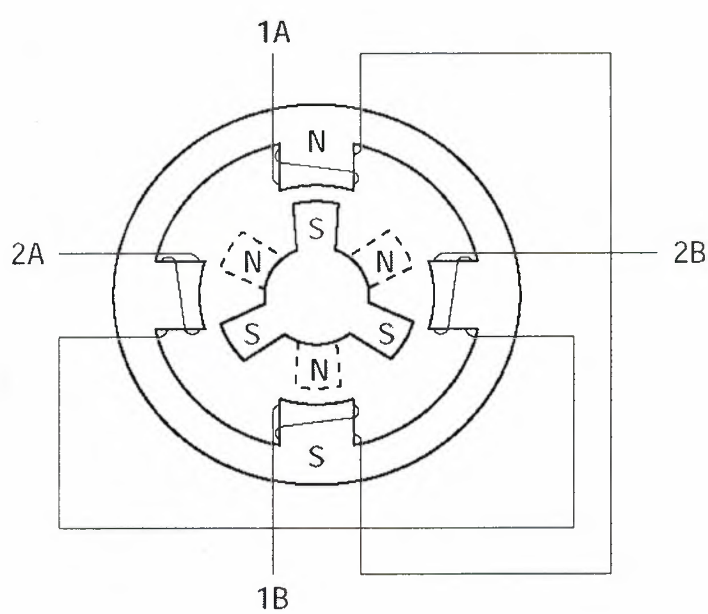


Fig. 1.4 12 step / rev hybrid motor

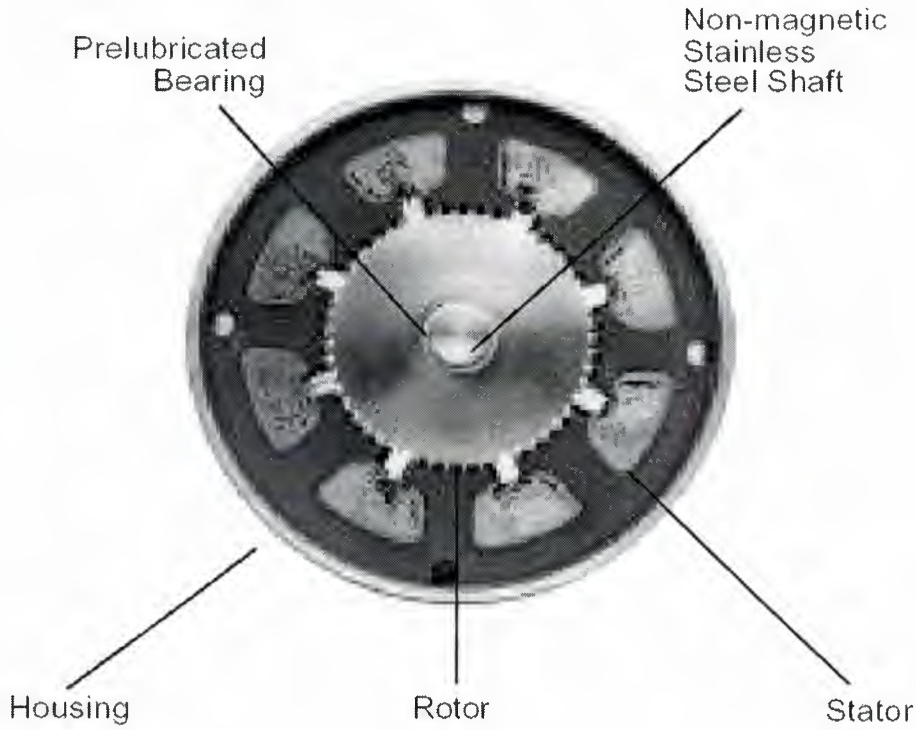


Fig. 1.3 Hybrid stepper motor

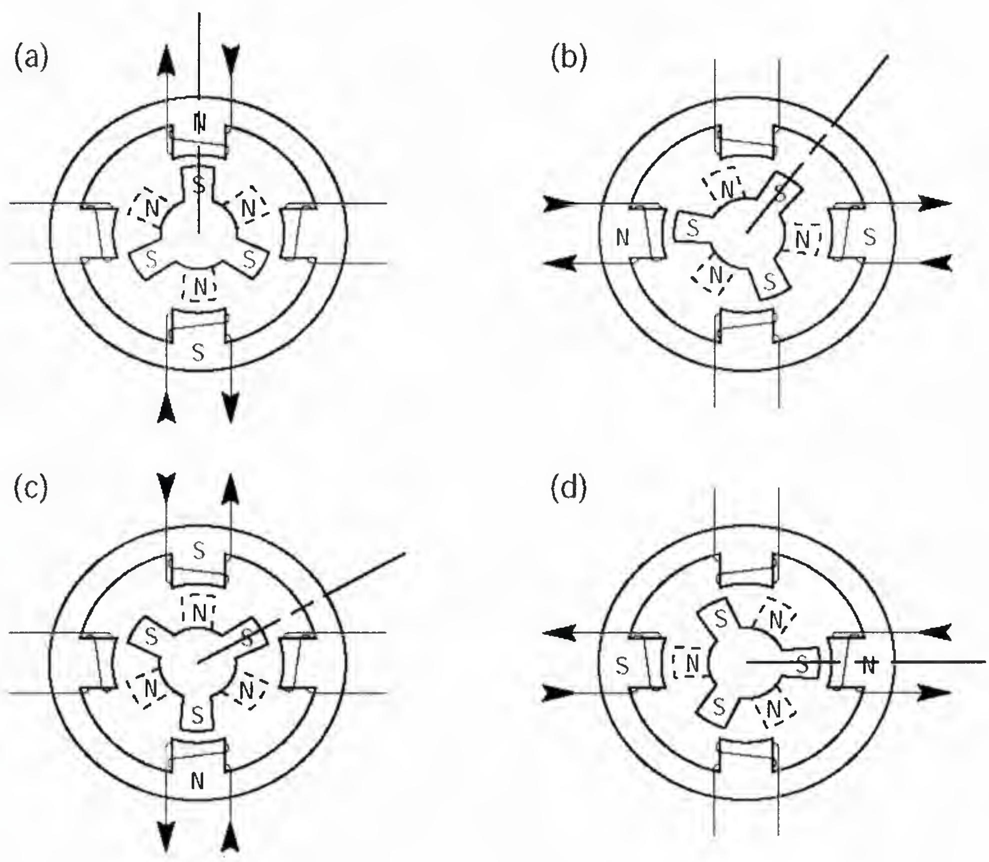


Fig. 1.5 Full stepping

By changing the current flow from the first to the second set of stator windings (b), the stator field rotates through 90° and attracts a new pair of rotor poles. This results in the rotor turning through 30°, corresponding to one full step. Reverting to the first set of stator windings but energizing them in the opposite direction, we rotate the stator field through another 90° and the rotor takes another 30° step (c). Finally, the second set of windings are energized in the opposite direction (d) to give a third step position. We can now go back to the first condition (a), and after these four steps the rotor will have moved through one tooth pitch. This simple motor therefore performs 12 steps per rev. Obviously, if the coils are energized in the reverse sequence, the motor will go round the other way. If two coils are energized simultaneously (Fig. 1.6), the rotor takes up an intermediate position since it is equally attracted to two stator poles. Greater torque is produced under these conditions because all the stator poles are influencing the rotor. The motor can be made to take a full step simply by reversing the current in one set of windings; this causes a 90° rotation of the stator field as before. In fact, this would be the normal way of driving the motor in the full-step mode, always keeping two windings energized and reversing the current in each winding alternately. motor and drive characteristics). In the half-step mode, we are alternately energizing two phases and then only one as shown in Fig. 1.9. Assuming the drive delivers the same winding current in each case, this will cause greater torque to be produced when there are two windings energized. In other words, alternate steps will be strong and weak. This does not represent a major deterrent to motor performance—the available torque is obviously limited by the weaker step, but there will be a significant improvement in low-speed smoothness over the full-step mode.

Clearly, we would like to produce approximately equal torque on every step, and this torque should be at the level of the stronger step. We can achieve this by using a higher current level when there is only one winding energized. This does not overdissipate the motor because the manufacturer's current rating assumes two phases to be energized (the current rating is based on the allowable case temperature). With only one phase energized, the same total power will be dissipated if the current is increased by 40%. Using this higher current in the one-phase-on state produces approximately equal torque on alternate steps (see Fig. 1.10).

If two coils are energized simultaneously (Fig. 1.6), the rotor takes up an intermediate position since it is equally attracted to two stator poles. Greater torque is produced under these conditions because all the stator poles are influencing the rotor.

The motor can be made to take a full step simply by reversing the current in one set of windings; this causes a 90° rotation of the stator field as before. In fact, this would be the normal way of driving the motor in the full-step mode, always keeping two windings energized and reversing the current in each winding alternately. By alternately energizing one winding and then two (Fig. 1.7), the rotor moves through only 15° at each stage and the number of steps per rev will be doubled. This is called half stepping, and most industrial applications make use of this stepping mode. Although there is sometimes a slight loss of torque, this mode results in much better smoothness at low speeds and less overshoot and ringing at the end of each step.

1.3. Current Patterns in the Motor Windings

When the motor is driven in its full-step mode, energizing two windings or "phases" at a time (see Fig. 1.8), the torque available on each step will be the same (subject to very small variations in the motor and drive characteristics). In the half-step mode, we are alternately energizing two phases and then only one as shown in Fig. 1.9. Assuming the drive delivers the same winding current in each case, this will cause greater torque to be produced when there are two windings energized. In other words, alternate steps will be strong and weak. This does not represent a major deterrent to motor performance—the available torque is obviously limited by the weaker step, but there will be a significant improvement in low-speed smoothness over the full-step mode.

Clearly, we would like to produce approximately equal torque on every step, and this torque should be at the level of the stronger step. We can achieve this by using a higher current level when there is only one winding energized. This does not overdissipate the motor because the manufacturer's current rating assumes two phases to be energized (the current rating is based on the allowable case temperature). With only one phase energized, the same total power will be dissipated if the current is increased by 40%. Using this higher current in the one-phase-on state produces approximately equal torque on alternate steps (see Fig. 1.10).

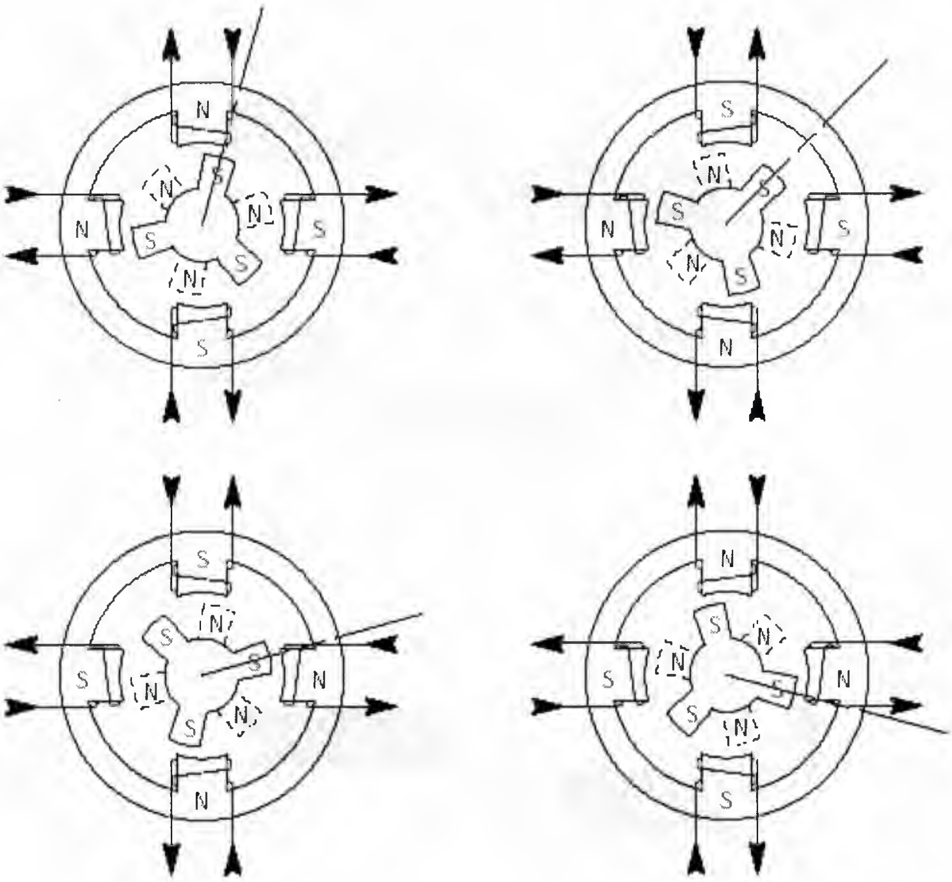


Fig 1.6 Full stepping, Two phase on

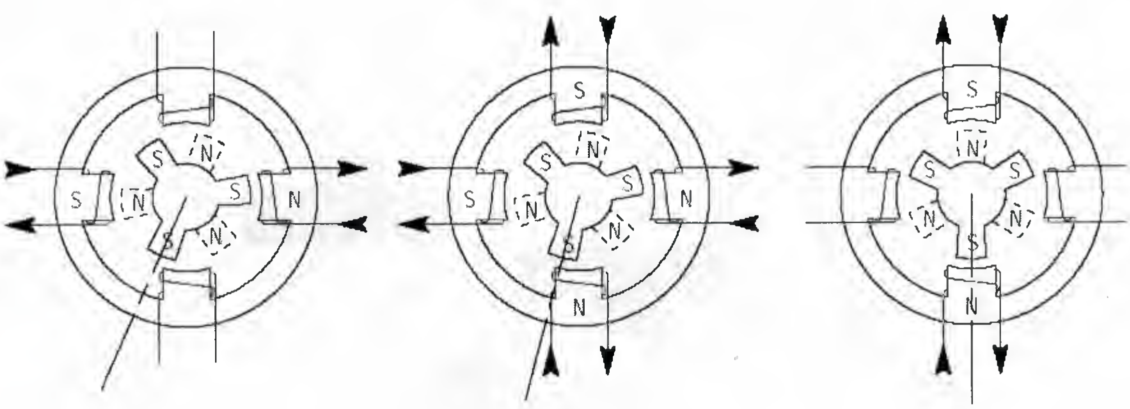


Fig 1.7 Half stepping

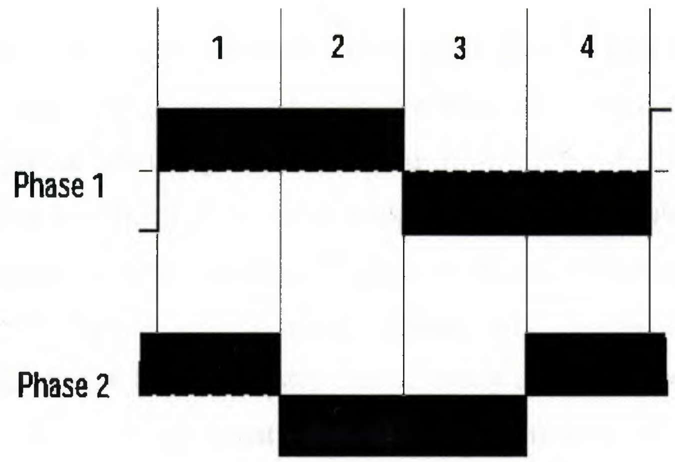


Fig 1.8 Full step current, 2-phase on

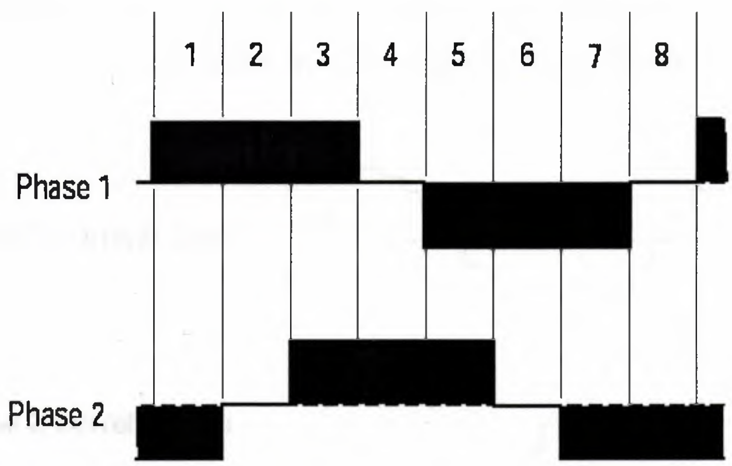


Fig 1.9 Half step current

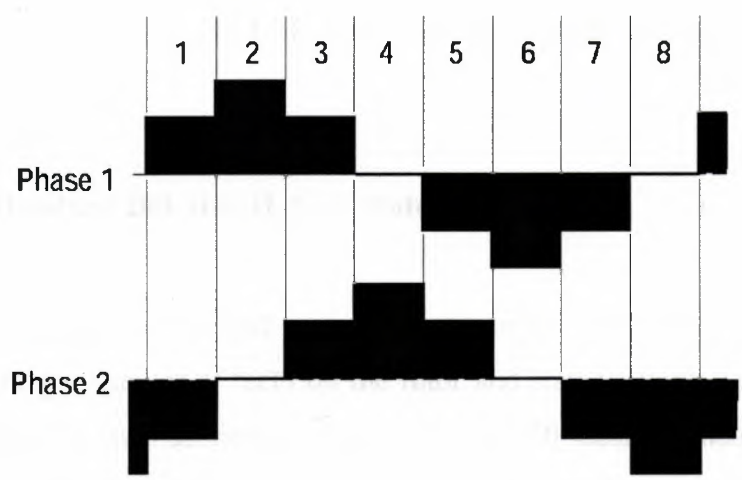


Fig 1.10 Half step current , profiled

We have seen that energizing both phases with equal currents produces an intermediate step position half-way between the one-phase-on positions. If the two phase currents are unequal, the rotor position will be shifted towards the stronger pole. This effect is utilized in the microstepping drive, which subdivides the basic motor step by proportioning the current in the two windings. In this way, the step size is reduced and the low-speed smoothness is dramatically improved. Highresolution microstep drives divide the full motor step into as many as 500 microsteps, giving 100,000 steps per revolution. In this situation, the current pattern in the windings closely resembles two sine waves with a 90° phase shift between them (see Fig. 1.11). The motor is now being driven very much as though it is a conventional AC synchronous motor. In fact, the stepper motor can be driven in this way from a 60 Hz-US (50Hz-Europe) sine wave source by including a capacitor in series with one phase. It will rotate at 72 rpm.

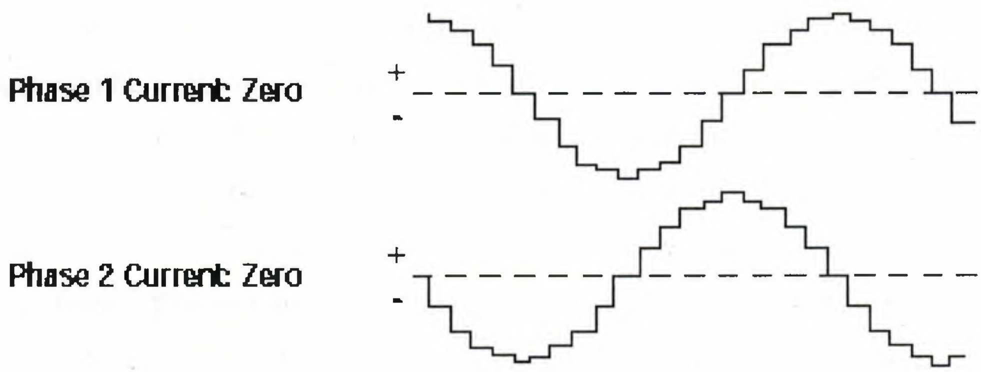


Fig 1.11 phase currents in micro step mode

1.3.1. Standard 200-Step Hybrid Motor

The standard stepper motor operates in the same way as our simple model, but has a greater number of teeth on the rotor and stator, giving a smaller basic step size. The rotor is in two sections as before, but has 50 teeth on each section. The halftooth displacement between the two sections is retained. The stator has 8 poles each with 5 teeth, making a total of 40 teeth (see Fig. 1.12).

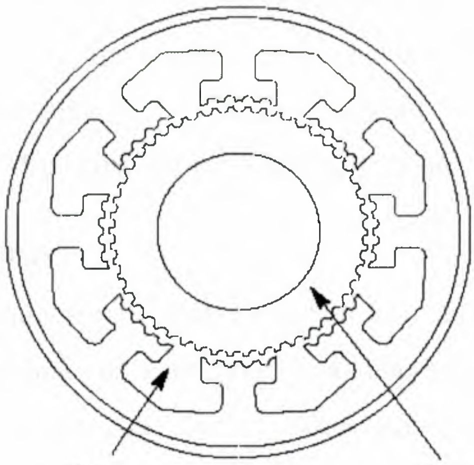


Fig 1.12 200 step hybrid motor

If we imagine that a tooth is placed in each of the gaps between the stator poles, there would be a total of 48 teeth, two less than the number of rotor teeth. So if rotor and stator teeth are aligned at 12 o'clock, they will also be aligned at 6 o'clock. At 3 o'clock and 9 o'clock the teeth will be misaligned. However, due to the displacement between the sets of rotor teeth, alignment will occur at 3 o'clock and 9 o'clock at the other end of the rotor. The windings are arranged in sets of four, and wound such that diametrically-opposite poles are the same. So referring to Fig. 1.12, the north poles at 12 and 6 o'clock attract the south-pole teeth at the front of the rotor; the south poles at 3 and 9 o'clock attract the north-pole teeth at the back. By switching current to the second set of coils, the stator field pattern rotates through 45°. However, to align with this new field, the rotor only has to turn through 1.8°. This is equivalent to one quarter of a tooth pitch on the rotor, giving 200 full steps per revolution.

