CHAPTER 7

DC MOTOR DRIVE

In this chapter, a DSP-based DC motor speed control system is described. The DC motor used is of permanent magnet type. The model of the permanent magnet DC motor is first developed, followed by the discussion on the mechanism that will be used to drive the motor.

7.1 Permanent Magnet DC Motor

A permanent magnet DC motor consists of a permanent magnet stator and armature windings in the rotor. The armature winding is supplied with a DC voltage that causes a DC current to flow in the windings. Interaction between the magnetic field produced by the armature current and that of the permanent magnet stator causes the rotor to rotate. The equivalent circuit of a PM DC motor is shown in Figure 7.1.

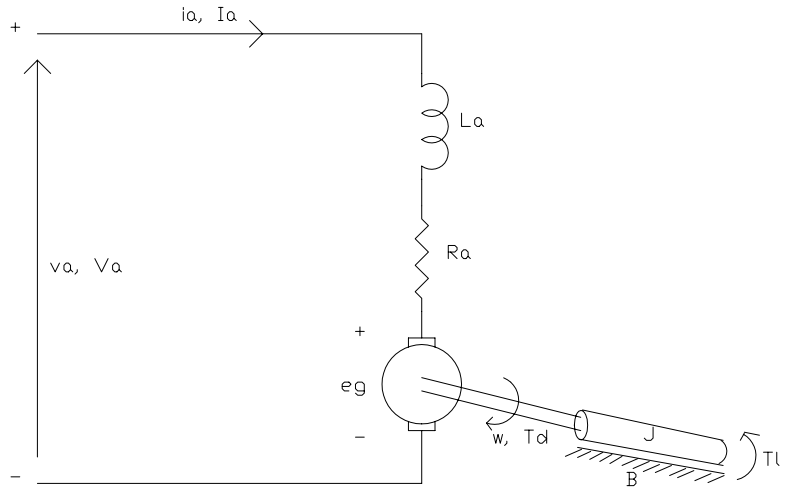


Figure 7.1. *Equivalent circuit of a permanent magnet DC motor*

The equivalent circuit in Figure 7.1 includes the resistance R_a and inductance L_a of the motor's armature windings. When a DC supply voltage V_a is applied to the

armature, current i_a flows in the armature and the motor torque to balance the load torque. Due to the movement of the armature in the magnetic field a back-emf voltage e_g is generated. The back-emf voltage opposes the voltage applied to the motor terminal and is proportional to the speed of the motor.

The equations describing the characteristic of a PM DC motor can be determined from Figure 7.1. The instantaneous armature current i_a can be found from:

$$v_a = R._a\, i_a + L_z \frac{di_a}{dt} + e_g \tag{7.1}$$

The back-emf is expressed as:

$$e_g = K_E \omega \tag{7.2}$$

The torque developed by the motor is proportional to the armature current and is expressed as

$$T_m = K_t i_a \tag{7.3}$$

The developed torque must be equal to the load torque:

$$T_m = J \frac{d\omega}{dt} + B\omega + T_L \tag{7.4}$$

where:

ω = motor speed, rad/s

B = viscous friction constant. N.m/rad/s

K_E =back emf constant

K_t = torque constant

L_a = armature circuit inductance, H

R_a = armature circuit resistance, ohm

T_L = load torque, N-m

T_m = motor developed torque, N-m

Under the steady state conditions, the time derivatives in these equations are zero and the steady state average quantities are:

$$E_g = K_E \omega \tag{7.5}$$

$$V_a = R_a I_a + E_g = R_a I_a + K_E \omega \tag{7.6}$$

$$T_m = K_t I_a = B\omega + T_L \tag{7.7}$$

From equation 7.6, the steady state speed of a permanent magnet motor can be found from:

$$\omega = \frac{V_a - I_a R_a}{K_E} \tag{7.8}$$

It can be noticed from Equation 7.8 that **the motor speed can be varied by controlling the armature voltage V_a or the armature current I_a.**

The direction of rotation of the motor can be reversed by reversing the polarity of the voltage applied to the terminal. If the voltage applied to the motor terminal is reversed, the direction of current flowing in the armature winding is also reversed. This will cause the motor to produce torque in the reverse direction. Note here that the polarity of the back-emf voltage is also reversed.

7.2. The Motor Drive

As described in the previous section, the speed of a permanent magnet DC motor can be altered by varying the voltage applied to its terminal. One way of varying the applied voltage is by using the pulse-width modulation (PWM) technique. Using this technique, a fixed frequency voltage signal with varying pulse-width is applied to the motor terminal. Figure 7.2 shows an example of a PWM signal where T is the signal period, t_d is the pulse-width, and V_m is the signal amplitude. The average voltage can be calculated from

$$V_{avg} = \frac{1}{T} \cdot \int_0^T v(t)\, dt = \frac{t_d}{T} \cdot V_m = k \cdot V_m \tag{7.9}$$

where k is the duty cycle defined as :

$$k = \frac{t_d}{T} \tag{7.10}$$

From equation (7.9) it can be seen that the average (DC component) of the voltage signal is linearly related to the pulse-width of the signal, or the duty cycle of the signal since the period is fixed. Therefore, varying the duty cycle of the signal can alter the voltage applied to the motor terminal.

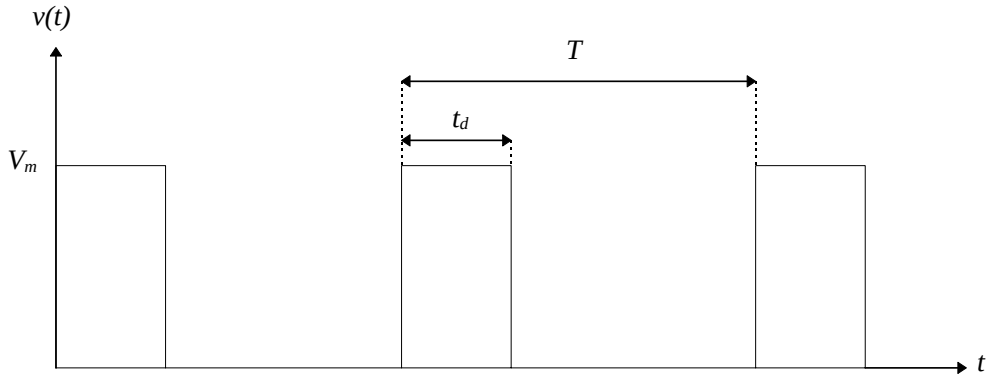


Figure 7.2 *A PWM signal*

The PWM voltage waveforms for the motor can be obtained using a special power electronic circuit called a *DC chopper*. A DC chopper basically uses power switching devices to switch a constant DC voltage on and off according to a specified switching scheme in order to obtain the required voltage and current waveforms. There are various types of DC chopper configurations, which can be found in textbooks on power electronics. Reference [1] given at the end of this chapter gives excellent explanations on different types of DC chopper circuits and their theories of operations.

In this section, we will discuss one type of DC chopper configuration called *bridge power converter* also known as *H-bridge converter*. The schematic diagram of this converter is shown in Figure 7.3. T1 to T4 are controlled switches that can be implemented using power semiconductor devices such as BJT, Power MOSFET, or IGBT. These devices provide low resistance for the current flow when they are turned on and very high resistance when turned off. Diodes D1 through D4 provide a path for preserving the continuity of the current flow when one or more of the switches are turned off. This is necessary to protect the power switches from excessive voltage spike due to

the inductive load presented by the DC motor. These diodes are also known as *freewheeling diodes.*

The DC voltage supply V_m can be obtained from a rectified ac signal or a DC voltage source such as a car battery.

