

## CHAPTER 6

### Stepper Motor Drives

#### 6.1 Stepper Motors

##### 6.1.1 Basic Operation

A stepper motor is an electric machine that rotates in discrete angular increments or steps. It is operated by applying pulses of a specific frequency to the input of the motor. Each pulse applied to the motor causes its shaft to move a certain angle of rotation, called a *stepping angle*. Figure 6.1 shows a simplified construction of *bifilar permanent magnet* stepper motor. The rotor of the motor is made of a permanent magnet material and has six teeth equally spaced around the circumference of the rotor with alternating north (N) and south(S) magnet polarities. The stator has four poles, each of which has a center-tapped winding. The windings on opposing poles are connected together so that only five wires - A, B, C, D, and V+ - leave the motor. A winding is excited by sending a current into the V+ wire and out one of the other wires. The windings are wound in the stator teeth in such way so that the following results are obtained.

- ❑ If winding B is excited, pole 1 is energized as North and pole 2 as South; if winding A is excited, pole 1 becomes South and pole 2 becomes North instead.
- ❑ If winding C is excited, pole 3 is energized as North and pole 4 as South; if winding D is excited, pole 3 becomes South and pole 4 becomes North instead.

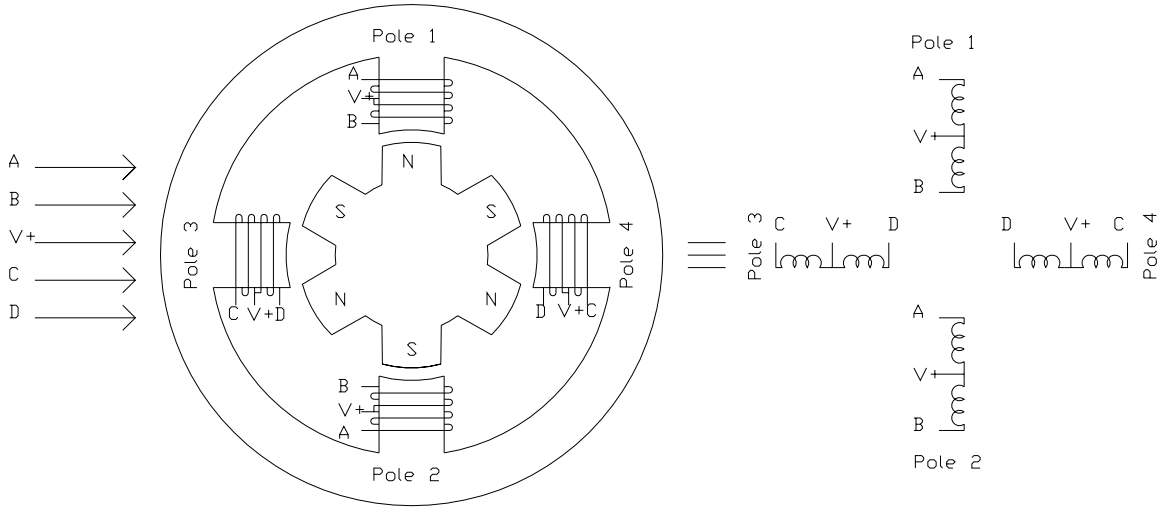


Figure 6.1 Construction of a bifilar PM stepper motor

The operation of the stepper motor relies on the simple principle that, opposite magnetic poles attract while like poles repel. If the windings are excited in a correct sequence, the rotor will rotate to a certain direction Figure 6.2 illustrates how the rotor rotates when the windings are excited with the sequence given in Table 1