Note that there are as many detent positions as there are full steps per rev, normally 200. The detent positions correspond with rotor teeth being fully aligned with stator teeth. When power is applied to a stepper drive, it is usual for it to energize in the "zero phase" state in which there is current in both sets of windings. The resulting rotor position does not correspond with a natural detent position, so an unloaded motor will always move by at least one half step at power-on. Of course, if the system was turned off other than in the zero phase state, or the motor is moved in the meantime, a greater movement may be seen at power-up.

Another point to remember is that for a given current pattern in the windings, there are as many stable positions as there are rotor teeth (50 for a 200-step motor). If a motor is de-synchronized, the resulting positional error will always be a whole number of rotor teeth or a multiple of 7.2°. A motor cannot "miss" individual steps – position errors of one or two steps must be due to noise, spurious step pulses or a controller fault.

1.3.2. Bifilar Windings

Most motors are described as being "bifilar wound", which means there are two identical sets of windings on each pole. Two lengths of wire are wound together as though they were a single coil. This produces two windings that are electrically and magnetically almost identical – if one coil were to be wound on top of the other, even with the same number of turns, the magnetic characteristics would be different. In simple terms, whereas almost all the flux from the inner coil would flow through the iron core, some of the flux from the outer coil would flow through the windings of the coil underneath.

The origins of the bifilar winding go back to the unipolar drive . Rather than have to reverse the current in one winding, the field may be reversed by transferring current to a second coil wound in the opposite direction. (Although the two coils are wound the same way, interchanging the ends has the same effect.) So with a bifilar-wound motor, the drive can be kept simple. However, this requirement has now largely disappeared with the widespread availability of the more-efficient bipolar drive. Nevertheless, the two sets of windings do give us additional flexibility, and we shall see that different connection methods can be used to give alternative torque-speed characteristics.

If all the coils in a bifilar-wound motor are brought out separately, there will be a total of 8 leads (see Fig. 1.13). This is becoming the most common configuration since it gives the greatest flexibility. However, there are still a number of motors produced with only 6 leads, one lead serving as a common connection to each winding in a bifilar pair. This arrangement limits the motor's range of application since the windings cannot be connected in parallel. Some motors are made with only 4 leads, these are not bifilar-wound and cannot be used with a unipolar drive. There is obviously no alternative connection method with a 4-lead motor, but in many applications this is not a drawback and the problem of insulating unused leads is avoided.

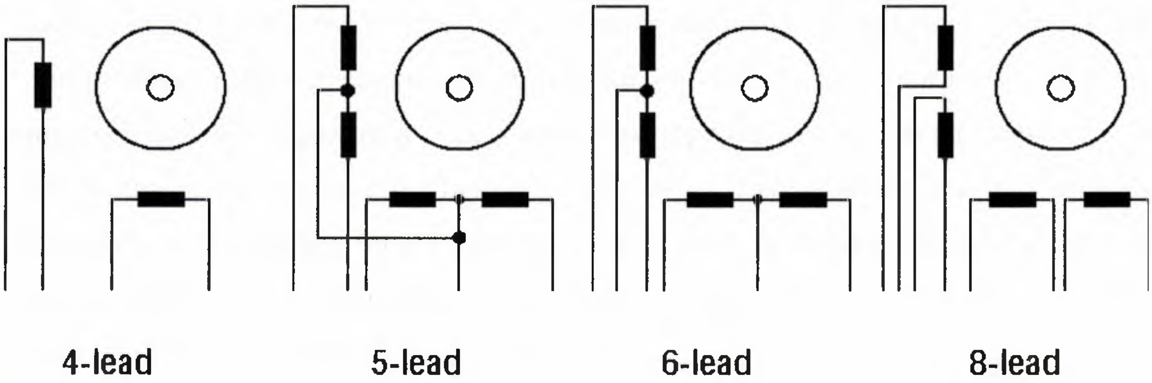


Fig 1.13 Motor lead configuration

Occasionally a 5-lead motor may be encountered. These are not recommended since they cannot be used with conventional bipolar drives requiring electrical isolation between the phases.

Looking at the motor longitudinal section (Fig. 1.14), we can see the permanent magnet in the rotor and the path of the flux through the pole pieces and the stator. The alternating flux produced by the stator windings flows in a plane at right angles to the page. Therefore, the two flux paths are at right angles to each other and only interact in the rotor pole pieces. This is an important feature of the hybrid motor – it means that the permanent

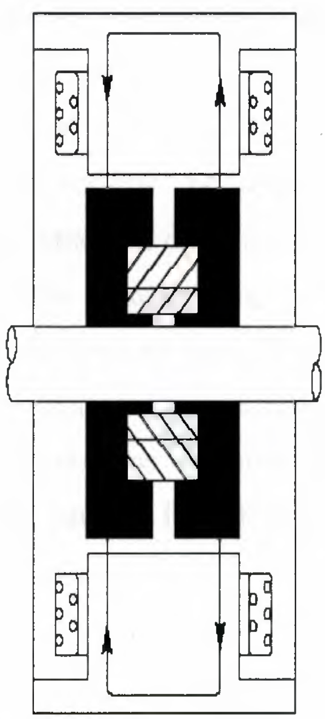


Fig 1.14 Longitudinal section through single stack motor

magnet in the rotor does not "see" the alternating field from the windings, hence it does not produce a demagnetizing effect. Unlike the DC servo motor, it is generally impossible to de-magnetize a stepper motor by applying excess current. However, too much current will damage the motor in other ways. Excessive heating may melt the insulation or the winding formers, and may soften the bonding material holding the rotor laminations. If this happens and the laminations are displaced, the effects can be the same as if the rotor had been de-magnetized

Fig. 1.14 also shows that the rotor flux only has to cross a small air gap (typically 0.1mm or 0.004") when the rotor is in position. By magnetizing the rotor after assembly, a high flux density is obtained that can be largely destroyed if the rotor is removed. Stepper motors should therefore not be dismantled purely to satisfy curiosity, since the useful life of the motor will be terminated. Because the shaft of the motor passes through the center of the permanent magnet, a non-magnetic material must be used to avoid a magnetic shortcircuit. Stepper shafts are therefore made of stainless steel, and should be handled with care. Small-diameter motors are particularly vulnerable if they are dropped on the shaft end, as this will invariably bend the shaft.

To produce a motor with a higher torque output, we need to increase the strength of both the permanent magnet in the rotor and the field produced by the stator. A stronger rotor magnet can be obtained by increasing the diameter, giving us a larger cross-sectional area. However, increasing the diameter will degrade the acceleration performance of the motor because the torque-to-inertia ratio worsens (to a first approximation, torque increases with diameter squared but inertia goes up by the fourth power). Nevertheless, we can increase torque output without degrading acceleration performance by adding further magnet sections or "stacks" to the same shaft (Fig. 1.15). A second stack will enable twice the torque to be produced and will double the inertia, so the torque-to-inertia ratio remains the same. Hence, stepper motors are produced in single-, two- and three-stack versions in each frame size.

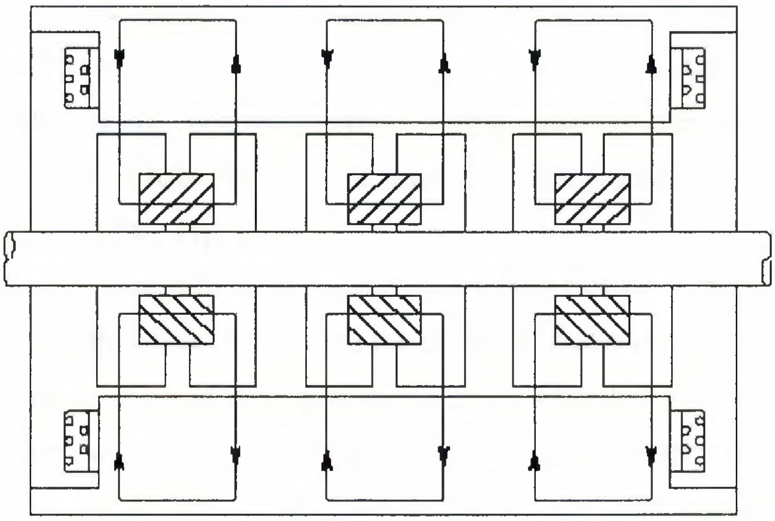


Fig 1.15 Three-stack hybrid stepping motor

As a guideline, the torque-to-inertia ratio reduces by a factor of two with each increase in frame size (diameter). So an unloaded 34-size motor can accelerate twice as rapidly as a 42-size, regardless of the number of stacks.

1.4. Linear stepping motors

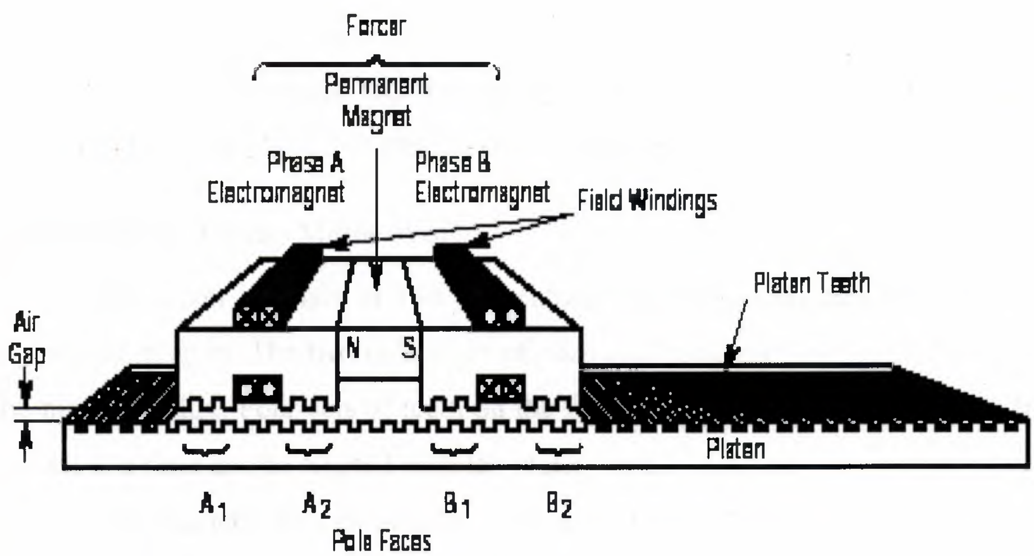


Fig 1.16 Linear stepping motor

The linear stepper is essentially a conventional rotary stepper that has been "unwrapped" so that it operates in a straight line. The moving component is referred to as the forcer and it travels along a fixed element or platen. For operational purposes, the platen is equivalent to the rotor in a normal stepper, although it is an entirely passive device and has no permanent magnet. The magnet is incorporated in the moving forcer together with the coils (see Fig. 1.16).

The forcer is equipped with 4 pole pieces each having 3 teeth. The teeth are staggered in pitch with respect to those on the platen, so that switching the current in the coils will bring the next set of teeth into alignment. A complete switching cycle (4 full steps) is equivalent to one tooth pitch on the platen. Like the rotary stepper, the linear motor can be driven from a microstep drive. In this case, a typical linear resolution will be 12,500 steps per inch.

The linear motor is best suited for applications that require a low mass to be moved at high speed. In a leadscrew-driven system, the predominant inertia is usually the leadscrew rather than the load to be moved. Hence, most of the motor torque goes to accelerate the leadscrew, and this problem becomes more severe the longer the travel required. Using a linear motor, all the developed force is applied directly to the load and the performance achieved is independent of the length of the move. A screw-driven system can develop greater linear force and better stiffness; however, the maximum speed may be as much as ten times higher with the equivalent linear motor. For example, a typical maximum speed for a linear motor is 100 in/sec. To achieve this with a 10-pitch ballscrew would require a rotary speed of 6,000 rpm. In addition, the linear motor can travel up to 12 feet using a standard platen.

1.4.1. How the Linear Motor Works

The forcer consists of two electromagnets (A and B) and a strong rare earth permanent magnet. The two pole faces of each electromagnet are toothed to concentrate the magnetic flux. Four sets of teeth on the forcer are spaced in quadrature so that only one set at a time can be aligned with the platen teeth.

The magnetic flux passing between the forcer and the platen gives rise to a very strong force of attraction between the two pieces. The attractive force can be up to 10 times the peak holding force of the motor, requiring a bearing arrangement to maintain precise clearance between the pole faces and platen teeth. Either mechanical roller bearings or air bearings are used to maintain the required clearance. When current is established in a field winding, the resulting magnetic field tends to reinforce permanent magnetic flux at one pole face and cancel it at the other. By reversing the current, the reinforcement and cancellation are exchanged.

Removing current divides the permanent magnetic flux equally between the pole faces. By selectively applying current to phase A and B, it is possible to concentrate flux at any of the forcer's four pole faces. The face receiving the highest flux concentration will attempt to align its teeth with the platen. Fig. 1.17 shows the four primary states or full steps of the forcer. The four steps result in motion of one tooth interval to the right. Reversing the sequence moves the forcer to the left.

Repeating the sequence in the example will cause the forcer to continue its movement. When the sequence is stopped, the forcer stops with the appropriate tooth set aligned. At rest, the forcer develops a holding force that opposes any attempt to displace it. As the resting motor is displaced from equilibrium, the restoring force increases until the displacement reaches one-quarter of a tooth interval. (See Fig. 1.18.) Beyond this point, the restoring force drops. If the motor is pushed over the crest of its holding force, it slips or jumps rather sharply and comes to rest at an integral number of tooth intervals away from its original location. If this occurs while the forcer is travelling along the platen, it is referred to as a stall condition.

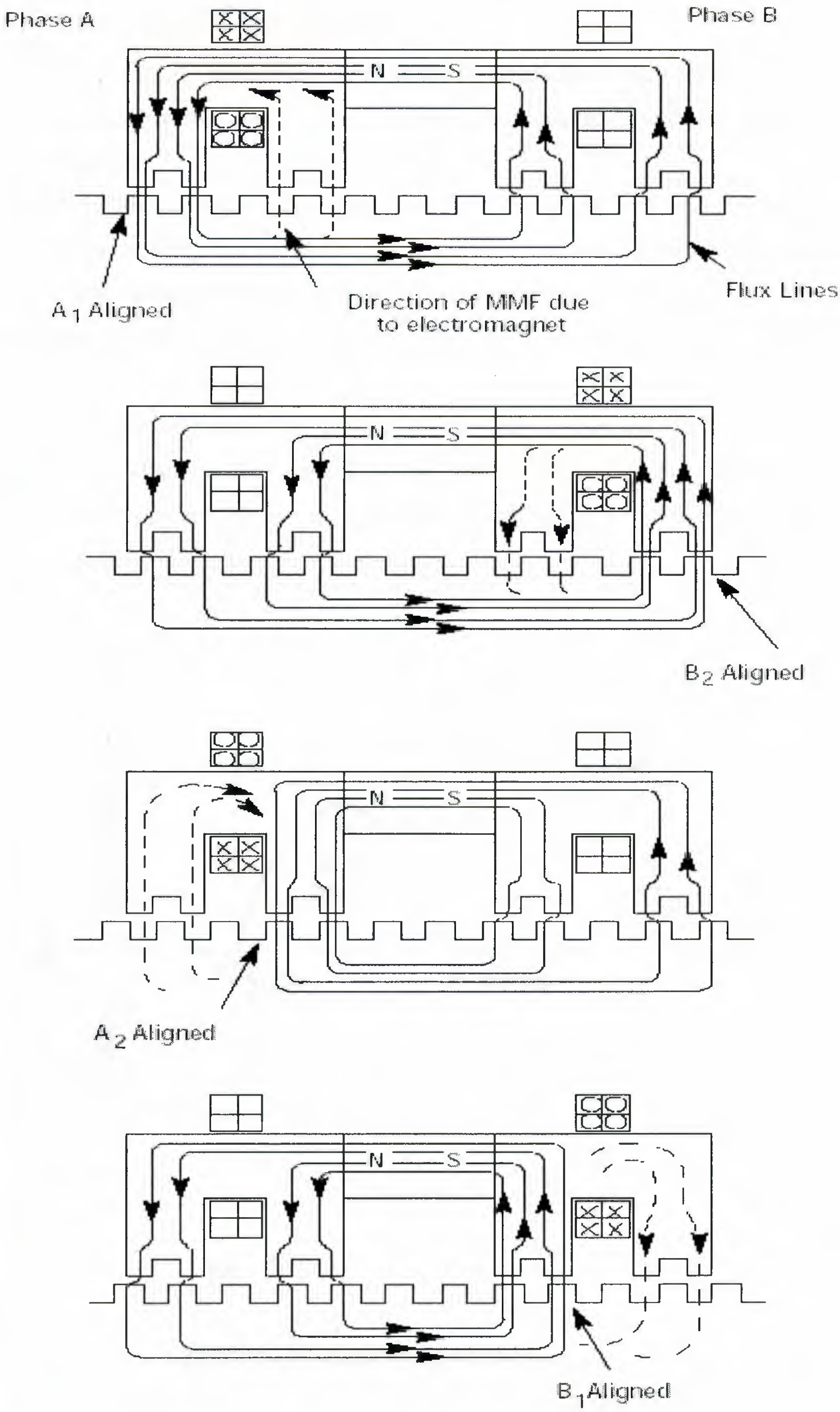


Fig. 1.17 The four cardinal states or full steps of the forcer

1.5. Step Motor Characteristics

There are numerous step motor performance characteristics that warrant discussion. However, we'll confine ourselves to those traits with the greatest practical significance.

Fig. 1.18 illustrates the static torque curve of the hybrid step motor. This relates to a motor that is energized but stationary. It shows us how the restoring torque varies with rotor position as it is deflected from its stable point. We're assuming that there are no frictional or other static loads on the motor. As the rotor moves away from the stable position, the torque steadily increases until it reaches a maximum after one full step (1.8°). This maximum value is called the holding torque and it represents the largest static load that can be applied to the shaft without causing continuous rotation. However, it doesn't tell us the maximum running torque of the motor – this is always less than the holding torque (typically about 70%).

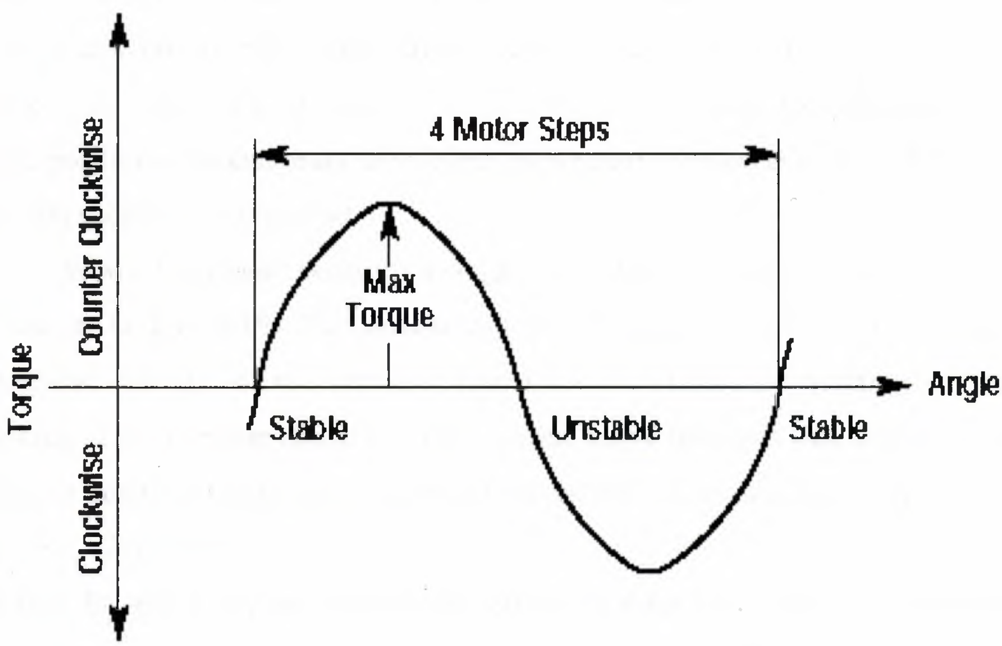


Fig 1.18 Static torque-displacement charecteristic

As the shaft is deflected beyond one full step, the torque will fall until it is again at zero after two full steps. However, this zero point is unstable and the torque reverses immediately beyond it. The next stable point is found four full steps away from the first, equivalent to one tooth pitch on the rotor or 1/50 of a revolution.

Although this static torque characteristic isn't a great deal of use on its own, it does help explain some of the effects we observe. For example, it indicates the static stiffness of the system, (i.e., how the shaft position changes when a torque load is applied to a stationary motor). Clearly the shaft must deflect until the generated torque matches the applied load. If the load varies, so too will the static position. Non-cumulative position errors will therefore result from effects such as friction or outof-balance torque loads. It is important to remember that the static stiffness is not improved by using a microstepping drive—a given load on the shaft will produce the same angular deflection. So while microstepping increases resolution and smoothness, it may not necessarily improve positioning accuracy.

Under dynamic conditions with the motor running, the rotor must be lagging behind the stator field if it is producing torque. Similarly, there will be a lead situation when the torque reverses during deceleration. Note that the lag and lead relate only to position and not to speed. From the static torque curve (Fig. 1.18), clearly this lag or lead cannot exceed two full steps (3.6°) if the motor is to retain synchronism. This limit to the position error can make the stepper an attractive option in systems where dynamic position accuracy is important.

