Application 5: Using the PRIMER to Regulate the Speed of a DC Motor

Purpose:

To introduce the student to one method of regulating the speed of a small DC motor.

Goals:

1. Study formulas, data, and waveforms relating to a DC motor.

2. Build an interface circuit that will allow the PRIMER to regulate the speed of a particular DC motor.

3. Build a motor holding fixture that will allow one motor to be mechanically coupled to another.

4. Load, run, and test a program that will allow the PRIMER via the interface circuit to:

   A. Regulate the speed of a particular DC motor.
   B. Accept desired speed input via the on-board DIP switches.
   C. Display motor speed and pulse width via the on-board 7-segment displays and LEDs respectively.

Equipment, Components, and Materials:

Equipment (required):

Qty.	Description	Source	Part Number
1	PRIMER	EMAC	E600-00
1	Solderless Breadboard	Radio Shack	276-175
1	PRIMER Interface Cable	EMAC	E600-15

Components and Materials:

Interface Circuit:

1	Transistor, 2N2222	Digi-Key	PN2222A-ND
1	Transistor, 2N2907	Digi-Key	PN2907A-ND
1	Resistor, 8.2K , ¼W, 5%, Carbon Film	Digi-Key	8.2KQ
1	Resistor, 1.8K , ¼W, 5%, Carbon Film	Digi-Key	1.8KQ
1	Resistor, 1K , ¼W, 5%, Carbon Film	Digi-Key	1.0KQ
1	Resistor, 390 , ¼W, 5%, Carbon Film	Digi-Key	390Q
1	Diode, 1N4005	Digi-Key	1N4005GI
1	Capacitor, 2200µF, 16V	Digi-Key	P1216

Motor Load Resistors:

1	Resistor, 1.0 , ½W, 5%, Carbon Film	Digi-Key	1.0H
1	Resistor, 3.3 , ½W, 5%, Carbon Film	Digi-Key	3.3H
1	Resistor, 8.2 , ½W, 5%, Carbon Film	Digi-Key	8.2H
1	Resistor, 33 , ½W, 5%, Carbon Film	Digi-Key	33H

Motor Holding Fixture: (optional)

Qty.	Description	Source	Part Number
1	Aluminum or Plexiglas Flat, 3.9" x 2.9" x 1/16-1/8"	-	-
2	Aluminum or Plexiglas Flat, 1.8" x 0.5" x 1/16-1/8"	-	-
8	Aluminum Spacers, Round Threaded, 4-40 x 0.75"	Digi-Key	J240
2	Perfboard, Glass epoxy, Pad per hole, 0.4" x 2.2"	-	-
2	Terminal Block, 2 position	Digi-Key	ED1631-ND
1	Tennis Racquet Grip Wrap (Motor Mounting Pads) (or equivalant)	SOFTGRIP	STG-X
12	Pan Head Screws, 4-40 x 1/4"	Digi-Key	H142
4	Pan Head Screws, 4-40 x 1/2"	Digi-Key	H146
16	Lock Washers, #4	Digi-Key	H236
2	Motor with Gear(1.5 to 4.5VDC, 65mA @ 4.5VDC, 3 pole, permanent anisotropic magnet, 1.5 oz.in. stall torque)	Radio Shack	273-237

General:

20"ea.	Wire, Stranded, 22 Ga., Red and Black	Radio Shack	278-1218
20"	Wire, Wire Wrap, 30 Ga.	Radio Shack	278-503

Introduction:

In this lab, we would like to program the PRIMER to regulate the speed of a DC motor. The PRIMER will adjust motor speed by varying the armature voltage applied to the motor. This will be accomplished by varying the amount of time a fixed voltage is applied to the armature within a fixed time frame. This technique is called pulse width modulation (PWM). The time when voltage is applied to the motor will be referred to as "motor on time" or pulse width (PW). The time remaining in the fixed time frame would be "motor off time." The PRIMER will read the speed of the motor by using the on-board analog to digital (A/D) converter to measure the voltage (back EMF) generated by the motor during motor off time. This voltage is directly proportional to motor speed. By comparing motor speed to the desired speed, input via the on-board DIP switches, the PRIMER can correctly adjust motor on time to keep motor speed constant. Before we get to the interface circuit and PRIMER program needed to regulate motor speed, it might be helpful to look at some basic information relative to DC motors in general and to the motor we will be regulating in particular.

Motor Formulas:

$$T = 7.04K\ Ia$$

$$Vg = K\ N$$

$$Ia = V - \frac{Vg}{Ra}$$

Where:
K = A constant for a particular motor.
= Field flux.
Ia = Armature Current.
Ra = Armature Resistance.
V = Armature Voltage.
Vg = Back or Counter EMF.
N = Motor Speed.

$$V - IaRa$$

N = --------   T = Motor Torque.
K

These formulas show that there is a linear relationship between applied armature voltage V and motor speed N for a given load. Since back EMF, Vg,is directly related to motor speed there is also a linear relationship between V and Vg. The formulas also show that:

1. Vg will always be less than V.
2. Ia, and therefore torque are greatest at low motor speed and both decrease as motor speed is increased.
3. When an increased load is applied to a motor it must supply more torque.
   This in turn means that Ia must increase. If Ia increases motor speed
   will decrease. The only way to return the motor to its original speed
   is to increase the armature voltage V.

The motor we will use in this lab is a permanent magnet type. Permanent magnets provide the field flux . Magnetic fields setup by current flowing in the armature windings cause the armature to rotate inside the magnetic fields set up by the permanent magnets. To maintain armature rotation, the direction of the armature magnetic fields must constantly change relative to the fixed direction of the magnetic fields of the permanent magnets. This function is provided by brushes riding on a commutator attached to the motor shaft that constantly changes the direction of current flow in the armature windings as the shaft rotates. In this mode of operation, we supply electrical energy to the motor in the form of armature current and the motor supplies mechanical energy in the form of shaft rotation. If we supply mechanical energy to the motor by rotating the shaft, the motor will supply electrical energy in the form of armature current. This armature current results from the armature windings cutting across the magnetic lines of force set up by the magnetic fields of the permanent magnets. This current as seen by an electrical load across the motor terminals would be alternating (AC) if not for the rectifying action of the commutator converting it to DC. In this mode of operation, the motor is acting as a generator and the resulting DC voltage measured across the motor terminals is called counter or back EMF. The amplitude of this voltage will depend on the electrical load attached to the motor terminals but for a given load, changes in this back EMF will be directly proportional to changes in the speed of the rotating armature.

Motor Waveforms:

If we use a pulse generator to apply pulse width modulation to the circuit of Figure 1 and observe the resulting A/D signal on an oscilloscope, we would see the waveforms of Figure 2. The three regions of interest in the waveforms are marked as A, B, and C. The period of the PWM signal is A + B + C. The motor on time is A and the motor off time is B + C. Region B in waveform B is a negative voltage generated by the collapsing magnetic field in the armature windings when armature current is cut off at the beginning of motor off time. If this voltage were not clamped by diode D1 to that coul about -0.7V, it would be a very large negative voltage d potentially damage the PRIMER A/D circuitry. Region C in Waveform B is the back EMF generated by the armature rotating in the magnetic field of the permanent magnets during motor off time. If the pulse width of the PWM signal is now increased we would see the waveforms of Figure 3. The motor speed will noticeably increase and the amplitude of the back EMF of Region C will be greater. Two things are of interest in observing the motor waveforms that will have a bearing on our motor controller program.