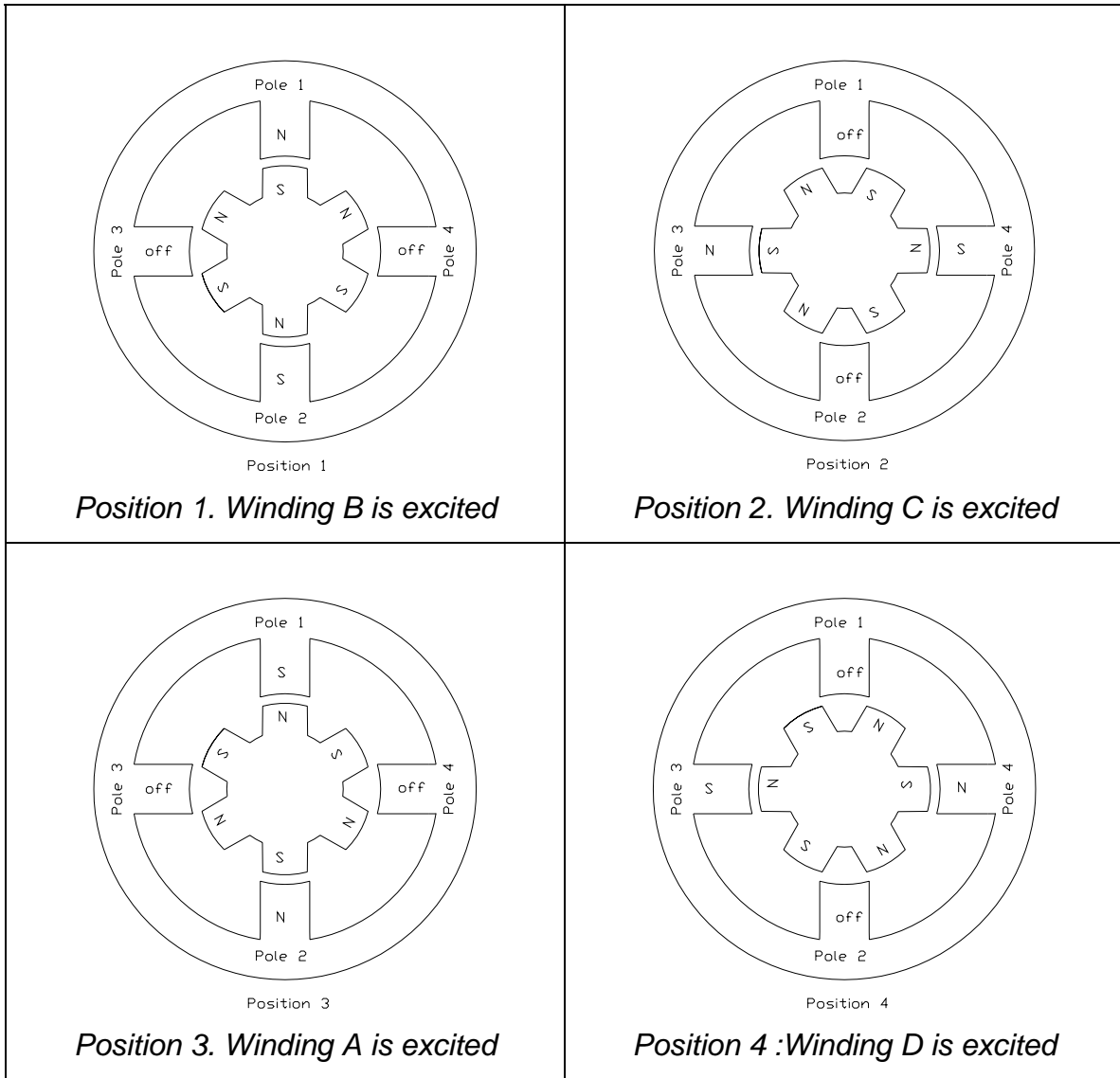


Figure 6.2 The rotor rotates in clockwise direction

	<b>Winding A</b>	<b>Winding B</b>	<b>Winding C</b>	<b>Winding D</b>
<b>Position 1</b>	Off	On	Off	Off
<b>Position 2</b>	Off	Off	On	Off
<b>Position 3</b>	On	Off	Off	Off
<b>Position 4</b>	Off	Off	Off	On

Table 1. *Winding excitation sequence for Figure 6.2*

As can be seen in Figure 6.2, the excitation sequence given in Table 1 causes the rotor to rotate in clockwise direction. If the excitation sequence is reversed, the direction of motion will also be reversed. If the excitation is removed, there is still some attraction between the poles and the teeth due to the permanent magnet in the rotor. As a result there is a residual *holding torque* even when there is no power applied to the motor.

