AN655

D/A Conversion Using PWM and R-2R Ladders to Generate Sine and DTMF Waveforms

Authors: Rob Stein and John Day
Microchip Technology Inc.

INTRODUCTION

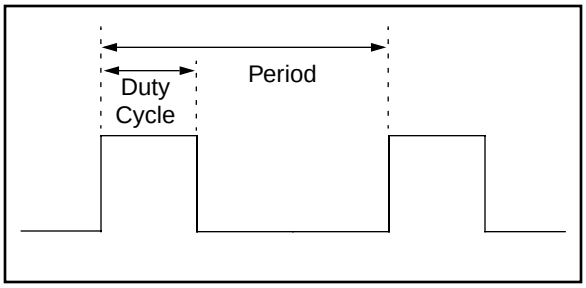
Many embedded applications require the generation of analog signals. Although separate D/A converter IC's exist on the market today, their extra price makes them prohibitive in cost sensitive embedded control designs. The following application note describes two DAC designs for generation of complex analog waveforms.: (PWM and R-2R Ladder). Both the Pulse Width Modulation (PWM) and resistor ladders require only a few external components to create a D/A converter with the PICmicro™ microcontroller series.

These techniques can be used to generate dual-tone multiplexed frequencies (DTMF) for telephone dialing, controlling the speed of a motor, generating sound and complex waveforms, and generating variable voltages in a power supply.

PULSE WIDTH MODULATION

Pulse Width Modulation (PWM) involves the generation of a series of pulses at a fixed period and frequency. The duty cycle defines the width of each pulse which is varied to generate waveforms. A simple low pass filter is then used to generate an output voltage directly proportional to the average time spent in the HIGH state (i.e., 50% duty cycle is equal to 2.5 volts when V_{DD} = 5.0V).

FIGURE 1: PWM WAVEFORM

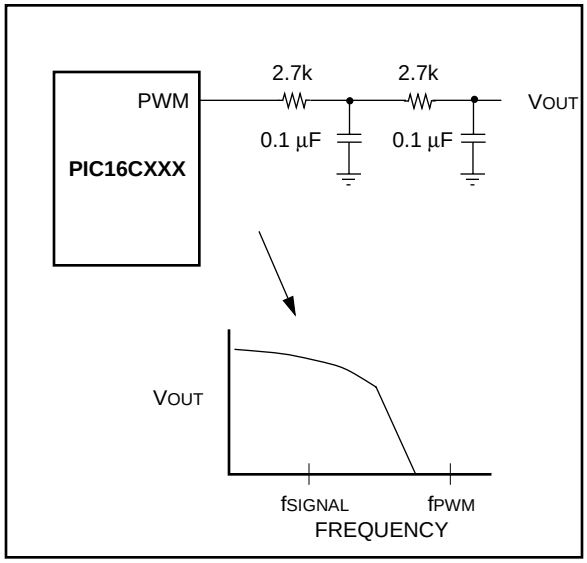


This D/A technique is used to produce slow moving analog outputs (from 0 to 100's of Hz). The software `SINE.ASM`, written for the PIC16C620, produces a 60 Hz sine function at a 5-bit resolution (32 steps). For each step that is generated the software allows ~ 10 PWM cycles for the low pass to settle. The number of PWM cycles for the low pass to settle varies from one application to another. The designer should experiment with this value to find the most optimal number for the application. These two criteria (sine frequency and # of steps) lead to the required PWM frequency:

PWM_Freq = (Sine_freq) • (# of steps) • 10
= 60 • 32 • 10
= 19.2 kHz

A simple RC low pass filter is the only external component needed to complete the circuit. A two pole filter is used in the example to pass the 60 Hz signal while eliminating the step (1920 Hz) and the PWM (19 khz) frequencies.

FIGURE 2: RC FILTER FREQUENCY RESPONSE



For this application note a software implementation was used to generate the PWM due to it's inherent lower cost. If greater resolution or frequency is required, a PICmicro family microcontroller with a hardware PWM should be used.

The TMR0 interrupt is used to generate precise PWM transitions. The time to service this interrupt will limit the minimum and maximum duty cycle for the PWM and hence the min/max voltage that can be produced. Therefore the interrupt service routine has been written to use as few cycles as possible.

A table lookup technique is used to generate one period of a sine function. Other waveforms, such as DC voltage ramps, triangles, sounds etc., can be produced

from this technique as well. The number of table entries is equal to the resolution required. The designer should carefully investigate the tradeoffs between PWM frequency, RC settling time, waveform resolution, and waveform frequency.

SINE.ASM SOURCE CODE

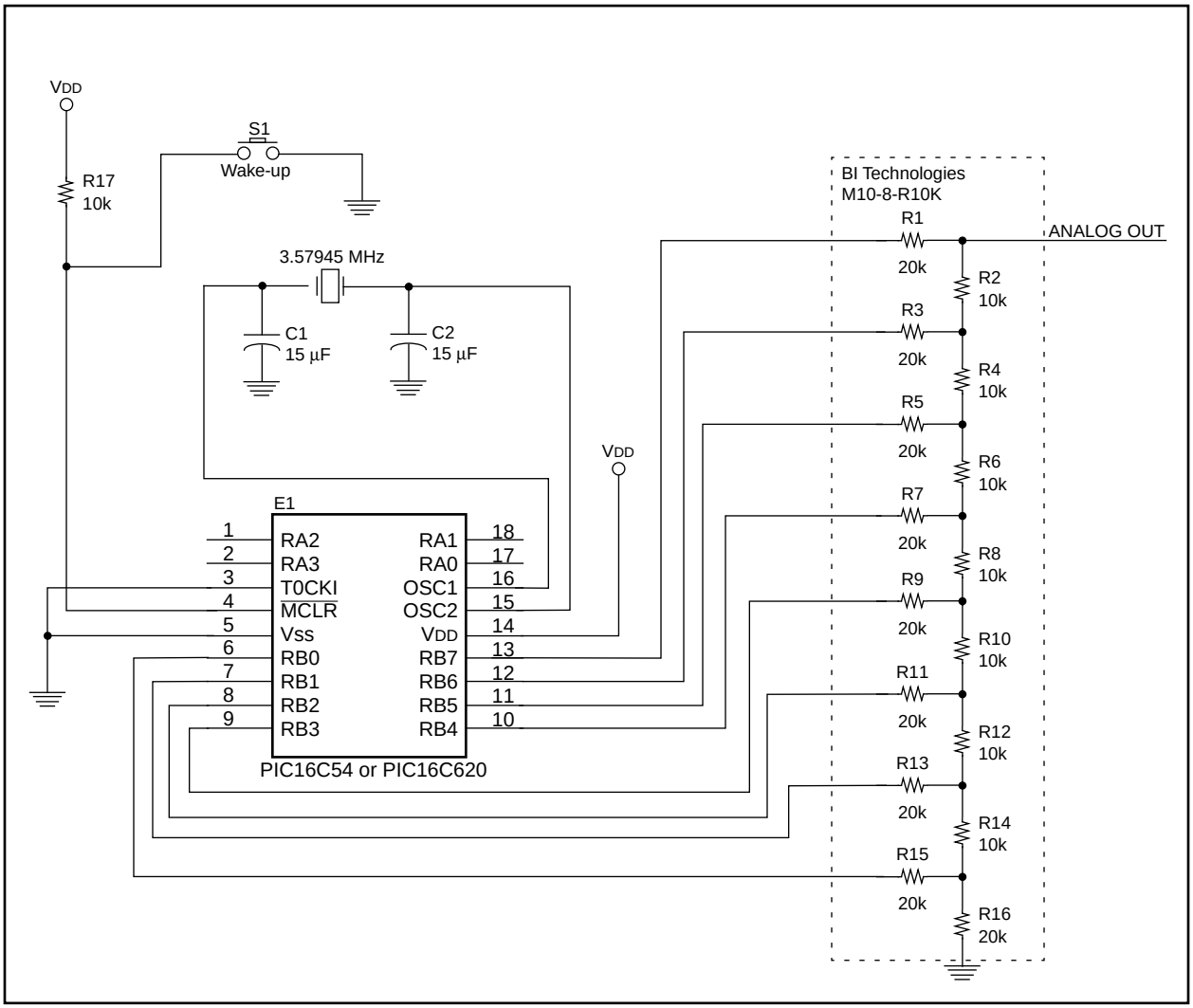
The `SINE.ASM` code example consists of four major code segments:

1. A main routine for counting and timing the 32 steps of the sine wave.
2. The `SetPWM` subroutine to pass a new PWM high and low counter value to the interrupt service routine at each step transition.
3. A table lookup to hold the 32 separate steps for the sine wave.
4. An interrupt service routine which creates the precise timing for the high and low PWM cycles.

R-2R LADDER

Many D/A converters are constructed with an internal resistor network driven by digital output drivers. The resistors are wired in a ladder fashion, with resistors of the same value being used for the rails, and resistors with 2x that value being used for the digital port outputs. Figure 3 shows the connection of a R2R ladder.

FIGURE 3: R2R LADDER SCHEMATIC



Implementation of a R2R ladder can be very inexpensive. BI Technologies sells a single SIP package for 8-bit and 10-bit R2R ladder networks. 8-bit R2R networks can also be manually constructed using 16 discrete resistors. Either way, R2R ladder networks have low EMI emissions, and reduced high frequency harmonics, eliminating the need for a low pass filter in most designs.

Microchip's microcontrollers have I/O pins with very low drive impedance (<75 Ohms at 5V) for both driving logic '1' and '0'. This allows direct connection to the R2R ladder without any external transistor drivers. To achieve 1 LSb accuracy using an 8-bit D/A, it is important to select resistors whose source impedance (in this case, 20k) is ≥ 256 times the driver source impedance.

To test the linearity of this design a +5V power supply was used. Each bit pattern was selected, then the output voltage was measured. The results show the transfer function in Figure 4.

SINE WAVE FUNCTIONS

The simplest way to generate sine waves is through a lookup table. One simply needs to determine the maximum allowable distortion along with the maximum operating frequency. Using these two criteria, the number of sample points needed can be determined:

Sample points = 100 / (% allowable voltage step inaccuracy)

Sample time = (Sample Points) • (Frequency)

Example:

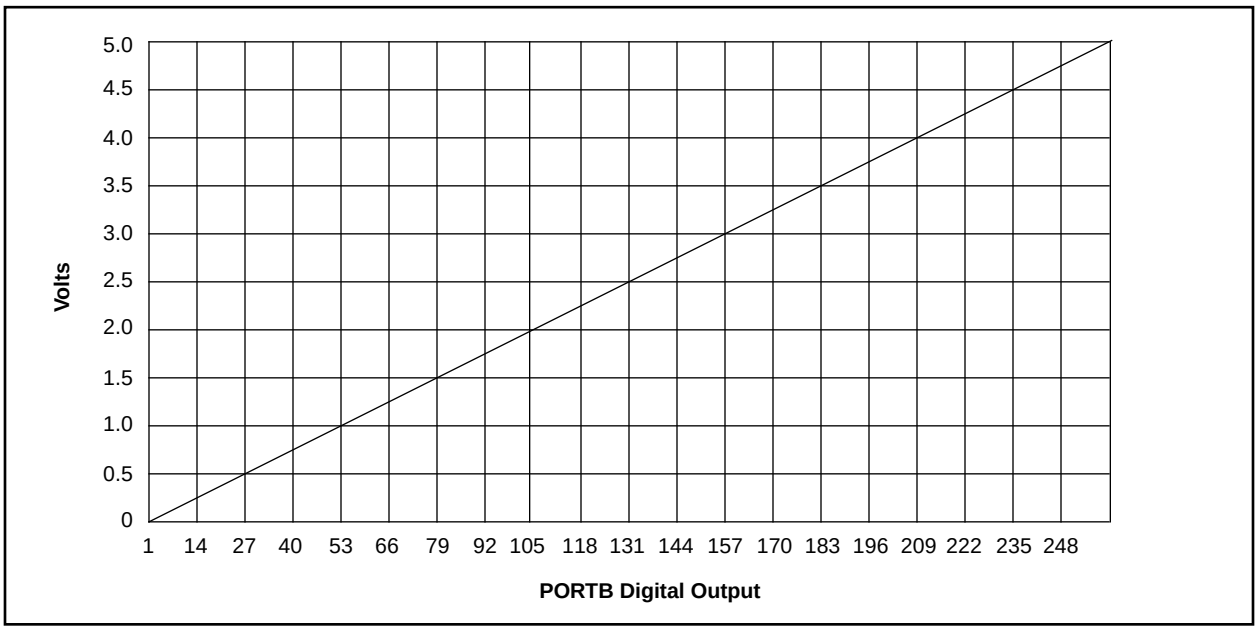
2 kHz Sine wave, <8% voltage step inaccuracy

Sample points = 100/5 = 20 points

Sample time = 20 • 2,000 = 40 kHz = 50 µs/Sample

To generate a single sine wave, the user simply needs to create a lookup table with the correct number of sample points. Next the software steps through each sample point and moves this value to the PORT output register, sending the correct voltage out of the R2R ladder. It is critical to ensure that the every sample point value be moved to the port output register using the same timing interval.

FIGURE 4: R2R LADDER TRANSFER FUNCTION



DTMF GENERATION

Many telecom applications, such as auto dialers, telephone keypads and security systems, require DTMF transmission for dialing and data transmission. Using the R2R ladder and seven sine lookup tables, an entire 12-key keypad of DTMF patterns can be generated with a 3.579545 MHz color burst crystal.