1. The back EMF voltage is not "straight line smooth" as we would like it to be, but rather is a varying signal riding on a DC level. The amplitude of the varying signal seems to increase with increasing motor speed (increased pulse width). We could filter this with our circuitry but it would be difficult since we would not want to filter the motor on time voltage. This would introduce an unwanted error in the back EMF. A better solution would be to digitally filter (average) the back EMF by totalling 16 back EMF samples and then dividing the total by 16.

2. The point in the PWM period where we will begin to sample the back EMF must be carefully chosen to avoid sampling the motor on time voltage or the negative voltage transition. A sample window must be set up that will start late enough to assure back EMF will be present during maximum PW, but not so late that the program can't finish executing the required amount of code before the start of the next PWM period.

Motor Speed vs. Pulse Width and the Motor as an Integrator:

If we applied increasing pulse widths to the circuit of Figure 1, allowed the motor to accelerate up to speed and recorded the back EMF for each pulse width for various motor loads and plotted the results we would get a graph similar to the one in Figure 4. You might be surprised to see that the relationship between applied pulse width and back EMF is not linear for many of the curves. The curves appear to go from logarithmic for an unloaded motor toward linear as motor load is increased. This seems to contradict the results we would predict if we use the motor formulas we looked at earlier. The reason for this is that we are asking the motor to integrate the PWM signal into an armature voltage. We would expect that:

This is a linear relationship but this relationship only holds up if the acceleration (charge) and deceleration (discharge) times in the motor (integrator) are close to equal. The acceleration time (charge time) will be much shorter than deceleration time at no motor load because we are driving the armature up to speed and then allowing the armature to decelerate at its own pace. Deceleration is strictly load dependent. If there is no load on the motor the deceleration time is long, (relative to acceleration time), the integrator discharge time is long, and the curve is logarithmic. As the motor load increases (decreasing RL), the acceleration (charge) and deceleration (discharge) times become more nearly equal, the motor begins to act more like a true integrator, the armature voltage to PW relationship becomes linear, and the graph becomes linear. To state the previous discussion another way, if the linear changes in PW were producing linear changes in armature voltage, the motor would be responding linearly. Look at the graph in Figure 5. Notice the motor speed response vs. pulse width increase is linear, independent of motor load. These plots were produced by integrating the PWM signal externally and applying the resulting voltage via a power op-amp to the motor. Now the motor is behaving as the formulas predict because it is not required to integrate the PWM signal. Since our program will allow the PRIMER to measure motor speed with the A/D converter and then adjust the pulse width to the value necessary to obtain the desired speed, you might imagine that nonlinearity in the motor speed curves is unimportant.

Nonlinearity can make it more difficult for our program to control motor speed. Consider the curve for an unloaded motor (motors uncoupled) in Figure 4. Notice that a pulse width change of only 1 count, say from 6 to 7, can cause a speed change of more than 10. This means it will be difficult if not impossible for our program to make fine adjustments in motor speed since it can only make incremental (not fractional) changes to pulse width. Now look at the curve in Figure 4 for a motor load of 8.1 ohms. Now incremental changes in pulse width result in incremental changes in motor speed and as a result much finer adjustment of motor speed will be possible. So even though our program will do a fair job controlling motor speed when the motor is operating on one of the non linear curves, it will do a much better job controlling speed when the motor is operating on a more linear curve.

Motor Interface Circuit Description and Assembly:

Capacitor C1 in Figure 6 provides energy during times of high armature current to prevent fluctuations of the 5V supply. Resistor R1 sets the base current of transistor Q1 when PWM is high. Transistor Q1 provides base current for transistor Q2 when PWM is high. Q2 base current is set by resistors R2 and R3. Resistor R2 prevents Q2 conduction as a result of Q1 leakage or low level transients. Q2 provides armature current for motor M1 when PWM is high. Diode D1 clamps the negative voltage spike generated by the collapsing magnetic field of the armature at Q2 turn off. Resistor R4 limits the current into the A/D converter during the negative voltage spike. Two advantages of using pulse width modulation applied directly to the motor to control motor voltage are:

1. Relatively simple interface circuitry.
2. There is much less power dissipation because the controlling devices are switches (on or off).

The circuit in Figure 6 consists of easily available, inexpensive components. The circuit can be constructed on a solderless breadboard and wired to the PRIMER and motor using the PRIMER Interface Cable. The PWM and A/D connections can be wire-wrapped from the PRIMER CN3 connector to wire-wrap posts or stiff wires pushed into the breadboard. The motor leads should be short lengths (10 in.max.) of 22 ga. wire soldered to the motor tabs (no polarity) and then tinned on the other end so they will push into the breadboard holes.

Motor Holding Fixture:

A convenient way of loading one motor is to have it drive another motor which can in turn feed generated current through various load resistors to increase the load on the driving motor. If the motor you are using has a gear attached to the shaft, two motors can be coupled as illustrated in the motor fixture drawing. If your motor does not have a gear on the shaft, you can try coupling two motors with a short length of plastic tubing that will slip onto and hold tightly to the motor shafts. With this scheme the motors will be mounted in-line instead of offset in the motor fixture. Other motor loading schemes can be used such as using the motor to drive a propeller or placing a friction load against the motor shaft (holding your finger against the shaft at different degrees of pressure will do). You can choose your own method for mounting, coupling, and loading the motors but remember to construct fixtures from non-ferrous material because of the permanent magnets in the motors.

Program Description:

Refer to flowcharts 1 and 2 for a discussion of the motor controller program. The program divides the PWM period into 64 time slices or t_slices. Each t_slice is 160µs long. The t_slices are numbered from 0-63. A variable called t_slice is incremented in an interrupt handler on every 7.5 interrupt. Continuous pulses 160µs apart from the timer chip initiate each 7.5 interrupt. This interrupt handler also manages the PWM output. If pulse width is less than time slice, PWM output (output port bit 0) is high, otherwise it's low. The scheduling of events is illustrated below:

```
New
Event       Minimum                      Maximum Sample
Period
              PW                            PW     Mark
Starts
        |      |                            |       |          |
|

---------------------------------------------------------------
        0      3                           50      52         63
0
                                  Time Slice
```

The time between time slice 0 and sample mark is used to display speed and pulse width. These are displayed on the 7-segment LED display and LEDs 7-1 respectively. Notice there are upper and lower limits for pulse width. The time between maximum PW and sample mark is reserved to allow the negative voltage spike to pass when PW is maximum. The time between sample mark and end of period is used to sample the back EMF, average 16 samples, and calculate a new pulse width based on the current speed and the desired speed (set with the PRIMER DIP switches). The program consists of two programs, a background program and a foreground program. The background program executes every time the microprocessor receives an interrupt pulse on the 7.5 interrupt pin. The timer chip is set by the initialization part of our program to provide a pulse to the 7.5 interrupt pin every 160µs. The background program has two functions.