From Figure 6.2 it can be seen, that the motor has a 30 degrees stepping angle, and it requires 12 steps to complete one revolution. The number of steps per revolution in a stepper motor can be increased by adding more teeth on the rotor and by having additional teeth machined into the stator poles. The stepping angle of a stepper motor can be made to be as small as 1.8 degree so that 200 steps are required per revolution.

The excitation scheme in Figure 6.2 is referred to as *single phase excitation* since only one of the four windings is excited at a time. At each step the rotor teeth is aligned exactly with the active stator teeth. It is, however, possible to operate the motor with two windings carrying current at the same time (*two-phase excitation*). In that case the rotor teeth align themselves between the two active stator teeth. Table 2 shows the actuation sequences and the rotor positions for single-phase and two phase excitation. Note that the stepping angles for the two kinds of excitation are the same but that the rotor positions are offset by half the stepping angle. These two actuation scheme are sometimes called *full-stepping* actuation modes

### ***Single Phase Excitation***

Rotor position	Winding A	Winding B	Winding C	Winding D
0	Off	On	Off	Off
$\theta$	Off	Off	On	Off
$2\theta$	On	Off	Off	Off
$3\theta$	Off	Off	Off	On

### *Two-Phase Excitation*

Rotor position	Winding A	Winding B	Winding C	Winding D
$\theta/2$	Off	On	On	Off
$3\theta/2$	On	Off	On	Off
$5\theta/2$	On	Off	Off	On
$7\theta/2$	Off	On	Off	On

Table 2. Full step actuation mode: single-phase and two-phase excitation

If the single-excitation and the two-phase actuation sequence are combined, a *half-step* mode results. In this mode the number of steps per revolution is doubled, so that a motor requiring 200 steps per revolution with the full-step mode will require 400 steps per revolution when operated in the half step mode. Table 3 shows the actuation sequence for the half step mode.

### *Half-step mode*

Rotor position	Winding A	Winding B	Winding C	Winding D
0	Off	On	Off	Off
$\theta/2$	Off	On	On	Off
$\theta$	Off	Off	On	Off
$3\theta/2$	On	Off	On	Off
$2\theta$	On	Off	Off	Off
$5\theta/2$	On	Off	Off	On
$3\theta$	Off	Off	Off	On
$7\theta/2$	Off	On	Off	On

Table 3. Half-step actuation mode

The stepper motor described above uses two windings with opposing magnetizing effect in each pole. This is the reason why it is called '*bifilar*' stepper motor. Some stepper motors use only one winding per pole, and are referred to as '*unifilar*' type. Unlike the bifilar type, the unifilar stepper motor requires a negative voltage to reverse the magnetic polarity of the pole. Besides unifilar and bifilar, stepper motors are also classified from the material used to build the rotor. There are some stepper motors that use a simple iron rotor with no permanent magnet. This type of stepper motor is called *variable reluctance* stepper motor. In this type of stepper motor, the rotor is still moved by the attraction of the rotor to the energized poles of the stator. However, a variable reluctance stepper motor has no residual holding torque when the winding is not energized.

### **6.1.2 Stepper-Motor Drive Circuits**

In the lab experiments, we will control the operation of a stepper motor directly from the DSP output port. For this purpose we need a drive circuit for the motor. The type of stepper motor used is very important in designing the drive circuit for the motor. Different type of stepper motor requires different drive circuit. In the laboratory experiment, we will use a bifilar permanent magnet type stepper motor, similar to the stepper motor in Figure 6.1, except that it has a stepping angle of 1.8 degree or 200 step/revolution. Other considerations in designing the drive circuit are the voltage and current required to energize the stator windings of the motor. The stepper motor we use in the lab requires a 4V voltage and 1A current for each phase winding. Figure 6.3 shows the stepper motor drive circuit we will use for the lab experiments.