The DTMF standard was originally developed by Bell Labs for use by AT&T in telephone systems. There are several specifications that have fallen out of the original standard which come from AT&T, CEPT, CCITT, etc,. The variations from one standard to another are typically deviations in frequency tolerance, power, power difference between two tones, and speech immunity. The CCITT standard is located in Recommendations Q.23 and Q.24 in Section 4.3 of the CCITT Red Book, Volume VI, Fascicle VI.1. The other standards are listed in the References section at the end of this chapter.

A telephone keypad is broken up into 4 rows and 3 columns for a total of 12 keys. Each row and column is represented by a frequency; therefore, each key is uniquely represented by the sum of a row and column as follows:

TABLE 1: 12-KEY KEYPAD FREQUENCY

	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

As an example, the ‘1’ key is represented by a tone of 697 and 1209 Hz simultaneously.

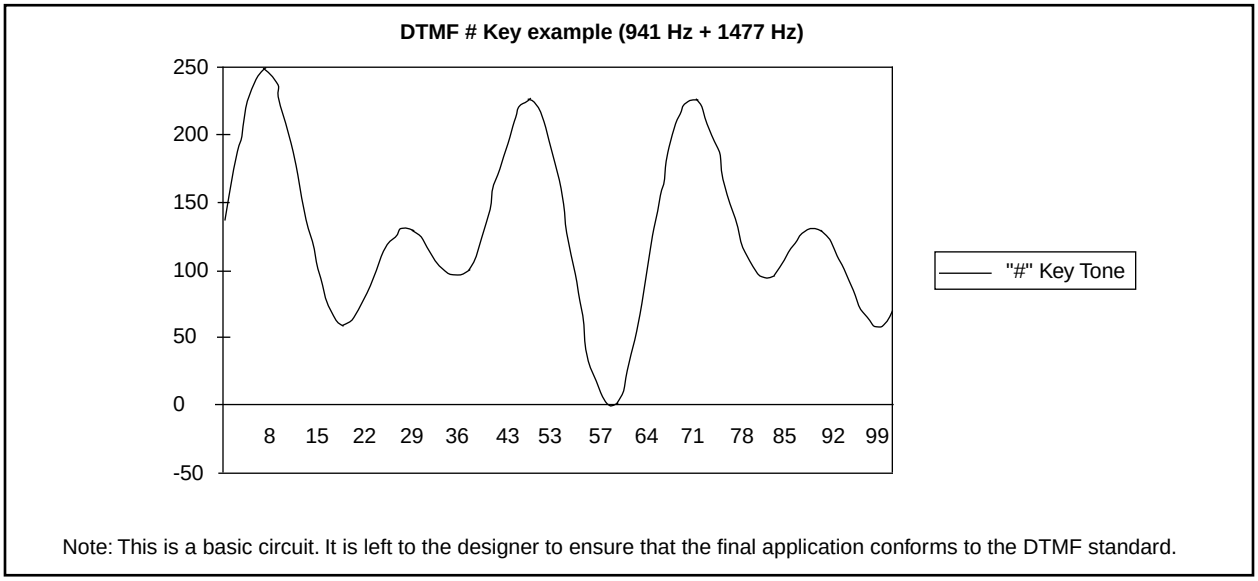
DTMF.ASM SOURCE CODE EXAMPLE

The source code `DTMF.ASM` shows how to generate the complete DTMF table for all keys. The complete DTMF generation algorithm requires only 285 words of code space, including all 7 sine lookup tables. The first 220 words of code space is used for the 7 sine lookup tables. Each sine wave lookup table ends with a value of 127 decimal to show the end of the sine wave pattern. The next section contains a look-up table for each key number which is used to look-up the sine wave pattern frequency number.

The `senddtmf` routine is used to beep the DTMF for 320 ms. Prior to calling the `senddtmf` routine, the key’s number is loaded into the W register. The sine wave addresses for that key are first calculated and the `WAVEABASE` and `WAVEBBASE` registers are loaded with the sine addresses. Next, each sine wave sample point is looked up and the sum of the two sine waves is moved to PORTB. Lastly, the pointer is reset to the sine base address if the value is equal to 127 decimal.

The test program loads each key into W and then calls `senddtmf`, showing all possible key patterns. The following figure shows a trace of the # key tones:

FIGURE 5: DTMF # KEY EXAMPLE



Note: This is a basic circuit. It is left to the designer to ensure that the final application conforms to the DTMF standard.

CONCLUSION

Using either PWM or R-2R ladder design techniques, many PICmicro microcontroller embedded designs will benefit from their low cost and component count. The following table compares the characteristics of the software and hardware PWM, R-2R ladder and external DAC design alternatives and their respective advantages and disadvantages.

TABLE 2: COST / PERFORMANCE TRADEOFF CHART

Cost / Performance Tradeoffs				
	S/W PWM	**H/W PWM**	**R2R Ladder**	**External DAC**
Cost	Low	Medium	Low	High
# of External Components	Low (1)	Low (1)	Low/Medium (2)	Low
I/Os Required	One	One	Eight (3)	Two/Nine (4)
Accuracy	Good (5)	Good (5)	Good (6)	Good/Excellent (7)
Resolution	Excellent (8)	Excellent (8)	Excellent (9)	Excellent (9)
Bandwidth	Low (10)	Low/Medium(10)	High	High
Harmonic Distortion	High	Medium	Low	Low

Note 1: One capacitor and one resistor for the low pass filter.
2: Multiply the required resolution times two for the number of external resistors or purchase one resistor SIP (16 resistors are required for 8-bit resolution).
3: The number of I/O's are directly proportional to resolution (8 I/O's for 8-bit resolution).
4: Separate DAC IC's are available with serial or parallel communication.
5: Absolute accuracy is dependent upon supply voltage.
6: Absolute accuracy is dependent upon supply voltage and resistor ladder tolerance.
7: Depends upon D/A reference source (supply voltage or a separate voltage reference).
8: Resolution is a dependent upon the PWM resolution.
9: Resolution is dependent upon the number of resistors in the ladder.
10: The waveform bandwidth is limited by the maximum PWM frequency and the settling time for the low pass filter. For an 8-bit resolution PWM the maximum frequency of operation is 19.5 kHz for a software PWM, and 80 kHz for a hardware PWM.

REFERENCE

Paul Horowitz and Winfield Hill:
"The Art of Electronics",
Cambridge University Press, New York 1989.

Contains a excellent practical description of analog filter design (as well as numerous other useful electronic subjects).

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).