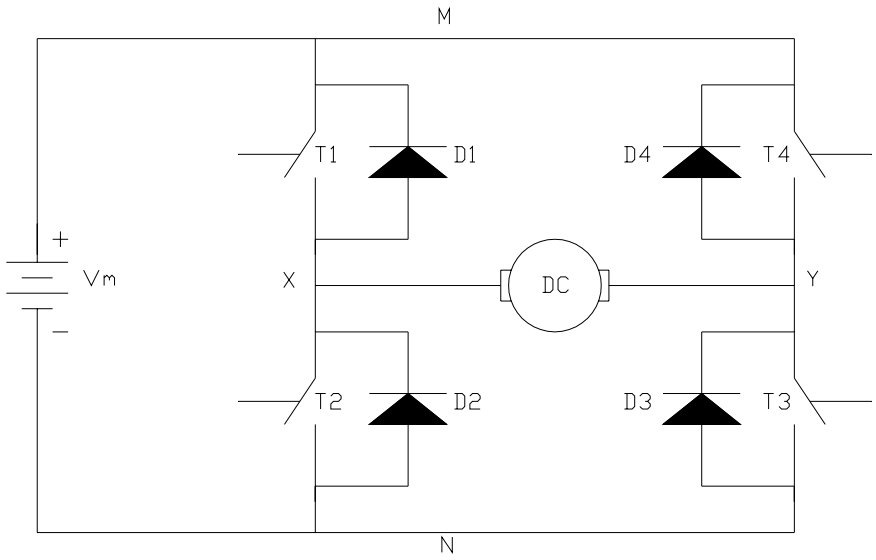
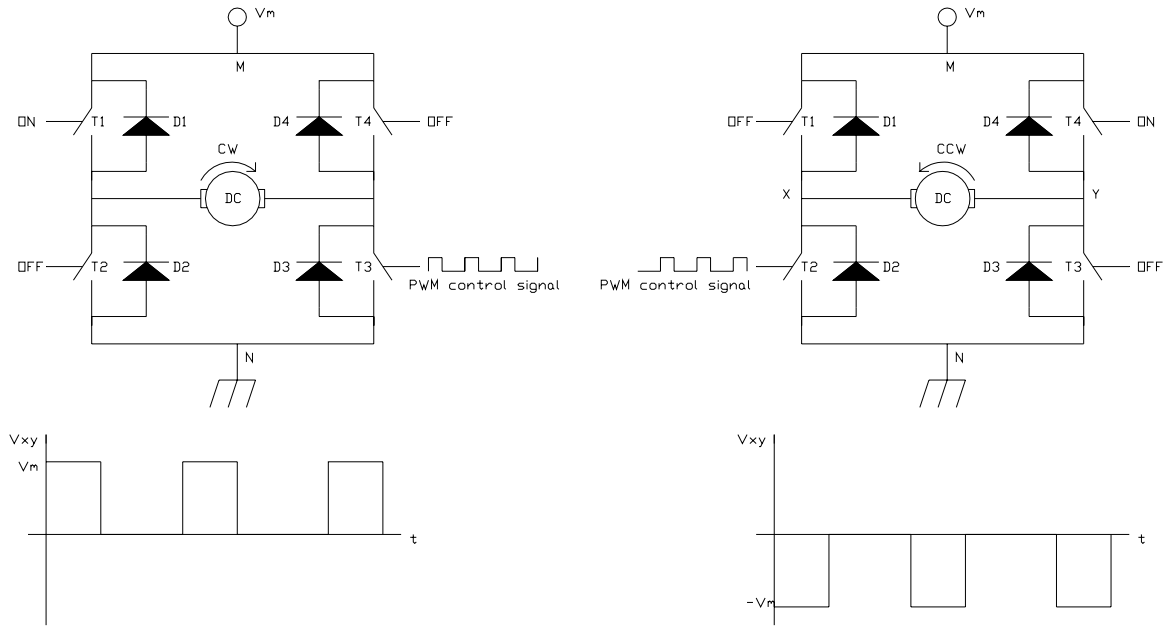


Figure 7.3 *H-bridge converter for DC motor drive*

In this section, we will examine two switching schemes for the bridge converter that can be used to drive the motor. Both schemes allow polarity reversal of the voltage applied across the motor and hence, provide for bi-directional control of the motor. The following paragraphs discuss the basic principles for the two switching schemes.

Scheme 1

This scheme uses only two of the switches of the bridge converter to drive the motor in one direction. Figure 7.4 shows how the switches are controlled to obtain the pulse-width modulated voltage waveform at the motor terminal. Figure 7.4(a) shows how the switches are controlled to drive the motor in one direction (say CW), while figure 4(b) to drive the motor in the other direction (CCW). The resulting voltages across the motor, for both cases are also shown in this figure.



(a) (b)

Figure 7.4 *Switching control for Scheme 1*
(a) *CW direction* ($V_{XY}>0$)
(b) *CCW direction* ($V_{XY}<0$)

In Figure 7.4 (a) switches T2 and T4 are kept off, and T1 is kept on all the time, while the PWM control signal is applied to switch T3. Figure 7.5 shows the flow of current when T3 is driven ON and OFF. In this figure the motor is replaced with its equivalent circuit and only devices that take part in current conduction are drawn. When T3 is driven ON, node Y is connected to ground which causes current to flow from the supply voltage to ground through T1, the motor, and T3. When T3 is turned OFF, the current will continue to flow due to the energy stored in the inductance of the motor. In this case, diode D4 will turn on and the current flows through closed path consisting of the T1, the motor, and D4.

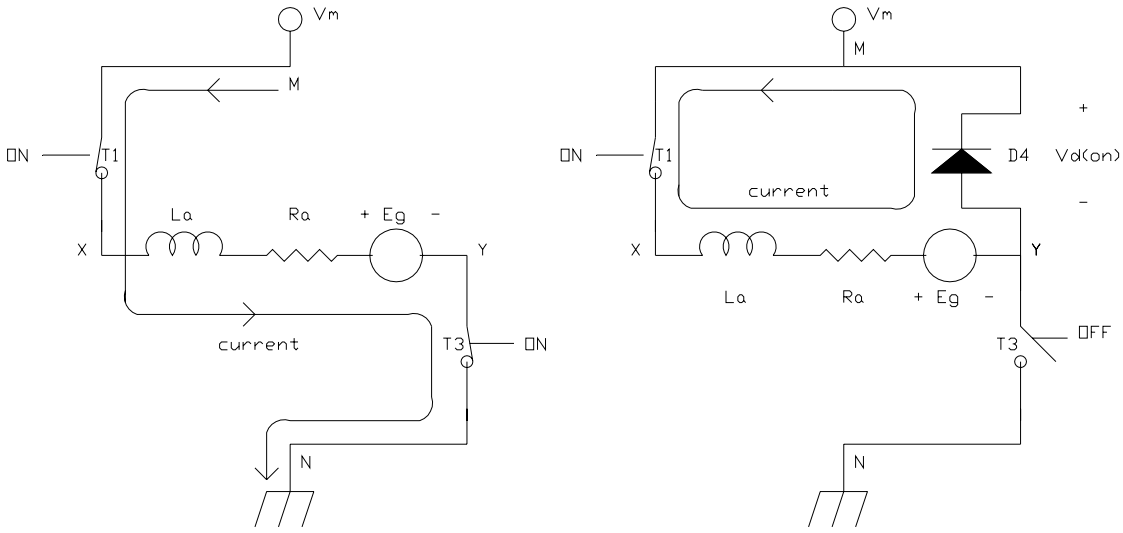


Figure 7.5 *Current flow paths when T3 is ON (a) and OFF (b) in Figure 7.4(a)*

The steady state voltage and current waveforms resulting from the switching scheme of Figure 7.4 (a) are shown in Figure 7.6. When T3 is ON, the current ramps up exponentially with a time constant determined by the motor inductance and the total resistance of the current flow path. The voltage across the motor terminal V_{XY}, in this case, is equal to the supply voltage V_m. When T3 is OFF, the current decays exponentially through T1 and D4 as in Figure 7.5 (b) and the voltage across the motor terminal V_{XY} will be equal to the negative of the diode turn on voltage (normally in the order of 0.6V - 0.8V).