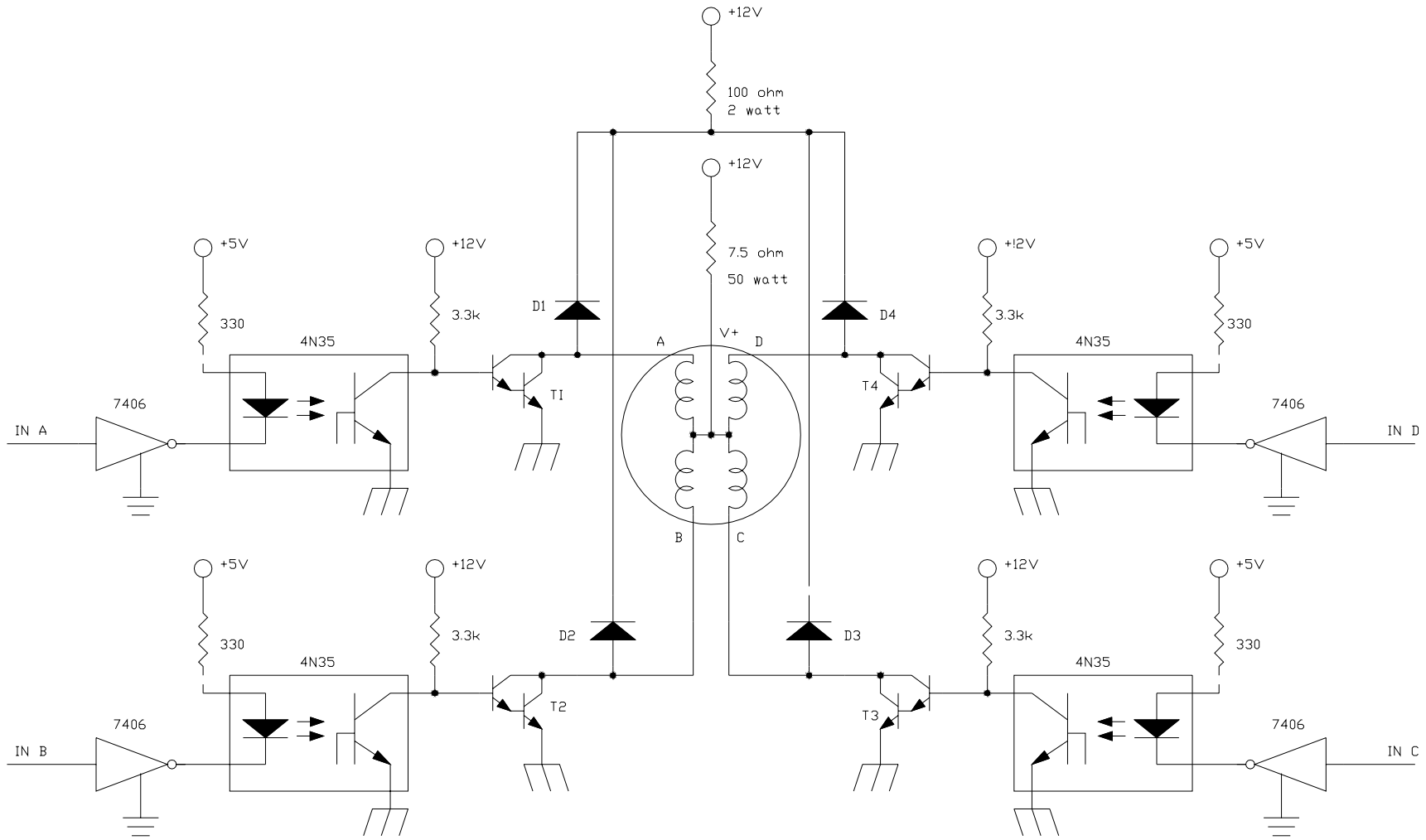


Figure 6.3 Stepper motor drive circuit

In Figure 6.3, wires A, B, C, and D from the stepper motor are connected to the Darlington power transistor T1, T2, T3, and T4 which act as switches. The V+ wire is connected to the +12V power supply via a series resistor. When the power transistor is switched on, a circuit is completed and current flows through one of the windings.

