SINGAPORE POLYTECHNIC
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

ET0104 Embedded Computer Systems Laboratory

Laboratory 5 - Controlling a stepper motor

Objectives

- To experiment with various types of stepper motor movements
- To move a stepper motor continuously and with acceleration

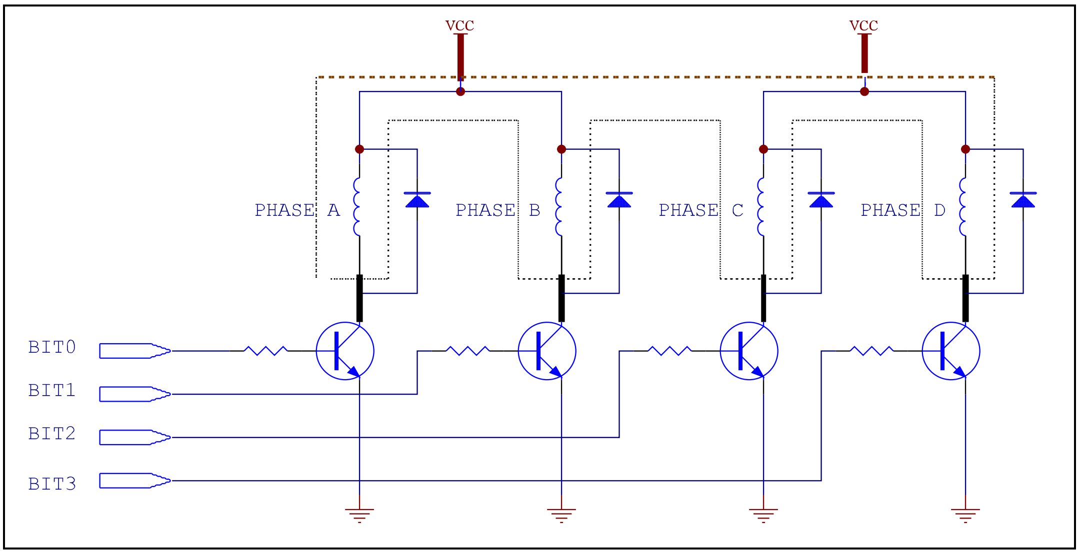
Interfacing to a stepper motor

Stepper motors are commonly used to provide motion control. Their popularity comes from their ease of use. To move it one step you send one pulse from a processor. To drive a load at a given speed, all that is needed is a properly timed set of pulses. The upper speed limit is controlled by its torque speed characteristics.

Stepper motors are rated by:
1) The angle the motor shaft rotates when one pulse is sent
2) The current it draws - which is a measure of the load it can drive

Also, we may drive the motor with one phase, two phase, or one-two phase on. We use only the lower four bits of a latch to turn the transistor on or off.

In the following diagram, the motor itself is represented by the dotted lines. It is connected to the IO board via a 6 pin connector which is plugged into the STPMTR connector located at the lower left of the board.



Schematic of a four phase stepper motor circuit

Postioning Applications

A table is given below for you to fill in. The processor will fetch one byte at a time from this table and output it to the Step Motor latch. This will turn on the matching transistor as shown in the earlier diagram. We are considering three types of drive schemes. For each of them, we will subjectively compare the torque the motor produces. To do so, hold the base of the motor mount and *lightly* hold the motor shaft to stop it from rotating, but *do not* force the motor!

Full step - one phase on

The first drive scheme we will consider is the full step one phase on motion. This means we want to make the motor move in full steps, one phase on at a time. A '1' turns the phase on.

Step	Phase D	Phase C	Phase B	Phase A	Value
1	0	0	0	1	%00000001
2					
3					
4					
1					

TABLE 1 Full-Step One phase on sequence for Clockwise Rotation.

Complete the table and enter the values into LAB5A.C at the section labelled "PTable". See the appendix. Though 8 spaces are given, use only what is needed.

For this application, we want to rotate the motor 360 degrees. The motor given is rated to rotate 1.8 degrees per step. The variable "NumSteps"in the program will control the number of steps the motor rotates. Calculate the value for one complete rotation.

NumSteps ____

Assemble the program, load and run it. See if the motor rotates one revolution. *Gently* feel the motor shaft as mentioned earlier, to get a feel of the torque generated.

Full step- one phase on-reverse

Now we want to make the motor rotate in the opposite direction for one revolution, one phase on at a time. Compute the values needed and modify LAB5A.C.

Step	Phase D	Phase C	Phase B	Phase A	Value
1	1	0	0	0	%00001000
2					
3					
4					

TABLE 2 Full-Step two phase on sequence for Anti-clockwise Rotation.

Full step- two phase on

Now we want to make the motor move full steps, two phases on at a time in the forward direction. Again, compute the values and modify LAB5A.C.

Step	Phase D	Phase C	Phase B	Phase A	Value
1	0	0	1	1	00000011
2					
3					
4					

TABLE 3 Full-Step two phase on sequence for Clockwise Rotation.

Gently feel the motor shaft as mentioned earlier, to get a feel of the torque generated - the two phase on drive should have more torque.

Half step

Finally, we make the motor move in half steps. Complete the table and proceed as before.

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

TABLE 4 Half-Step Sequence for Clockwise Rotation.

Run the program again, and *lightly* touch the motor shaft. Does it seem to have more vibration as compared to the previous movements?

Speed applications

To move at a given speed, we need to introduce a time delay in between steps. While a motor moves in discrete steps, motor speed is normally given in terms of revolutions per minute (rpm). Thus we have to perform some conversions.

In this part of the lab, we want the motor to act as the "second" hand of a clock. That is, it should turn one revolution in one minute. Again, one step takes 1.8 degrees. Complete the following calculations

Number of steps sent per minute ______________________

Time duration between each step ______________________

Look at program LAB5A.C. The Delay subroutine uses the value PulseDly to act as a delay loop. You are also given the machine cycle timing for each instruction. Since we know the timing between pulses, calculate the value of PulseDly so we can get the required time delay.

____

PulseDly ________________________________

Modify the program with the new value of PulseDly. Since the motor will be moving continuously we do not need to check how many steps we moved. So we jump to check for end of a phase sequence.

Make all these changes to LAB5A.C. Load and test it.Do you have an accurate “clock”?

Appendix

Postioning applications

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#include <time.h>


#define         SMPort          0x___
#define         NumSteps        50
#define         PtableLen       8

unsigned char Ptable[]={____, ____, ____, ____, ____, ____, ____, ____};


void main(void)
{
      int i,j;
      i=0;
      for(j=NumSteps;j>0;j--)
      {
            _outp(SMPort,Ptable[i]);     /* output to port */

            Sleep(300);
                i++;
                if (i>=PtableLen) i=0;

      }
}
```

Acceleration

```
/* stepper motor driver program       */
/* uses acceleration                  */

#define         SMPort          0x335
```