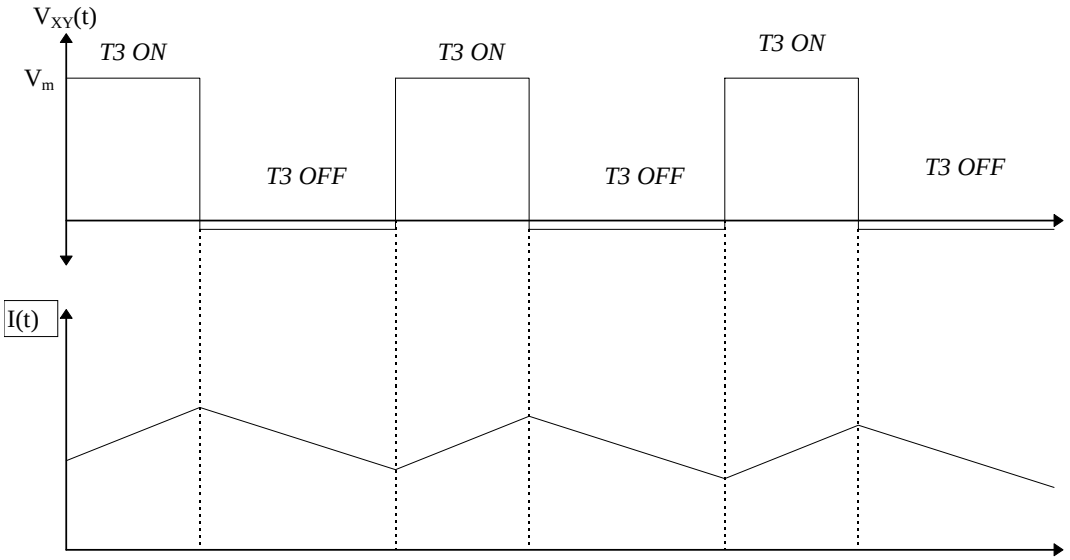


Figure 7.6. *Voltage and current waveforms for Figure 7.4 (a)*

When the motor is driven in the CCW direction, switches T1 and T3 are turned OFF, and T4 is turned ON all the time, while the PWM control signal is applied to switch T2. When T2 is turned ON the current now flows from the supply voltage to the ground through T4 the motor, and T2. This causes the motor terminal voltage V_{XY} to be equal to the negative of the supply voltage V_m. When T2 is turned OFF, the current flows through diode D1, and the voltage across the motor is clamped to the diode turn-on voltage. The voltage and current waveforms for the case when the motor is driven in the CCW direction (Figure 7.4(a)) are the same as in Figure 7.6 except that the polarities are reversed.

Scheme 2

In this scheme, all the four switches of the bridge converter are utilized to drive the motor in one direction. Looking at the bridge configuration in Figure 7.3, we can see that there are two ways to force the voltage at the motor terminal (V_{XY}) to zero i.e. either by turning on T1 and T4 or by turning on T2 and T3 simultaneously. To drive the motor in the CW direction the switches are controlled as shown in Figure 7.7. The current flow paths are also shown in Figure 7.7.

The scheme uses four states of the switches to obtain the PWM voltage waveform at the motor terminal. To make $V_{XY} = V_m$, switches T1 andT3 are turned ON while T2 and

T4 are turned off (state I and state III). To force V_{XY}=0, either T1 and T4 are turned ON (state II) or T2 and T3 are turned ON (state IV). The four switches states are summarized in Table 7.1 below.