1. To increment the time slice each time it executes. The only exception to this is when time slice reaches a maximum count of 63 at which time it is set back to zero.
2. To set the PWM signal (output port bit 0) high or low. If time slice is less than pulse width the output is high, otherwise it is low.

The foreground program monitors time slice and waits till it's 0. Then it displays motor speed on the leftmost four 7-segment LED digits and it displays pulse width in a bar graph fashion on LEDs 7-1 as follows:

Pulse Width   LEDs On

0-7	(0% - 11%)	1
8-15	(12% - 23%)	1, 2
16-23	(24% - 36%)	1, 2, 3
24-31	(37% - 48%)	1, 2, 3, 4
32-39	(49% - 61%)	1, 2, 3, 4, 5
40-47	(62% - 73%)	1, 2, 3, 4, 5, 6
48-50	(74% - 78%)	1, 2, 3, 4, 5, 6, 7

The foreground program then waits for time slice to equal sample mark.

Sample mark is set to accommodate the longest possible pulse width plus time for the negative voltage transition (after motor current cutoff) to expire. At sample mark the back EMF is sampled and added to a total of 16 such samples. If 16 samples have not yet been totaled the program repeats by going back and waiting for time slice to equal 0. When 16 samples have been totaled, the total is divided by 16 to produce an average speed (it is this average speed that will later be displayed on the 7-segment display after time slice 0). The average speed is then subtracted from the speed set on the PRIMER DIP switches to produce an error term. If the error is < -1, the pulse width is decremented. If the error is > 1, the pulse width is incremented. If the error is -1, 0, or 1, the pulse width is unchanged. The pulse width is then range checked. If the pulse width is less than minimum (3), it is set to minimum. If the pulse width is greater than maximum (50), it is set to maximum. Otherwise the pulse width is unchanged.

The entire process then repeats by going back and again waiting for time slice 0. To test the motor speed program wire the circuit of Figure 6 and connect the PRIMER and drive motor M1 to the circuit as previously described. Couple the second motor M2 if available to the drive motor M1. Motor M2 if used should be unloaded (no RL across its terminals). Set the PRIMER DIP switches for a speed of 20. Load the motor control program into the PRIMER and run the program. The motor will accelerate to speed and the PW and average speed will be displayed as previously described. Load the drive motor by placing an 8.2 , ½W resistor across the terminals of motor M2 or by hand friction. The motor speed will decrease at first, as indicated by the 7-segment LED display. Then the PW will increase, as indicated by the 7 LEDs, to bring the motor speed back to 20. Now remove the 8.2 load resistor from motor M2 or the friction source. The speed of the drive motor will increase suddenly and the PW will begin to decrease to bring the motor speed back to 20.

Use the curves of Figure 4 and load resistors for various speeds set in on the DIP switches to exercise the motor speed control program. Notice from the curves of Figure 4 that there are limits on the maximum speed attainable for various motor loads. If you try to request a motor speed greater than the motor can provide for a given load, the program will simply increase the pulse width to maximum to get the maximum speed possible. Note that the following program text can be cut out and assembled.

```
;---------------------------------------------------------------
; This program regulates the speed of a DC motor by....
; [1] Averaging 16 samples of back EMF during motor off time.
; [2] Generating an error term (DIP switch - average EMF).
; [3] Using the error term to adjust the pulse width.
; [4] Using the resulting pulse width to pulse width modulate
;     (PWM) the motor.
```

```
;
; WARNING:  Use a 9V supply with a current limit of 1000 mA or
;         more with this lab.  The standard 500mA supply will
;         be damaged if it is used with this lab.
;
MOS:          EQU    1000H           ;MOS SERVICES ADDRESS.
PWM_PORT:     EQU    11H             ;DIGITAL OUTPUT PORT.
DIP_SW:       EQU    12H             ;DIP SWITCH PORT.
SERV09:       EQU    09H             ;MOS SERVICE.ADCIN => L.
SERV13:       EQU    13H             ;MOS SERVICE.DE => 7-SEG DISPLAY.
PW_MIN:       EQU    03H             ;MINIMUM PW. T=160uS X PW_MIN
PW_MAX:       EQU    32H             ;MAXIMUM PW. T=160uS X PW_MAX
MAX_SLICE:    EQU    3FH             ;MAXIMUM NUMBER OF TIME SLICES.
                                     ;SETS PWM PERIOD.
                                     ;T=160uS X MAX_SLICE.
SMARK:        EQU    34H             ;TIME SLICE WHERE BACK EMF
                                     ;SAMPLE WILL BE TAKEN.
VEC7HLF:      EQU    0FFE9H          ;7.5 INTERRUPT VECTOR.
SCALELO:      EQU    35H             ;MODE/SCALER FOR TIMER,
SCALEHI:      EQU    11000000B       ;CONTINUOUS PULSES EVERY 160uS.
TIMERLO:      EQU    14H             ;TIMER PORT.
TIMERHI:      EQU    15H             ;TIMER PORT.
TIMCMD:       EQU    0CDH            ;TIMER CONTROL COMMAND.
CMDREG:       EQU    10H             ;TIMER CONTROL PORT.
INTMASK:      EQU    1AH             ;INTERRUPT MASK.

              ORG    0FF01H

              DI
              LXI    H,SLICER        ;POINT 7.5 INTERRUPT
              SHLD   VEC7HLF         ;VECTOR TO SLICER.
              MVI    A,SCALELO       ;SET UP TIMER FOR
              OUT    TIMERLO         ;CONTINUOUS PULSES
              MVI    A,SCALEHI       ;AT DESIRED INTERRUPT
              OUT    TIMERHI         ;RATE.
              MVI    A,TIMCMD
              OUT    CMDREG
              MVI    A,INTMASK
              SIM
              EI

PWM_MOTOR:
              LXI    H,0000H         ;REG H = TOTAL
              MVI    B,10H           ;REG B = SAMPLE COUNT.

CHKZERO:
              LDA    T_SLICE         ;TIME SLICE = 0 ?
              CPI    00H
              JNZ    CHKZERO         ;NO.GO CHECK SMARK.
              MVI    D,00H           ;DISPLAY SPEED.
              MOV    E,C             ;C = SPEED.
              PUSH   B
              MVI    C,SERV13
              CALL   MOS
```

```
             POP    B
             LDA    PULSE_WIDTH
             MOV    D,A             ;DISPLAY PW.
             MVI    E,0FFH          ;E = MASK.
             ORA    E               ;CLEAR CARRY.
ROT_MASK:
             RAL                    ;ROTATE 0 TO MASK.
             MOV    E,A             ;SAVE MASK.
             MOV    A,D             ;GET PW.
             SUI    08H             ;PW = PW - 8.
             MOV    D,A             ;SAVE RESULT TO D.
             MOV    A,E             ;GET MASK.
             JNC    ROT_MASK        ;PW STILL POS. ?
             DI                     ;DISABLE INTERRUPT.
             LDA    IMAGE           ;GET IMAGE.
             RAR                    ;SAVE BIT 0.
             MOV    A,E             ;GET MASK.
             RAL                    ;7 BITS MASK + BIT 0.
             STA    IMAGE           ;TO IMAGE.
             EI                     ;ENABLE INTERRUPT.
CHK_SMARK:
             LDA    T_SLICE
             CPI    SMARK           ;TIME SLICE = SMARK ?
             JNZ    CHK_SMARK       ;NO.WAIT TILL IT IS.
             XCHG                   ;DE = TOTAL.
             PUSH   B               ;SAMPLE BACK EMF.
             MVI    C,SERV09
             CALL   MOS
             POP    B
             MVI    H,00H           ;HL = SAMPLE.
             DAD    D               ;HL = TOTAL + SAMPLE.
             DCR    B               ;DEC. SAMPLE COUNT.
             JNZ    CHKZERO         ;IF NOT 0, CHK 0 T_SLICE.

DIV_MORE:    DAD    H               ;HL*16/256=HL/16, SO...
             DAD    H               ;...4 DAD H's MAKES HL*16...
             DAD    H               ;..AFTER THIS H=HL/256 (THINK ABOUT IT)
             DAD    H               ;SPEED=TOTAL / MAX SAMP (16).
             MOV    C,H             ;STORE SPEED.
             IN     DIP_SW          ;GET DESIRED SPEED.
             ANI    00111111B       ;DES.SPEED 6 BITS MAX.
             SUB    H               ;SWITCH-SPEED=ERROR.
             LXI    H,PULSE_WIDTH
             JM     DECPW_CHK       ;ERROR = -. DEC PW ?
             CPI    2               ;ERROR < 2 ?
             JC     PW_RANGE        ;YES. NO PW CHANGE.
             INR    M               ;NO. INC PW.
             JMP    PW_RANGE        ;RANGE CHECK PW.
DECPW_CHK:
             CPI    0FFH            ;ERROR = -1.
             JZ     PW_RANGE        ;YES. RANGE CHECK PW.
             DCR    M               ;NO. DEC PW.
PW_RANGE:
             MVI    A,PW_MIN        ;PW < MIN ?
```