The winding in a stepper motor has substantial inductance. This inductance presents two problems. First, when a winding is switched on, the equation for the current flowing in the winding is:

$$i(t) = \frac{V}{R} \left( 1 - \exp \frac{-t \cdot R}{L} \right)$$

where  $V$  = supply voltage

$R$  = total circuit resistance

$L$  = total circuit inductance

The current needs approximately  $3L/R$  seconds (three time constants) to reach 95 percent of its final value. The stepper motor we use in the lab has a winding resistance of 4 ohms (4V/1A) and winding inductance of approximately 20 mH. Therefore, the time required for the current to reach its final value will be about 15 ms. This certainly limits the stepping rate of the motor. For this reason a 7.5-ohm series resistor is added in the V+ line in Figure 4. This series resistor provides additional resistance to the circuit so that the time constant is decreased. Note that the addition of the series resistance is responsible for the higher supply voltage used in the circuit. Assuming the collector-emitter saturation voltage is 0.5V and the winding voltage drop is 4V, the required 1 A winding current causes 7.5 V voltage drop on the series resistor. Therefore, a +12V (0.5V+4V+7.5V) supply is used.

The second problem created by the winding inductance occurs when a winding is switched off. Recall that when a circuit containing an inductive load is switched off, the collapsing field attempts to maintain the flow of current. If no path is provided for the current. Damaging voltage spikes will be generated across the switching devices. Therefore, freewheeling diodes (D1 to D4) must be included in the driver circuit. The freewheeling diode circuit used in Figure 6.3 includes a series resistor to increase the decay rate of the winding currents. If this resistor is not used, the maximum stepping rate will be limited by the time it takes the winding current to decay.



The opto-coupler and buffer circuits that drive the base of the power transistors are the same as those we used previously for the relay drive circuit. The opto-couplers are used to provide the isolation between the microcontroller ground and the high power circuit ground, while the inverting buffers (7406) are used to supply enough current to the opto-coupler diodes. *Note that the stepper motor drive circuit in Figure 6.4 has an inverting property. This means that logic 1 at an input terminal will cause the corresponding winding to turn off, while logic 0 causes it to turn on.*

## 6.2 Generation of the stepper motor drive signals

### Program:

Write a program that drives the stepper motor in clockwise direction for 180° in 5 sec.

### Solution:

The stepper motor operates with 200 steps per revolution. Thus, to move through 180 degrees, 100 steps are required. This is to be achieved in 5 seconds, thus a step is output to the motor every  $5/100 = 50\text{msec}$ . Timer1 is set up in continuous up-counting mode to interrupt every 50msec. Timer 1 ISR increments the step counter and the state counter and outputs the proper sequence. After 100 steps have been output, the ISR stops the motor and disables the timer interrupt so that no more commands are sent to the motor.

The complete program listing is as below-

$$\text{Timer 1 period value} = \text{CPUCLK/Prescaler/freq} = 30 \times 10^6 / 32 / 20 = 46875$$

```

;*****
; File Name:      ch6_e1.asm
; Target System: C240x Evaluation Board
; Description:    This program moves the stepper motor by 180 deg.
;                in 5 seconds, given the motor has a stepping
;                angle of 1.8 deg or 200 step/revolution
;                T1 - T4 -> IOPB0-3 (Connector P1 Pin 5, 6, 7, 8)
;                TIMER 1 PERIOD INTERRUPT FOR 50MS I.E. 46875
;
;*****
;*****
;                SYSTEM OPTIONS
;*****
real_time   .set  0      ; 1 for real time mode, otherwise set 0
T1PERIOD    .set  46875  ; for 50ms, T1PR=30000000/32/20=46875,
                       ; 20Hz=1/50ms, given 30MHz 2407 DSP,
                       ; T1 prescaler=32