APPENDIX A: DTMP.ASM

```
MPASM 01.40.01 Intermediate      DTMF.ASM   3-31-1997  11:00:08         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00001 ; Filename: DTMF.ASM
                     00002 ; ****************************************
                     00003 ; * Author:   John Day                    *
                     00004 ; *           Sr. Field Applications Engineer  *
                     00005 ; *           Microchip Technology        *
                     00006 ; * Revision: 1.1                         *
                     00007 ; * Date      December 20, 1995           *
                     00008 ; * Part:     PIC16C54                    *
                     00009 ; * Compiled using MPASM V1.40            *
                     00010 ; ****************************************
                     00011 ; * Include files:                        *
                     00012 ; *           NONE (used by DTMF.ASM)     *
                     00013 ; ****************************************
                     00014 ; * Fuses:    OSC:  XT (3.579545 Mhz xtal)     *
                     00015 ; *           WDT:  OFF                   *
                     00016 ; *            CP:  OFF                   *
                     00017 ; ********************************************************************************
                     00018 ; * This program uses and external R2R ladder network to generate complete *
                     00019 ; * DTMF dial tones used for telphone dialing.                           *
                     00020 ; ********************************************************************************
                     00021 ; * Program Memory:                                                     *
                     00022 ; *            220 Words - sine wave look-up table (7 sine waves total)    *
                     00023 ; *             25 Words - keypad sine address matrix look-up             *
                     00024 ; *             37 Words - DTMF sine wave base initialization/generation   *
                     00025 ; *              3 Words - Initialization                                 *
                     00026 ; *             25 Words - Test sample code                               *
                     00027 ; * RAM Memory:                                                          *
                     00028 ; *              8 Bytes                                                  *
                     00029 ; ********************************************************************************
                     00030         list     p=16C54, r=dec
                     00031         #include <p16c5x.inc>
                     00001         LIST
                     00002 ; P16C5X.INC  Standard Header File, Version 3.30 Microchip Technology, Inc.
                     00224         LIST
0FFF 0FF9            00032         __CONFIG _XT_OSC&_WDT_OFF&_CP_OFF
                     00033
  00000010           00034 WAVEABASE       EQU     10h     ; Base address of sine A waveform
  00000011           00035 POINTERA        EQU     11h     ; Pointer to current position in sine A
  00000012           00036 WAVEBBASE       EQU     12h     ; Base address of sine B waveform
  00000013           00037 POINTERB        EQU     13h     ; Pointer to current position in sine B
  00000014           00038 NEXTVALUE       EQU     14h     ; Sum register to store Sine A + Sine B
  00000015           00039 SINECOUNT       EQU     15h     ; LSB counter for time to output DTMF
  00000016           00040 SINECOUNTH      EQU     16h     ; MSB counter for time to output DTMF
  00000017           00041 TEMP            EQU     17h     ; Temporary storage
  0000007F           00042 ENDSINE         EQU     .127    ; Value to show the end of a sine table
                     00043 ; ********************************************************************************
                     00044 ; * sinelookup                                        *
                     00045 ; * This is the look-up table for the (4x3) keypad matrix sine wave table  *
                     00046 ; * There are (7) sine waves stored here and adding any two from the       *
                     00047 ; * matrix will product a DTMF signal for the appropriate key             *
                     00048 ; * Crystal Frequency:  3.579545 Mhz                                      *
                     00049 ; * Instructions/Loop:  35                                               *
                     00050 ; * Base Frequency:     1209 1336 1477  697  770  852  941 Hz            *
                     00051 ; * Actual Frequency:   1217 1345 1475  691  774  852  946 Hz            *
```

```
                     00052 ; * Error                0.7  0.7 -0.1 -0.9  0.5    0  0.5 %             *
                     00053 ; * Num Table Entries:    21   19   52   37   33   30   27                 *
                     00054 ; * Total Table Entries: 219                                               *
                     00055 ; * Program Memory:                                                        *
                     00056 ; *         220 Words - Used for (7) sine look-up entries                  *
                     00057 ; * RAM Memory:                                                            *
                     00058 ; *     NONE - Look-up table only                                         *
                     00059 ; ************************************************************************
0000                 00060 sinelookup                ; Used as address table to call look-up table
0000 01E2            00061         addwf     PCL,F   ; Add sine offset to PC to jump into table
0001                 00062 sineoffset                ; Used to calculate offset value address
0001                 00063 sinerow1                  ; Address for sine wave in row 1
0001 0895            00064         retlw     149
0002 08AA            00065         retlw     170
0003 08BE            00066         retlw     190
0004 08D0            00067         retlw     208
0005 08E0            00068         retlw     224
0006 08EC            00069         retlw     236
0007 08F6            00070         retlw     246
0008 08FD            00071         retlw     253
0009 08FF            00072         retlw     255
000A 08FE            00073         retlw     254
000B 08FA            00074         retlw     250
000C 08F2            00075         retlw     242
000D 08E6            00076         retlw     230
000E 08D8            00077         retlw     216
000F 08C7            00078         retlw     199
0010 08B4            00079         retlw     180
0011 08A0            00080         retlw     160
0012 088A            00081         retlw     138
0013 0875            00082         retlw     117
0014 085F            00083         retlw      95
0015 084B            00084         retlw      75
0016 0838            00085         retlw      56
0017 0827            00086         retlw      39
0018 0819            00087         retlw      25
0019 080D            00088         retlw      13
001A 0805            00089         retlw       5
001B 0801            00090         retlw       1
001C 0800            00091         retlw       0
001D 0802            00092         retlw       2
001E 0809            00093         retlw       9
