6.1 Introduction

Stepper motors are useful in applications where a high degree of positional control is required. This is true in cornputer and instrumentation applications, where they are used in printers, robots, floppy and hard disk drives, for example. This is in contrast to conventional DC or AC motors which rotate continuously. Stepper motors rotate to move a load to a given position, then stop. In the case of the floppy drive, a motor may be commanded to rotate a given number of steps to position the magnetic head. The head will then remain stationary, reading the data above that particular track. It is not possible for many types of motors to hold a load in position when it stops.

6.1.1 Ease of application

These motors are particularly useful in computer applications. This is because by just issuing a series of pulses, the motor will rotate a fixed number of steps. By changing the order of application, the motor can be made to reverse direction. To move at a given speed, we issue the pulses at a given rate. In the same way acceleration may be controlled as well. In summary, most of the common tasks of motion control can be accomplished in software. This reduces hardware costs, which can be high for precision mechanical applications.

6.1.2 Types of Stepper Motors

Three basic types of stepper motors are in common use, namely the Permanent magnet (PM), Variable reluctance (VR) and Hybrid types.We will look at the PM type.

6.1.3 Characteristics

As in other types of motors, there is a rotating part called the rotor. PM motors have a permanent magnet on the rotor. The rotor resides in a casing which physically surrounds it. In the casing there are several coils of wire called the stator. The user drives the rotor by sending current into the stator in a defined sequence. Stepper motors are classified by physical size, torque, winding resistance and step size. The size, torque and resistance determine the load to be moved and will not be considered here.

Step size

The step size is the amount the motor rotates in one step. This is purely a function of the mechanical construction of the motor. One stepper motor could have a step size of 1.8 degrees, another of a step angle of 18 degrees and so on.

Phase

This is the number of sets of windings on the stator. The most common type of motors use four phases. This will also determine the number of wires required to drive the motor. Another way of looking at the phase is the number of electrical signals which have to be applied to the motor for one electrical cycle.

Relation between phase and step size

The relationship between a phase and the actual distance a motor moves depends on the motor construction and how the motor is electrically connected. For example, switching the current from one phase to the next results in the motor moving one step.

6.2 Basic motor control

Here we will look at how a stepper motor moves a series of steps. The angle moved in a step depends on the actual physical construction of a motor. Motors commonly used in a floppy drive have step angles of 1.8 degrees. Looking at a motor with such a small angle is difficult. For convenience, we will consider a motor with a step angle of 90 degrees. In reality, a motor with such a large angle is not common. We will also assume it has four coils on its stator. This corresponds to four phases - A, B, C, D and a permanent magnet rotor with a single magnet.

Let phase A be activated first as a SOUTH pole.The rotor will align itself so the poles attract. After this, phase A is turned off and phase B is energized, as shown. The rotor will turn by 90 degrees. The magnetic force produced by the stator coils pulls the rotor to align with it.