When the stepper performs a single step, the nature of the response is oscillatory as shown in Fig. 1.19. The system can be likened to a mass that is located by a "magnetic spring", so the behavior resembles the classic mass-spring characteristic. Looking at it in simple terms, the static torque curve indicates that during the step, the torque is positive during the full forward movement and so is accelerating the rotor until the new stable point is

reached. By this time, the momentum carries the rotor past the stable position and the torque now reverses, slowing the rotor down and bringing it back in the opposite direction. The amplitude, frequency and decay rate of this oscillation will depend on the friction and inertia in the system as well as the electrical characteristics of the motor and drive. The initial overshoot also depends on step amplitude, so half-stepping produces less overshoot than full stepping and microstepping will be better still. Attempting to step the motor at its natural oscillation frequency can cause an exaggerated response known as resonance. In severe cases, this can lead to the motor desynchronizing or "stalling." It is seldom a problem with half-step drives and even less so with a microstepper. The natural resonant speed is typically 100-200 full steps/sec. (0.5-1 rev/sec).

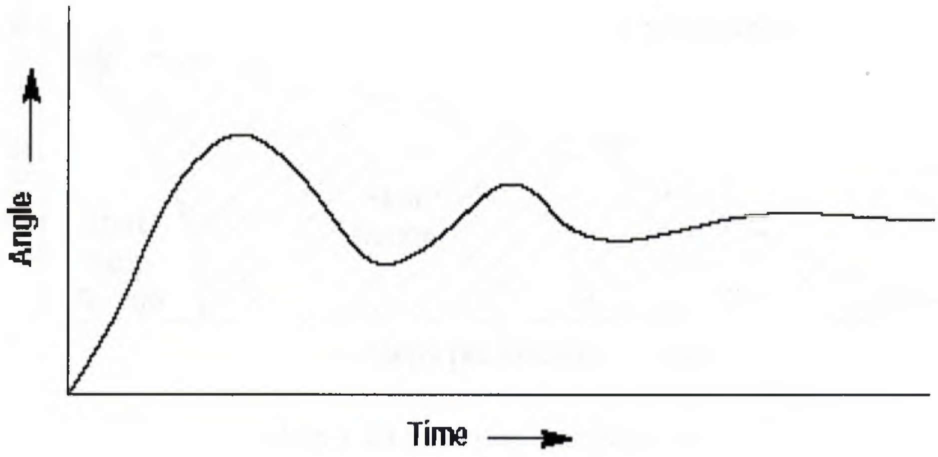


Fig. 1.19 Single step response

Under full dynamic conditions, the performance of the motor is described by the torque-speed curve as shown in Fig. 1.20. There are two operating ranges, the start/stop (or pull in) range and the slew (or pull out) range. Within the start/stop range, the motor can be started or stopped by applying index pulses at constant frequency to the drive. At speeds within this range, the motor has sufficient torque to accelerate its own inertia up to synchronous speed without the position lag exceeding 3.6°. Clearly, if an inertial load is added, this speed range is reduced. So the start/ stop speed range depends on the load inertia. The upper limit to the start/stop range is typically between 200 and 500 full steps/sec (1-2.5 revs/sec).

To operate the motor at faster speeds, it is necessary to start at a speed within the start/stop range and then accelerate the motor into the slew region. Similarly, when stopping the motor, it must be decelerated back into the start/stop range before the clock pulses are terminated. Using acceleration and deceleration "ramping" allows much higher speeds to be achieved, and in industrial applications the useful speed range extends to about 3000 rpm (10,000 full steps/sec). Note that continuous operation at high speeds is not normally possible with a stepper due to rotor heating, but high speeds can be used successfully in positioning applications.

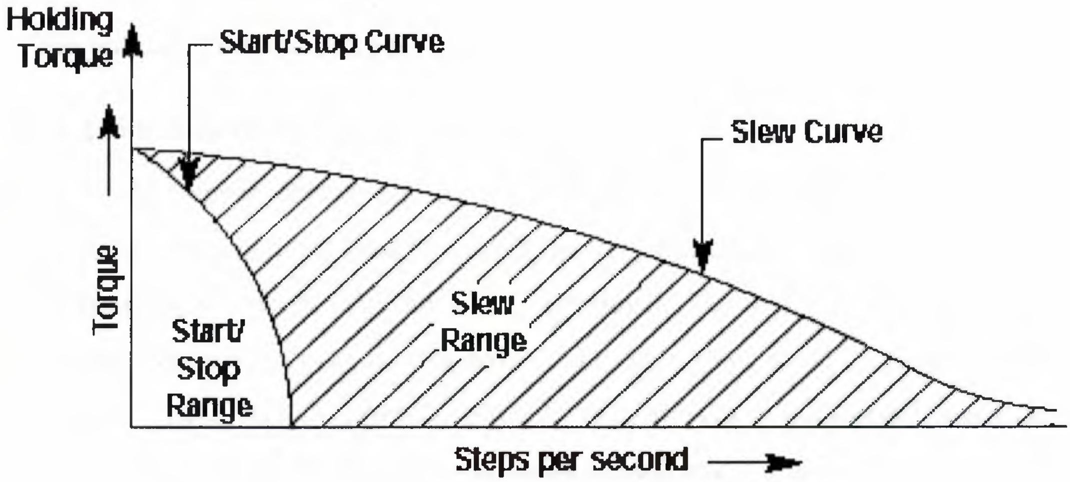


Fig. 1.20 Start/stop and slew curves

The torque available in the slew range does not depend on load inertia. The torque-speed curve is normally measured by accelerating the motor up to speed and then increasing the load until the motor stalls. With a higher load inertia, a lower acceleration rate must be used but the available torque at the final speed is unaffected.

1.5.1. Size and Power

In addition to being classified by their step angle stepper motors are also classified according to frame sizes which correspond to the diameter of the body of the motor. For instance a size 11 stepper motor has a body diameter of approximately 1.1 inches. Likewise a size 23 stepper motor has a body diameter of 2.3 inches (58 mm), etc. The body length may however, vary from motor to motor within the same frame size classification. As a general rule the available torque output from a motor of a particular frame size will increase with increased body length.

Power levels for IC-driven stepper motors typically range from below a watt for very small motors up to 10 – 20 watts for larger motors. The maximum power dissipation level or thermal limits of the motor are seldom clearly stated in the motor manufacturers data. To determine this we must apply the relationship P =V x I For example, a size 23 step motor may be rated at 6V and 1A per phase. Therefore, with two phases energized the motor has a rated power dissipation of 12 watts. It is normal practice to rate a stepper motor at the power dissipation level where the motor case rises 65°C above the ambient in still air. Therefore, if the motor can be mounted to a heatsink it is often possible to increase the allowable power dissipation level. This is important as the motor is designed to be and should be used at its maximum power dissipation ,to be efficient from a size/output power/cost point of view.

1.5.2.When to Use a Stepper Motor

A stepper motor can be a good choice whenever controlled movement is required. They can be used to advantage in applications where you need to control rotation angle, speed, position and synchronism. Because of the inherent advantages listed previously, stepper motors have found their place in many different applications. Some of these include printers, plotters, highend office equipment, hard disk drives, medical equipment, fax machines, automotive and many more.

1.5.3. The Rotating Magnetic Field

When a phase winding of a stepper motor is energized with current a magnetic flux is developed in the stator. The direction of this flux is determined by the "Right Hand Rule" which states:

"If the coil is grasped in the right hand with the fingers pointing in the direction of the current in the winding (the thumb is extended at a 90° angle to the fingers), then the thumb will point in the direction of the magnetic field."

Figure 5 shows the magnetic flux path developed when phase B is energized with winding current in the direction shown. The rotor then aligns itself so that the flux opposition is minimized. In this case the motor would rotate clockwise so that its south pole aligns with the north pole of the stator B at position 2 and its north pole aligns with the south pole of stator B at position 6. To get the motor to rotate we can now see that we must provide a sequence of energizing the stator windings in such a fashion that provides a rotating magnetic flux field which the rotor follows due to magnetic attraction.

1.5.4. Torque Generation

The torque produced by a stepper motor depends on several factors.

- The step rate
- The drive current in the windings
- The drive design or type

In a stepper motor a torque is developed when the magnetic fluxes of the rotor and stator are displaced from each other. The stator is made up of a high permeability magnetic material. The presence of this high permeability material causes the magnetic flux to be confined for the most part to the paths defined by the stator structure in the same fashion that currents are confined to the conductors of an electronic circuit. This serves to concentrate the flux at the stator poles. The torque output produced by the motor is proportional to the intensity of the magnetic flux generated when the winding is energized.

The basic relationship which defines the intensity of the magnetic flux is defined by:

$H = (N \times i) / l$ where:

N = The number of winding turns

i = current

H = Magnetic field intensity

l = Magnetic flux path length

This relationship shows that the magnetic flux intensity and consequently the torque is proportional to the number of winding turns and the current and inversely proportional to the length of the magnetic flux path. From this basic relationship one can see that the same frame size stepper motor could have very different torque output capabilities simply by changing the winding parameters. More detailed information on how the winding parameters affect the output capability of the motor can be found in the application note entitled "Drive Circuit Basics".

1.5.5. Phases, Poles and Stepping Angles

Usually stepper motors have two phases, but three- and five-phase motors also exist. A bipolar motor with two phases has one winding/phase and a unipolar motor has one winding, with a center tap per phase. Sometimes the unipolar stepper motor is referred to as a "fourphase motor", even though it only has two phases.

Motors that have two separate windings per phase also exist—these can be driven in either bipolar or unipolar mode.

A pole can be defined as one of the regions in a magnetized body where the magnetic flux density is concentrated. Both the rotor and the stator of a step motor have poles. Figure 2 contains a simplified picture of a two-phase stepper motor having 2 poles (or 1 pole pairs) for each phase on the stator, and 2 poles (one pole pair) on the rotor. In reality several more poles are added to both the rotor and stator structure in order to increase the number of steps per revolution of the motor, or in other words to provide a smaller basic (full step) stepping angle. The permanent magnet stepper motor contains an equal number of rotor and stator pole pairs. Typically the PM motor has 12 pole pairs. The stator has 12 pole pairs per phase. The hybrid type stepper motor has a rotor with teeth. The rotor is split into two parts, separated by a permanant magnet—making half of the teeth south poles and half north poles.The number of pole pairs is equal to the number of teeth on one of the rotor halves. The stator of a hybrid motor also has teeth to build up a higher number of equivalent poles (smaller pole pitch, number of equivalent poles = 360/teeth pitch) compared to the main poles, on which the winding coils are wound. Usually 4 main poles are used for 3.6 hybrids and 8 for 1.8- and 0.9-degree types.

It is the relationship between the number of rotor poles and the equivalent stator poles, and the number the number of phases that determines the full-step angle of a stepper motor. Step angle=360 , (NPh x Ph)=360/N

NPh = Number of equivalent poles per phase = number of rotor poles

Ph = Number of phases

N = Total number of poles for all phases together

If the rotor and stator tooth pitch is unequal, a more-complicated relationship exists.

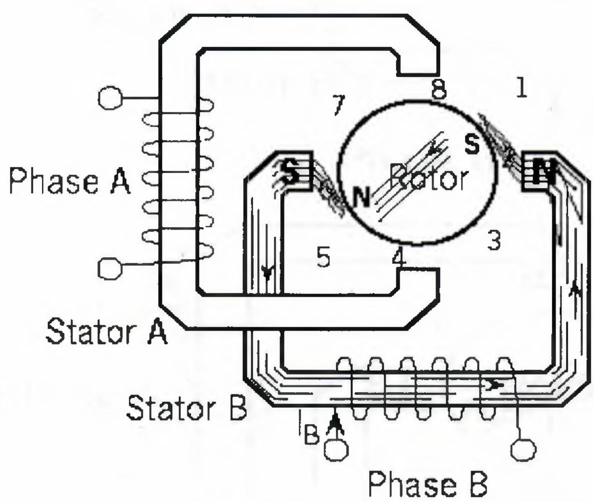


Fig 1.21 magnetic flux path through a two pole stepper motor with a lag between rotor and stator

1.5.6. Stepping Modes

The following are the most common drive modes.

• Wave Drive (1 phase on)

• Full Step Drive (2 phases on)

• Half Step Drive (1 & 2 phases on)

• Microstepping (Continuously varying motor currents)

For the following discussions please refer to the figure 1.22. In Wave Drive only one winding is energized at any given time. The stator is energized according to the sequence A > B A ® B and the rotor steps from position 8 ® 2 ® 4 ® 6. For unipolar and bipolar wound motors with the same winding parameters this excitation mode would result in the same mechanical position. The disadvantage of this drive mode is that in the unipolar wound motor you are only using 25% and in the bipolar motor only 50% of the total motor winding at any given time. This means that you are not getting the maximum torque output from the motor.

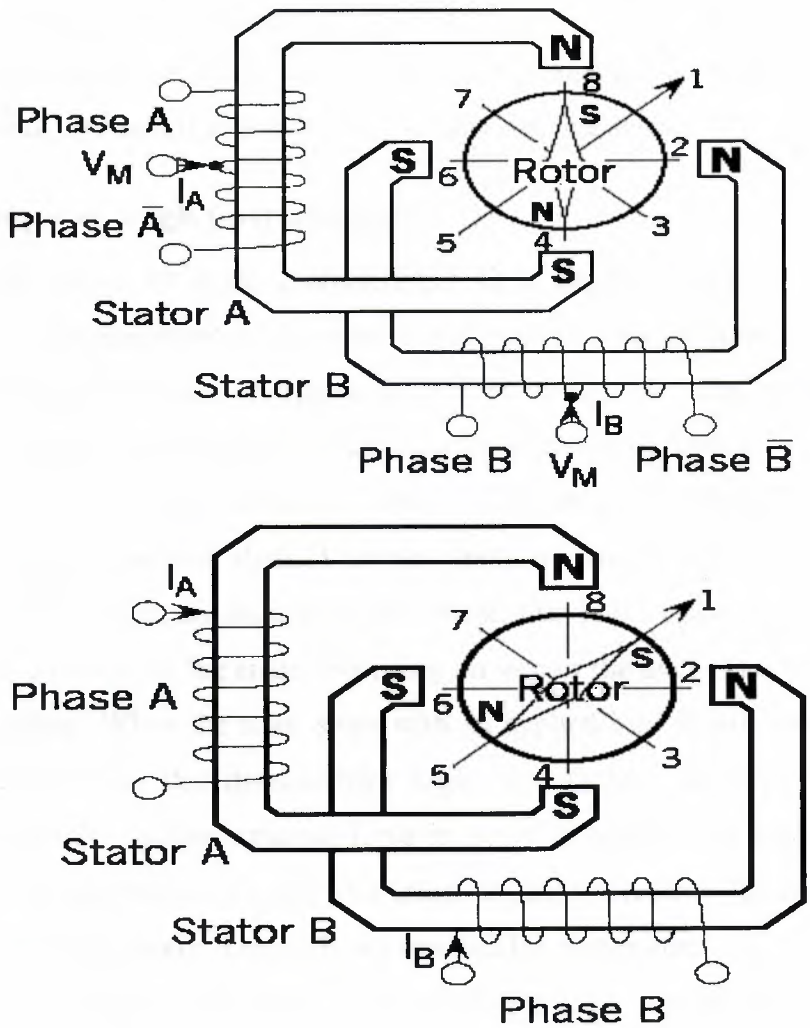


Fig 1.22 Unipolar and bipolar wound stepper motors

In Full Step Drive you are energizing two phases at any given time. The stator is energized according to the sequence AB ® AB ® AB ® AB and the rotor steps from position 1 ® 3 ® 5 ® 7 . Full step mode results in the same angular movement as 1 phase on drive but the mechanical position is offset by one half of a full step. The torque output of the unipolar wound motor is lower than the bipolar motor (for motors with the same winding parameters) since the unipolar motor uses only 50% of the available winding while the bipolar motor uses the entire winding.

Half Step Drive combines both wave and full step (1&2 phases on) drive modes. Every second step only one phase is energized and during the other steps one phase on each stator. The stator is energized according to the sequence AB ® B ® AB ® A ® AB ® B ® AB ® A and the rotor steps from position 1 ® 2 ® 3 ® 4 ® 5 ® 6 ® 7 ® 8. This results in angular movements that are half of those in 1- or 2-phases-on drive modes. Half stepping can reduce a phenomena referred to as resonance which can be experienced in 1- or 2- phases-on drive modes.

In Microstepping Drive the currents in the windings are continuously varying to be able to break up one full step into many smaller discrete steps.

1.5.7. Torque vs, Angle Characteristics

The torque vs angle characteristics of a stepper motor are the relationship between the displacement of the rotor and the torque which applied to the rotor shaft when the stepper motor is energized at its rated voltage. An ideal stepper motor has a sinusoidal torque vs displacement characteristic as shown in figure1.24

Positions A and C represent stable equilibrium points when no external force or load is applied to the rotor shaft. When you apply an external force Ta to the motor shaft you in essence create an angular displacement, Qa. This angular displacement, Qa, is referred to as a lead or lag angle depending on wether the motor is actively accelerating or decelerating. When the rotor stops with an applied load it will come to rest at the position defined by this displacement angle. The motor develops a torque, Ta, in opposition to the applied external force in order to balance the load. As the load is increased the displacement angle also increases until it reaches the maximum holding torque, Th, of the motor. Once Th is exceeded the motor enters an unstable region. In this region a torque is the opposite direction is created and the rotor jumps over the unstable point to the next stable point.

The displacement angle is determined by the following relationship:

X = (Z , 2p) ′ *sin*(Ta , Th) where:

Z = rotor tooth pitch

Ta = Load torque

Th = Motors rated holding torque

X = Displacement angle.

Therefore if you have a problem with the step angle error of the loaded motor at rest you can improve this by changing the "stiffness" of the motor. This is done by increasing the holding torque of the motor. We can see this effect shown in the figure 1.21. Increasing the holding torque for a constant load causes a shift in the lag angle from Q2 to Q1.

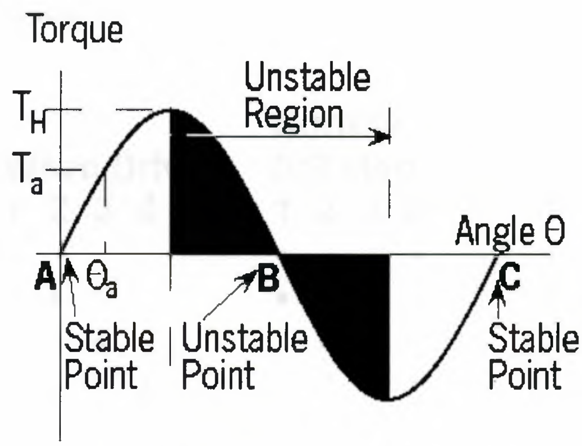


Fig 1.23 Torque vs. rotor angular position

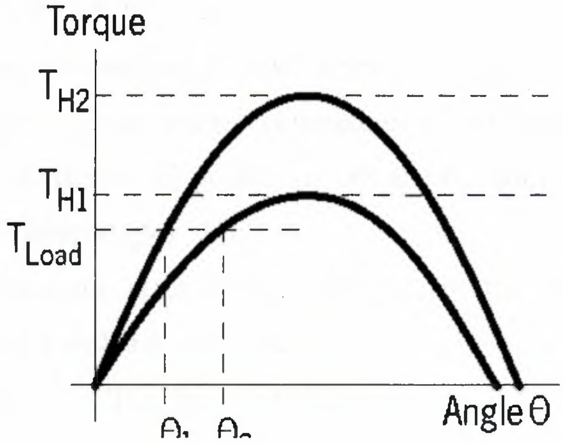


Fig 1.24 Torque vs.rotor angle position at different holding torque

1.5.8. Step Angle Accuracy

One reason why the stepper motor has achieved such popularity as a positioning device is its accuracy and repeatability. Typically stepper motors will have a step angle accuracy of 3 – 5% of one step. This error is also noncumulative from step to step. The accuracy of the stepper motor is mainly a function of the mechanical precision of its parts and assembly.

Figure 9 shows a typical plot of the positional accuracy of a stepper motor.

1.5.8.1. Step Position Error

The maximum positive or negative position error caused when the motor has rotated one step from the previous holding position.

Step position error = measured step

angle - theoretical angle

	Wave Drive				Normal full step				Half-step drive							
Phase	1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	8
A	•				•			•	•						•	•
B		•			•	•			•	•	•					
$\overline{A}$			•			•	•				•	•	•			
$\overline{B}$				•			•	•						•	•	•

Table 1 Excitation sequences for different drive mode

1.5.8.2. Positional Error

The motor is stepped N times from an initial position (N = 360°/step angle) and the angle from the initial position is measured at each step position. If the angle from the initial position to the N-step position is QN and the error is DQN where:

DQN = DQN - (step angle) ´ N.

The positional error is the difference of the maximum and minimum but is usually expressed with a ± sign. That is:

positional error = ±1/2(DQMax - DQMin)

1.5.8.3. Hysteresis Positional Error

The values obtained from the measurement of positional errors in both directions.

1.6. Mechanical Parameters, Load, Friction, Inertia

The performance of a stepper motor system (driver and motor) is also highly dependent on the mechanical parameters of the load. The load is defined as what the motor drives. It is typically frictional, inertial or a combination of the two.Friction is the resistance to motion due to the unevenness of surfaces which rub together. Friction is constant with velocity. A minimum torque level is required throughout the step in over to overcome this friction (at least equal to the friction). Increasing a frictional load lowers the top speed, lowers the acceleration and increases the positional error. The converse is true if the frictional load is lowered Inertia is the resistance to changes in speed. A high inertial load requires a high inertial starting torque and the same would apply for braking. Increasing an inertial load will increase speed stability, increase the amount of time it takes to reach a desired speed and decrease the maximum self start pulse rate. The converse is again true if the inertia is decreased.