State	T1	T2	T3	T4
I	ON	OFF	ON	OFF
II	ON	OFF	OFF	ON
III	ON	OFF	ON	OFF
IV	OFF	ON	ON	OFF

Table 7.1 *Switches states for CW drive*

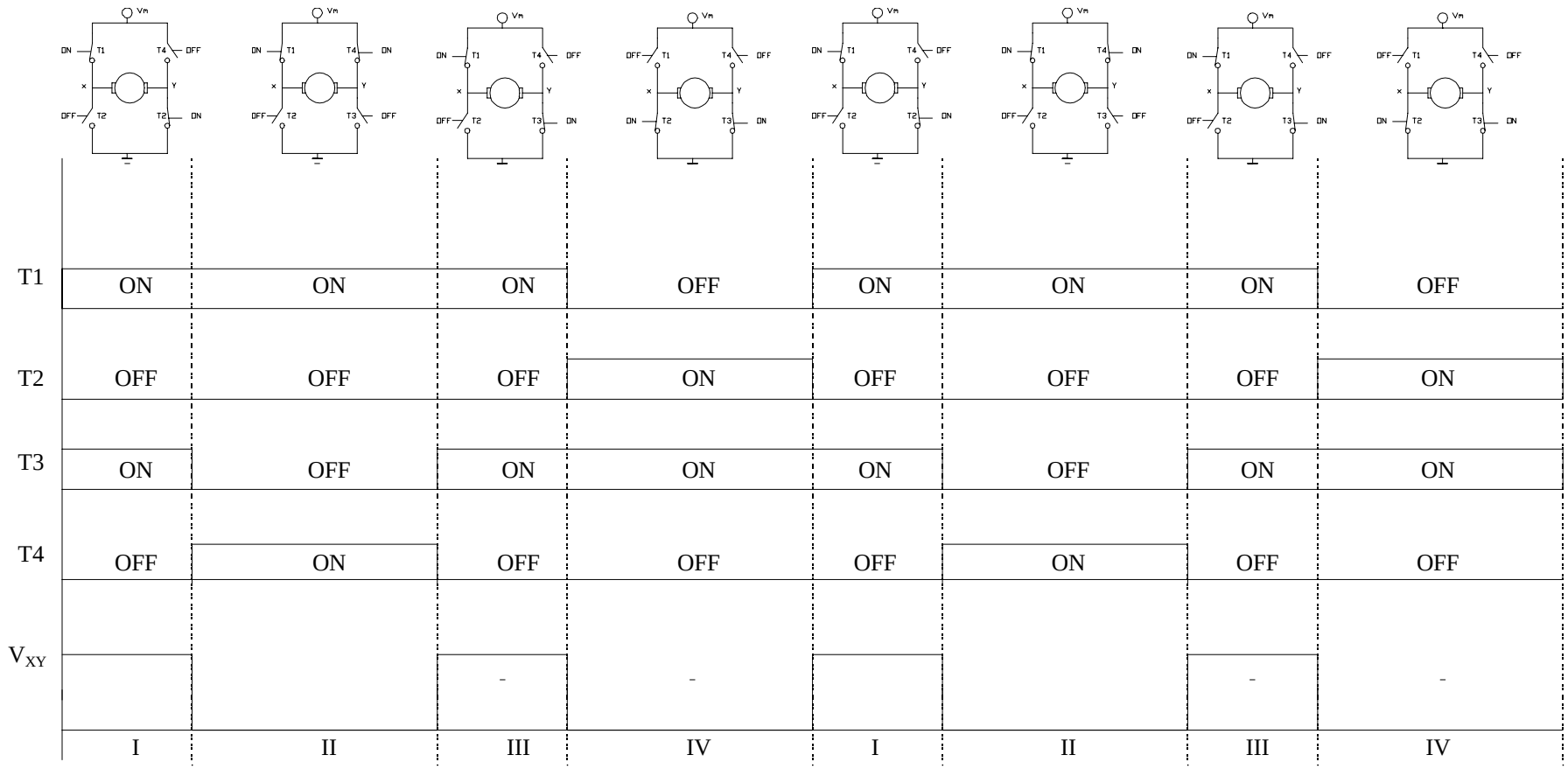


Figure 7.7 *Switches control for CW drive (scheme 2)*

To drive the motor in the other direction, we need to obtain V_{XY} = - V_m during the high period of the signal (state I and state III). This can be accomplished by turning on switches T2 and T4. The switch states for this case are shown in Table 7.2.

State	T1	T2	T3	T4
I	OFF	ON	OFF	ON
II	ON	OFF	OFF	ON
III	OFF	ON	OFF	ON
IV	OFF	ON	ON	OFF

Table 7.2 *Switches states for CCW drive*

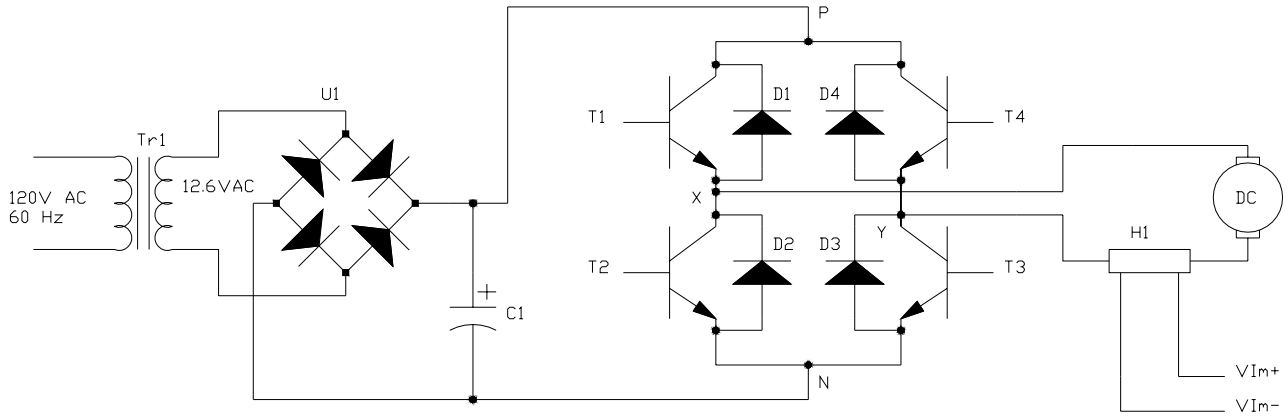
Note that in the above schemes, two switches that reside in the same leg of the bridge (T1 and T2 or T3 and T4) should not be turned on at the same time since this would result in the supply voltage being short circuited to ground.

7.3 Circuits realizations

As mentioned earlier, the switches in the H-bridge are implemented using power semiconductor devices. Below the realization of an H-bridge power converter using BJT power transistors as the switching devices is discussed. In this section, we will also discuss how to interface an H-bridge power converter to the DSP ports. In this case, a special interface circuit called *base-drive circuit* will be developed.

7.3.1 BJT Bridge Converter

Figure 7.8 shows the H-bridge converter implemented using BJT power transistors. It uses 4 BJT power transistors of the type 2N6547. The 2N6547 is an NPN power transistor which can handle collector current up to 10 amp and is specifically designed for use with inductive loads such motor or solenoid. The freewheeling diodes are implemented using MUR410, 10A ultra-fast switching diodes.




```
Component list:
T1, T2, T3, T4: 2N6547              Tr1 : 12.6V,3 A transformer
D1,D2,D3,D4   : MUR410              C1  : 5000uF electrolyte capacitor
U1 : KBL01 diode bridge rectifier   H1  : SY-05 current sensor.
```

Figure 7.8 *H-Bridge Power Converter*

In Figure 7.8, the DC supply for the motor is obtained from a rectified ac-signal. The 120 V ac voltage source is first converted to a lower level ac voltage using a step down transformer. A diode bridge rectifier (U1) then rectifies the secondary output of the transformer. A large capacitor C1 in Figure 7.8 is used to filter the high frequency content of the rectified signal.

The circuit in Figure 7.8 also includes an SY-05 Hall effect current sensor (H1) which can be used to detect or observed the motor current. This device produces an output voltage proportional to the current flowing through its sensing terminal.

The data-sheets for the devices used in the circuit are given at the end of the chapter.

7.3.2 Interface circuit (Base-drive Circuit)

The base drive circuit is used to interface the bridge converter to the DSP. Following are functions that must be carried out by the base drive circuits:

1) The base drive circuit must be able to supply sufficient current and apply necessary voltage to the base of each transistor in the bridge converter to ensure proper turn on and off of the devices. Note that the emitters of the upper transistors T1 and T4 are connected to the floating nodes X and Y respectively, while the emitter of the lower transistors T2 and T3 are both connected to node N. This interconnection requires that the voltage signal that drives the base of each transistor be referenced to its corresponding emitter nodes.
2) The base drive circuit must provide a mechanism that prevents the transistors that reside at the same leg of the bridge converter to turn on at the same instant of time. Recall that in theory the switching schemes described earlier never turns on the transistors at one leg at the same time. In the scheme 2, for example, T1 and T2 (or T3 and T4) are always switched in the opposite states, one being ON and the other OFF. However, a problem might occur during the transitions between the ON and OFF states, due to the fact that the switching devices used in practice can not turn ON or turn OFF instantaneously. To prevent short-circuiting during this period, we must allow one transistor to turn OFF completely before turning ON the other transistor by delaying the turn ON process. A circuit that accomplishes this is called a *lock out circuit*. The implementation of the base drive circuit, therefore, must include the lock out circuits.
3) The base drive circuit must provide isolation between the power converter that carries high power signals, and the DSP circuitry that carries low level digital signals. This can be accomplished using opto-couplers as we have seen in the previous chapters.
4) The base drive circuit must ensure that currents drawn from the DSP output ports do not exceed the maximum permissible current dictated by the DSP chip manufacturer. The circuit, therefore, must contain buffer devices that enhance the DSP output current capability.

Figure 7.9 shows a base drive circuit that can accomplish the requirements stated above. The base drive contains four drives each for the four transistors in the bridge converter. The output stage of each drive consists of a fast high power op-amp, OPA633 (U3), which can supply current up to 100mA maximum. To satisfy the requirement that the emitter node of each transistor is referenced to the different nodes at the converter, three pairs of power supplies with three separate ground references are used to drive the output stages. Each power supply pair denoted by VPPx (+5V) and VNNx (-5V) (x=1,2,3) is referenced to its own ground point denoted by GNDx (x=1,2,3). VPP1 and VNN1 are used for T1 drive (GND1=node X), VPP2 and VNN2 are used for T4 drive (GND2=node Y), while VPP3 and VNN3 are used for T3 and T4 which share a common emitter node (node N at the bridge converter = GND3).