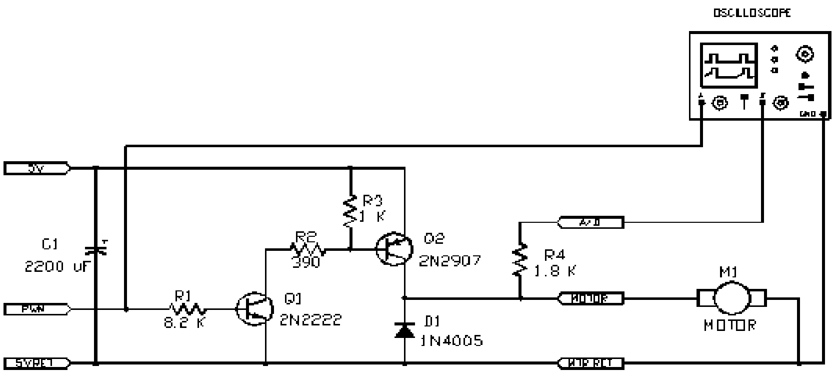
```
             CMP    M
             JC     MAX_CHK        ;NO. CHECK MAX.
             MOV    M,A            ;YES. PW = MIN.
MAX_CHK:
             MVI    A,PW_MAX       ;PW > MAX ?
             CMP    M
             JNC    PWM_MOTOR      ;NO. PW OK.
             MOV    M,A            ;YES. PW = MAX.
             JMP    PWM_MOTOR      ;START AGAIN.

;---------------------------------------------------------------
;.......................SLICER..................................
;SLICER IS AN INTERRUPT HANDLER FOR THE 7.5 INTERRUPT.
;SLICER CONTROLS A TIME MARKER (T_SLICE) BY ADJUSTING IT FROM
;0 TO MAX_SLICE IN EQUAL TIME INCREMENTS ON EACH 7.5 INTERRRUPT.
;SLICER ALSO CONTROLS THE WIDTH OF THE PULSE USED TO DRIVE THE
;MOTOR BY COMPARING THE VALUE OF PULSE_WIDTH TO THAT OF T_SLICE
;TO DETERMINE IF THE PULSE SHOULD BE HIGH OR LOW.
;PULSE HIGH => T_SLICE < PULSE_WIDTH.
;PULSE LOW  => T_SLICE >=PULSE_WIDTH.
;---------------------------------------------------------------

SLICER:
             PUSH   PSW            ;SAVE REGISTERS.
             PUSH   H
             LXI    H,T_SLICE      ;H POINTS TO T_SLICE.
             INR    M              ;INCREMENT T_SLICE
             MVI    A,MAX_SLICE
             CMP    M              ;T_SLICE = MAX_SLICE ?
             JNZ    PWM            ;NO. T_SLICE OK.
             MVI    M,00H          ;YES. T_SLICE = 0.
PWM:
             MOV    A,M            ;A = T_SLICE.
             LXI    H,PULSE_WIDTH  ;M = PULSE WIDTH.
             CMP    M              ;T_SLICE < PULSE WIDTH ?
             LXI    H,IMAGE        ;M = IMAGE.
             MOV    A,M            ;GET IMAGE.
             RAR                   ;CY => BIT 7.
             RLC                   ;BIT 7 => BIT 0.
             MOV    M,A            ;STORE IMAGE.
             OUT    PWM_PORT       ;OUTPUT IMAGE.
             POP    H              ;RECOVER REGISTERS.
             POP    PSW
             EI
             RET                   ;RETURN

T_SLICE:     DB     00H
PULSE_WIDTH: DB     PW_MIN
IMAGE:       DS     01H
             END
;----------------------------------------------------------

OBJECT/MACHINE CODE

ADDRESS  DATA   INSTRUCTION
FF01     F3     DI
FF02     21     LXI   H, FF92
FF03     92
FF04     FF
FF05     22     SHLD  FFE9
```

```
FF06     E9
FF07     FF
FF08     3E      MVI    A, 35
FF09     35
FF0A     D3      OUT    14
FF0B     14
FF0C     3E      MVI    A, C0
FF0D     C0
FF0E     D3      OUT    15
FF0F     15
FF10     3E      MVI    A, CD
FF11     CD
FF12     D3      OUT    10
FF13     10
FF14     3E      MVI    A, 1A
FF15     1A
FF16     30      SIM
FF17     FB      EI
FF18     21      LXI    H, 0000
FF19     00
FF1A     00
FF1B     06      MVI    B, 10
FF1C     10
FF1D     3A      LDA    FFB2
FF1E     B2
FF1F     FF
FF20     FE      CPI    00
FF21     00
FF22     C2      JNZ    FF1D
FF23     1D
FF24     FF
FF25     16      MVI    D, 00
FF26     00
FF27     59      MOV    E,C
FF28     C5      PUSH   B
FF29     0E      MVI    C, 13
FF2A     13
FF2B     CD      CALL   1000
FF2C     00
FF2D     10
FF2E     C1      POP    B
FF2F     3A      LDA    FFB3
FF30     B3
FF31     FF
FF32     57      MOV    D,A
FF33     1E      MVI    E, FF
FF34     FF
FF35     B3      ORA    E
FF36     17      RAL
FF37     5F      MOV    E,A
FF38     7A      MOV    A,D
FF39     D6      SUI    08
FF3A     08
FF3B     57      MOV    D,A
FF3C     7B      MOV    A,E
FF3D     D2      JNC    FF36
FF3E     36
FF3F     FF
FF40     F3      DI
FF41     3A      LDA    FFB4
```

```
FF42     B4
FF43     FF
FF44     1F     RAR
FF45     7B     MOV   A,E
FF46     17     RAL
FF47     32     STA   FFB4
FF48     B4
FF49     FF
FF4A     FB     EI
FF4B     3A     LDA   FFB2
FF4C     B2
FF4D     FF
FF4E     FE     CPI   34
FF4F     34
FF50     C2     JNZ   FF4B
FF51     4B
FF52     FF
FF53     EB     XCHG
FF54     C5     PUSH  B
FF55     0E     MVI   C, 09
FF56     09
FF57     CD     CALL  1000
FF58     00
FF59     10
FF5A     C1     POP   B
FF5B     26     MVI   H, 00
FF5C     00
FF5D     19     DAD   D
FF5E     05     DCR   B
FF5F     C2     JNZ   FF1D
FF60     1D
FF61     FF
FF62     29     DAD   H
FF63     29     DAD   H
FF64     29     DAD   H
FF65     29     DAD   H
FF66     4C     MOV   C,H
FF67     DB     IN    12
FF68     12
FF69     E6     ANI   3F
FF6A     3F
FF6B     94     SUB   H
FF6C     21     LXI    H,FFB3
FF6D     B3
FF6E     FF
FF6F     FA     JM     FF7B
FF70     7B
FF71     FF
FF72     FE     CPI    02
FF73     02
FF74     DA     JC     FF81
FF75     81
FF76     FF
FF77     34     INR    M
FF78     C3     JMP    FF81
FF79     81
FF7A     FF
FF7B     FE     CPI    FF
FF7C     FF
FF7D     CA     JZ     FF81
```

```
FF7E    81
FF7F    FF
FF80    35     DCR    M
FF81    3E     MVI    A,03
FF82    03
FF83    BE     CMP    M
FF84    DA     JC     FF88
FF85    88
FF86    FF
FF87    77     MOV    M,A
FF88    3E     MVI    A,32
FF89    32
FF8A    BE     CMP    M
FF8B    D2     JNC    FF18
FF8C    18
FF8D    FF
FF8E    77     MOV    M,A
FF8F    C3     JMP    FF18
FF90    18
FF91    FF
FF92    F5     PUSH   PSW
FF93    E5     PUSH   H
FF94    21     LXI    H,FFB2
FF95    B2
FF96    FF
FF97    34     INR    M
FF98    3E     MVI    A,3F
FF99    3F
FF9A    BE     CMP    M
FF9B    C2     JNZ    FFA0
FF9C    A0
FF9D    FF
FF9E    36     MVI    M,00
FF9F    00
FFA0    7E     MOV    A,M
FFA1    21     LXI    H,FFB3
FFA2    B3
FFA3    FF
FFA4    BE     CMP    M
FFA5    21     LXI    H,FFB4
FFA6    B4
FFA7    FF
FFA8    7E     MOV    A,M
FFA9    1F     RAR
FFAA    07     RLC
FFAB    77     MOV    M,A
FFAC    D3     OUT    11
FFAD    11
FFAE    E1     POP    H
FFAF    F1     POP    PSW
FFB0    FB     EI
FFB1    C9     RET
FFB2    00     (time slice)
FFB3    03     (pulse width)
FFB4    xx     (output port, undefined leave blank)
```