The rotor oscillations of a stepper motor will vary with the amount of friction and inertia load. Because of this relationship unwanted rotor oscillations can be reduced by mechanical damping means however it is more often simpler to reduce these unwanted oscillations by electrical damping methods such as switch from full step drive to half step drive.

1.6.1.Torque vs, Speed Characteristics

The torque vs speed characteristics are the key to selecting the right motor and drive method for a specific application. These characteristics are dependent upon (change with) the motor, excitation mode and type of driver or drive method. A typical "speed – torque curve" is shown in figure 1.25 .

To get a better understanding of this curve it is useful to define the different aspect of this curve.

1.6.2.1. Holding torque

The maximum torque produced by the motor at standstill.

1.6.1.2. Pull-In Curve

The pull-in curve defines a area refered to as the start stop region. This is the maximum frequency at which the motor can start/stop instantaneously, with a load applied, without loss of synchronism.

1.6.1.3. Maximum Start Rate

The maximum starting step frequency with no load applied.

1.6.1.4. Pull-Out Curve

The pull-out curve defines an area refered to as the slew region. It defines the maximum frequency at which the motor can operate without losing synchronism. Since this region is outside the pull-in area the motor must ramped (accelerated or decelerated) into this region.

1.6.1.5. Maximum Slew Rate

The maximum operating frequency of the motor with no load applied. The pull-in characteristics vary also depending on the load. The larger the load inertia the smaller the pull-in area. We can see from the shape of the curve that the step rate affects the torque output capability of stepper motor The decreasing torque output as the speed increases is caused by the fact that at high speeds the inductance of the motor is the dominant circuit element.

The shape of the speed – torque curve can change quite dramatically depending on the type of driver used. The bipolar chopper type drivers which Ericsson Components produces will maximum the speed – torque performance from a given motor. Most motor manufacturers provide these speed - torque curves for their motors. It is important to understand what driver type or drive method the motor manufacturer used in developing their curves as the torque vs. speed characteristics of an given motor can vary significantly depending on the drive method used.

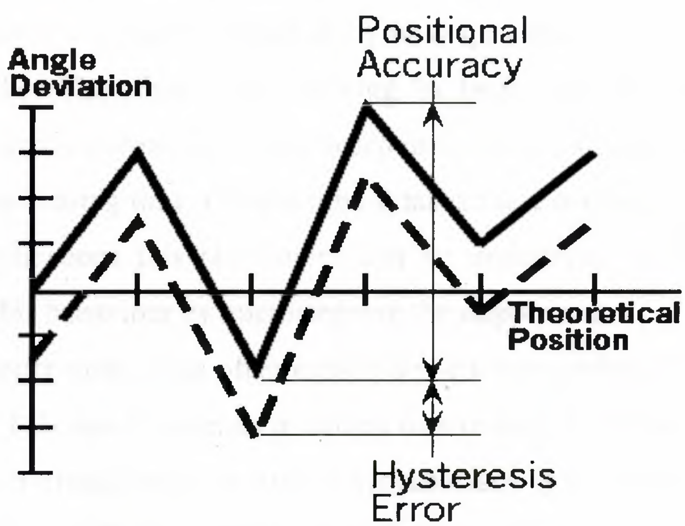


Fig 1.25 Positional accuracy of a stepper motor.

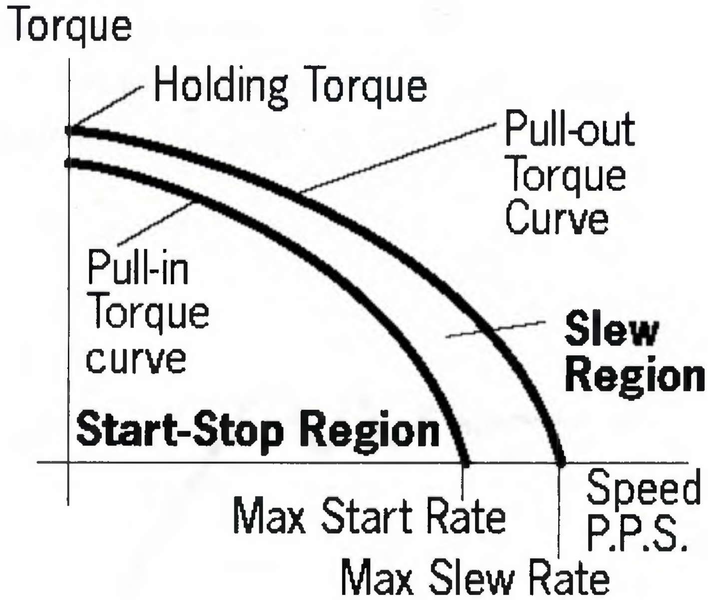


Fig 1.26 Torque vs. Speed charecteristic of a stepper motor

1.7. Single Step Response and Resonances

The single-step response characteristics of a stepper motor is shown in figure 1.27 When one step pulse is applied to a stepper motor the rotor behaves in a manner as defined by the above curve. The step time t is the time it takes the motor shaft to rotate one step angle once the first step pulse is applied. This step time is highly dependent on the ratio of torque to inertia (load) as well as the type of driver used.

Since the torque is a function of the displacement it follows that the acceleration will also be. Therefore, when moving in large step increments a high torque is developed and consequently a high acceleration. This can cause overshots and ringing as shown. The settling time T is the time it takes these oscillations or ringing to cease. In certain applications this phenomena can be undesirable. It is possible to reduce or eliminate this behaviour by microstepping the stepper motor.

Stepper motors can often exhibit a phenomena refered to as resonance at certain step rates. This can be seen as a sudden loss or drop in torque at certain speeds which can result in missed steps or loss of synchronism. It occurs when the input step pulse rate coincides with the natural oscillation frequency of the rotor. Often there is a resonance area around the 100 – 200 pps region and also one in the high step pulse rate

region. The resonance phenomena of a stepper motor comes from its basic construction and therefore it is not possible to eliminate it
completely. It is also dependent upon the load conditions. It can be reduced by driving the motor in half or microstepping modes.

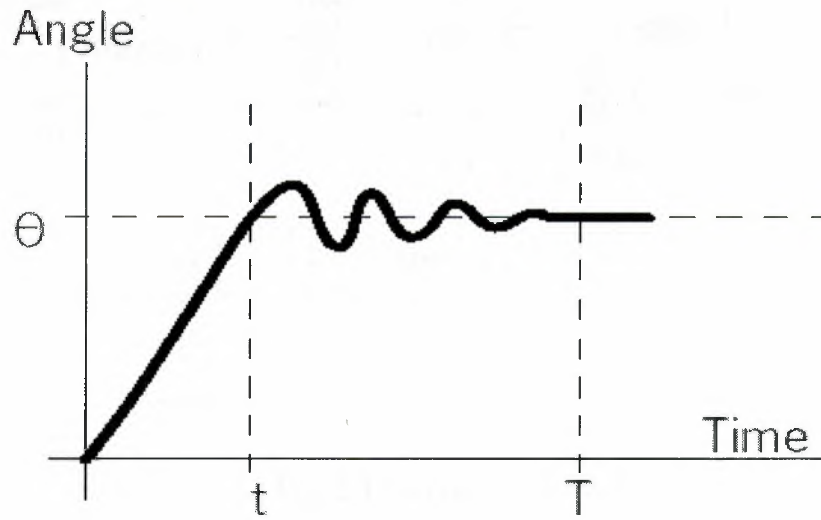


Fig 1.27 Single step vs time .

2. Step Motor Drive Technologies

2.1. Stepping Motor Drives

The stepper drive delivers electrical power to the motor in response to low-level signals from the control system. The motor is a torque-producing device, and this torque is generated by the interaction of magnetic fields. The driving force behind the stator field is the magneto-motive force (MMF), which is proportional to current and to the number of turns in the winding. This is often referred to as the amp-turns product. Essentially, the drive must act as a source of current. The applied voltage is only significant as a means of controlling the current.

Input signals to the stepper drive consist of step pulses and a direction signal. One step pulse is required for every step the motor is to take. This is true regardless of the stepping mode. So the drive may require 200 to 101,600 pulses to produce one revolution of the shaft. The most commonly-used stepping mode in industrial applications is the halfstep mode in which the motor performs 400 steps per revolution. At a shaft speed of 1800 rpm, this corresponds to a step pulse frequency of 20kHz. The same shaft speed at 25,000 steps per rev requires a step frequency of 750 kHz, so motion controllers controlling microstep drives must be able to output a much higher step frequency.

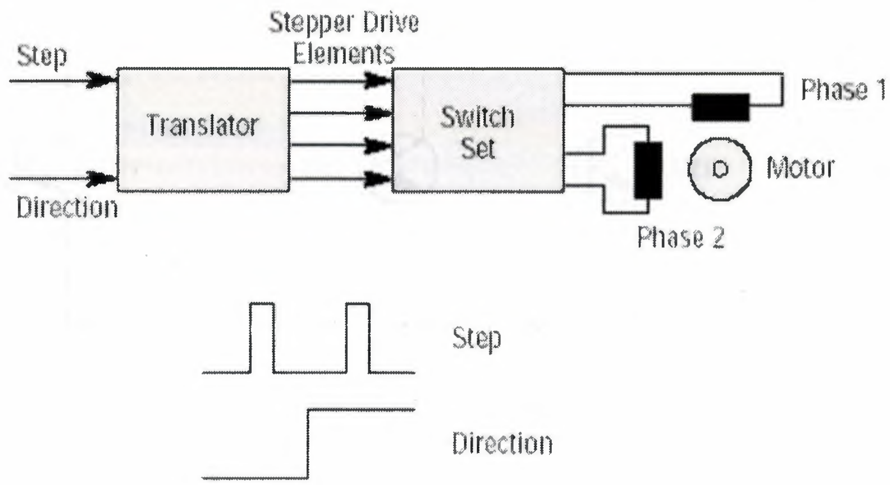


Fig. 2.1 Stepper drive elements

The logic section of the stepper drive is often referred to as the translator. Its function is to translate the step and direction signals into control waveforms for the switch set (see Fig. 2.1). The basic translator functions are common to most drive types, although the translator is necessarily more complex in the case of a microstepping drive. However, the design of the switch set is the prime factor in determining drive performance, so we will look at this in more detail.

The simplest type of switch set is the unipolar arrangement shown in Fig. 2.2. It is referred to as a unipolar drive because current can only flow in one direction through any particular motor terminal. A bifilar-wound motor must be used since reversal of the stator field is achieved by transferring current to the second coil. In the case of this very simple drive, the current is determined only by the motor winding resistance and the applied voltage.

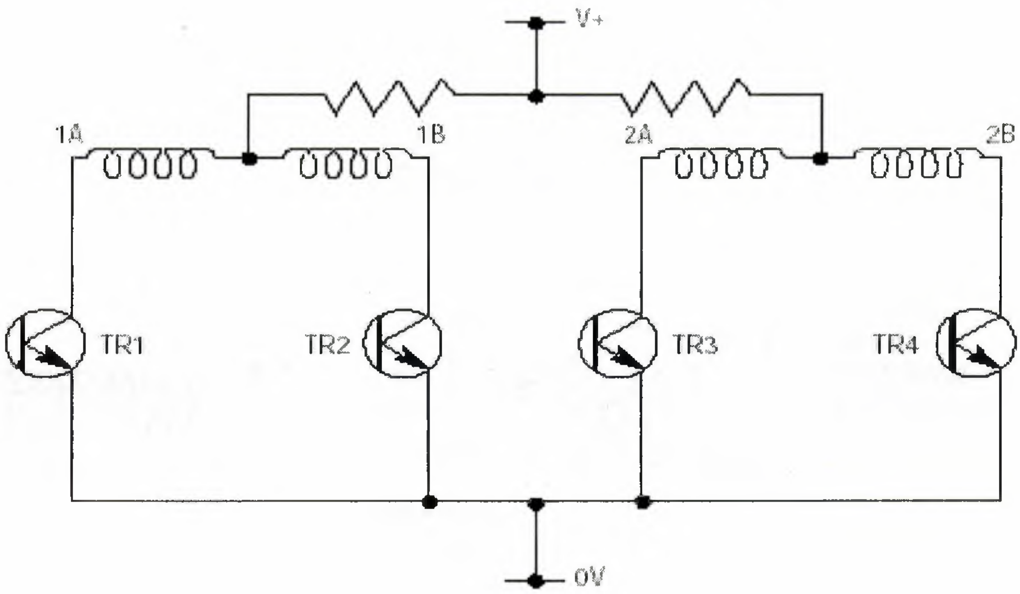


Fig. 2.2 Basic unipolar drive

Such a drive will function perfectly well at low stepping rates, but as speed is increased, the torque will fall off rapidly due to the inductance of the windings.

2.1.1. Inductance/Water Analogy

For those not familiar with the property of inductance, the following water analogy may be useful (Fig. 2.3). An inductor behaves in the same way as a turbine connected to a flywheel. When the tap is turned on and pressure is applied to the inlet pipe, the turbine will take time to accelerate due to the inertia of the flywheel. The only way to increase Applying a voltage to the terminals of an inductor produces a similar effect. With a pure inductance (i.e., no resistance), the current will rise in a linear fashion for as long as the voltage is applied. The rate of rise of current depends on the inductance and the applied voltage, so a higher voltage must be applied to get the current to rise more quickly. In a practical inductor possessing resistance, the final current is determined by the resistance and the applied voltage. Once the turbine has been accelerated up to speed, stopping it again is not a simple matter. The kinetic energy of the flywheel has to be dissipated, and as soon as the tap is turned off, the flywheel drives the turbine like a pump and tries to keep the water flowing. This will set up a high pressure across the inlet and outlet pipes in the reverse direction. The equivalent energy store in the inductor is the magnetic field. As this field collapses, it tries to maintain the current flow by generating a high reverse voltage.

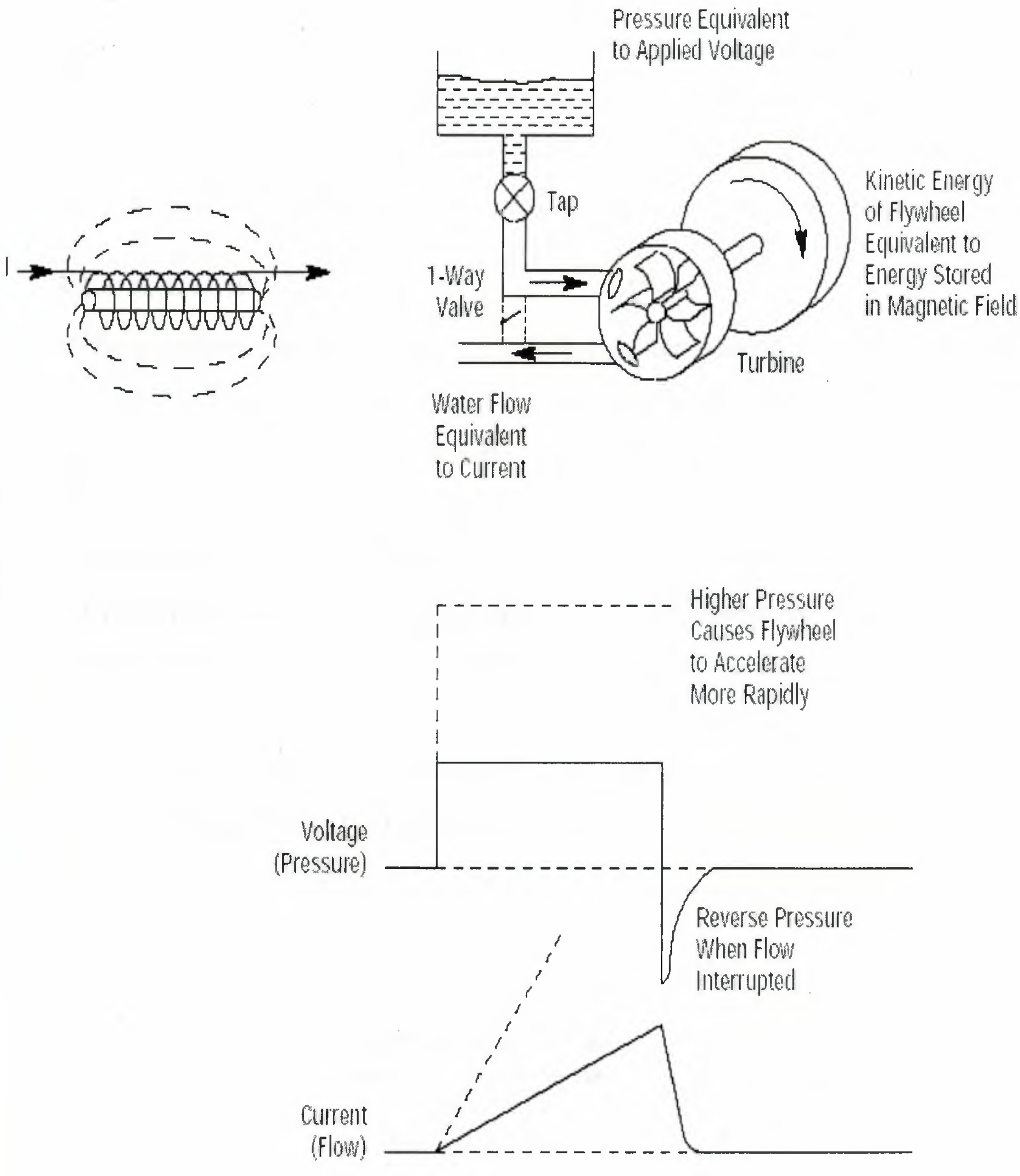


Fig. 2.3 Inductance water analogy

the acceleration rate is to increase the applied pressure. If there is no friction or leakage loss in the system, acceleration will continue indefinitely for as long as the pressure is applied. In a practical case, the final speed will be determined by the applied pressure and by friction and the leakage past the turbine blades. By including a one-way valve across the turbine connections, the water is allowed to continue circulating when the tap is turned off. The energy stored in the flywheel is now put to good use in maintaining the flow. We use the same idea in the recirculating chopper drive, in which a diode allows the current to recirculate after it has built up. Going back to our simple unipolar drive, if we look at the way the current builds up (Fig. 2.4) we can see that it follows an exponential shape with its final value set by the voltage and the winding resistance. To get it to build up more rapidly, we could increase the applied voltage, but this would also increase the final current level. A simple way to alleviate this problem is to add a resistor in series with the motor to keep the current the same as before.

2.1.2. R-L Drive

The principle described in the Inductance/Water Analogy (p. A24) is applied in the resistance-limited (R-L) drive see Fig. 2.4. Using an applied voltage of 10 times the rated motor voltage, the current will reach its final value in one tenth of the time. If you like to think in terms of the electrical time constant, this has been reduced from L/R to L/10R, so we'll get a useful increase in speed. However we're paying a price for this extra performance. Under steady-state conditions, there is 9 times as much power dissipated in the series resistor as in the motor itself, producing a significant amount of heat. Furthermore, the extra power must all come from the DC power supply, so this must be much larger. R-L drives are therefore only suited to low-power applications, but they do offer the benefits of simplicity, robustness and low radiated interference.

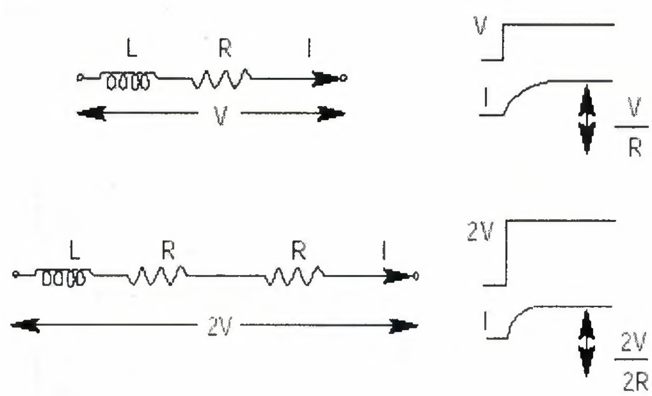


Fig. 2.4 Principle of the R-L drive

2.1.3. Unipolar Drive

A drawback of the unipolar drive is its inability to utilize all the coils on the motor. At any one time, there will only be current flowing in one half of each winding. If we could utilize both sections at the same time, we could get a 40% increase in ampturns for the same power dissipation in the motor. To achieve high performance and high efficiency, we need a bipolar drive (one that can drive current in either direction through each motor coil) and a better method of current control. Let's look first at how we can make a bipolar drive.