001F 0813            00094         retlw      19
0020 081F            00095         retlw      31
0021 082F            00096         retlw      47
0022 0841            00097         retlw      65
0023 0855            00098         retlw      85
0024 086A            00099         retlw     106
0025 087F            00100         retlw     127      ; End of this sine wave
0026                 00101 sinerow2                  ; Address for sine wave in row 2
0026 0898            00102         retlw     152
0027 08AF            00103         retlw     175
0028 08C5            00104         retlw     197
0029 08D8            00105         retlw     216
002A 08E8            00106         retlw     232
002B 08F4            00107         retlw     244
002C 08FC            00108         retlw     252
002D 08FF            00109         retlw     255
002E 08FE            00110         retlw     254
002F 08F8            00111         retlw     248
0030 08EE            00112         retlw     238
0031 08E0            00113         retlw     224
0032 08CF            00114         retlw     207
0033 08BA            00115         retlw     186
0034 08A4            00116         retlw     164
0035 088C            00117         retlw     140
```

```
0036 0873     00118         retlw      115
0037 085B     00119         retlw       91
0038 0845     00120         retlw       69
0039 0830     00121         retlw       48
003A 081F     00122         retlw       31
003B 0811     00123         retlw       17
003C 0807     00124         retlw        7
003D 0801     00125         retlw        1
003E 0800     00126         retlw        0
003F 0803     00127         retlw        3
0040 080B     00128         retlw       11
0041 0817     00129         retlw       23
0042 0827     00130         retlw       39
0043 083A     00131         retlw       58
0044 0850     00132         retlw       80
0045 0867     00133         retlw      103
0046 087F     00134         retlw      127      ; End of this sine wave
0047          00135 sinerow3                    ; Address for sine wave in row 3
0047 089A     00136         retlw      154
0048 08B4     00137         retlw      180
0049 08CB     00138         retlw      203
004A 08DF     00139         retlw      223
004B 08EE     00140         retlw      238
004C 08F9     00141         retlw      249
004D 08FF     00142         retlw      255
004E 08FF     00143         retlw      255
004F 08F9     00144         retlw      249
0050 08EE     00145         retlw      238
0051 08DF     00146         retlw      223
0052 08CB     00147         retlw      203
0053 08B4     00148         retlw      180
0054 089A     00149         retlw      154
0055 0880     00150         retlw      128
0056 0865     00151         retlw      101
0057 084B     00152         retlw       75
0058 0834     00153         retlw       52
0059 0820     00154         retlw       32
005A 0811     00155         retlw       17
005B 0806     00156         retlw        6
005C 0800     00157         retlw        0
005D 0800     00158         retlw        0
005E 0806     00159         retlw        6
005F 0811     00160         retlw       17
0060 0820     00161         retlw       32
0061 0834     00162         retlw       52
0062 084B     00163         retlw       75
0063 0865     00164         retlw      101
0064 087F     00165         retlw      127        ; End of this sine wave
0065          00166 sinerow4                      ; Address for sine wave in row 4
0065 089D     00167         retlw      157
0066 08B9     00168         retlw      185
0067 08D2     00169         retlw      210
0068 08E6     00170         retlw      230
0069 08F5     00171         retlw      245
006A 08FE     00172         retlw      254
006B 08FF     00173         retlw      255
006C 08FA     00174         retlw      250
006D 08EE     00175         retlw      238
006E 08DD     00176         retlw      221
006F 08C6     00177         retlw      198
0070 08AB     00178         retlw      171
0071 088E     00179         retlw      142
0072 0871     00180         retlw      113
0073 0854     00181         retlw       84
0074 0839     00182         retlw       57
0075 0822     00183         retlw       34
```

```
0076 0811     00184         retlw       17
0077 0805     00185         retlw        5
0078 0800     00186         retlw        0
0079 0801     00187         retlw        1
007A 080A     00188         retlw       10
007B 0819     00189         retlw       25
007C 082D     00190         retlw       45
007D 0846     00191         retlw       70
007E 0862     00192         retlw       98
007F 087F     00193         retlw      127        ; End of this sine wave
0080          00194 sinecolumna                   ; Address for sine wave in column A
0080 08A5     00195         retlw      165
0081 08C8     00196         retlw      200
0082 08E4     00197         retlw      228
0083 08F7     00198         retlw      247
0084 08FF     00199         retlw      255
0085 08FC     00200         retlw      252
0086 08EE     00201         retlw      238
0087 08D7     00202         retlw      215
0088 08B7     00203         retlw      183
0089 0893     00204         retlw      147
008A 086C     00205         retlw      108
008B 0848     00206         retlw       72
008C 0828     00207         retlw       40
008D 0811     00208         retlw       17
008E 0803     00209         retlw        3
008F 0800     00210         retlw        0
0090 0808     00211         retlw        8
0091 081B     00212         retlw       27
0092 0837     00213         retlw       55
0093 085A     00214         retlw       90
0094 087F     00215         retlw      127         ; End of this sine wave
0095          00216 sinecolumnb                    ; Address for sine wave in column B
0095 08A9     00217         retlw      169
0096 08CE     00218         retlw      206
0097 08EB     00219         retlw      235
0098 08FC     00220         retlw      252
0099 08FF     00221         retlw      255
009A 08F5     00222         retlw      245
009B 08DE     00223         retlw      222
009C 08BC     00224         retlw      188
009D 0895     00225         retlw      149
009E 086A     00226         retlw      106
009F 0843     00227         retlw       67
00A0 0821     00228         retlw       33
00A1 080A     00229         retlw       10
00A2 0800     00230         retlw        0
00A3 0803     00231         retlw        3
00A4 0814     00232         retlw       20
00A5 0831     00233         retlw       49
00A6 0856     00234         retlw       86
00A7 087F     00235         retlw      127          ; End of this sine wave
00A8          00236 sinecolumnc            ; Address for sine wave in column C (double sine wave)
00A8 08AD     00237         retlw      173
00A9 08D4     00238         retlw      212
00AA 08F1     00239         retlw      241
00AB 08FF     00240         retlw      255
00AC 08FC     00241         retlw      252
00AD 08E9     00242         retlw      233
00AE 08C8     00243         retlw      200
00AF 089E     00244         retlw      158
00B0 0870     00245         retlw      112
00B1 0844     00246         retlw       68
00B2 0820     00247         retlw       32
00B3 0808     00248         retlw        8
00B4 0800     00249         retlw        0
```

```
00B5 0808     00250         retlw        8
00B6 0820     00251         retlw       32
00B7 0844     00252         retlw       68