Schematic 1



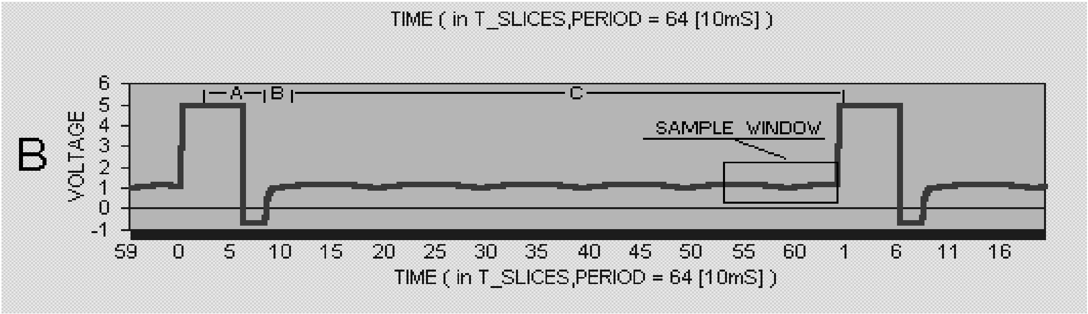
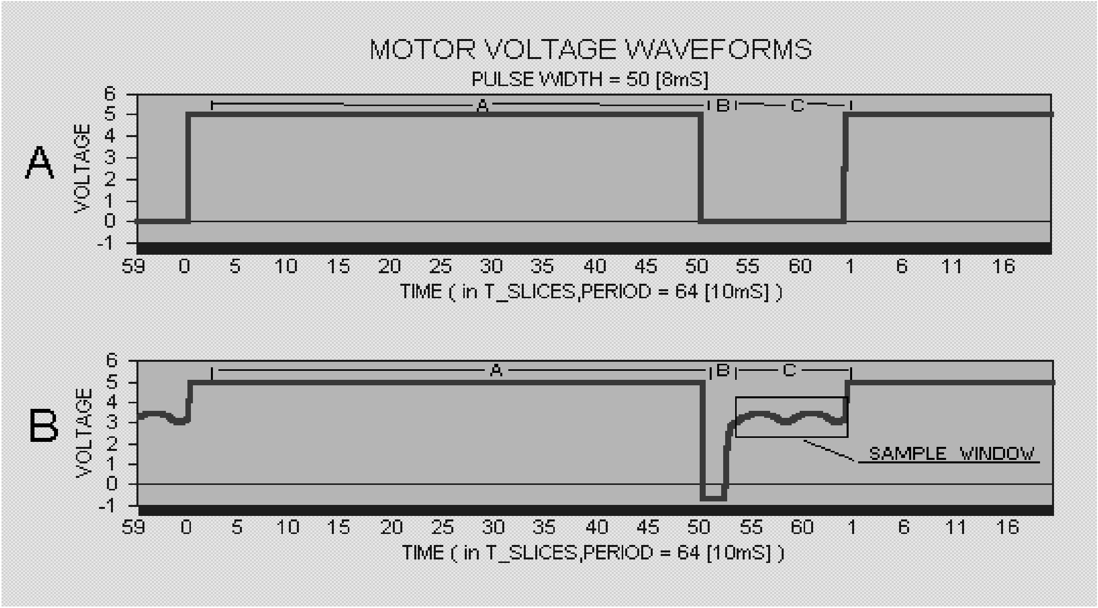
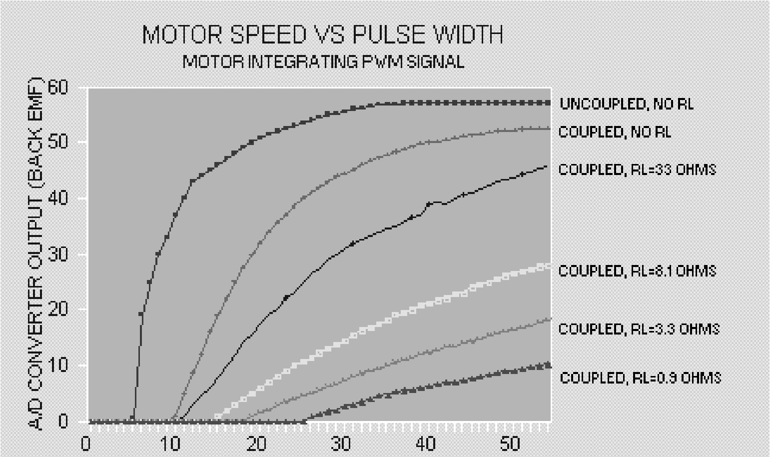