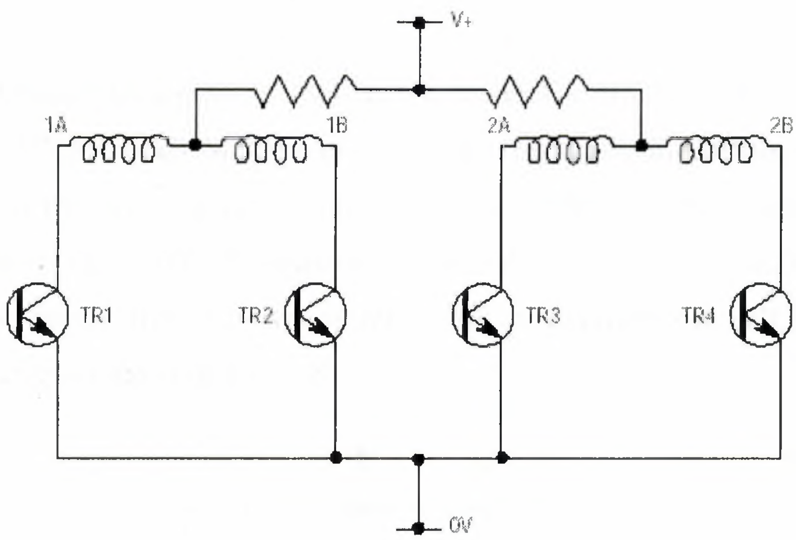


Fig . 2.5. Basic Unipolar Drive

2.1.4. Bipolar Drive

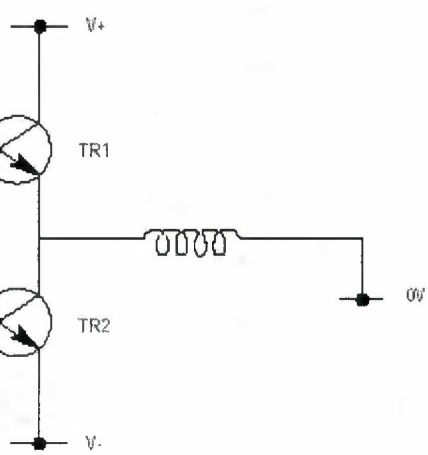


Fig 2.6. Simple Bipolar Drive

An obvious possibility is the simple circuit shown in Fig. 2.6, in which two power supplies are used together with a pair of switching transistors. Current can be made to flow in either direction through the motor coil by turning on one transistor or the other. However, there are distinct drawbacks to this scheme. First, we need two power supplies, both of which must be capable of delivering the total current for both motor phases. When all the current is coming from one supply the other is doing nothing at all, so the power supply utilization is poor. Second, the transistors must be rated at double the voltage that can be applied across the motor, requiring the use of costly components.

The standard arrangement used in bipolar motor drives is the bridge system shown in Fig. 2.7 Although this uses an extra pair of switching transistors, the problems associated with the previous configuration are overcome. Only one power supply is needed and this is fully utilized; transistor voltage ratings are the same as that available for driving the motor. In low-power systems, this arrangement can still be used with resistance limiting as shown in Fig. 2.8.

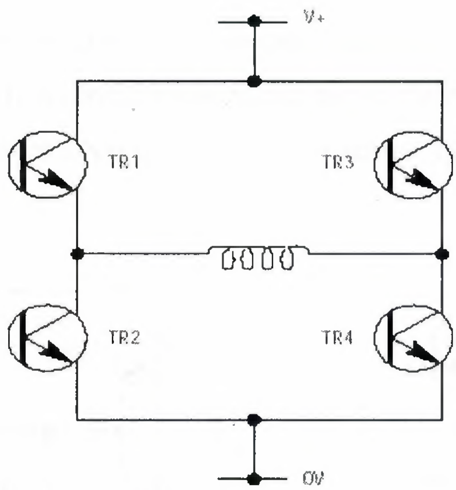


Fig. 2.7 Bipolar bridge

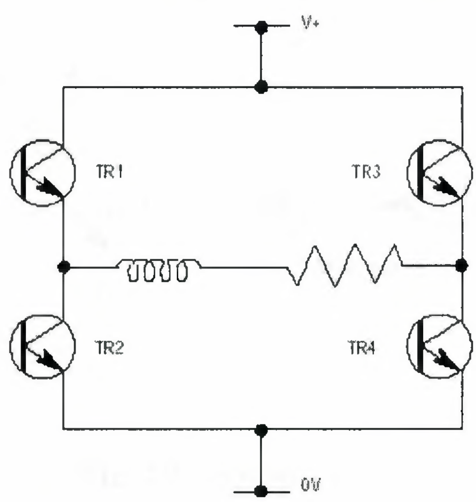


Fig. 2.8. Bipolar R-L Drive

2.1.5. Recirculating Chopper Drive

The method of current control used in most stepper drives is the recirculating chopper (Fig. 2.9). This approach incorporates the four-transistor bridge, recirculation diodes, and a sense resistor. The resistor is of low value (typically 0.1 ohm) and provides a feedback voltage proportional to the current in the motor. Current is injected into the winding by turning on one top switch and one bottom switch, and this applies the full supply voltage across the motor. Current will rise in an almost linear fashion and we can monitor this current by looking across the sense resistor. When the required current level has been reached, the top switch is turned off and the stored energy in the coil keeps the current circulating via the bottom switch and the diode. Losses in the system cause this current to slowly decay, and when a pre-set lower threshold is reached, the top switch is turned back on and the cycle repeats. The current is therefore maintained at the correct average value by switching or "chopping" the supply to the motor.

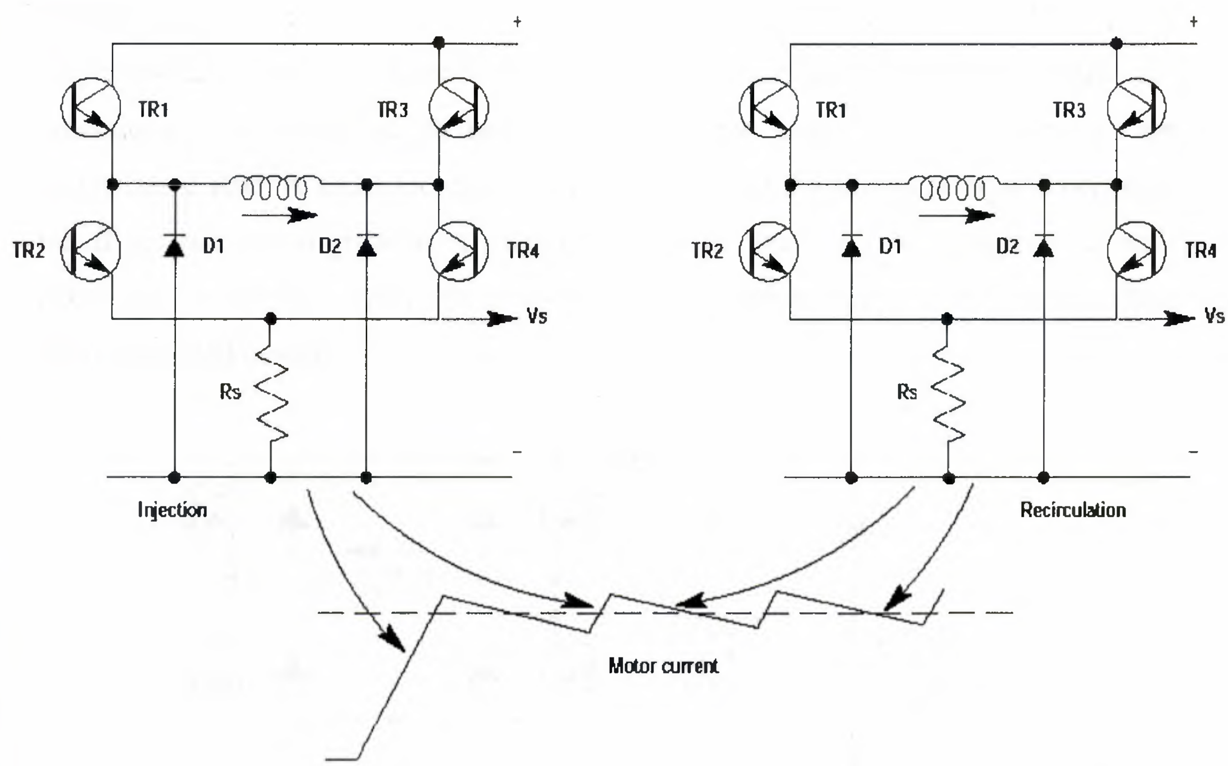


Fig. 2.9 Recirculating chopper drive

This method of current control is very efficient because very little power is dissipated in the switching transistors other than during the transient switching state. Power drawn from the power supply is closely related to the mechanical power delivered by the shaft (unlike theR-L drive, which draws maximum power from the supply at standstill). A variant of this circuit is the regenerative chopper. In this drive, the supply voltage is applied across the motor winding in alternating directions, causing the current to ramp up and down at approximately equal rates. This technique tends to require fewer components and is consequently lower in cost, however, the associated ripple current in the motor is usually greater and increases motor heating.

2.1.6. Regeneration and Power Dumping

Like other rotating machines with permanent magnets, the step motor will act as a generator when the shaft is driven mechanically. This means that the energy imparted to the load inertia during acceleration is returned to the drive during deceleration. This will increase the motor current and can damage the power switches if the extra current is excessive. A threshold detector in the drive senses this increase in current and momentarily turns off all the bridge transistors (Fig. 2.10). There is now a path for the regenerated current back to the supply capacitor, here it increases the supply voltage. During this phase, the current is no longer flowing through the sense resistors, so the power switches must be turned on again after a short period (typically 30µS) for conditions to be eassessed. If the current is still too high, the drive returns to the regenerative state. A small increase in supply voltage during regeneration is acceptable, but if the rise is too great the switches may be damaged by overvoltage rather than excessive current. To resolve this problem, we use a power dump circuit that dissipates the regenerated power.

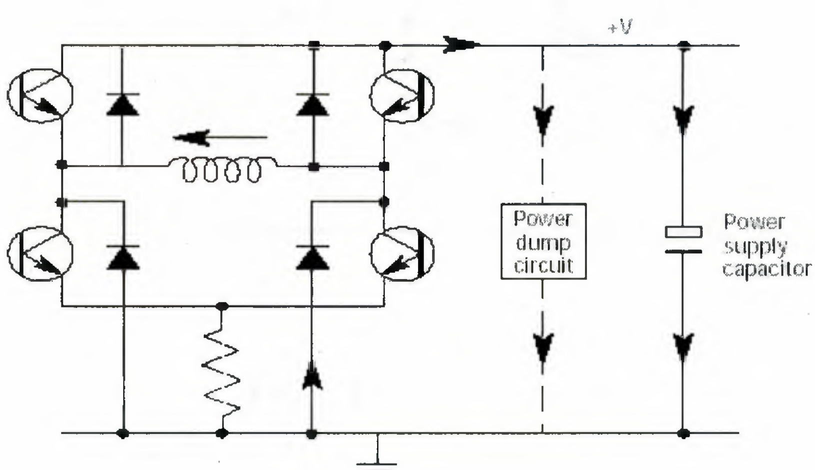


Fig. 2.10 Current flow during regeneration

The circuit of a simple power dump is shown in Fig. 2.11. A rectifier and capacitor fed with C from the supply transformer provide a reference voltage equal to the peak value of the incoming AC. Under normal conditions this will be the same as the drive supply voltage. During excess regeneration, the drive supply voltage will rise above this reference, and this ill turn on the dump transistor connecting the 33-ohm resistor across the power supply. When the supply voltage has decreased sufficiently, the transistor is turned back off. Although the instantaneous current flowing through the dump resistor may be relatively high, the average power dissipated is usually small since the dump period is very short. In applications where t the regenerated power is high, perhaps caused by frequent and rapid deceleration of a high inertia, a supplementary high-power dump resistor may be necessary.

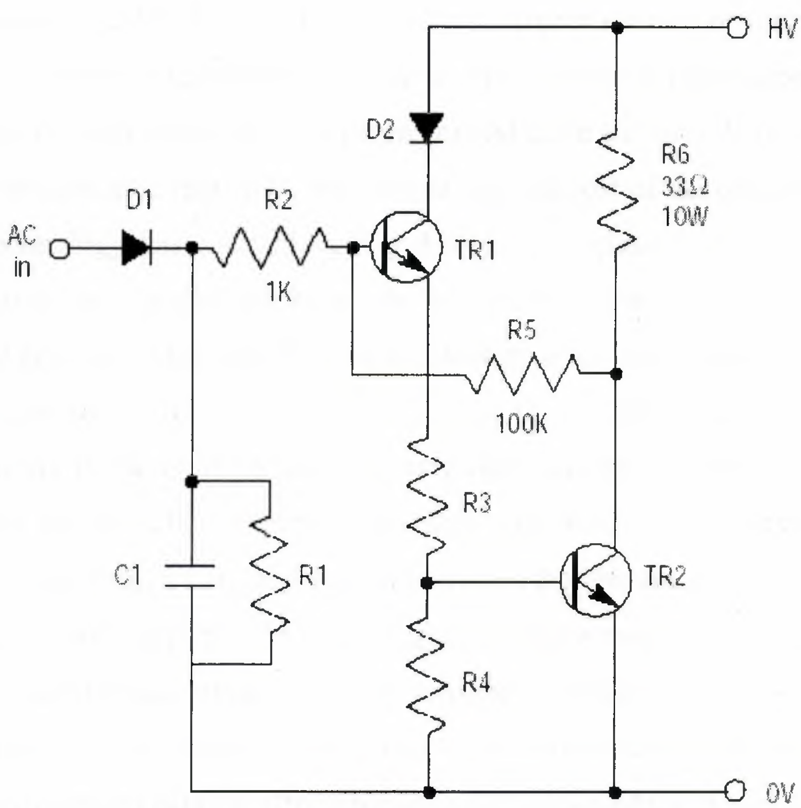


Fig. 2.11 Power dump circuit

2.1.7. Stepper Drive Technology Overview

Within the various drive technologies, there is a spectrum of performance. The uni-polar resistancelimited (R-L) drive is a relatively simple design, but it lacks shaft power performance and is very inefficient. A uni-polar system only uses half of the motor winding at any instant. A bi-polar design allows torque producing current to flow in all motor windings, sing the motor more efficiently, but increasing the complexity of the drive. A bi-polar R-L drive improves shaft performance, but is still very inefficient—generating a lot of wasted eat. An alternative to resistance-limiting is to control current by means of chopper regulation. ÊA chopper regulator is very efficient since it does not waste power by dropping voltage through a resistor. However, good current control in the motor is essential to deliver optimum shaft power. Pulse width modulation (PWM) and threshold modulation are two types of chopper regulation techniques. PWM controls the average of the motor current and is very good for precise current control, while threshold modulation controls current to a peak level. Threshold modulation can be applied to a wider range of motors, but it does suffer greater loss of performance than PWM when the motor has a large resistance or long motor cables are used. Both chopper regulation techniques can use recirculating current control, which improves the power dissipation in the motor and drive and overall system efficiency. As system performance increases, the complexity and cost of the drive increases. Stepper drive technology has evolved—being driven by machine builders that require more shaft power in smaller packages, higher speed capability, better efficiency, and improved accuracy. One trend of the technology is towards microstepping, a technique that divides each full step of the motor into smaller steps. This is achieved electronically in the drive by proportioning the current between the motor windings. The higher the resolution, the more precision is required in the current control circuits. In its simplest form, a half-step system increases the resolution of a standard 1.8° full-step motor to 400 steps/rev. Ministepping drives have more precise current control and can increase the resolution to 4,000 steps/rev. Microstep drives typically have resolutions of 50,000 steps/rev, and in addition to improved current control, they often have adjustments to balance offsets between each phase of the motor and to optimize the current profile for the particular motor being used.

2.1.8. Full-Step and Half-Step Systems

Full-step and half-step systems do not have the resolution capability of the ministepping or microstepping systems. However, the drive technology is not as complex and the drives are relatively inexpensive. Full-step and half-step systems will not have the same low-speed s smoothness as higher resolution systems. An inherent property of a stepper motor is its lowspeed resonance, which may de-synchronize a motor and cause position loss. Full-step and halfstep drives are more prone to resonance effects and this may limit their application in low-speed systems. Full-step and half-step systems can be operated at speeds above the motor's resonant speed without loss of synchronization. For this reason, full-step and half- step systems are normally applied in high-speed, point-to-point positioning applications. In these types of applications, the machine designer is primarily concerned with selecting a motor/drive system capable of producing the necessary power output. Since power is the product of torque and speed, a high-torque system with low-speed capability may not produce is much power as a low-torque, highspeed system. Sizing the system for torque only may not provide the most cost-effective solution, selecting a system based on power output will make the most efficient use of the motor and drive. Step motor systems typically require the motor to accelerate to reach high speed. If a motor was requested to run instantaneously at 3000 r rpm, the motor would stall immediately. At slow speeds, it is possible to start the motor without position loss by applying unramped step pulses. The maximum speed at which synchronization will occur without ramping is called the *start/stop velocity.* The start/stop v velocity is inversely proportional to the square-root of the total inertia. The start/stop capability provides a benefit for applications that require high-speed point-to-point positioning—since the acceleration to the start/stop velocity is almost instantaneous, the move-time will be reduced. No additional time is required to accelerate the motor from zero to the start/stop velocity. While the move-time can be reduced, it is generally more complicated for the controller or indexer to calculate the motion profile and implement a start/stop velocity. In most applications, using start/ stop velocities will eliminate the need to run the motor at its resonant frequency and prevent desynchronization.

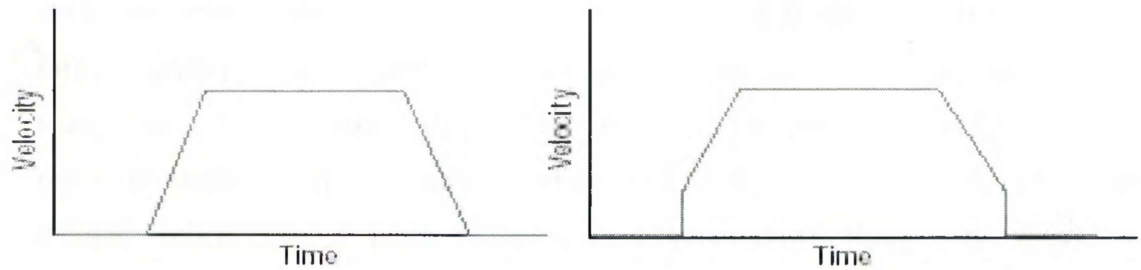


Fig 2.11. Start /Stop velocity (velocity vs. time)

2.1.9. Ministep Systems

Applications that require better low-speed smoothness than a half-step system should consider using a microstepping or ministepping solution. Microstepping systems, with resolutions of 50,000 steps/rev, can offer exceptional smoothness, without requiring a gear-reducer. Ministepping systems typically do not have wavetrimming capability or offset adjustment to achieve the optimum smoothness, but offer a great improvement over full-step and half-step systems. Ministepping systems have resolutions between 1,000 and 4,000 steps/rev. The motor is an important element in providing good smoothness. Some motor designs are optimized for high-torque output rather than smooth rotation. Others are optimized for smoothness rather than high torque. Ministepping systems are typically offered with a motor as a “packaged” total solution, using a motor that has been selected for its premium smoothness properties. Ministep systems are sometimes selected to improve positional accuracy. However, with an open-loop system, friction may prevent the theoretical unloaded accuracy from being achieved in practice.

2.1.10. Microstepping Drives

As we mentioned earlier, subdivision of the basic motor step is possible by proportioning the current in the two motor windings. This produces a series of intermediate step positions between the onephase- on points. It is clearly desirable that these intermediate positions are equally spaced and produce approximately equal torque when the motor is running. Accurate microstepping places increased demands on the accuracy of current control in the drive, particularly at low current levels. A small phase imbalance that may be barely detectable in a halfstep drive can produce unacceptable positioning errors in a microstep system. Pulse-width modulation is frequently used to achieve higher accuracy than can be achieved using a simple threshold system. The phase currents necessary to produce the intermediate steps follow an approximately sinusoidal profile as shown in Fig. 2.12. However the same profile will not give the optimum response with all motors. Some will work well with a sinusoidal shape, whereas others need a more filled-out or trimmed-down shape (Fig. 2.12). So a microstep drive intended to operate with a variety of motors needs to have provision for adjusting the current profile. The intermediate current levels are usually stored as data in an EPROM, with some means of selecting alternative data sets to give different profiles. The change in profile may be thought of in terms of adding or subtracting a third-harmonic component to or from the basic sine wave.

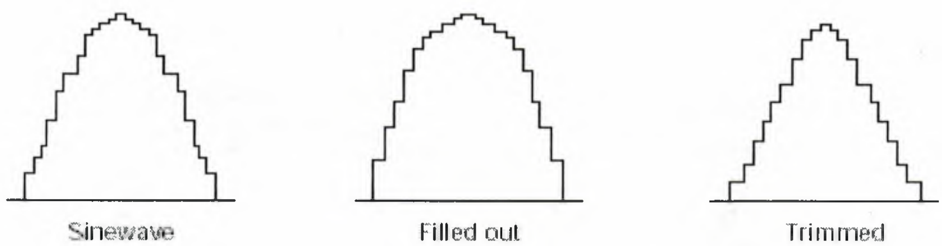


Fig. 2.12 Microstep current profile

In the case of high-resolution microstep drives producing 10,000 steps per rev or more, the best performance will only be obtained with a particular type of motor. This is one in which t the stator teeth are on a 7.5° pitch, giving 48 equal pitches in 360°. In most hybrid steppers, the stator teeth have the same pitch as the rotor teeth, giving equal increments of 7.2°. This latter arrangement tends to give superior torque output, but is less satisfactory as a microstepper since the magnetic poles are “harder” – there is no progressive transfer of tooth alignment from one pole to the next. In fact, with this type of motor, it can be quite difficult to find a current profile that gives good static positioning combined with smooth low-speed rotation. An alternative to producing a 7.5°-pitch stator is to incorporate a slight skew in the motor teeth. This produces a similar effect and has the benefit of using standard 7.2° laminations throughout.

Skewing is also used in DC brush motors as a means of improving smoothness. Due to this dependence on motor type for performance, it is usual for high-resolution microstep systems to be supplied as a matched motor-drive package.