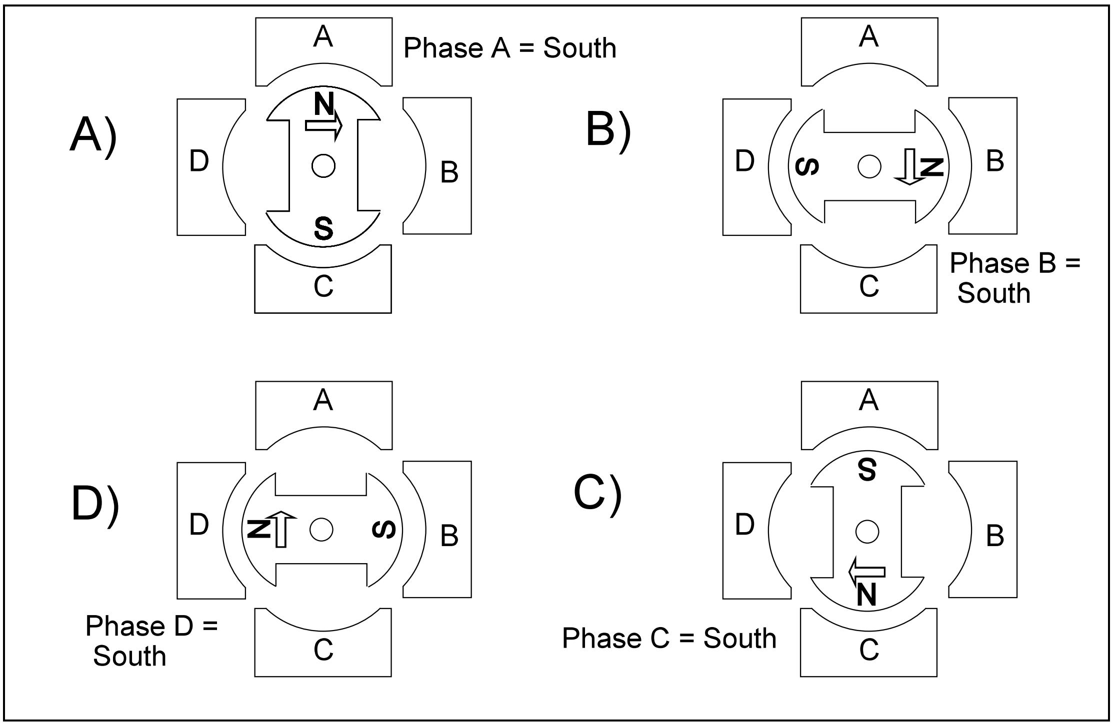

Fig 6-1 Stepper motor movement

Now phases A and B are turned off, and phase C turned on. The rotor moves another 90 degrees and so on for phase D.

Note that in this case, turning on one phase moves the motor by 90 degrees. This is not always the case

Now a microcontroller can control the current in the phases easily, by sending binary valued voltages to it.By doing so, it controls the position. By timing the signals, it will control the speed and acceleration.

The following is a schematic of a typical stepper motor circuit.

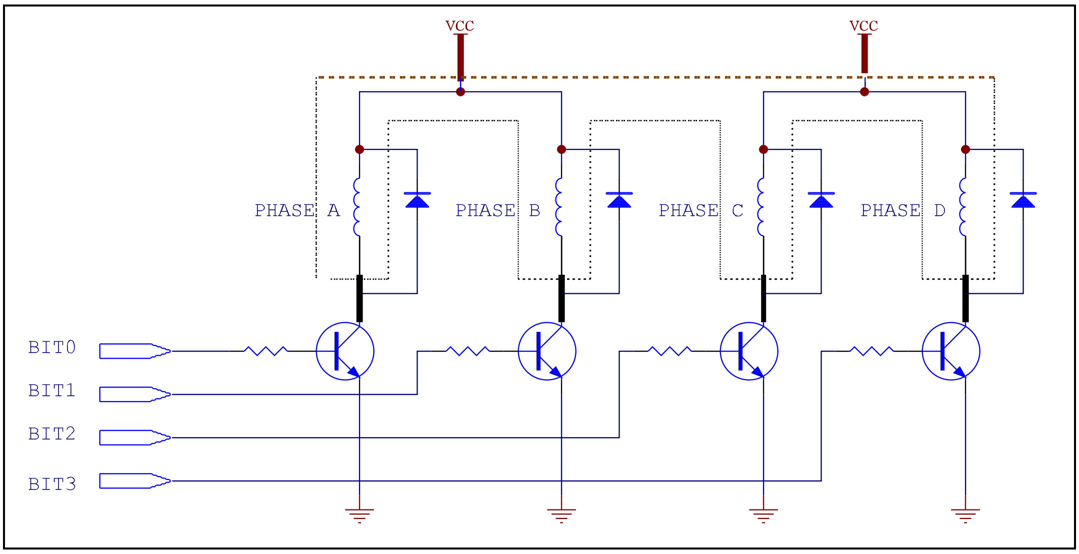

Fig 6-2 Schematic of a four phase stepper motor circuit

In this circuit, the lowest four bits of a port are used to control each phase. The motor itself is represented by the dotted lines. It is connected to the rest of the circuit by the thick lines. Altogether these represent six wires. Fly back diodes are used to redirect the current back to the power supply when the transistors turn off. Note that two coils are connected to the supply voltage. This fact may be used if the wiring of a stepper motor is not given.