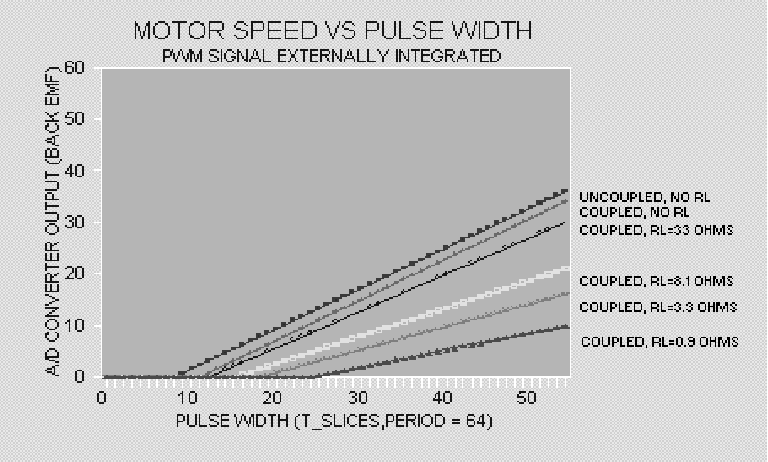
Figure 2



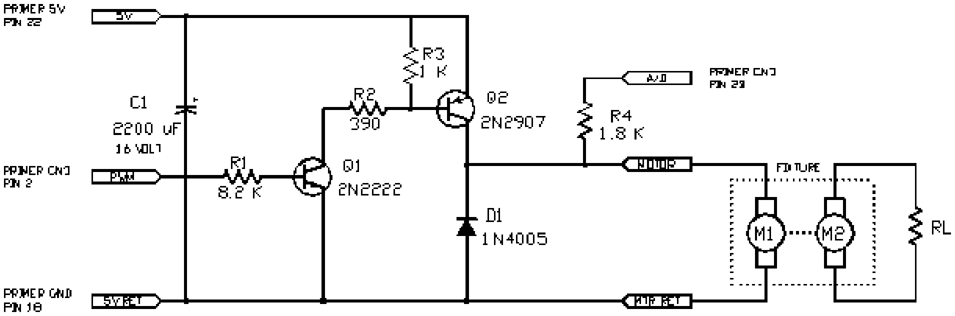
→

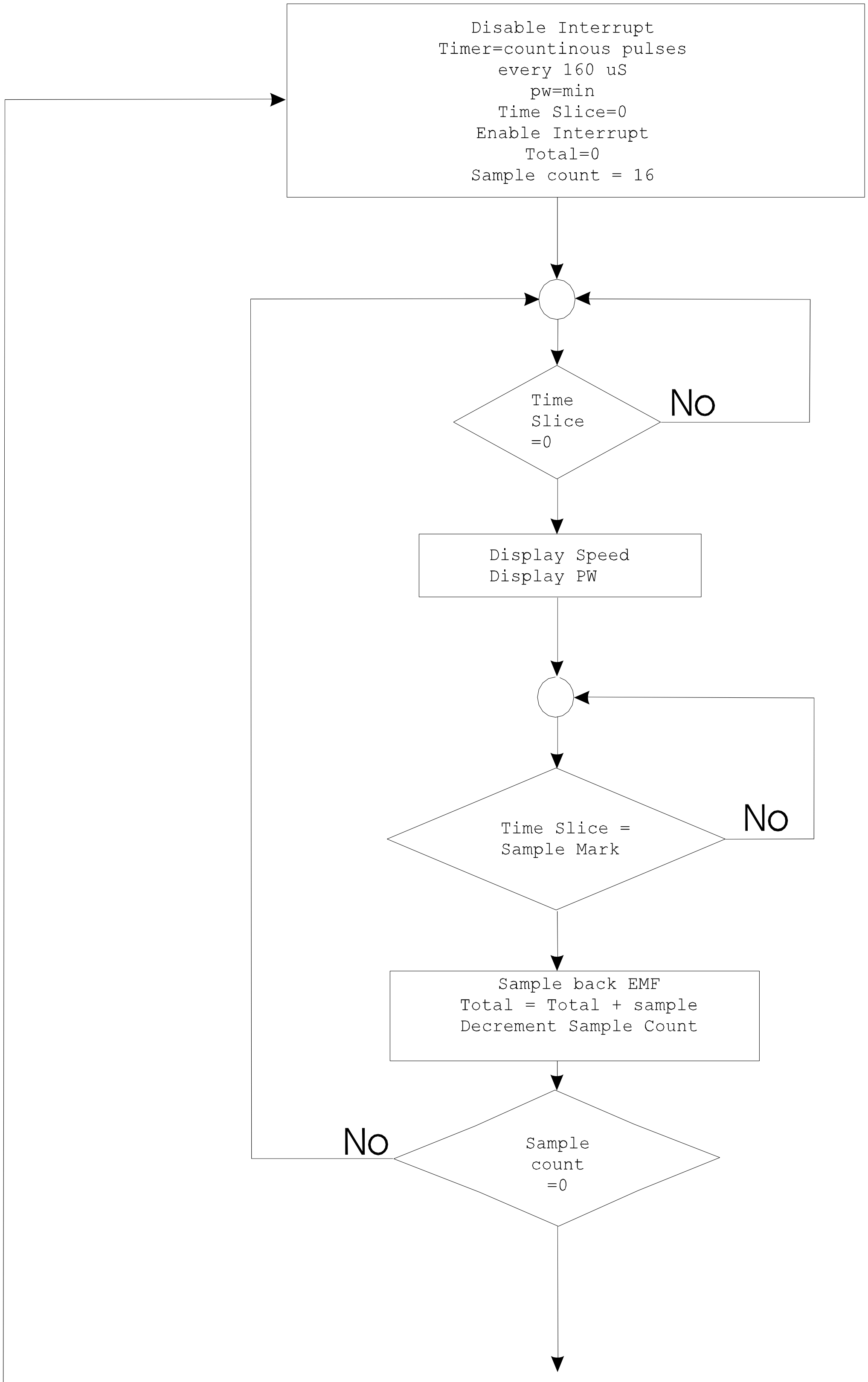
Figure 3


MOTOR SPEED VS PULSE WIDTH
MOTOR INTEGRATING PWM SIGNAL
A/D CONVERTER OUTPUT (BACK EMF)
0
10
20
30
40
50
60
0
10
20
30
40
50
UNCOUPLED, NO RL
COUPLED, NO RL
COUPLED, RL=33 OHMS
COUPLED, RL=8.1 OHMS
COUPLED, RL=3.3 OHMS
COUPLED, RL=0.9 OHMS


MOTOR SPEED VS PULSE WIDTH
PWM SIGNAL EXTERNALLY INTEGRATED
A/D CONVERTER OUTPUT (BACK EMF)
0
10
20
30
40
50
60
0
10
20
30
40
50
PULSE WIDTH (T_SLICES,PERIOD = 64)
UNCOUPLED, NO RL
COUPLED, NO RL
COUPLED, RL=33 OHMS
COUPLED, RL=8.1 OHMS
COUPLED, RL=3.3 OHMS
COUPLED, RL=0.9 OHMS