2.2. The Stepper Torque/Speed Curve

We have seen that motor inductance is the factor that opposes rapid changes of current and therefore makes it more difficult to drive a stepper at high speeds. Looking at the torque- speed curve in Fig. 2.13, we can see what is going on. At low speeds, the current has plenty of time to reach the required level and so the average current in the motor is very close to the regulated value from the drive. Changing the regulated current setting or changing to a drive with a different current rating will affect the available torque accordingly.

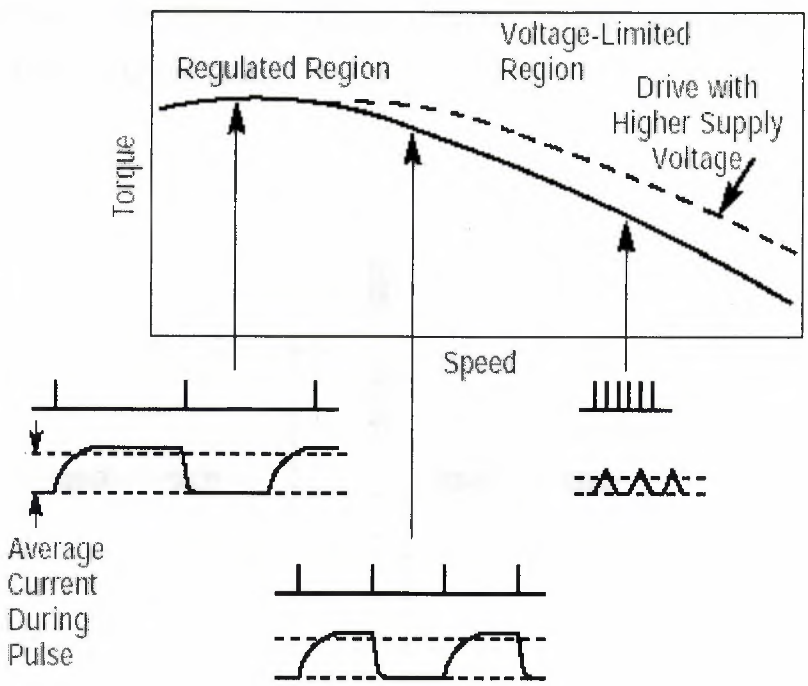


Fig 2.13 Regulated and voltage-limited regions of the torque-speed curve

As speed increases, the time taken for the current to rise becomes a significant proportion of the interval between step pulses. This reduces the average current level, so the torque starts to fall off. As speed increases further, the interval between step pulses does not allow the current time to reach a level where the chopping action can begin. Under these conditions, the final value of current depends only on the supply voltage. If the voltage is increased, the current will increase more rapidly and hence will achieve a higher value in the available time. So this region of the curve is described as "voltage limited", as a change in the drive current setting would have no effect. We can conclude that at low speeds the torque depends on the drive current setting, whereas at high speeds it depends on the drive supply voltage. It is clear that highspeed performance is not affected by the drive current setting. Reducing the current simply "flattens out" the torque curve without restricting the ability to run at high speeds. When performance is limited by the available high-speed torque, there is much to be said for running at the lowest current that gives an adequate torque margin. In general, dissipation in motor and drive is reduced and lowspeed performance in particular will be smoother with less audible noise. With a bipolar drive, alternative possibilities exist for the motor connections as shown in Fig. 2.14. An 8-lead motor can be connected with the two halves of each winding either in series or in parallel. With a 6-lead motor, either one half-winding or both half- windings may be connected in series. The alternative connection schemes produce different torque-speed characteristics and also affect the motor's current rating.

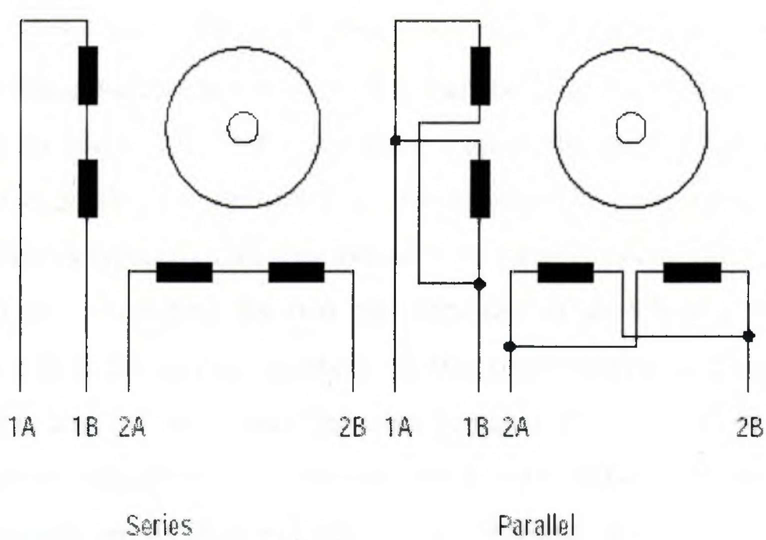


Fig. 2.14 Series & parallel connections

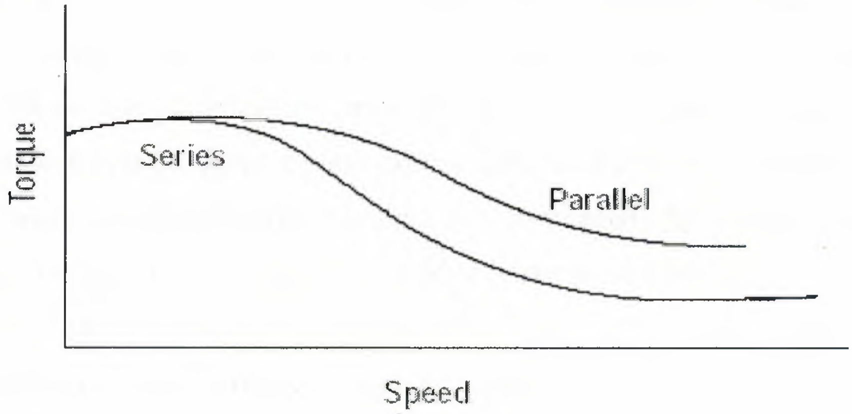


Fig. 2.15 Series & parallel torque/speed curves

Compared with using one half-winding only, connecting both halves in series requires the drive current to flow through twice as many turns. For the same current, this doubles the "amp-turns" and produces a corresponding increase in torque. In practice, the torque increase is seldom as high as 100% due to the non-linearity of the magnetic material. Equally, the same torque will be produced at half the drive current when the windings are in series. However, having doubled the effective number of turns in the winding means that we have also increased the inductance by a factor of 4. This causes the torque to drop off much more rapidly as speed is increased, and as a result, the series mode is most useful at low speeds. The maximum shaft power obtainable in series is typically half that available in parallel (using the same current setting on the drive). Connecting the two half-windings of an 8-lead motor in parallel allows the current to divide itself between the two coils. It does not change the effective number of turns and the inductance therefore remains the same. So at a given drive current, the torque characteristic will be the same for two half-windings in parallel as for one of the windings on its own. For this reason, "parallel" in the context of a 6-lead motor refers to the use of one half-winding only. As has already been mentioned, the current rating of a step motor is determined by the allowable temperature rise. Unless the motor manufacturer's data states otherwise, the rating is a "unipolar" value and assumes both phases of the motor are energized simultaneously. So a current rating of 5A means that the motor will accept 5A flowing in each half-winding. When the windings of an 8-lead motor are connected in parallel, half of the total resistance is produced. For the same power dissipation in the motor, the current may now be increased by 40%. Therefore, the 5A motor will accept 7A with the windings in parallel, giving a significant increase in available torque. Conversely, connecting the windings in series will double the total resistance and the current rating is reduced by a factor of 1.4, giving a safe current of 3.5A for our 5A-motor in series. As a general rule, parallel is the preferred connection method as it produces a flatter torque curve and greater shaft power (Fig. 2.15). Series is useful when high torque is required at low speeds, and it allows the motor to produce full torque from a lower-current drive. Care should be taken to avoid overheating the motor in series since its current rating is lower in this mode. Series configurations also carry a greater likelihood of resonance due to the high torque produced in the low-speed region.

2.3. Common Questions and Answers About Step Motors

1. Why do step motors run hot?

Two reasons: 1. Full current flows through the motor windings at standstill. 2. PWM drive designs tend to make the motor run hotter. Motor construction, such as lamination material and riveted rotors, will also affect heating.

2. What are safe operating temperatures?

The motors have class B insulation, which is rated at 130°C. Motor case temperatures of 90°C will not cause thermal breakdowns. Motors should be mounted where operators cannot come into contact with the motor case.

3. What can be done to reduce motor heating?

Many drives feature a "reduce current at standstill" command or jumper. This reduces current when the motor is at rest without positional loss.

4. What does the absolute accuracy specification mean?

This refers to inaccuracies, non-cumulative, encountered in machining the motor.

5. How can the repeatability specification be better than that of accuracy?

Repeatability indicates how precisely a previous position can be re-obtained. There are no inaccuracies in the system that affect a given position, returning to that position, the same inaccuracy is encountered.

6. Will motor accuracy increase proportionately with the resolution?

No. The basic absolute accuracy and hysteresis of the motor remain unchanged.

7. Can I use a small motor on a large load if the torque requirement is low?

Yes, however, if the load inertia is more than ten times the rotor inertia, cogging and extended ringing at the end of the move will be experienced.

8. How can end of move "ringing" be reduced?

Friction in the system will help damp this oscillation. Acceleration/deceleration rates could be increased. If start/stop velocities are used, lowering or eliminating them will help.

9. Why does the motor stall during no load testing?

The motor needs inertia roughly equal to its own inertia to accelerate properly. Any resonances developed in the motor are at their worst in a no-load condition.

10. Why is motor sizing important, why not just go with a larger motor?

If the motor's rotor inertia is the majority of the load, any resonances may become more pronounced. Also, productivity would suffer as excessive time would be required to accelerate the larger rotor inertia. Smaller may be better.

11. What are the options for eliminating resonance?

This would most likely happen with full step systems. Adding inertia would lower the resonant frequency. Friction would tend to dampen the modulation. Start/stop velocities higher than the resonant point could be used. Changing to half step operation would greatly help. Ministepping and microstepping also greatly minimize any resonant vibrations. Viscous inertial dampers may also help.

12. Why does the motor jump at times when it's turned on?

This is due to the rotor having 200 natural detent positions. Movement can then be ±3.6° in either direction.

13. Do the rotor and stator teeth actually mesh?

No. While some designs used this type of harmonic drive, in this case, an air gap is very carefully maintained between the rotor and the stator.

14. Does the motor itself change if a microstepping drive is used?

The motor is still the standard 1.8° stepper. Microstepping is accomplished by proportioning currents in the drive (higher resolutions result). Ensure the motor's inductance is compatible.

15. A move is made in one direction, and then the motor is commanded to move the same distance but in the opposite direction. The move ends up short, why?

Two factors could be influencing the results. First, the motor does have magnetic hysteresis that is seen on direction changes. This is in the area of 0.03°. Second, any mechanical backlash in the system to which the motor is coupled could also cause loss of motion.

16. Why are some motors constructed as eightlead motors?

This allows greater flexibility. The motor can be run as a six-lead motor with unipolar drives. With bipolar drives, the windings can then be connected in either series or parallel.

17. What advantage do series or parallel connection windings give?

With the windings connected in series, lowspeed torques are maximized. But this also gives the most inductance so performance at higher speeds is lower than if the windings were connected in parallel.

18. Can a flat be machined on the motor shaft?

Yes, but care must be taken to not damage the bearings. The motor must not be disassembled. Compumotor does not warranty the user's work.

19. How long can the motor leads be?

For bipolar drives, 100 feet. For unipolar designs, 50 feet. Shielded, twisted pair cables are required.

20. Can specialty motors, explosion-proof, radiation-proof, high-temperature, low-temperature, vacuum-rated, or waterproof, be provided?

Compumotor is willing to quote on most requirements with the exception of explosion proof.

21. What are the options if an explosion-proof motor is needed?

Installing the motor in a purged box should be investigated.

3. Pic 16F84A

3.1. High Performance RISC CPU Features:

• Only 35 single word instructions to learn

• All instructions single-cycle except for program branches which are two-cycle

• Operating speed: DC - 20 MHz clock input DC - 200 ns instruction cycle

• 1024 words of program memory

• 68 bytes of Data RAM

• 64 bytes of Data EEPROM

• 14-bit wide instruction words

• 8-bit wide data bytes

• 15 Special Function Hardware registers

• Eight-level deep hardware stack

• Direct, indirect and relative addressing modes

• Four interrupt sources:

- External RB0/INT pin
- TMR0 timer overflow
- PORTB<7:4> interrupt-on-change
- Data EEPROM write complete

3.2. Peripheral Features:

• 13 I/O pins with individual direction control

• High current sink/source for direct LED drive

- 25 mA sink max. per pin
- 25 mA source max. per pin

• TMR0: 8-bit timer/counter with 8-bit programmable prescaler

3.3. Special Microcontroller Features:

• 10,000 erase/write cycles Enhanced FLASH Program memory typical

• 10,000,000 typical erase/write cycles EEPROM Data memory typical

• EEPROM Data Retention > 40 years

• In-Circuit Serial Programming™ (ICSP™) – via two pins

• Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)

• Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation

• Code protection

• Power saving SLEEP mode

• Selectable oscillator options

3.4. Pin Diagrams

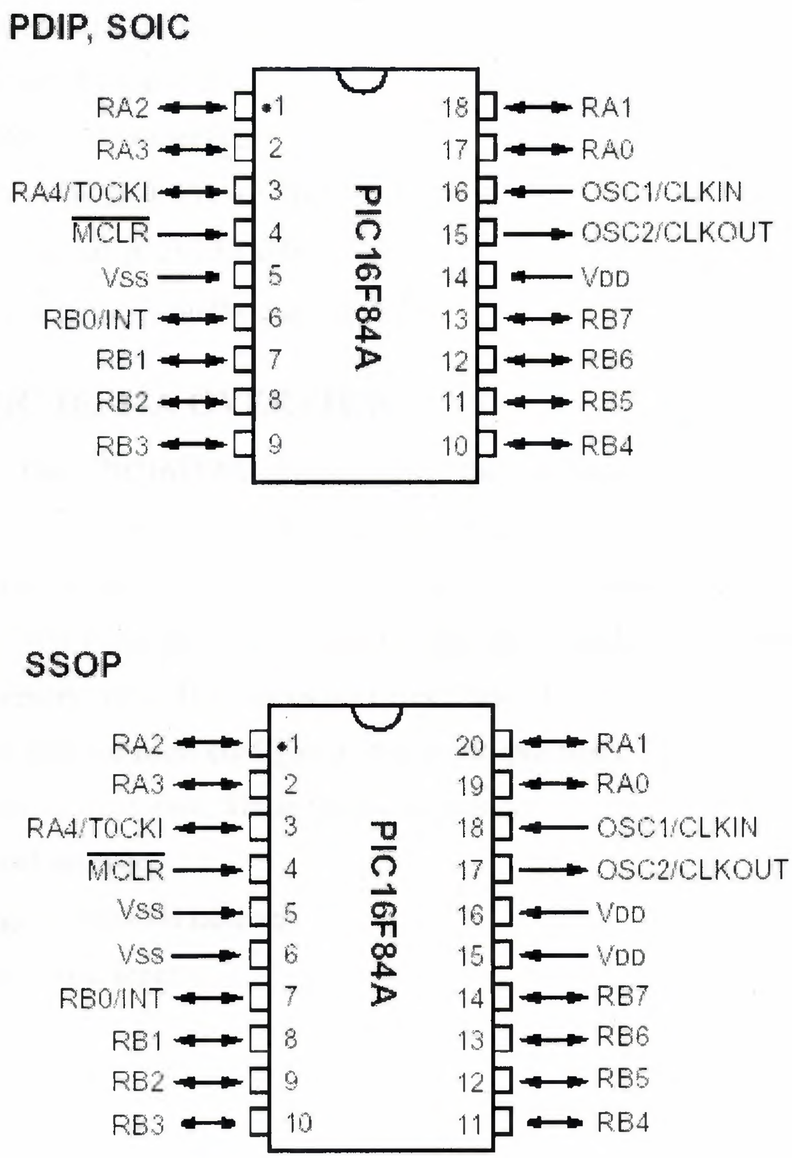


Fig 3.1.1 Pic 16f84 pin diagrams

3.5. CMOS Enhanced FLASH/EEPROM Technology:

• Low power, high speed technology

• Fully static design

• Wide operating voltage range:

- Commercial: 2.0V to 5.5V

- Industrial: 2.0V to 5.5V

• Low power consumption:

- < 2 mA typical @ 5V, 4 MHz

- 15 μA typical @ 2V, 32 kHz

- < 0.5 μA typical standby current @ 2V

3.6. PIC 16F84A OVERVIEW

The PIC16F84A belongs to the mid-range family of the PICmicro microcontroller devices. A block diagram of the device is shown in Figure 3.2.

The program memory contains 1K words, which translates to 1024 instructions, since each 14-bit program memory word is the same width as each device instruction. The data memory (RAM) contains 68 bytes. Data EEPROM is 64 bytes. There are also 13 I/O pins that are user- configured on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

• External interrupt

• Change on PORTB interrupt

• Timer0 clock input

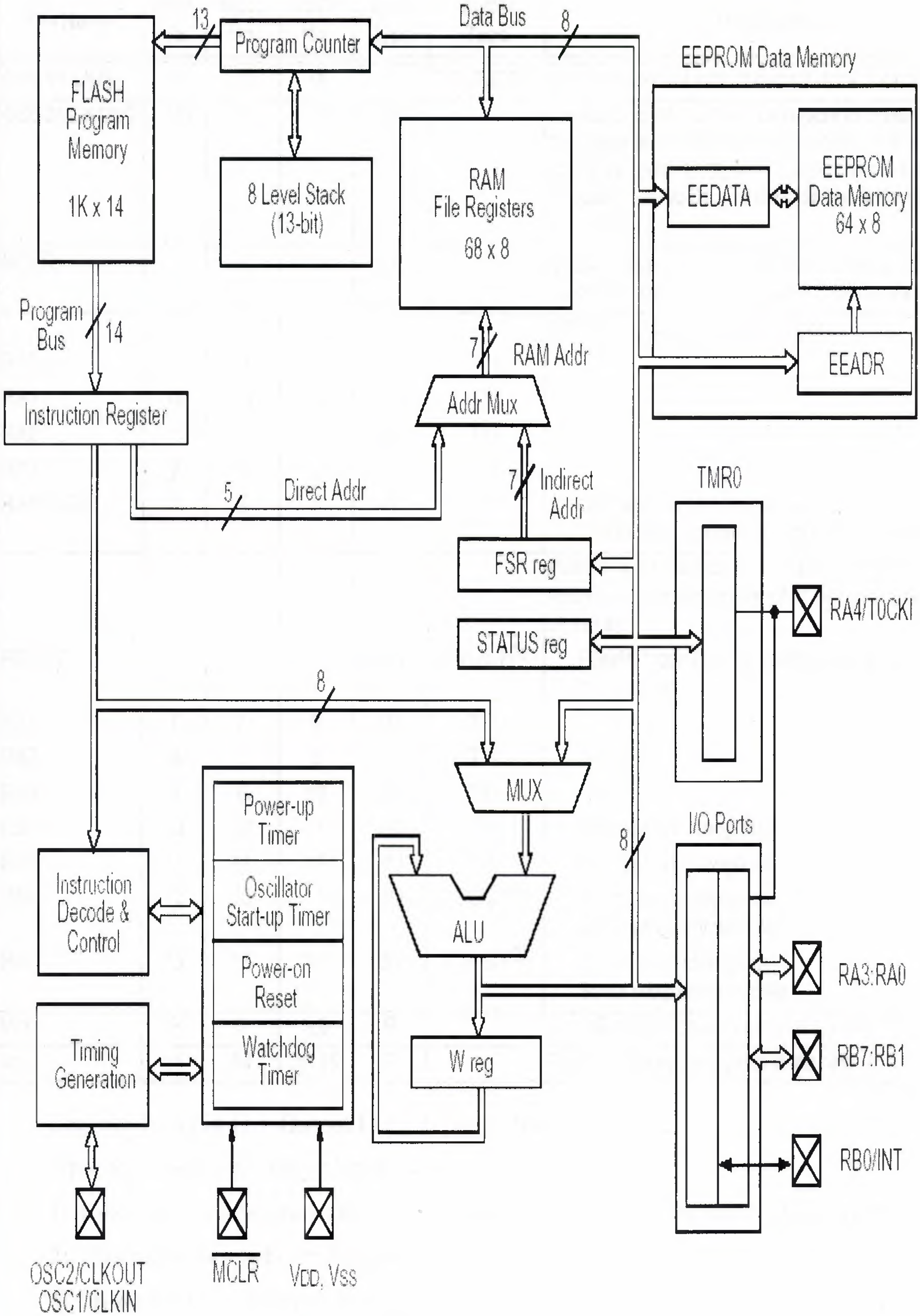


Fig 3.2. Pic 16f84 A block diagram

Table 1-1 details the pinout of the device with descriptions and details for each pin

Pin Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	18	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0	17	17	19	I/O	TTL	
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	7	I/O	TTL/ST(1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	Interrupt-on-change pin.
RB5	11	11	12	I/O	TTL	Interrupt-on-change pin.
RB6	12	12	13	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming clock.
RB7	13	13	14	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming data.
Vss	5	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = Output I/O = Input/Output P = Power — = Not used TTL = TTL input ST = Schmitt Trigger input

1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Table 1.1. PIC16F84A Pinout description

3.7. MEMORY ORGANIZATION

There are two memory blocks in the PIC16F84A. These are the program memory and the data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle. The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module. The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0h-3Fh.