```

```

;*****
;-----
; External references
;-----
        .include      "f2407.h"
;       .global       MON_RT_CNFG
        .ref          SYS_INIT
;-----
; Local Variable Declarations
;-----
        .def          GPR0                ;General purpose register.

        .bss          GPR0,1             ;General purpose register.
        .bss          STEPCTR,1          ;variable for step counter
        .bss          STATECTR,1        ;variable for state counter
        .bss          STATEVAL,1        ;variable for state value

;=====
; V E C T O R   T A B L E   ( including RT monitor traps )
;=====
;       .include "c200mmrt.i" ; Include conditional assembly options.

        .global _c_int0,PHANTOM,GISR1,GISR2,GISR3,GISR4,GISR5,GISR6

;=====
; M A I N   C O D E   - starts here
;=====
        .text
_c_int0:  CALL        SYS_INIT            ; DSP initialization
;-----
; Initialise the Real time monitor
;-----

        POINT_PG0
;---Real Time option-----
        .if (real_time)
            CALL        MON_RT_CNFG      ;For Real-Time
        .endif
;-----

;-----
; System Interrupt Init.
;-----

;---Real Time option -----
        .if (real_time)
            SPLK #0000000001000010b,IMR ;En Int lvl 2 & 7 for T1, RT
            ;|||||!!!!|!!!!|
            ;5432109876543210
        .endif

        .if (real_time != 1)
            SPLK #0000000000000010b,IMR ;Enable Int 2 only for T1
            ;|||||!!!!|!!!!|!!!!
            ;5432109876543210
        .endif

        SPLK #0FFFFh, IFR                ; Clear any pending Ints

```

```

;-----
;
;
;           RESET SECTION - BEGINS
;-----
LDP      #DP_EVA                ;set data page
SPLK    #0FFFFh, EVAIFRA        ;clear all EVA group A interrupts
SPLK    #0FFFFh, EVAIFRB        ;clear all EVA group B interrupts
SPLK    #0FFFFh, EVAIFRC        ;clear all EVA group C interrupts
SPLK    #00000h, EVAIMRA        ;enabled desired EVA group A interrupts
SPLK    #00000h, EVAIMRB        ;enabled desired EVA group B interrupts
SPLK    #00000h, EVAIMRC        ;enabled desired EVA group C interrupts

LDP      #DP_EVA                ;set data page
SPLK    #0, GPTCONA
SPLK    #0, T1CON
SPLK    #0, T2CON

SPLK    #0, COMCONA
SPLK    #0, ACTRA
SPLK    #0, DBTCONA
SPLK    #0, CAPCONA
;-----
;           RESET SECTION - ENDS
;-----
LDP      #STATECTR
SPLK    #0, STATECTR            ;Initialize state counter
SPLK    #0, STEPCTR            ;Initialize step counter

LDP      #DP_PF2
SPLK    #0000h, MCRA            ;Set pins to be I/O
SPLK    #0F0Fh, PBDATDIR        ;Select IOPB0-3 as outputs

LDP      #DP_EVA
SPLK    #0055h, GPTCONA
SPLK    #T1PERIOD, T1PR        ;Set timer 1 period 50ms
SPLK    #0, T1CNT
SPLK    #0, T2CNT

SPLK    #150Ah, T1CON           ;Set PRESCALER=32

SPLK    #0080h, EVAIMRA

SBIT1 T1CON,          0000000001000000b ;Enable timer 1 by setting
;                   ||||| Bit6 of T1CON to 1
;                   5432109876543210

CLRC    INTM                    ;Enable interrupts
END     B      END

;=====
; I S R - GISR2
; Description:   Determines the current state and outputs the
;               appropriate sequence to the transistors every
;               50ms. It also keeps track of the number of steps
;               and stops the motor when it has covered 180 degrees.
; Modifies:    STATEVAL, STATECTR, STEPCTR, PBDATDIR, EVIMRA
;=====
;Context save regs
GISR2    MAR    *,AR1            ; AR1 is stack pointer