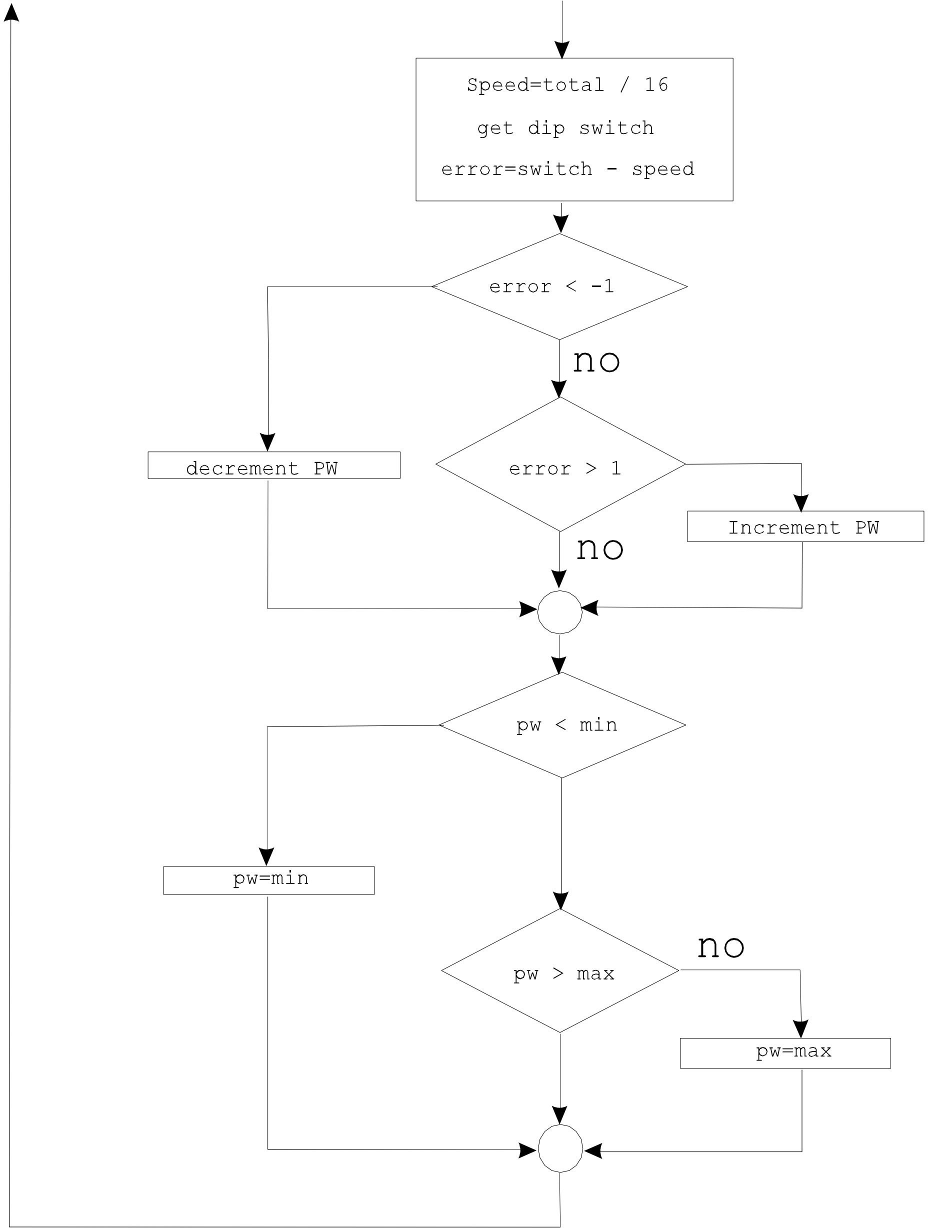
Schematic 2

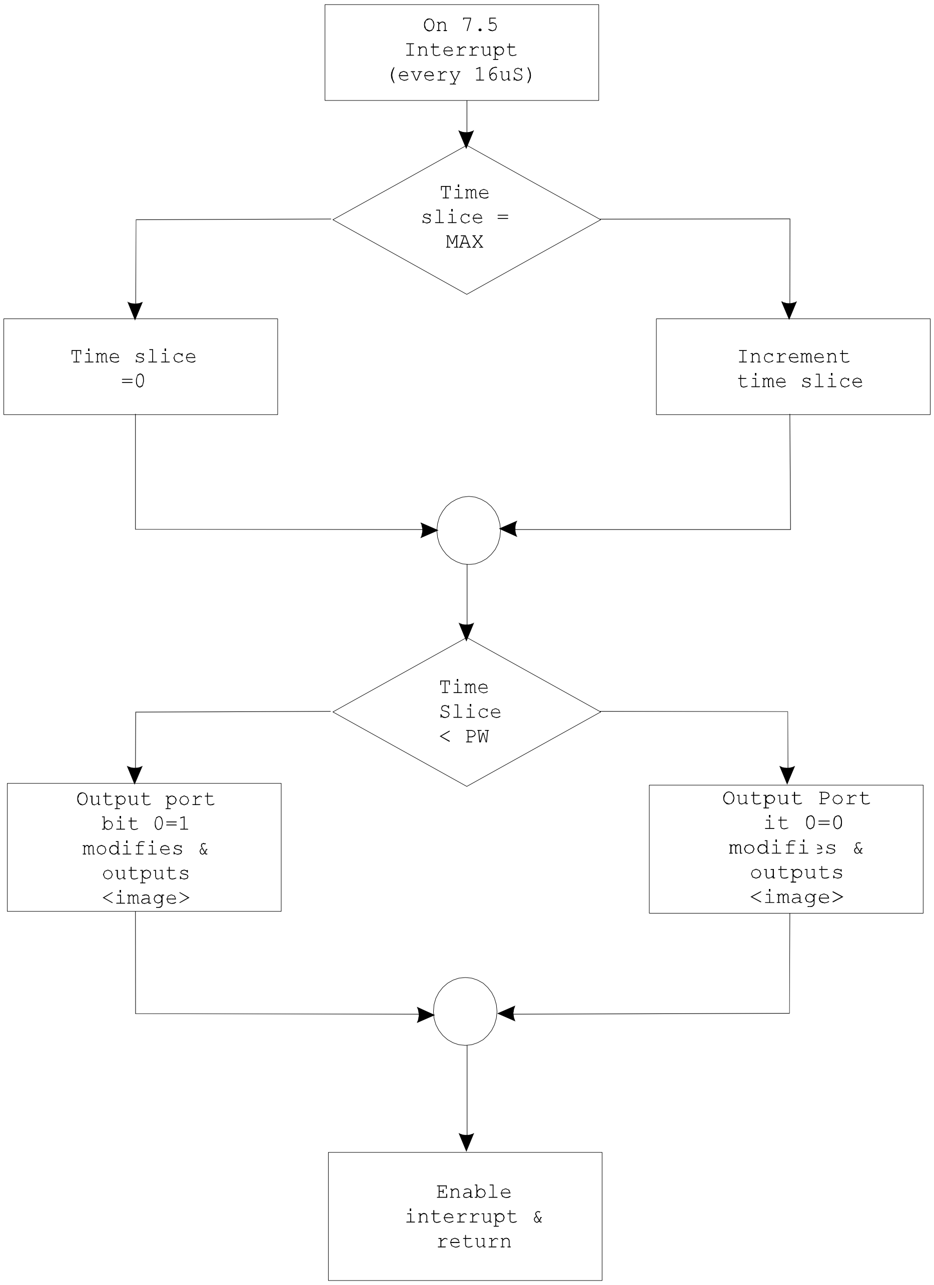

PRIMER 5V
PIN 22
5V
C1
2200 uF
16 VOLT
PRIMER CN3
PIN 2
PWM
R1
8.2 K
Q1
2N2222
R2
390
R3
1 K
Q2
2N2907
D1
1N4005
PRIMER GND
PIN 18
5V RET
A/D
PRIMER CN3
PIN 29
R4
1.8 K
MOTOR
MTR RET
FIXTURE
M1
M2
RL

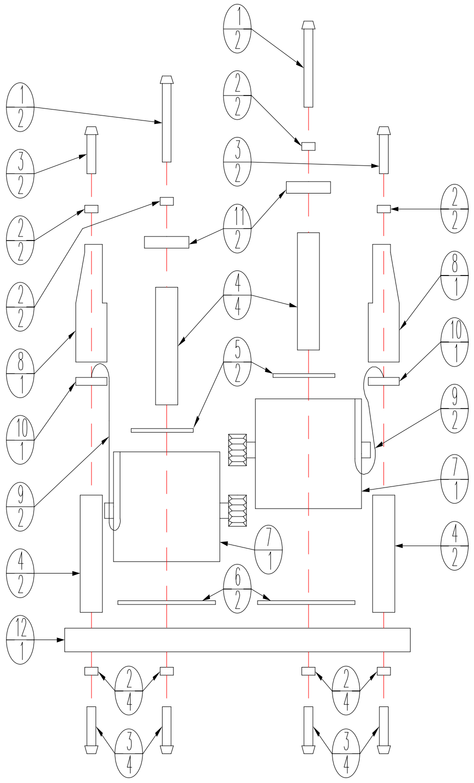


continued on next page

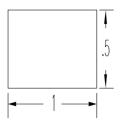
Continued from previous page



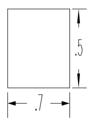

On 7.5 Interrupt (every 16uS)
Time slice = MAX
Time slice =0
Increment time slice
Time Slice < PW
Output port bit 0=1 modifies & outputs <image>
Output Port it 0=0 modifies & outputs <image>
Enable interrupt & return