3.7.1. Program Memory Organization

The PIC16FXX has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F84A, the first 1K x 14 (0000h-03FFh) are physically implemented (Figure 3.3.). Accessing a location above the physically implemented address will cause a wraparound. For example, for locations 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h, and 1C20h, the instruction will be the same. The RESET vector is at 0000h and the interrupt vector is at 0004h.

3.7.2. Data Memory Organization

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area, while the second is the General Purpose Registers (GPR) area. The SFRs control the operation of the device. Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM. The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. Figure 3.4. shows the data memory map organization. Instructions `MOVWF` and `MOVF` can move values from the W register to any location in the register file ("F"), and vice-versa. The entire data memory can be accessed either directly using the absolute address of each register file or indirectly through the File Select Register (FSR) Indirect addressing uses the present value of the RP0 bit for access into the banked areas of data memory. Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the RP0 bit (STATUS<5>). Setting the RP0 bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The first twelve locations of each Bank are reserved for the Special Function Registers.

3.8. GENERAL PURPOSE REGISTER FILE

Each General Purpose Register (GPR) is 8-bits wide and is accessed either directly or indirectly through the FSR. The GPR addresses in Bank 1 are mapped to addresses in Bank 0. As an example, addressing location 0Ch or 8Ch will access the same GPR.

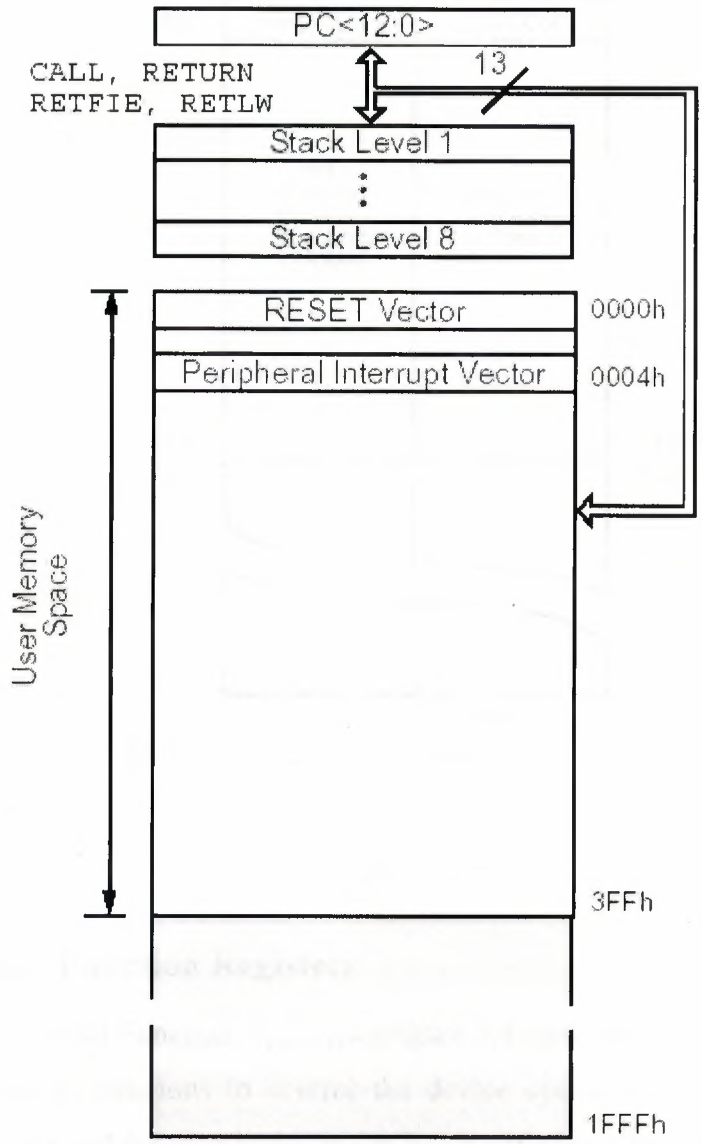


Fig 3.3. Program memory map and stack PIC 16F84A

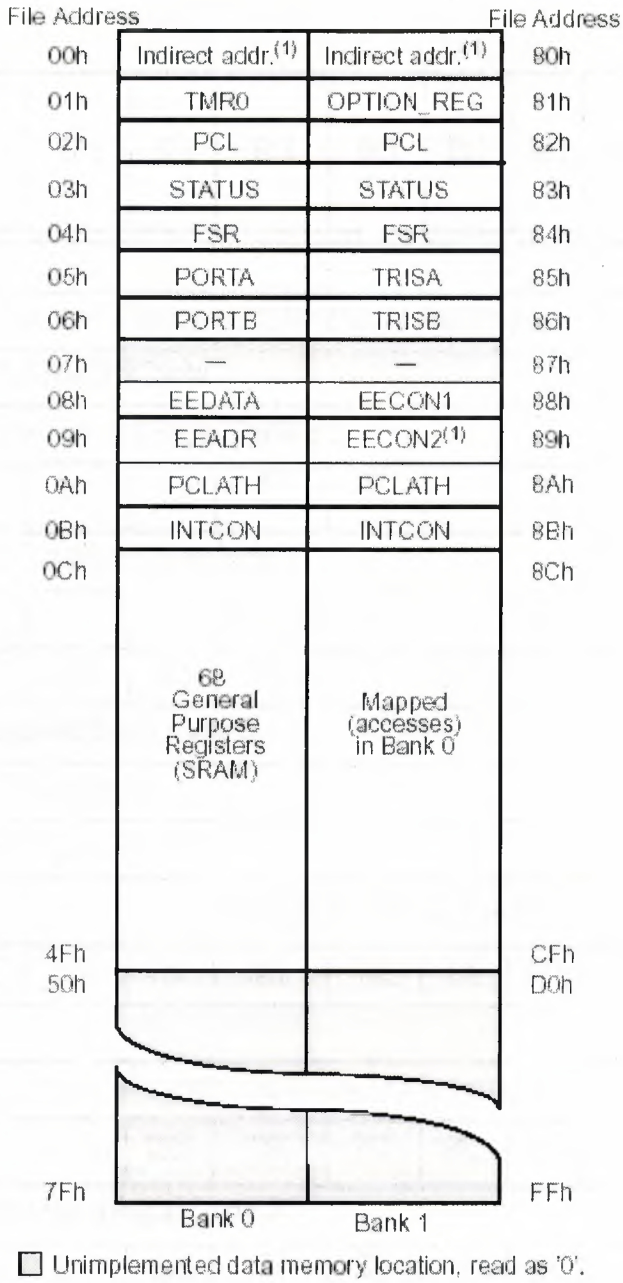


Fig 3.4 Register file map

PIC16F84A

3.9. Special Function Registers

The Special Function Registers (Figure 3.4. and Table 2-1) are used by the CPU and Peripheral functions to control the device operation. These registers are static RAM. The special function registers can be classified into two sets, core and peripheral. Those associated with the core functions are described in this section. Those related to the operation of the peripheral features are described in the section for that specific feature.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page
Bank 0											
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- ----	11
01h	TMR0	8-bit Real-Time Clock/Counter								xxxx xxxx	20
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11
03h	STATUS(2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11
05h	PORTA(4)	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	16
06h	PORTB(5)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	18
07h	—	Unimplemented location, read as '0'								—	—
08h	EEDATA	EEPROM Data Register								xxxx xxxx	13,14
09h	EEADR	EEPROM Address Register								xxxx xxxx	13,14
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC(1)					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10
Bank 1											
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
83h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	16
86h	TRISB	PORTB Data Direction Register								1111 1111	18
87h	—	Unimplemented location, read as '0'								—	—
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
89h	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	14
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC(1)					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10

Table 2.1. Special function register file

Legend: x = unknown, u = unchanged. - = unimplemented, read as '0', q = value depends on condition

1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> are never transferred to PCLATH.

2: The TO and PD status bits in the STATUS register are not affected by a MCLR Reset.

3: Other (non power-up) RESETS include: external RESET through MCLR and the Watchdog Timer Reset.

4: On any device RESET, these pins are configured as inputs.

5: This is the value that will be in the port output latch.

3.9.1. Status Register

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bit for data memory. As with any register, the STATUS register can be the destination for any instruction. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to device logic. Furthermore, the TO and PD bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended. For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged). Only the `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions should be used to alter the STATUS register, because these instructions do not affect any status bit.

bit 7-6 **Unimplemented:** Maintain as '0'

bit 5 **RP0**: Register Bank Select bits (used for direct addressing)

01 = Bank 1 (80h - FFh)

00 = Bank 0 (00h - 7Fh)

bit 4 **TO**: Time-out bit

1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction

0 = A WDT time-out occurred

bit 3 **PD**: Power-down bit

1 = After power-up or by the CLRWDT instruction

0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for borrow, the polarity is reversed)

1 = A carry-out from the 4th low order bit of the result occurred

0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for borrow, the polarity is reversed)

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

Note: A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'

- n = Value at POR '1'= Bit is set '0' = Bit is cleared x = Bit is unknown

3.9.2. Option Register

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull- ups on PORTB.

bit 7 **RBPU:** PORTB Pull-up Enable bit

1 = PORTB pull-ups are disabled

0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit

1 = Interrupt on rising edge of RB0/INT pin

0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit

1 = Transition on RA4/T0CKI pin

0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit

1 = Increment on high-to-low transition on RA4/T0CKI pin

0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit

1 = Prescaler is assigned to the WDT

0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

3.9.3. Intcon Register

The INTCON register is a readable and writable register that contains the various enable bits for all interrupt sources.

bit 7 **GIE:** Global Interrupt Enable bit

1 = Enables all unmasked interrupts

0 = Disables all interrupts

bit 6 **EEIE**: EE Write Complete Interrupt Enable bit

1 = Enables the EE Write Complete interrupts

0 = Disables the EE Write Complete interrupt

bit 5 **T0IE**: TMR0 Overflow Interrupt Enable bit

1 = Enables the TMR0 interrupt

0 = Disables the TMR0 interrupt

bit 4 **INTE**: RB0/INT External Interrupt Enable bit

1 = Enables the RB0/INT external interrupt

0 = Disables the RB0/INT external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit

1 = Enables the RB port change interrupt

0 = Disables the RB port change interrupt

bit 2 **T0IF**: TMR0 Overflow Interrupt Flag bit

1 = TMR0 register has overflowed (must be cleared in software)

0 = TMR0 register did not overflow

bit 1 **INTF**: RB0/INT External Interrupt Flag bit

1 = The RB0/INT external interrupt occurred (must be cleared in software)

0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF**: RB Port Change Interrupt Flag bit

1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)

0 = None of the RB7:RB4 pins have changed state

3.9.4. PCL and PCLATH

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 13 bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<12:8> bits and is not directly readable or writable. If the program counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`. All updates to the PCH registergo through the PCLATH register. The stack allows a combination of up to 8 program calls and interrupts to occur. The stack contains the return address from this branch in program execution. Mid-range devices have an 8 level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not modified when the stack is PUSHed or POPed. After the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push .

3.9.5. Indirect Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). This is indirect addressing.

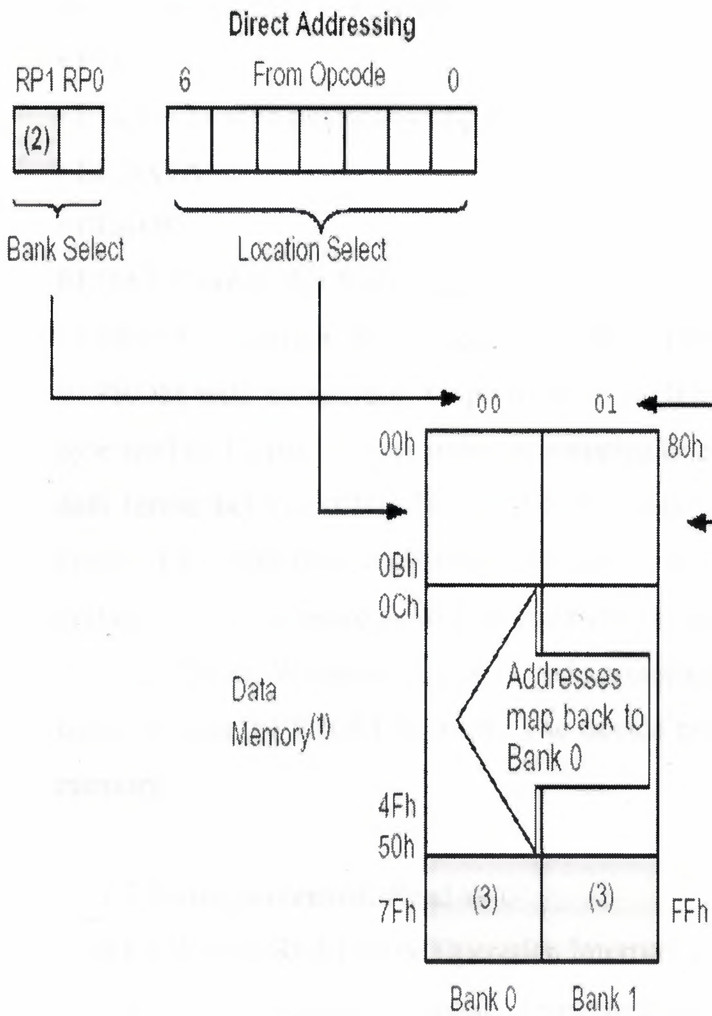


Fig 3.5. Direct / Indirect addressing

Note

1: For memory map detail, see Figure 3.3

2: Maintain as clear for upward compatibility with future products.

3: Not implemented.

3.10. DATA EEPROM MEMORY

The EEPROM data memory is readable and writable during normal operation . This memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers. There are four SFRs used to read and write this memory. These registers are:

- EECON1
- EECON2 (not a physically implemented register)
- EEDATA
- EEADR

EEDATA holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. PIC16F84A devices have 64 bytes of data EEPROM with an address range from 0h to 3Fh. The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The writetime will vary with voltage and temperature as well as from chip to chip. Please refer to AC specifications for exact limits. When the device is code protected, the CPU may continue to read and write the data EEPROM memory. The device programmer can no longer access this memory.

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit

1 = The write operation completed (must be cleared in software)

0 = The write operation is not complete or has not been started

bit 3 **WRERR:** EEPROM Error Flag bit

1 = A write operation is prematurely terminated (any MCLR Reset or any WDT Reset during normal operation)

0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit

1 = Allows write cycles

0 = Inhibits write to the EEPROM

bit 1 **WR:** Write Control bit

1 = Initiates a write cycle. The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.

0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit

1 = Initiates an EEPROM read RD is cleared in hardware. The RD bit can only be set (not cleared) in software.

0 = Does not initiate an EEPROM read

3.11. Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next cycle, in the EEDATA register; therefore, it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

3.12. Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte.

3.13. Write Verify

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM should be verified to the desired value to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit. Generally, the EEPROM write failure will be a bit which was written as a '0', but reads back as a '1' (due to leakage off the bit).

3.14. I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

3.15. PORTA and TRISA Registers

PORTA is a 5-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin). Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read. This value is modified and then written to the port data latch. Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

Note: On a Power-on Reset, these pins are configured as inputs and read as '0'.

3.16. PORTB and TRISB Registers

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin). Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBPU (OPTION<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset. Four of PORTB's pins, RB7:RB4, have an interrupt-onchange feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupton- change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>). This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.

b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared. The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

3.17. TIMER0 MODULE

The Timer0 module timer/counter has the following features:

• 8-bit timer/counter

• Readable and writable

• Internal or external clock select

• Edge select for external clock

• 8-bit software programmable prescaler

• Interrupt-on-overflow from FFh to 00h

Figure 5-1 is a simplified block diagram of the Timer0 module.

3.17.1. Timer0 Operation

Timer0 can operate as a timer or as a counter. Timer mode is selected by clearing bit T0CS (OPTION_REG<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register. Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed below. When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock . Also, there is a delay in the actual incrementing of Timer0 after synchronization.

3.17.2. Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP since the timer is shut-off during SLEEP.

3.18. Oscillator Configurations

3.18.1.Oscilator Types

The PIC16F84A can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

3.19. Programmer Circuit

0This circuit is a very simple which is use for PIC 16f84a (in fig 3.6.) . Also it needs a programer software. I used picprog2 program written by Tord Andersson. It has a serial interface connecting to the chip. It is kind of SQTP (Serially Quick Turn Programmable circuit) chip.I used LM 7805 which is +5 voltage regulator. The capacitors are for filtering the voltage.

The Programming steps;

I got a compiler from microchip called MPLAB. It coverts my assambler code to hex code. Then i load the hex code to pic programmer software (picprog2). It loads the hex coe to the pic circuit's eeprom. The pic circuit is now ready to run.

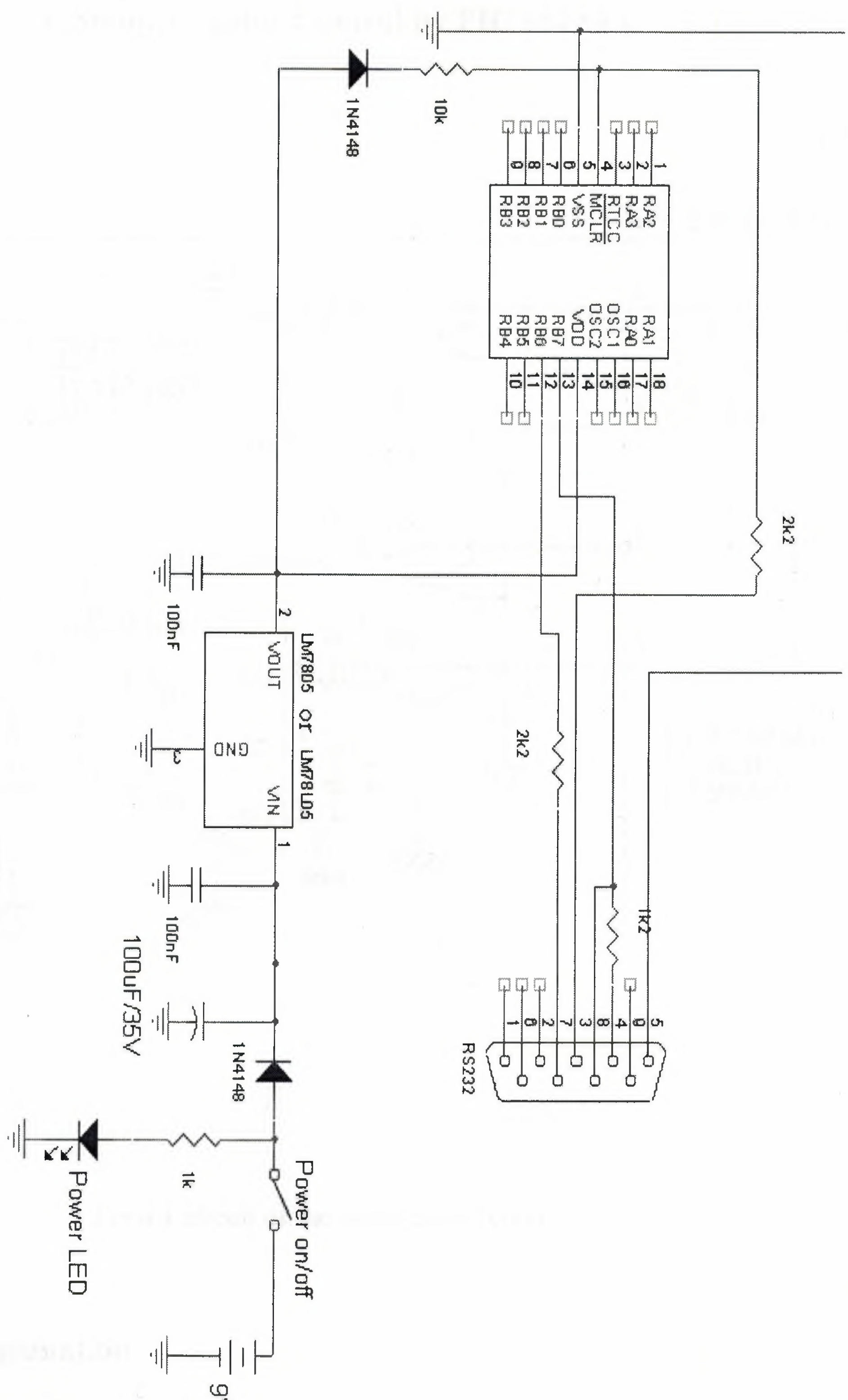


Fig 3.6. PIC 16F84A programer circuit

4. Stepper Motor Control by PIC 16F84A

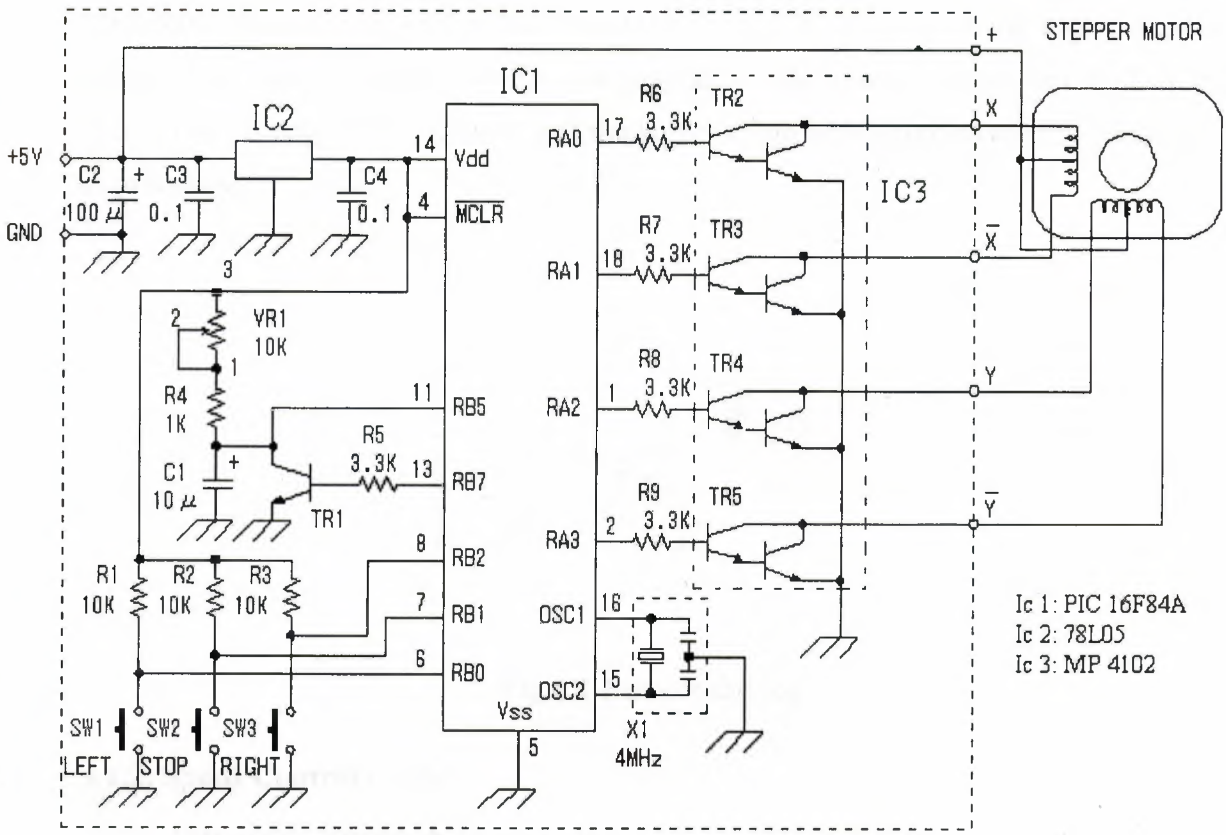


Fig 4.1 circuit of the steper motor control

4.1. Parts explanation

4.1.1. Motor driving circuit

This is the circuit which drives the coil of the stepper motor. There are circuits which drive X coil, X̄ coil, Y coil and Ȳ coil respectively. Darlington connection-type transistor is used for the drive of the coil. As for the Darlington connection, 2 stages of transistors are connected inside in series. The "hfe" of this transistor is the multiplication of the "hfe" of each transistor inside. In case of MP4102 which was used this time, the hfe is over 4000. Because the ratio of the input electric current and the output current is big, the rising edge and the falling edge of the control signal can be made sharp. The diode to be putting between the collector and the power is for the protection of the transistor. When the transistor becomes OFF from ON, the coil of the motor tries to continue to pass an electric current and generates high voltage. An electric current by this voltage is applied to the diode and the high voltage which applies over the transistor is prevented.