```

```

MAR    *+                ; skip one position
SST    #1, *+            ; save ST1
SST    #0, *+            ; save ST0
SACH   *+                ; save ACC high
SACL   *                 ; save ACC low

LDP    #DP_EVA
SPLK   #0FFFFh,EVAIFRA   ; Clear all Group A interrupt flags
;=====
;Start main section of ISR
;=====
LDP    #STEPCTR
LACC   STEPCTR
ADD    #1
SACL   STEPCTR           ;Increment step counter
SUB    #101
BCND   NXT1,NEQ          ;If number of steps < 100, goto NXT1
LDP    #DP_PF2           ;Load DP for I/O registers
SPLK   #0F0Fh,PBDATDIR   ;Turn off all transistors,
                        ;stop the motor if step=100
LDP    #DP_EVA           ;Load DP for event manager
SPLK   #0000h, EVAIMRA   ;Disable timer 1 period interrupt
B      FIN
NXT1   LDP    #STATECTR
LACC   #STATE
ADD    STATECTR          ;ACC=address of current state
TBLR   STATEVAL          ;Move the state to a vairable
LACC   STATEVAL          ;Load the state
LDP    #DP_PF2
SACL   PBDATDIR          ;Output the current state

LDP    #STATECTR
LACC   STATECTR
SUB    #3
BCND   NXT, NEQ          ;If state counter < 3, jump to NXT
SPLK   #0,STATECTR      ;Reset state counter
B      FIN
NXT    LACC   STATECTR
ADD    #1
SACL   STATECTR          ;Increment state counter

;=====
;End main section of ISR
;=====
;Context restore regs
FIN    POINT_PG0
MAR    *, AR1            ;make stack pointer active
LACL   *-                ;Restore Acc low
ADDH   *-                ;Restore Acc high
LST    #0, *-            ;load ST0
LST    #1, *-            ;load ST1
EINT                                ;Enable all interrupts, clear INTM to 0.
RET

;=====
; I S R - PHANTOM
;
; Description:    Dummy ISR, used to trap spurious interrupts.
;=====

```

PHANTOM	B	PHANTOM
GISR1	RET	
;GISR2	RET	
GISR3	RET	
GISR4	RET	
GISR5	RET	
GISR6	RET	
STATE	.word 0F0Dh	; Position 2
	.word 0F0Bh	; Position 1
	.word 0F0Eh	; Position 4
	.word 0F07h	; Position 3

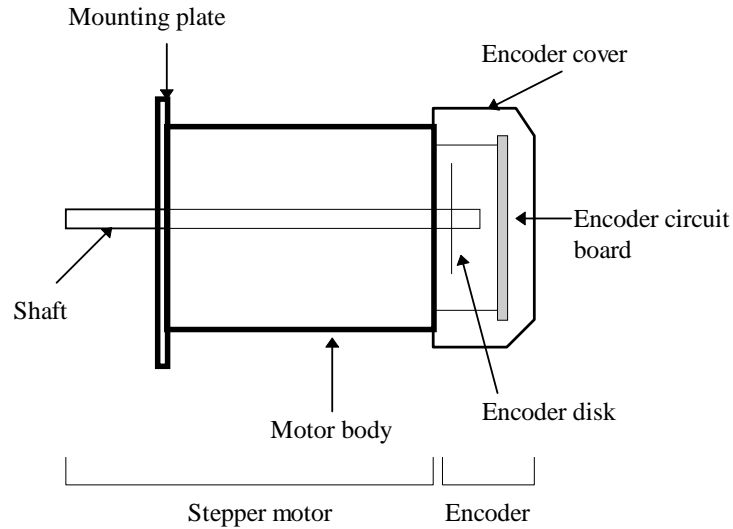
## **LABORATORY EXPERIMENT 5          Stepper Motor Drives**

### **Objectives**

In this lab, the students will learn the basic operation of a stepper motor and its interfaces to a DSP. Upon completion of this lab the students should be able to understand and write assembly language programs to control the stepper motor.