6.3 Standard methods of position control

There are several ways to improve on motor positioning. We can improve on the torque, or turning force applied to the rotor. This can be done by increasing the current to the coils. We will consider only computer control here.

6.3.1 Full step

As described earlier, the motor moves 90 degrees per step as we activate the phases. This is called full stepping, where the rotor moves the entire step it was designed for.

Another way is by turning on two coils at the same time. As shown, if we turn on phases A and B at the same time, the rotor will rest at a position halfway between the two. Although twice the current flows, torque increases 1.414 times only. Then phases B and C are turned on and the motor moves another 90 degrees.

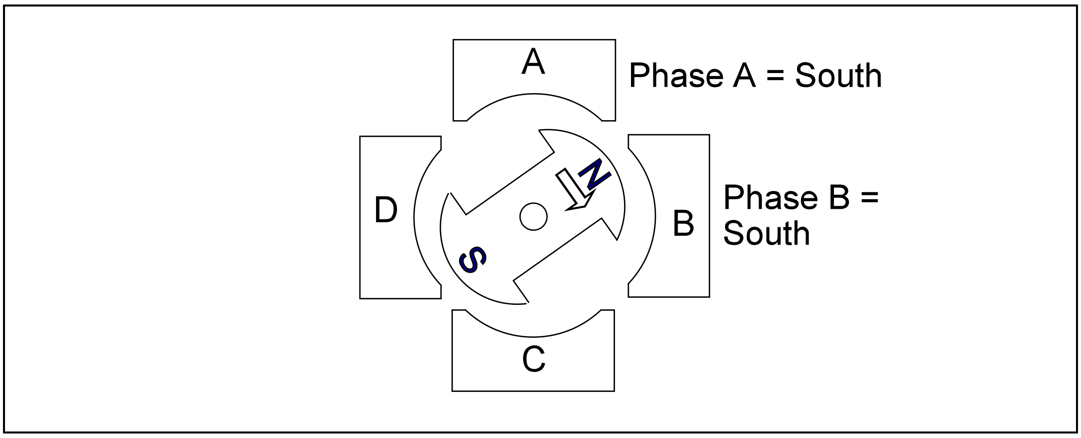

Fig 6-3 Full step - halfway between phases

We can control the movement and torque produced both by activating the proper phases and by the motor electrical connection.

Unipolar drive
So far, we have connected up the stator coils so current flow only one way through the transistor. This is called Unipolar Drive.

Bipolar drive
This works by controlling a phase with up to four transistors and diodes so current can flow both ways into a phase. For example, Phase A is activated as a South pole and later it is activated as a North pole. Some motors which have only four windings, without a common power connection, have to use this drive arrangement. Also, this makes more efficient use of electrical power. When a motor coil is turned off, the diodes conduct. This allows current to flow back to the power supply. In effect, energy is returned to the supply.

6.3.2 Microprocessor control

With the circuit above, all a microprocessor has to do it to output the required four bits to make the motor move. The table below represents the phases to be turned on to make the motor rotate one revolution, with the hex value to be output to the port above. Remember for this example the lowest four bits are connected as in the schematic shown.

Step	Phase D	Phase C	Phase B	Phase A	Value
1	0	0	0	1	1
2	0	0	1	0	2
3	0	1	0	0	4
4	1	0	0	0	8
1	0	0	0	1	1

Full-Step Sequence for Clockwise Rotation.

6.3.3 Half step / other steps

If we vary the phases so that A turns on, then A and B, then B and so on, we can make the motor move in 45 degree steps. This is called half stepping and effectively increases the resolution of the motor. The sequence of phase activation to move a complete revolution in half stepping mode is shown below.

Step	Phase D	Phase C	Phase B	Phase A	Value
1	0	0	0	1	1
2	0	0	1	1	3
3	0	0	1	0	2
4	0	1	1	0	6
5	0	1	0	0	4
6	1	1	0	0	C
7	1	0	0	0	8
8	1	0	0	1	9
1	0	0	0	1	1

Half-Step Sequence for Clockwise Rotation.