00B8 0870     00253         retlw      112
00B9 089E     00254         retlw      158
00BA 08C8     00255         retlw      200
00BB 08E9     00256         retlw      233
00BC 08FC     00257         retlw      252
00BD 08FF     00258         retlw      255
00BE 08F1     00259         retlw      241
00BF 08D4     00260         retlw      212
00C0 08AD     00261         retlw      173
00C1 0880     00262         retlw      128
00C2 0852     00263         retlw       82
00C3 082B     00264         retlw       43
00C4 080E     00265         retlw       14
00C5 0800     00266         retlw        0
00C6 0803     00267         retlw        3
00C7 0816     00268         retlw       22
00C8 0837     00269         retlw       55
00C9 0861     00270         retlw       97
00CA 088F     00271         retlw      143
00CB 08BB     00272         retlw      187
00CC 08DF     00273         retlw      223
00CD 08F7     00274         retlw      247
00CE 08FF     00275         retlw      255
00CF 08F7     00276         retlw      247
00D0 08DF     00277         retlw      223
00D1 08BB     00278         retlw      187
00D2 088F     00279         retlw      143
00D3 0861     00280         retlw       97
00D4 0837     00281         retlw       55
00D5 0816     00282         retlw       22
00D6 0803     00283         retlw        3
00D7 0800     00284         retlw        0
00D8 080E     00285         retlw       14
00D9 082B     00286         retlw       43
00DA 0852     00287         retlw       82
00DB 087F     00288         retlw      127         ; End of this sine wave
              00289 ; ************************************************************************
              00290 ; * sineaddress                                                          *
              00291 ; * This subroutine is used to calculate actual address in the sinelookup   *
              00292 ; * table for two DTMF waveforms.  It is only called by setdtmfbase. W      *
              00293 ; * is loaded with the key number and this routine returns the sine         *
              00294 ; * lookup address the sine waves.                                        *
              00295 ; * RAM used:   W                                                         *
              00296 ; *     PROGRAM MEM:    25 Words                                          *
              00297 ; ************************************************************************
00DC          00298 sineaddress                              ; Lookup table for sine address
00DC 01E2     00299         addwf     PCL,F                  ; Add to PC to jump into table
00DD          00300 keyoffset
00DD 0800     00301 k1      retlw     sinerow1-sineoffset    ; Offset for Row 1 sine wave
00DE 087F     00302         retlw     sinecolumna-sineoffset ; Offset for Column A sine wave
00DF 0800     00303 k2      retlw     sinerow1-sineoffset    ; Offset for Row 1 sine wave
00E0 0894     00304         retlw     sinecolumnb-sineoffset ; Offset for Column A sine wave
00E1 0800     00305 k3      retlw     sinerow1-sineoffset    ; Offset for Row 1 sine wave
00E2 08A7     00306         retlw     sinecolumnc-sineoffset ; Offset for Column A sine wave
00E3 0825     00307 k4      retlw     sinerow2-sineoffset    ; Offset for Row 2 sine wave
00E4 087F     00308         retlw     sinecolumna-sineoffset ; Offset for Column A sine wave
00E5 0825     00309 k5      retlw     sinerow2-sineoffset    ; Offset for Row 2 sine wave
00E6 0894     00310         retlw     sinecolumnb-sineoffset ; Offset for Column B sine wave
00E7 0825     00311 k6      retlw     sinerow2-sineoffset    ; Offset for Row 2 sine wave
00E8 08A7     00312         retlw     sinecolumnc-sineoffset ; Offset for Column B sine wave
00E9 0846     00313 k7      retlw     sinerow3-sineoffset    ; Offset for Row 3 sine wave
00EA 087F     00314         retlw     sinecolumna-sineoffset ; Offset for Column B sine wave
00EB 0846     00315 k8      retlw     sinerow3-sineoffset    ; Offset for Row 3 sine wave
```

```
00EC 0894     00316         retlw     sinecolumnb-sineoffset    ; Offset for Column B sine wave
00ED 0846     00317 k9      retlw     sinerow3-sineoffset       ; Offset for Row 3 sine wave
00EE 08A7     00318         retlw     sinecolumnc-sineoffset    ; Offset for Column C sine wave
00EF 0864     00319 k10     retlw     sinerow4-sineoffset       ; Offset for Row 4 sine wave
00F0 087F     00320         retlw     sinecolumna-sineoffset    ; Offset for Column C sine wave
00F1 0864     00321 k11     retlw     sinerow4-sineoffset       ; Offset for Row 4 sine wave
00F2 0894     00322         retlw     sinecolumnb-sineoffset    ; Offset for Column C sine wave
00F3 0864     00323 k12     retlw     sinerow4-sineoffset       ; Offset for Row 4 sine wave
00F4 08A7     00324         retlw     sinecolumnc-sineoffset    ; Offset for Column C sine wave
  00000000    00325 key1=k1-keyoffset       ; Calculation for sine addr for keypad 1
  00000002    00326 key2=k2-keyoffset       ; Calculation for sine addr for keypad 2
  00000004    00327 key3=k3-keyoffset       ; Calculation for sine addr for keypad 3
  00000006    00328 key4=k4-keyoffset       ; Calculation for sine addr for keypad 4
  00000008    00329 key5=k5-keyoffset       ; Calculation for sine addr for keypad 5
  0000000A    00330 key6=k6-keyoffset       ; Calculation for sine addr for keypad 6
  0000000C    00331 key7=k7-keyoffset       ; Calculation for sine addr for keypad 7
  0000000E    00332 key8=k8-keyoffset       ; Calculation for sine addr for keypad 8
  00000010    00333 key9=k9-keyoffset       ; Calculation for sine addr for keypad 9
  00000012    00334 keystar=k10-keyoffset   ; Calculation for sine addr for keypad *
  00000014    00335 key0=k11-keyoffset      ; Calculation for sine addr for keypad 0
  00000016    00336 keypound=k12-keyoffset  ; Calculation for sine addr for keypad #
              00337 ; ************************************************************************
              00338 ; * senddtmf                                                             *
              00339 ; * This subroutine is used to calculate the offset address for          *
              00340 ; * the two sine waves to be sent and initialize the WAVEABASE and WAVEBBASE*
              00341 ; * file registers.  The key number (key0 - key9 or keystar or keypound) is *
              00342 ; * loaded into W before this routine is called.  Next, the DTMF for that   *
              00343 ; * key is sent to the R2R ladder through portB for 320 mS                 *
              00344 ; * Example:                                                             *
              00345 ; * movlw   key1                                                        *
              00346 ; * call    senddtmf                                                    *
              00347 ; * RAM used:        8 bytes                                            *
              00348 ; * PROGRAM MEM:    37 Words                                            *
              00349 ; ************************************************************************
00F5          00350 senddtmf
00F5 0037     00351         movwf   TEMP            ; Initialize temp with key number
00F6 09DC     00352         call    sineaddress     ; Get the addr for sine wave a for this key
00F7 0030     00353         movwf   WAVEABASE       ; Initialize sine wave A base address