The output stage of each drive that carries relatively high current is separated from the rest of the circuits using a high-speed opto-coupler HP2200 (U1). The output of the each opto-coupler is pulled to the corresponding VPPx and VNNx supplies. This causes the magnitudes of the opto-coupler output voltages to be $\pm 5V$. The $\pm 5V$ applied to the input of the power op-amps will cause the op-amps to saturate and produce output voltages of magnitude approximately $\pm 3.5V$ that will turn on and off the power transistors.

The lock out function in each drive is implemented using a circuit consisting of an IC comparator (U2: LM339) and a network of resistors, capacitors and a diode. The operation of the lock out circuit can be explained by looking at Figure 7.10 where one of the lockout circuit is redrawn. Figure 7.11 shows the effect of the lockout circuit on the output of the opto-coupler that drives the power op-amp.

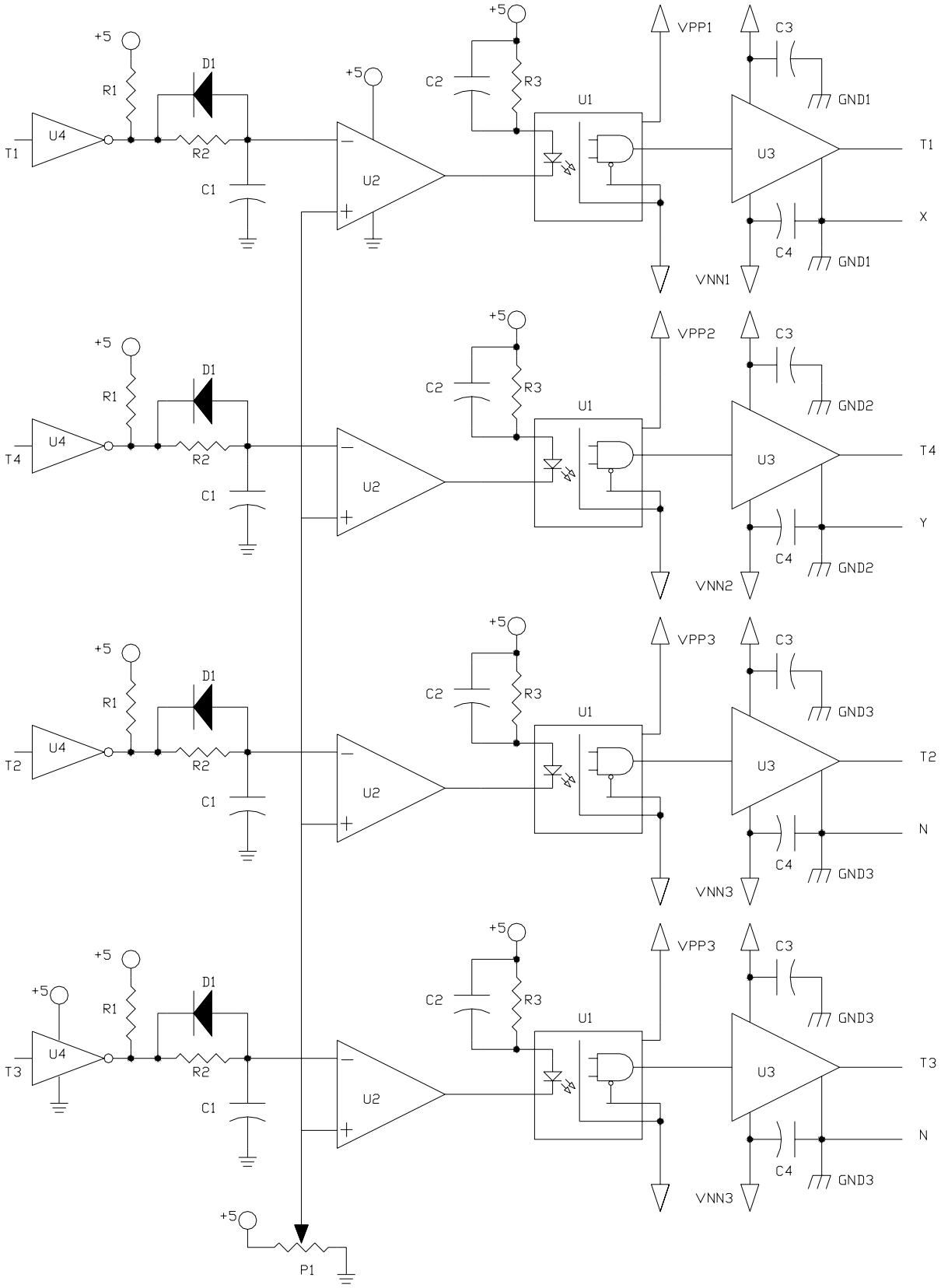


Figure 7.9. *Base drive circuit for the BJT H-bridge converter*

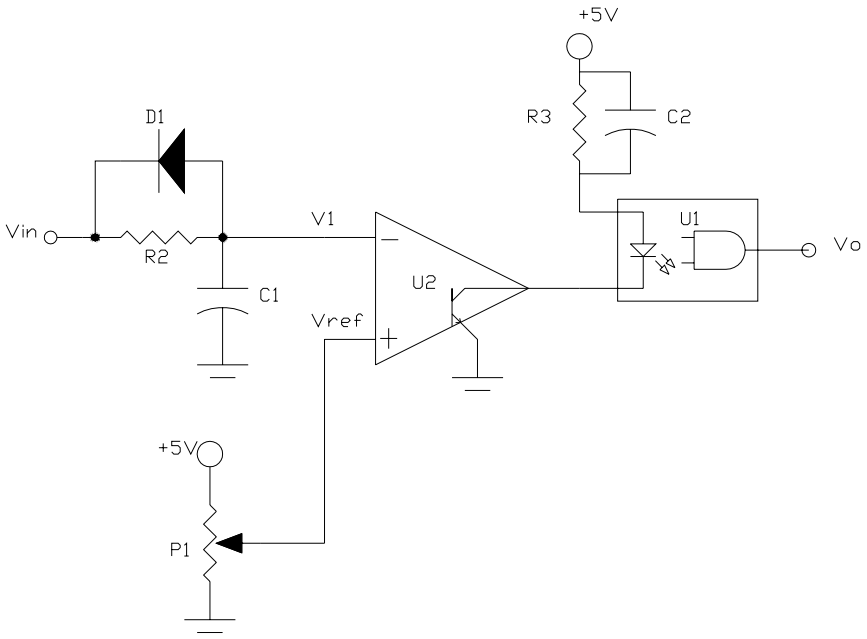


Figure 7.10 L*ockout circuit*

The IC comparator LM339 (U2) used in the lockout circuit has an open collector output as shown in Figure 7.10. When the difference between the non-inverting and the inverting input of the comparator (V_{ref} - V_1) is positive the output transistor inside the IC is turned ON pulling the output to ground. This causes the LED of the opto-coupler to turn ON and the output of the opto-coupler becomes +5V (VPP). When the difference is negative the output transistor is turned off causing the LED to turn OFF and the output of the opto-coupler becomes -5V(VNN).