In full and half step movements, the full current is applied to the stator coils. If we send a fraction of the current, the rotor will come to a position in between the two phases. This technique is called microstepping.

If we activate the phases in the opposite sequence - for example D,C, B, A in full stepping, we will make the motor move in reverse.

6.4 Speed control

Although a stepper motor performs positioning duties well, it can move loads as well. It is not possible to drive a stepper motor at an arbitrary speed and acceleration. At high speeds, the back EMF generated by the moving rotor reduces the voltage available to drive current through the motor coils. Some areas of concern in controlling movement are discussed.

Loss of synchronisation

If the rate at which the coils are energized is too high, the motor may not rotate in synchronism with the pulse rate, and may even stop moving. For example, if the microcontroller sends out pulses at 10 microseconds intervals, we expect the motor to move at a rate of 1/10 microseconds or 10,000 pulses per second. If the motor is not capable of this, it may stall - that is, stop moving or it will move at a lower rate.

If we stop sending pulses, we expect the motor to stop immediately. Again, if the speed is too high, the motor and load may continue moving even though the pulses have stopped.

Resonance

At other speeds, the motor may vibrate or even rotate in reverse all by itself. These effects are known as resonance and are due to the fact that the stepper motor is a very nonlinear device. The interaction of the various elements of the motor and its driving circuit may cause such effects to occur.

6.4.1 Torque speed curve

An important characteristic for a stepper motor in motion control is the torque speed curve. For a given motor, and a given phase voltage, the motor is given a load which is proportional to the amount of torque it can produce. A low stepping rate is applied. The motor should not lose synchronisation. When the pulses stop, the motor should stop immediately. The speed is now increased by a small amount and the same check for motor synchronism performed. This goes on until the motor just loses synchronism.

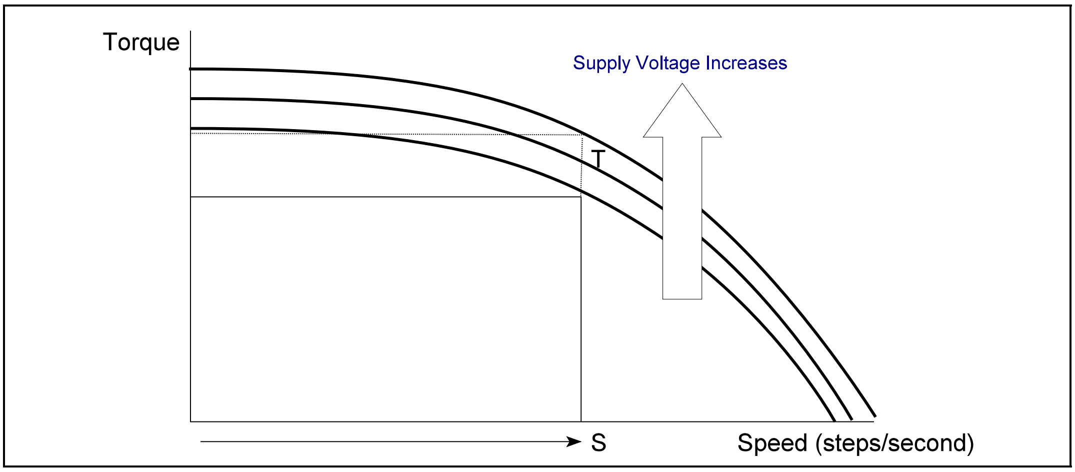

Fig 6-4 Torque Speed curve

If we look at the torque speed curve, at point T, what this means is that to produce a torque of T, the motor may be driven at any speed up to S steps per second for a given supply voltage. The motor must remain in synchronism to the pulse rate. If we increase the supply voltage the torque speed curve shows that there is more torque for the same speed.

At high stepping rates, the rotor produces higher back EMF. This acts against the applied voltage thus reducing the motor current. As a result the torque produced will drop. Consequently if we want to move a load at a high stepping rate, we need to start slowly and gradually increase the rate, so the motor does not lose synchronisation.

One common difficulty is to determine what torque is needed to drive a certain load. Also, if the system is not properly constructed mechanically, the load will not be even.
In this case a trial and error method may be needed.