00F8 03B0     00354         swapf   WAVEABASE,F   ; swap WAVEABASE so that Z bit is not affected
00F9 0031     00355         movwf   POINTERA        ; Initialize sine wave A pointer address
00FA 0297     00356         incf    TEMP,W          ; Now we get the second sine wave address...
00FB 09DC     00357         call    sineaddress     ; Get the addr for sine wave b for this key
00FC 0032     00358         movwf   WAVEBBASE       ; Initialize sine wave B base address
00FD 03B2     00359         swapf   WAVEBBASE,F    ; swap WAVEBBASE so that Z bit is not affected
00FE 0033     00360         movwf   POINTERB        ; Initialize sine wave B pointer address
00FF 0C20     00361         movlw   20h             ; Place 32 decimal into W for loop counter
0100 0036     00362         movwf   SINECOUNTH      ; Initialize loop counter to 20
0101          00363 loopsine2cyc
0101 0B02     00364         goto    loopsine   ; Waste two cycles to maintain 35 cycle loop count
0102          00365 loopsine
0102 0211     00366         movf    POINTERA,W      ; Place sine wave address into W
0103 0900     00367         call    sinelookup      ; Lookup first sine wave
0104 0034     00368         movwf   NEXTVALUE       ; Place first sine wave into NEXTVALUE
0105 0F7F     00369         xorlw   ENDSINE         ; Update Z bit
0106 0390     00370         swapf   WAVEABASE,W     ; Restore to beginning of sine wave
0107 0643     00371         btfsc   STATUS,Z        ; Skip if not at end of sine wave
0108 0031     00372         movwf   POINTERA        ; Restore start address if at end of sine
0109 02B1     00373         incf    POINTERA,F      ; Move to the next place in the wave
010A 0213     00374         movf    POINTERB,W      ; Place sine wave address into W
010B 0900     00375         call    sinelookup      ; Lookup second sine wave
010C 01F4     00376         addwf   NEXTVALUE,F     ; Add second sine wave into NEXTVALUE
010D 0F7F     00377         xorlw   ENDSINE         ; Update Z bit
010E 0392     00378         swapf   WAVEBBASE,W     ; Pointer of sine wave beginning -> W
010F 0643     00379         btfsc   STATUS,Z        ; Skip if not at end of sine wave
0110 0033     00380         movwf   POINTERB        ; Restore start address if at end of sine
0111 02B3     00381         incf    POINTERB,F      ; Move to next place in the wave
```

```
0112 0314     00382         rrf     NEXTVALUE,W     ; Divide by 2, Place output into PORTB
0113 0026     00383         movwf   PORTB           ; Update PORTB with new R2R value
0114 0B15     00384         goto    waste2cyc       ; Waste (2) cycles for 35 total
0115          00385 waste2cyc
0115 02F5     00386         decfsz  SINECOUNT,F     ; Skip if we are done
0116 0B01     00387         goto    loopsine2cyc    ; Do it again! (add 2 cycles as well)
0117 02F6     00388         decfsz  SINECOUNTH,F    ; Skip if we are done
0118 0B02     00389         goto    loopsine        ; Do it again!
0119 0800     00390         retlw   0               ; Return from sine output
              00391 ; ************************************************************************
              00392 ; * init                                                                 *
              00393 ; * This code is used to initialize the PIC.  PORTB is set to zero and all  *
              00394 ; * pins are set to outputs                                               *
              00395 ; * RAM used:       0 bytes                                             *
              00396 ; * PROGRAM MEM:    3 Words                                             *
              00397 ; ************************************************************************
011A          00398 init
011A 0066     00399         clrf    PORTB           ; Init output latches for port B to 0
011B 0040     00400         clrw                    ; Clear W register
011C 0006     00401         tris    PORTB           ; Set all of PORT B to outputs
              00402 ; ************************************************************************
              00403 ; * testallkeys                                                          *
              00404 ; * This code is used to test all possible keys in the keypad.  First each  *
              00405 ; * key address is loaded and then senddtmf is called.                   *
              00406 ; * RAM used:       0 bytes                                             *
              00407 ; * PROGRAM MEM:    25 Words                                            *
              00408 ; ************************************************************************
011D          00409 testallkeys
011D 0C00     00410         movlw   key1        ; Place key “1” address into W
011E 09F5     00411         call    senddtmf    ; Transmit DTMF tone for this key
011F 0C02     00412         movlw   key2        ; Place key “2” address into W
0120 09F5     00413         call    senddtmf    ; Transmit DTMF tone for this key
0121 0C04     00414         movlw   key3        ; Place key “3” address into W
0122 09F5     00415         call    senddtmf    ; Transmit DTMF tone for this key
0123 0C06     00416         movlw   key4        ; Place key “4” address into W
0124 09F5     00417         call    senddtmf    ; Transmit DTMF tone for this key
0125 0C08     00418         movlw   key5        ; Place key “5” address into W
0126 09F5     00419         call    senddtmf    ; Transmit DTMF tone for this key
0127 0C0A     00420         movlw   key6        ; Place key “6” address into W
0128 09F5     00421         call    senddtmf    ; Transmit DTMF tone for this key
0129 0C0C     00422         movlw   key7        ; Place key “7” address into W
012A 09F5     00423         call    senddtmf    ; Transmit DTMF tone for this key
012B 0C0E     00424         movlw   key8        ; Place key “8” address into W
012C 09F5     00425         call    senddtmf    ; Transmit DTMF tone for this key
012D 0C10     00426         movlw   key9        ; Place key “9” address into W
012E 09F5     00427         call    senddtmf    ; Transmit DTMF tone for this key
012F 0C12     00428         movlw   keystar     ; Place key “*” address into W
0130 09F5     00429         call    senddtmf    ; Transmit DTMF tone for this key
0131 0C14     00430         movlw   key0        ; Place key “0” address into W
0132 09F5     00431         call    senddtmf    ; Transmit DTMF tone for this key
0133 0C16     00432         movlw   keypound    ; Place key “#” address into W
0134 09F5     00433         call    senddtmf    ; Transmit DTMF tone for this key
0135 0B1D     00434         goto    testallkeys ; jump back and do it again!
0136          00435 resetvector
01FF          00436         ORG     1ffh        ; The RESET vector of a 54 is at 1FFh
01FF 0B1A     00437         goto    init        ; Jump to initialion routine
              00438         END
```

```
MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0080 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
00C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0100 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXX----------
01C0 : ---------------- ---------------- ---------------- ---------------X
0FC0 : ---------------- ---------------- ---------------- ---------------X

All other memory blocks unused.

Program Memory Words Used:   311
Program Memory Words Free:   201

Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,     0 suppressed
```