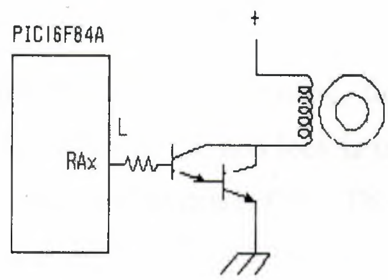


Fig 4.1.1 motor driving

4.1.2. Speed Control Circuit

This is the circuit which controls the rotational speed of the motor.TR1 becomes ON condition when RB7 becomes H level. In this condition, the electric charge of capacitor C1 flows through the transistor and the voltage of the both edges of the capacitor becomes 0 V almost. When RB7 becomes an L level, the transistor becomes OFF condition. In this condition, the electric current flows through VR1 and R4 into capacitor C1 and the charging to the capacitor begins. The voltage of the both edges of the capacitor becomes high gradually as charging is done. The voltage of the capacitor is detected by RB5. The software of PIC interrupts the control of the motor until it checks RB5 after making RB7 an L level and RB5 becomes H level. When making the value of VR1 small, the charging time of the capacitor is short and the control of the motor becomes quick. The control of the motor becomes slow when making VR1 big. The speed control range can be changed by changing the value of the capacitor.

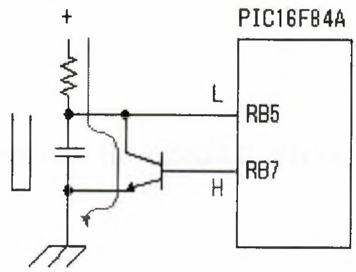


Fig 4.1.2 Speed control of step motor

4.1.3. Start / Stop Circuit

This is the circuit for the clockwise rotating, the counterclockwise rotating or stopping a motor. The baton switch of the non lock is used. Pull-up resistor is used for the port to become H level when the switch is OFF. The RB port of PIC16F84A has an internal pull up feature. However, because RB5 is used for the voltage detection of the capacitor at the circuit this time, an internal pull up feature isn't used. If using RA port for the voltage detection of the capacitor, the RB internal pull up feature can be used. The circuit this time put an external pull-up resistor in the relation of the pattern.

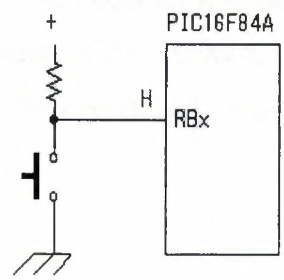


Fig 4.1.3 Start stop position

4.1.4.Oscillator

4-MHz resonator is used because the circuit this time doesn't need high-speed operation.

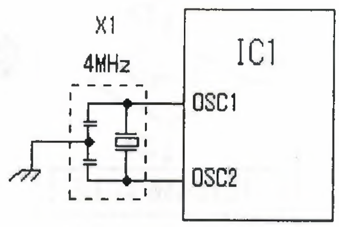


Fig 4.1.4 oscilator circuit

4.1.5. Power supply circuit

The purpose of this circuit is to keep power supply voltage to PIC to 5V when the power of the stepper motor is more than 5V. Because the operating voltage of the stepper motor to be using this time is about 5V, the power supply voltage is +5V. In this case, the voltage which is applied to PIC becomes less than 5V because of the voltage drop (about 1V) in the regulator. In case of PIC16F84A, the operation is possible even if the power falls to about 3V because the operating voltage range is from 2V to 5.5V. It is enough in the 100-mA type.

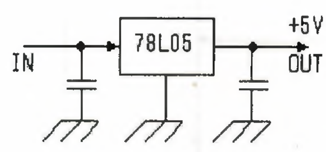


Fig 4.1.5 Power supply circuit

4.2. Flowchart of the program

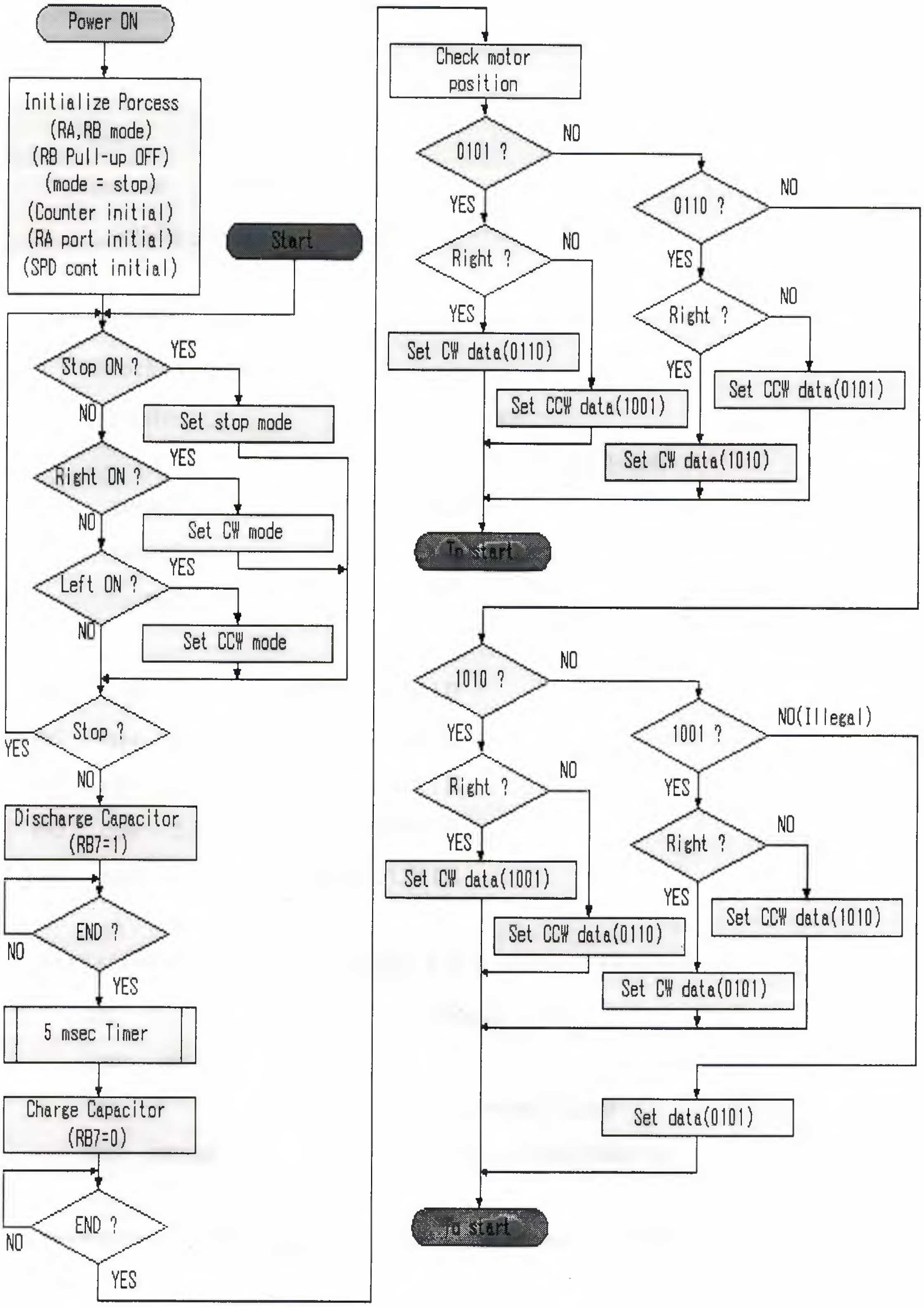


Fig 4.1.6 Flowchart

4.3. Source Code of Stepper Control Program

```
        list        p=pic16f84a
        include        p16f84a.inc
        __config _hs_osc & _wdt_off & _pwrte_on & _cp_off

;**************** Label Definition *******************
        cblock  h'0c'
          mode                  ;Operation mode
                                        ;0=stop 1=right 2=left
          count1                ;Wait counter
          count2                        ;Wait counter(for 1msec)
             endc

rb0    equu    0        ;RB0 of PORTB
rb1    equ     1        ;RB1 of PORTB
rb2    equ     2        ;RB2 of PORTB
rb5    equ     5        ;RB5 of PORTB
rb7    equ     7        ;RB7 of PORTB

;**************** Program Start **********************
        org     0                       ;Reset Vector
        goto    init
        org     4                       ;Interrupt Vector
        clrf    intcon                  ;Clear Interruption reg

;**************** Initial Process ********************
init
        bsf     status,rp0              ;Change to Bank1
        clrf    trisa                   ;Set PORTA all OUT
        movlw   b'00100111'             ;RB0,1,2.5=IN RB7=OUT
```

```
        movwf   trisb                   ;Set PORTB
init
        bsf     status,rp0              ;Change to Bank1
        clrf    trisa                   ;Set PORTA all OUT
        movlw   b'00100111'             ;RB0,1,2.5=IN RB7=OUT
        movwf   trisb                   ;Set PORTB
        movlw   b'10000000'             ;RBPU=1 Pull up not use
        movwf   option_reg              ;Set OPTION_REG
        bcf     status,rp0              ;Change to Bank0
        clrf    mode                    ;Set mode = stop
        clrf    count1                  ;Clear counter
        clrf    count2                  ;Clear counter
        movlw   b'00000101'             ;Set PORTA initial value
        movwf   porta                   ;Write PORTA
        bsf     portb,rb7               ;Set RB7 = 1
        btfsc   portb,rb5               ;RB5 = 0 ?
        goto    $-1                     ;No. Wait
start
;************* Check switch condition *****************
        btfsc   portb,rb1               ;RB1(stop key) = ON ?
        goto    check1                  ;No. Next
        clrf    mode                    ;Yes. Set stop mode
        goto    drive                   ;No. Jump to motor drive
check1
        btfsc   portb,rb2               ;RB2(right key) = ON ?
        goto    check2                  ;No. Next
        movlw   d'1'                    ;Yes. Set right mode
        movwf   mode                    ;Save mode
        goto    drive                   ;No. Jump to motor drive
check2
        btfsc   portb,rb0               ;RB0(left key) = ON ?
        goto    drive                   ;No. Jump to motor drive
        movlw   d'2'                    ;Yes. Set left mode
        movwf   mode                    ;Save mode
```

```
;******************** Motor drive ********************
drive
        movf    mode,w              ;Read mode
        bz      start               ;mode = stop
        bsf     portb,rb7           ;Set RB7 = 1
        btfsc   portb,rb5           ;RB5 = 0 ?
        goto    $-1                 ;No. Wait
        movlw   d'5'                ;Set loop count(5msec)
        movwf   count1              ;Save loop count
loop    call    timer               ;Wait 1msec
        decfsz  count1,f            ;count - 1 = 0 ?
        goto    loop                ;No. Continue
        bcf     portb,rb7           ;Set RB7 = 0
        btfss   portb,rb5           ;RB5 = 1 ?
        goto    $-1                 ;No. Wait
        movf    porta,w             ;Read PORTA
        sublw   b'000000101'        ;Check motor position
        bnz     drive2              ;Unmatch
        movf    mode,w              ;Read mode
        sublw   d'1'                ;Right ?
        bz      drive1              ;Yes. Right
        movlw   b'00001001'         ;No. Set Left data
        goto    drive_end           ;Jump to PORTA write
drive1
        movlw   b'00000110'         ;Set Right data
        goto    drive_end           ;Jump to PORTA write
drive2
        movf    porta,w             ;Read PORTA
        sublw   b'000000110'        ;Check motor position
        bnz     drive4              ;Unmatch
        movf    mode,w              ;Read mode
        sublw   d'1'                ;Right ?
        bz      drive3              ;Yes. Right
```

```
        movlw   b'00000101'          ;No. Set Left data
        goto    drive_end            ;Jump to PORTA write
drive3
        movlw   b'00001010'          ;Set Right data
        goto    drive_end            ;Jump to PORTA write
drive4
        movf    porta,w              ;Read PORTA
        sublw   b'000001010'         ;Check motor position
        bnz     drive6               ;Unmatch
        movf    mode,w               ;Read mode
        sublw   d'1'                 ;Right ?
        bz      drive5               ;Yes. Right
        movlw   b'00000110'          ;No. Set Left data
        goto    drive_end            ;Jump to PORTA write
drive5
        movlw   b'00001001'          ;Set Right data
        goto    drive_end            ;Jump to PORTA write
drive6
        movf    porta,w              ;Read PORTA
        sublw   b'000001001'         ;Check motor position
        bnz     drive8               ;Unmatch
        movf    mode,w               ;Read mode
        sublw   d'1'                 ;Right ?
        bz      drive7               ;Yes. Right
        movlw   b'00001010'          ;No. Set Left data
        goto    drive_end            ;Jump to PORTA write
drive7
        movlw   b'00000101'          ;Set Right data
        goto    drive_end            ;Jump to PORTA write

drive8
        movlw   b'00000101'          ;Compulsion setting
drive_end
        movwf   porta                ;Write PORTA
```

```
    goto    start                    ;Jump to start

;************* 1msec Timer Subroutine *****************
timer
    movlw   d'200'                   ;Set loop count
    movwf   count2                   ;Save loop count
tmlp    nop                          ;Time adjust
    nop                              ;Time adjust
    decfsz  count2,f                 ;count - 1 = 0 ?
    goto    tmlp                     ;No. Continue
    return                           ;Yes. Count end

;*****************************************************
;       END of Stepper Motor controller
;*****************************************************
    end
```

4.3.1. Label definition

```
;****************  Label Definition  ********************
         cblock   h'0c'
```

The data area is automatically assigned from 0ch by CBLOCK directive. ENDC is used for he ending of assignment. The purpose of each data area is shown below.

Label	Purpose
Mode :	This is the area which manages the condition of the motor control 0=stop, 1=clockwise , 2=counterclockwise
count1:	This is the count area to make control waiting time it counts 1 msec five times and 5 msec are made
count2 :	This is the counter to make 1 msec.

4.3.2. The program start

```
;****************   Program Start   ***********************
```

Instruction is executed from Zero addresses of the program memory when making the power ON of the PIC. When there is interruption processing, processing is begun from the addresse 4. Because it isn't using interruption this time, there is not program execution from the addresse 4. It makes the interruption prohibition condition if the interruption occurs. It isn't necessary to do this processing.

4.3.3. The initialization process

```
;****************   Initial Process   ********************
```

The following processing is done as the processing of being initialized after the turning on.

The initialization of the mode of port A

All ports are set to output mode.

The initialization of the mode of port B

RB0,1,2 and 5 are set to input mode. And RB7 is set to output mode.

Port B pull-ups are disabled (RPBU=1)

Because RB5 is used as the high impedance input at the circuit this time, the RB pull up feature should not be used

Setting of a stop mode

Immediately after turned on, it sets a motor to the stop mode. When there is not this step, the original value of mode becomes 0. It is set for the safety.

Counters for the control waiting time are initialized

There is not a problem even if there is not these processing. They are set for the safety.

Port A initialization

It sets 0101 as the initial state of port A. Because it drives with the transistor, the logic reverses. It is in the condition, $\overline{Y}$=H Y=L $\overline{X}$=H X=L, from the bit on the left.

Discharging of the capacitor for the speed control

It makes RB7 H level and it makes TR1 ON and discharging in the electric charge of the capacitor for the speed control. The end of the discharge is confirmed in RB5.

4.3.4. The switch condition confirmation process

```
;*************   Check switch condition   ****************
```

It detects the ON condition of the stop switch, the RRC switch, the RLC switch. A condition is set to mode according to the kind of the switch which was made ON. The order of the detection is a stop, a RRC, a RLC. When more than one switch is pushed at the same time, the switch which detected ON earlier is effective. This processing is done every time it controls 1 step of motor.

4.3.5. The motor drive process

```
;********************Motordrive*******************
```

A stop mode is checked first. In case of the stop mode, it doesn't drive the motor and it jumps to the switch condition confirmation process. In case of not being a stop mode, the following process is done.

Discharging of the capacitor for the speed control

Discharging the capacitor as the preparation to make the timing of the speed control.

The wait processing of 5 milliseconds

In the high-speed control, the rotor doesn't follow the change of the magnetic pole and the step motor doesn't rotate normally. It sets a timer value to turn a full speed normally.In case of the motor which was used this time, it doesn't rotate normally when making less than 5 milliseconds.

The charging of a capacitor for the speed control and the confirmation process

It makes RB7 an L level and it begins charging the capacitor. It confirms that the charging completes in RB5. It is completion if RB5 becomes H level. Correctly, it is not charging completion and it is the fact that the voltage of the capacitor became above the threshold voltage of RB5.

The motor drive process

After the speed control timing, a motor is driven. The control state of the motor is confirmed first. This is done by reading the condition of port A. Next, whether it is a clockwise mode or a counterclockwise mode is judged. The following control state which should drive a motor by the result is set to port A. Because there are four conditions, processing is done in each condition. After the motor drive process, it jumps again to the switch condition confirmation process.

The stationary torque of the stepper motor is large. However, as the turn becomes fast, the turn torque falls. The stepper motor can not do a high-speed turn. A stepper motor is made to control a turn position correctly. It is to control a turn position correctly like the drive motor of the printer and so on. The circuit this time controlled the number of rotations of the motor by the charging of the capacitor but can control a turn angle in the drive number of times

CONCLUSION

Stepper motor is the main equipment of the Control Industry. Lots kind of stepper motor are being used in worldwide industries under different techniques. My aim is to prove the PIC circuit can be useful in stepper motor applications. Although kinds of stepper motor drive tecniques are available, the advantage of controlling with an eeprom is to adapt any application or any factory automation. However maintanance and debugging is very easy and costs cheaper. The main needs of an application is generally consists of memory manipulation, input/output process, math routines, real-time processes. The PIC coveres all these techinques.

Control is a huge world. And the PIC makes it more manageable.

References

Handbook of Small Electric Motors
William H.Yeadon and Alan W, Yeadon, eds. McGraw-Hill, c2001.

Stepping motors: a guide to modern theory and practice
Acarnley, P. P.
P. Peregrinus on behalf of the IEE, 1984, c1982.

Stepping motors and their microprocessor controls
Kenjo, Takashi
Oxford University Press, c1984.

Design with Microprocessors for Mechanical Engineers
Stiffler McGraw-Hill

Electronic Control Systems in Mechanical Engineering
Bolton Addison Wesley

The Art of Electronics
Horowitz and Hill
Cambridge University Press

Electro-mechanical Engineering - An Integrated Approach
by Fraser and Milne IEEE