A reference input V_{ref} obtained from the +5V supply and a variable resistor P1 is applied to the non-inverting output of the comparator. When the input voltage Vin is LOW (~0V), the inverting input is also zero and the output of the opto-coupler is -5V. When the input voltage changes from 0 to +5V, diode D1 turns off (since it is reversed biased), causing the current to flow through resistor R_2 and charge the capacitance C_1. This causes the voltage at the inverting input of the comparator (V_1) to rise exponentially according to the equation:

$$V_1 = +5 \cdot \left(1 - e^{-\frac{t}{R_2 C_1}} \right)$$

When V_1 is still less than V_{ref}, the output transistor of the comparator is turned off preventing the opto-coupler output V_o to change immediately. Only when V_1 raises to a value greater than V_{ref}, will the output transistor turn ON and the output V_o change to +5V as shown in Figure 7.11. This way the process of turning on the power transistor is delayed and the amount of delay is determined by the time constant R_2 C_1 and the voltage reference V_{ref}. Since the values of the resistor R_2 and capacitor C_1 are fixed, the delay time can be adjusted by varying the value of V_{ref} using the potentiometer P1. Larger value of V_{ref} results in longer delay time, lower value causes shorter delay time.

On the other hand, if the input voltage V_{in} changes from HIGH (~5V) to LOW (~0V), diode D1 turns ON (since it becomes forward biased) providing a low resistance path for the charge stored in capacitor C_1 to discharge quickly to ground. This causes the value of V_1 to drop quickly below the reference voltage $V_{ref.}$ This mechanism speeds up the change of output voltage V_o to VNN (- 5V) and the turning OFF process the power transistor.

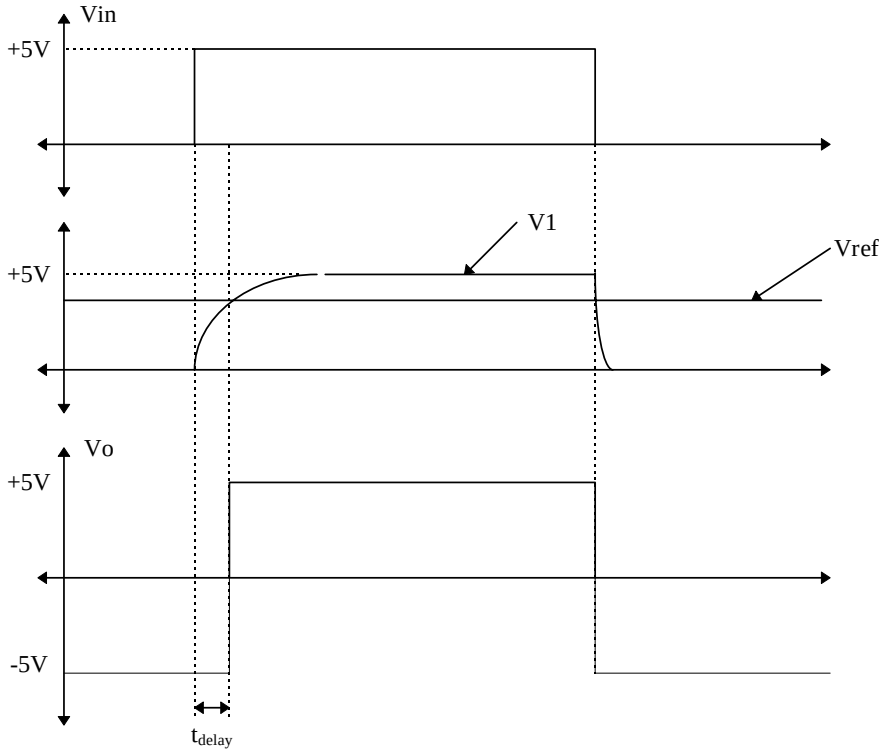


Figure 7.11. *Effects of the lock-out circuit on the turning ON and turning OFF processes*

The last requirement the base drive circuit must perform is to ensure the current drive capability of the DSP port is not exceeded. For this purpose the base drive circuit in Figure 7.9 includes inverting buffer gates U4 (7406 IC). ***Note that the inclusion of the inverting buffers at the input of the base drive circuit inverts the logic for turning on the power transistor. Logic 0 at one of the inputs of the base drive circuit causes the corresponding power transistor to turn ON, logic 1 causes it to turn OFF.***

7.4 Example Programs

To control the operation of the H-bridge converter for the DC motor drive, the DSP must generate the required base drive signals. The base drive signals for the switching scheme 1 is relatively simple to implement, since it requires only a single PWM control signal to drive the motor in one direction. For this switching scheme the power converter and the base drive circuit may be connected to the DSP ports as shown in Figure 7.12. Since the lower transistors T2 and T3 need to be driven by PWM control signals, the drives for these transistors are connected to the output pins of the DSP T1PWM and T2PWM. The upper transistors T1 and T4 do not require PWM control signals and therefore can be connected to any general purpose port pins. In Figure 7.12, T1 and T4 are connected to the I/O port B pins IOPB2 and IOPB3.

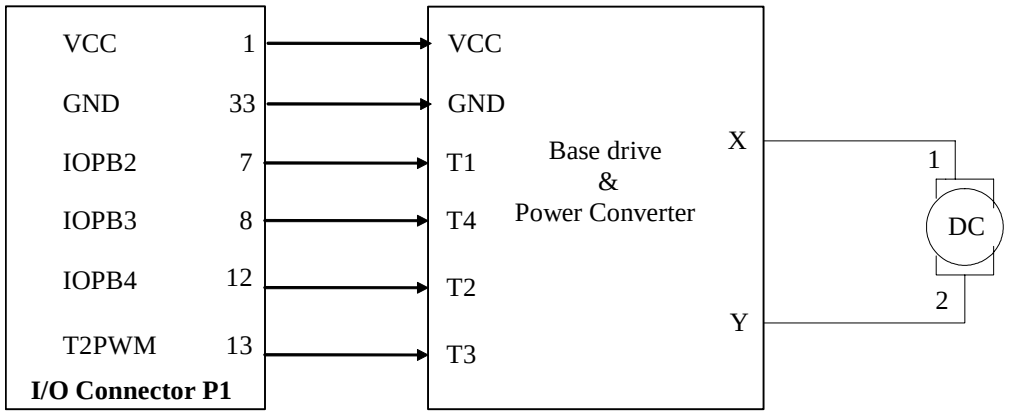


Figure 7.12 *DSP EVB ports interconnections for switching scheme 1*

Below is a program for the DSP that can be used to drive the motor in CW direction using the switching scheme 1 and the port interconnections as shown in Figure 7.12. The PWM signal used has 50% duty cycle and 20 kHz frequency.

Program 1

The basic scheme is to output a "0" at T1 to turn it ON. A PWM wave of duty cycle 50% and frequency 20 kHz at T1PWM. This PWM wave is generated using Timer 1 in continuous up-counting mode. A "1" is output at IOPB3 and IOPB5 in order to turn OFF T1 and T4.