### **Discussion**

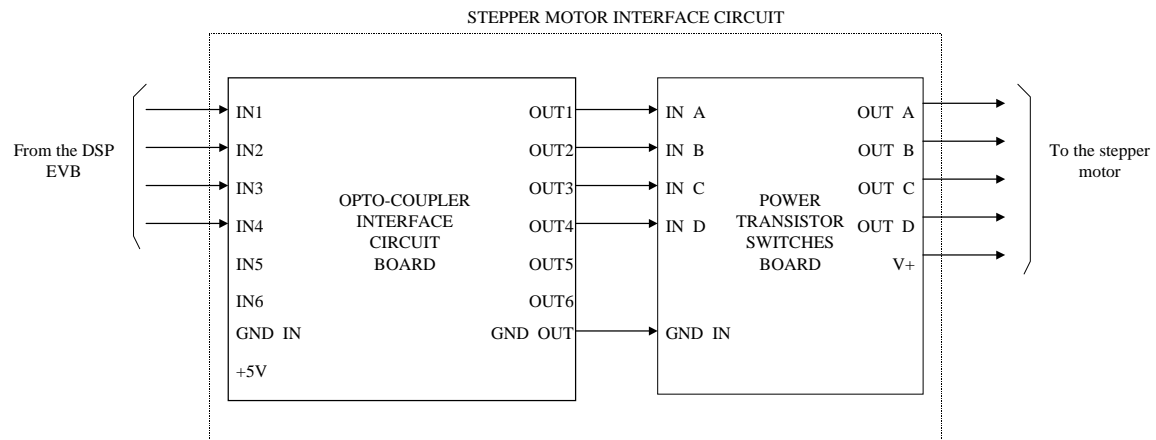
In this lab, we will use an ASTROSYS C245 stepper motor. This motor is a 4-phase bifilar stepper motor with stepping angle of 1.8 degree and 1 A per phase current. It is equipped with a SHARP 3GP1R26W bi-directional optical incremental encoder which has 49 pulse/ revolution resolution and a zero index signal. The construction of the stepper motor and its encoder is shown in Figure 1.



**Figure 1 Construction of the ASTROSYS stepper motor**

An interface circuit is needed to drive the stepper motor from the DSP port. Figure 2 shows the schematic diagram of the stepper motor interface circuit. In the lab the interface circuit is broken down into two separate boards as shown in Figure 3: *the opto-coupler interface circuit board* and *the power transistor switches circuit board*. The opto-coupler interface circuit board is the same as the one we have used for the relay interface in the previous lab, while the power transistor switches circuit board contain the rest of the stepper motor interface circuit.





**Figure 3** *The opto-coupler interface circuit board and the power transistor switch board*

## **Procedure**

### **Setup**

1. Make sure that the EVM system has been properly setup as in the previous lab.
2. Connect the power transistor switches circuit board to the stepper motor using the connectors and wires provided.
3. Establish the interconnection between the power transistor switches circuit board and the opto-coupler interface circuit board.
4. Connect the opto-coupler board inputs IN1 to IN4 to pins IOPB0 to IOPB4 of the EVM respectively and the GND and +5V connections.
5. Turn on the EVM power supply and the power for all the boards.
6. Launch Code Composer and run the program.



## **Laboratory Assignments**

### **Stepper Motor Drive**

1. Write a program that drives the stepper motor to rotate in CW direction using the single-phase excitation scheme and perform the following experiments :
  - a) Drive the stepper motor to rotate for 18 degrees in 2 sec
  - b) Drive the stepper motor to rotate for 90 degrees in 1 sec
  - c) Drive the stepper motor to rotate for 1 revolution in 5 sec
  - d) Drive the stepper motor to rotate for 10 revolution in 10 sec
  - e) Drive the motor to run continuously with stepping rate of 100 ms/step
  - f) Drive the motor to run continuously with stepping rate of 2 ms/stepVerify the result of your program and observe the output wave-forms from the EVM using the oscilloscope.
2. Repeat 1 and 2 using the dual-phase excitation scheme.
3. Repeat 1 and 2 using the half-step-mode excitation scheme.
4. Repeat step 1 to 3 but this time for driving the motor in CCW direction