6.5 Other considerations

In stepper motor movement applications, the motor will be accelerated to a constant speed, which is also called the slewing speed. Before coming to a stop it will be decelerated. At all times the timing routine is used. These are all processor intensive activities. For situations where many motors must be controlled simultaneously additional pieces of hardware may be used. These are translators and indexers. These may be single chip devices, or a printed circuit board depending on the application.

Translator
When interfacing to this piece of hardware, the motor still has to send out steps. It this way the motor still controls the speed. But the direction of travel, whether to use half or full stepping is done by setting some pins.

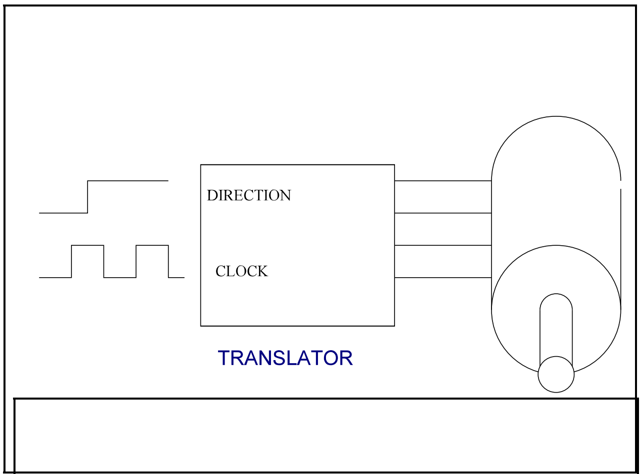

Fig 6-5 Translator block diagram

Indexer
This hardware accepts commands from the processor specifying the start/stop, acceleration and deceleration rates, slew rate, and the final position. It then generates the required stepping rates to achieve the speed profile and informs the processor when the final position has been reached. This may be done by using an interrupt or by setting a bit in a status register. The processor does not send out timing signals.

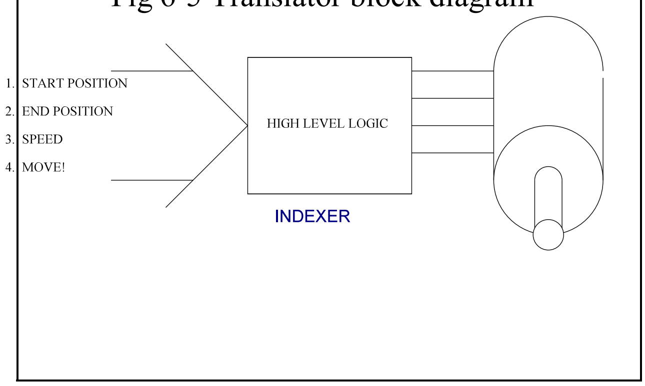

Fig 6-6 Indexer block diagram

As can be seen, an indexer has a lot of functions and may include a translator circuit as part of it.

6.6 Sample application

A stepper motor rotates 1.8 degree per step. What step rate is needed to make it rotate at 10 revolutions per minute (rpm) ? Assume that the program execution time is negligible. You also have access to a routine called Delay1ms which provides a 1 millisecond delay. Use the motor in FULL step mode.

<u>Calculation:</u>
We need to convert from rpm to steps per second.

10 rpm = 360 degrees/minute * 10

This is equal to 3600 / 60 degrees/second
Since one step is 1.8 degrees, we need to issue 60 / 1.8 = 33.33 steps per second

Or 1/33.33 seconds per step = 0.03 which is 30 milliseconds per step.

Use the lookup table for the full step motion.
The following is a sample C program that moves the motor in halfsteps.

```
#define     SMPort      0x335
#define     NumSteps    200
#define     PtableLen   8

unsigned char Ptable []={0x01,0x02,0x04,0x08,0x03,0x06,0x0c,0x09};

void main (void)
      {
      int i, j;
      i=0;
      for (j=NumSteps;j>0;j--)
            {
            _outp (SMPort,Ptable [i]);     /* output to port */
            Sleep (300);                   /* step rate */
            i++;
            if (i>=PtableLen) i=0;
            }
      }
```