The complete code is given below -

```
;*********************************************************************
; File Name:       ch7_e1.asm
; Target System: C240x Evaluation Board
; Description:    DC MOTOR CLOCKWISE ROTATION SCHEME 1:
;                 Apply PWM on only one of the four power switches.
;                 These four switches are driven by I/O/TxPWM pins:
;           IOPB2 - T1; IOPB3 – T4; IOPB4/T1PWM – T3; IOPB5/T2PWM – T2
;                       PWM frequency is 5 kHz
;*********************************************************************

;*********************************************************************
;                              SYSTEM OPTIONS
;*********************************************************************
real_time   .set  0           ; 1 for real time mode, otherwise set 0
clockwise   .set  1           ; 1=clockwise, 0=counterclockwise
per_val     .set  6000  ;Value to be loaded in the GPT period register
                        ;30000000/1/5000=6000
cmpr_val    .set  3000  ;Value to be loaded in the GPT compare register

;*********************************************************************

;---------------------------------------------------------------------
; 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.

;=====================================================================
; V E C T O R    T A B L E    ( including RT monitor traps )
```

```
;=========================================================================
;        .include "c200mnrt.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         #DP_PF2
      SPLK  #0000h, MCRA      ;Set pins to be I/O
      SPLK  #3C3Ch,PBDATDIR   ;Turn off T1, T2, T3 and T4 by
                                    ;outputting 1 on all pins.

      .if clockwise=0
      SPLK  #1000h, MCRA      ;Configure the o/p pins for T1PWM(IOPB4),
                              ;IOPB2, IOPB3, IOPB5. i.e. output PWM
                              ;at input of T3.
      SPLK  #2C28h,PBDATDIR   ;Turn on T1

      LDP   #DP_EVA
      SPLK  #0045h,GPTCONA    ;Enable Compare outputs of all GPTs
                              ;GPT1 and 2 compare outputs - active low
      SPLK  #cmpr_val,T1CMPR  ;Load compare register
      SPLK  #per_val, T1PR    ;Load period register
      SPLK  #1042h, T1CON     ;Select continuous up counting mode
                              ;Prescaler = 1, enable timer compare
                              ;operation, enable timer operation.
      .else
      SPLK  #2000h, MCRA      ;Configure the o/p pins for T2PWM(IOPB5),
                              ;IOPB2, IOPB3, IOPB4. i.e. output PWM
                              ;at input of T2.


      SPLK  #1C14h,PBDATDIR   ;Turn on T4

      LDP   #DP_EVA
      SPLK  #0045h,GPTCONA    ;Enable Compare outputs of all GPTs
                              ;GPT1 and 2 compare outputs - active low
      SPLK  #cmpr_val,T2CMPR  ;Load compare register
      SPLK  #per_val, T2PR    ;Load period register
      SPLK  #1042h, T2CON     ;Select continuous up counting mode
                              ;Prescaler = 1, enable timer compare
                              ;operation, enable timer operation.
      .endif

END   B     END               ;End of program


;=======================================================================
; 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
```

It can be seen that the program that drives the motor using scheme 1 is very straightforward and simple. This is due to the simplicity of programming the PWM unit in the DSP. For switching scheme 2, this advantage cannot be utilized since we need to control the switching of the four transistors at the same time. In this case the base drive and power converter can be connected as shown in Figure 7.13.

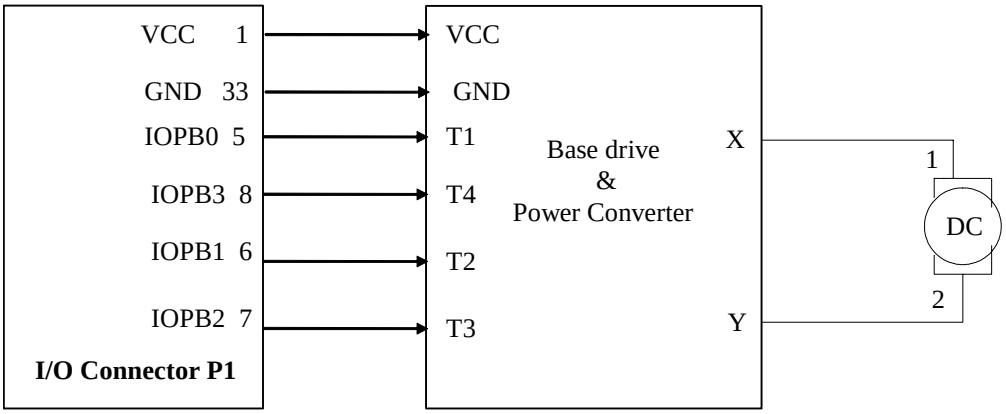


Figure 7.13 *DSP EVB ports interconnections for switching scheme 2*

Example 2:

Write a program to drive the motor in clockwise direction using scheme 2. The frequency of the PWM is 1 kHz at a duty cycle of 50%. Also, compute the speed by reading the input of an incremental encoder. The encoder outputs 2950 pulses/revolution. Store the number of pulses generated per second in a memory location SPEED.

Solution:

The pulses are output in four states as per the following sequence:

State	**T1**		**T2**		**T3**		**T4**	
	Command	**Value**	**Command**	**Value**	**Command**	**Value**	**Command**	**Value**
I	ON	0	OFF	1	ON	0	OFF	1
II	ON	0	OFF	1	OFF	1	ON	0
III	ON	0	OFF	1	ON	0	OFF	1
IV	OFF	1	ON	0	ON	0	OFF	1

Thus in each period, there are four states. The period interval is split into four equal sub-intervals. At the end of each interval, the pattern for a state is output. Timer T1 keeps track of each sub-interval. The period of the timer is set to a value corresponding to a frequency of 1 kHz / 4 = 250Hz. The timer period interrupt is set. Thus, the timer period ISR decides which sub-interval it is, and accordingly outputs the right pattern.

The incremental encoder is connected to the capture unit 1 i.e. the CAP1 input. The capture unit is programmed to generate an interrupt every time a rising edge is detected at CAP1. The CAP1_ISR counts the pulses from the encoder. At the end of every 1 second interval, the timer 1 ISR stores the number of pulses in SPEED and resets the pulse counter. The actual speed of the motor in rpm can be computes as -