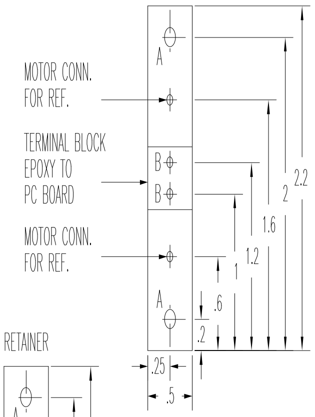
BOTTOM MOUNTING PAD



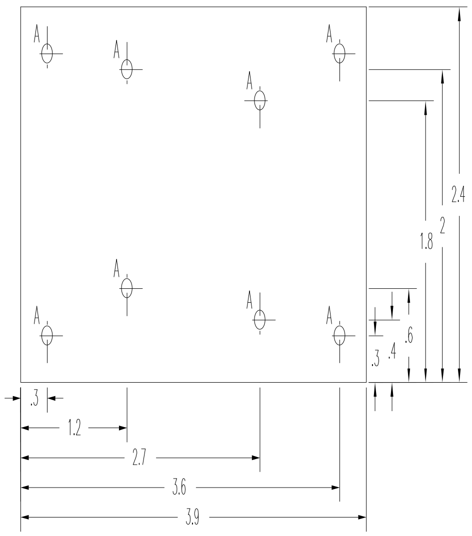
TOP MOUNTING PAD



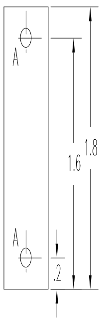
PC BOARD



BASE PLATE



RETAINER



ITEM	QTY	DESCRIPTION
1	4	SCREW, PAN HEAD 4-40 X 1/2"
2	16	LOCK WASHER #4
3	12	SCREW, PAN HEAD 4-40 X 1/4"
4	8	SPACER ROUND T 4-40 X 1/4" X
5	2	MOTOR MOUNTING PAD, TOP
6	2	MOTOR MOUNTING PAD, BOTTOM
7	2	MOTOR, DC, PM, 1.5V-4.5V, WIGEA
8	2	TERMINAL BLOCK, 2 POSITION
9	4	WIRE, MOTOR-PC 22 GA. STRANDED
10	2	PC BOARD, .4" X GLASS EPOXY
11	2	RETAINER, AL FLA 0.5" X 1.8" X 1/
12	1	BASE PLATE, AL F 2.9" X 3.9" X 1/

HOLE	QTY	DESCRIPTION
A	12	1/8" THRU
B	2	0.035" (#65) THR

TITLE
MOTOR MOUNTING

BY
DEREK JOHNSTON

REV. A

DWG. NO. FIG.