RPM = (SPEED/2950) * 60

The program listing is as below -

```
;*******************************************************************************
; File Name:       ch7_e2.asm
; Target System: C240x Evaluation Board
; Description:    DC MOTOR CLOCKWISE ROTATION SCHEME 2
;                 PWM frequency is 1 kHz, duty cycle is 50%
;                 Capture the encoder pulses and calculate for the speed
;                 the incremental encoder outputs 2950 pulses/revolution
;                 The four power switches are driven by I/O pins:
;                 IOPB0 - T1; IOPB1 - T2; IOPB2 - T3; IOPB3 - T4
;*******************************************************************************

;*******************************************************************************
;                                SYSTEM OPTIONS
;*******************************************************************************
real_time   .set  0      ; 1 for real time mode, otherwise set 0
STATE1      .set  0F0Ah
STATE2      .set  0F06h
STATE3      .set  0F0Ah
STATE4      .set  0F09h
T1PERIOD    .set  234    ; 30000000/32/4000 = 234 to make PWM freq=1 kHz

;*******************************************************************************

;-------------------------------------------------------------------------------
; 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        ENCDR_CNT,1       ;Encoder pulse counter
            .bss        SEC_CTR,1         ;Second counter
            .bss        SPEED,1           ;Value of speed
            .bss        CTR,1             ;State counter
```

```
;=====================================================================
; V E C T O R    T A B L E    ( including RT monitor traps )
;=====================================================================
;     .include "c200mnrt.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  #0000000001001010b,IMR  ;En Int lvl 2, 4 & 7
                    ;|||||||||!||||||
                      ;5432109876543210
      .endif


      .if (real_time != 1)
            SPLK  #0000000000001010b,IMR  ;Enable Int 2 & Int 4
                      ;||||!!!!!!!!!!!!
                      ;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             ;Reset EVA registers
      SPLK   #0, T1CON
      SPLK   #0, T2CON
      SPLK   #0, COMCONA
      SPLK   #0, ACTRA
      SPLK   #0, DBTCONA
      SPLK   #0, CAPCONA

      LDP   #ENCDR_CNT
      SPLK  #0, ENCDR_CNT           ;Reset variables
      SPLK  #0, CTR
      SPLK  #0, SEC_CTR
      SPLK  #0, SPEED
;-----------------------------------------------------------------------
;           RESET SECTION - ENDS
;-----------------------------------------------------------------------

      LDP   #DP_PF2
      SPLK  #0008h, MCRA            ;Select IOB0-3 as outputs
                                    ;Select CAP1 function of pin

      LDP   #DP_EVA
      SPLK  #0055h, GPTCONA
      SPLK  #T1PERIOD, T1PR
      SPLK  #0, T1CNT
      SPLK  #0, T2CNT

      SPLK  #150Ah, T1CON     ;SET PRESCALER =32, continuous up mode,
                              ;Disable timer, disable timer compare

      SPLK   #02080h, CAPCONA ;Enable capture units 1 and 2
                              ;Select GP Timer 2 as time base for cap1
                              ;This time base is not used here
                              ;Detect falling edge
      SPLK   #00FFh, CAPFIFOA    ;Clear the capture unit FIFO initially
      SPLK   #0080h, EVAIMRA     ;Enable timer1 period interrupt
      SPLK   #0001h, EVAIMRC     ;Enable capture unit 1 interrupt

      SBIT1 T1CON,0000000001000000b ;Enable timer 1 by setting Bit6
            ;     ||||||||||||||||
            ;     5432109876543210
      CLRC  INTM

END   B     END

;=====================================================================
; GISR2  -        Timer interrupt service routine
; Description:    Outputs the proper sequence for all the transistors.
;                 Loads the number of pulses in SPEED at the end of
;                 every 0.1 sec interval.
```

```
; Modifies:       SPEED, ENCDR_CNT, SEC_CTR
;=====================================================================
GISR2 LDP   #SEC_CTR
            LACC  SEC_CTR
            ADD   #1
            SUB   #400
            BCND  NXT, NEQ                ;Check if 0.1 sec
            LACC  ENCDR_CNT               ;If 0.1 sec then
SPEED=ENCDR_CNT
            SACL  SPEED
            SPLK  #0, ENCDR_CNT           ;Reset ENCDR_CNT and SEC_CTR
            SPLK  #0, SEC_CTR
            B     NXT1
NXT         LACC  SEC_CTR           ;If not 1 sec increment SEC_CTR
            ADD   #1
            SACL  SEC_CTR
NXT1        LACC  CTR
            BCND  ST2, NEQ          ;Check the state number if 0
            LDP   #DP_PF2
            SPLK  #STATE1, PBDATDIR ;Output sequence for STATE1
            LDP   #CTR
            SPLK  #1,CTR            ;Update state counter
            B     FIN
ST2         SUB   #1
            BCND  ST3,NEQ           ;Check the state number if 1
            LDP   #DP_PF2
            SPLK  #STATE2, PBDATDIR ;Output sequence for STATE2
            LDP   #CTR
            SPLK  #2,CTR            ;Update state counter
            B     FIN
ST3         SUB   #1
            BCND  ST4,NEQ           ;Check the state number if 2
            LDP   #DP_PF2
            SPLK  #STATE3, PBDATDIR ;Output sequence for STATE3
            LDP   #CTR
            SPLK  #3,CTR            ;Update state counter
            B     FIN
ST4         LDP   #DP_PF2
            SPLK  #STATE4, PBDATDIR ;Output sequence for STATE4
            LDP   #CTR
            SPLK  #0, CTR           ;Reset state counter
FIN         LDP   #DP_EVA           ;Set data page
            SPLK  #0FFFFh, EVAIFRA  ;clear all EVA group A interrupts
            CLRC  INTM              ;Enable maskable interrupts
            RET

;=====================================================================
; GISR4  -  Capture Unit interrupt service routine
; Description:    Counts the number of pulses from encoder.
; Modifies: ENCDR_CNT
;=====================================================================
GISR4 LDP   #ENCDR_CNT
            LACC  ENCDR_CNT
            ADD   #1
            SACL  ENCDR_CNT         ;Increment the pulse counter
            LDP   #DP_EVA
            SPLK   #0FFFFh, EVAIFRC ;clear all EVA group C interrupts
            CLRC  INTM              ;Enable maskable interrupts
            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
```

Laboratory Experiment 6

Objectives

To understand the principles of DC motor drives using an H-bridge converter and its interface to the DSP (base drive circuit)

To write programs for the DSP to control the speed of the DC motor using the H-bridge converter and its base drive circuit (open loop control of DC-motor)

Discussion

The DC motor used in the experiments is a Pittman PM DC Motor with HP incremental encoder. The PM DC motor can be operated from 1.5V to 24V terminal voltage and is reversible.

The motor is equipped with an incremental encoder from HP, HEDS9100. The encoder produces a single channel output of 2950 pulses per revolution of the shaft. The encoder is powered from a +5V DC voltage and has output voltage compatible with the TTL logic devices.

Procedure

I. Setup

1. Make sure that the EVM system has been properly setup as in the previous lab.
2. Make sure that the connections between the base drive and power converter are properly installed. Do not make any connections from the base drive circuit to the EVM yet!
3. Connect the motor terminal to the X and Y outputs of the power converter.
4. Turn on the EVM power supply and the power for all the boards.
5. Run Code Composer.

II. Reset conditions of the DSP ports

1. Using a multimeter measure the output voltages at the following DSP ports: IOPB2, IOPB3, T2PWM and IOPB5.
2. The states of the output ports you measured in 1 represent the reset condition of the ports. Using the information obtained, can you explained the reason why the base drive circuit is designed to have an inverting logic (0 = Transistor ON, 1 = Transistor OFF)

Laboratory Assignments

I. Switching scheme 1

1. Connect the base drive and power converter circuit to the DSP as shown in Figure 7.12
2. Power On the Base drive & Power converter circuit.
3. Adjust the voltage V_{ref} of the lock out circuit in the base drive to zero. Briefly explain why for switching scheme 1, delaying the turn ON process is not necessary!
4. Write a program for the DSP that drives the motor with a PWM signal with the following duty cycles at frequency of 1 kHz
   a) 25 %
   b) 50%
   c) 75%
5. Download and run your programs in the EVM and observe the following using the oscilloscope :
   a) voltage at the motor terminal (V_{xy})
   b) output waveforms of the encoder.

For each case determine the speed of the motor from the period of the encoder output signal. Answer the following question:

Does the speed change proportionally with the duty cycle?

6. Repeat step 4 and 5 for the following PWM signal frequencies:
   (a) 2.0 kHz
   (b) 4.0 kHz

(c) 8.0 kHz

(d) 16.0 kHz

7. Rewrite your program to drive the motor in the other direction and experiment with different duty cycles and frequencies

II. Switching scheme 2

1. Connect the base drive and power converter circuit to the DSP as shown in Figure 7.13
2. Calculate the voltage V_{ref} required to obtain a 10 microseconds turn ON delay for the lockout circuit. Adjust V_{ref} in the base drive circuit to this calculated value.
3. Write a program for the DSP that drives the motor with a PWM signal with the following duty cycles at frequency of 1.0 kHz.
   a) 25 %
   b) 50%
   c) 75%
5. Download and run your programs in the EVM and observe the following using the oscilloscope:
   a) voltage at the motor terminal (V_{xy})
   b) output waveforms of the encoder.

For each case determine the speed of the motor from the period of the encoder output signal. Answer the following questions:

Does the speed change proportionally with the duty cycle?

6. Repeat step 4 and 5 for the following PWM signal frequencies:
   (a) 2.kHz
   (b) 4 kHz
   (c) 8 kHz
7. Rewrite your program to drive the motor in the other direction and experiment with different duty cycles and frequencies