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).

APPENDIX B: SINE.ASM

```
MPASM 01.40.01 Intermediate      SINE.ASM   3-31-1997  10:59:22         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

             00001 ;*******************************************************************
             00002         TITLE "PWM based sine wave generator"
             00003         LIST P=16C620, R=DEC
             00004
             00005         INCLUDE         <P16C620.INC>
             00001         LIST
             00002 ; P16C620.INC  Standard Header File, Version 1.01  Microchip Technology, Inc.
             00164         LIST
2007 3FB1    00006         __CONFIG        _BODEN_OFF&_CP_OFF&_PWRTE_ON&_WDT_OFF&_XT_OSC
             00007 ;
             00008 ;*******************************************************************
             00009 ;       File:           SINE.ASM
             00010 ;       Author:         Rob Stein
             00011 ;       Date:           12/20/95
             00012 ;       Assembler:      MPASM V01.40
             00013 ;       Xtal:           20 Mhz
             00014 ;       Inst Clk:       5 Mhz (200nSec)
             00015 ;*******************************************************************
             00016 ;       Description:
             00017 ;       Outputs a 60 Hz sythesized sine wave (32 step) via a general
             00018 ;       purpose I/O pin (RB1) into a low pass filter. A software PWM
             00019 ;       routine is used to create 32 separate sinewave steps.   This
             00020 ;       software was prototyped with the PICDEM1 board.
             00021 ;
             00022 ;       Circuit Diagram:
             00023 ;
             00024 ;                   2.7k            2.7k
             00025 ;       RB1  ___/\  /\  /\______/\  /\  /\________ Analog Output
             00026 ;                 \/  \/    |     \/  \/     |
             00027 ;                           |                 |
             00028 ;                        ----- 0.1uF       ----- 0.1uF
             00029 ;                        -----             -----
             00030 ;                          |                 |
             00031 ;                         GND               GND
             00032 ;
             00033 ;       ROM Usage: 98 words
             00034 ;
             00035 ;       RAM Usage: 6 bytes
             00036 ;
             00037 ;************************ Constant Definition   *********************
             00038
  01312D00   00039 FXTAL           EQU     .20000000       ; Crystal Frequency
  004C4B40   00040 FINST           EQU     FXTAL/4         ; Instruction Cycle Frequency
  0000003C   00041 FSINE           EQU     .60             ; Sine function frequency
  00000020   00042 STEP#           EQU     .32             ; Number of steps
  00000780   00043 FSTEP           EQU     FSINE * STEP#   ; Step frequency
             00044
             00045 ;************************  Register Definition  *********************
             00046
  00000020   00047 TEMPW           EQU     0x20            ; Temporary interrupt storage for W
  00000021   00048 DELAYCNT1       EQU     0x21            ; Delay routine counter low
  00000022   00049 DELAYCNT2       EQU     0x22            ; Delay routine counter high
  00000023   00050 STEPCOUNT       EQU     0x23            ; Sine step counter
  00000024   00051 OUTLOW          EQU     0x24            ; PWM low cycle load for TMR0
```

```
  00000025    00052 OUTHIGH EQU     0x25            ; PWM high cycle load for TMR0
              00053
              00054 ;************************     Bit Definition    ********************
              00055
  00000001    00056 PWM     EQU     0x01            ; RB1 used for PWM output
              00057
              00058 ;*****************************************************************
              00059 ;                                Reset Vector
              00060 ;*****************************************************************
              00061
0000          00062         org     0x000
0000 2816     00063         goto    Start           ; Begining of Program
              00064
              00065 ;*****************************************************************
              00066 ;                Interupt Vector and Service Routine
              00067 ;       This interrupt routine is entered via an overflow of TMR0 from
              00068 ;       0xFF to 0x00.  A test of RB1 determines if the next time state
              00069 ;       is a high or low cycle.  The next interrupt will occur based the
              00070 ;       TMR0 reload value (OUTLOW or OUTHIGH).
              00071 ;
              00072 ;       The interrupt routine was designed to use a minimial number of
              00073 ;       instruction cycles.  This was done to maximize the PWM duty cycle
              00074 ;       range (ie. a 5 % to 95 % range is achievable with this ISR).  Note
              00075 ;       that 'swapf' instructions are used to perform register moves without
              00076 ;       effecting the STATUS flags (this saves instruction cycles by
              00077 ;       eliminating the need to temporarily save the STATUS register).
              00078 ;
              00079 ;*****************************************************************
              00080
0004          00081         org     0x004         ; Interrupt vector location
0004          00082 IntVector
0004 00A0     00083         movwf   TEMPW         ; Temporarily save W
0005 1886     00084         btfsc   PORTB,PWM     ; Was this a Low cycle ?
0006 280F     00085         goto    PWMLow        ; No ...
0007          00086 PWMHigh
0007 0E25     00087         swapf   OUTHIGH,W     ; Yes... Load hi-time w/o affecting STATUS flags
0008 1486     00088         bsf     PORTB,PWM
0009 0000     00089         nop                   ; Delay to equalize high/low TMR0 load cycles
000A 0081     00090         movwf   TMR0          ; Load next edge interrupt time
000B 110B     00091         bcf     INTCON,T0IF   ; Clear TMR0 overflow flag
000C 0EA0     00092         swapf   TEMPW,F       ; Swap saved W
000D 0E20     00093         swapf   TEMPW,W       ; Restore W
000E          00094 IntEndHi
000E 0009     00095         retfie                ; Return from Interrupt
000F          00096 PWMLow
000F 1086     00097         bcf     PORTB,PWM
0010 0E24     00098         swapf   OUTLOW,W      ; Load low time
0011 0081     00099         movwf   TMR0          ; Load next edge interrupt time
0012 110B     00100         bcf     INTCON,T0IF   ; Clear TMR0 overflow flag
0013 0EA0     00101         swapf   TEMPW,F       ; Swap saved W
0014 0E20     00102         swapf   TEMPW,W       ; Restore W
0015          00103 IntEndLo
0015 0009     00104         retfie                ; Return from Interrupt
              00105
              00106 ;*****************************************************************************
              00107 ;                             Main Routine
              00108 ;*****************************************************************************
0016          00109 Start
0016 0183     00110         clrf    STATUS        ; Intitialize STATUS & select bank 0
0017 1683     00111         bsf     STATUS,RP0    ; Select register bank 1
0018 3088     00112         movlw   0x88
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0019 0081     00113         movwf   OPTION_REG    ; 1:1 TMR0 prescaler, PORTB pull-ups disabled
001A 30FF     00114         movlw   0xFF
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
001B 0085     00115         movwf   TRISA         ; Set Port_A as inputs
```

```
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
001C 0186     00116         clrf    TRISB         ; Set Port_B as outputs
001D 1283     00117         bcf     STATUS,RP0    ; Select register bank 0
001E 0086     00118         movwf   PORTB         ; PORT_B pins high
001F 0181     00119         clrf    TMR0          ; Initialize TMR0
0020 30A0     00120         movlw   0xA0
0021 008B     00121         movwf   INTCON        ; Enable TMRO and global interrupt
0022          00122 ResetStep
0022 3020     00123         movlw   STEP#
0023 00A3     00124         movwf   STEPCOUNT     ; Load counter for 32 steps
0024          00125 StepLoop
0024 2059     00126         call    Delay         ; Software delay
0025 0823     00127         movf    STEPCOUNT,W   ; Pass table offset via W
0026 2037     00128         call    SineTable     ; Get table value
0027 202B     00129         call    SetPWM        ; Set-up low & high PWM values
0028 0BA3     00130         decfsz  STEPCOUNT,F   ; Next step
0029 2824     00131         goto    StepLoop
002A 2822     00132         goto    ResetStep
              00133
              00134 ;*************************************************************************
              00135 ;                              Set PWM Subroutine                       *
              00136 ;       The following calculates the next low and high PWM time values.  *
              00137 ;       The two time values, OUTLOW and OUTHIGH, will be passed to the   *
              00138 ;       interrupt service routine.                                       *
              00139 ;*************************************************************************
002B          00140 SetPWM
002B 138B     00141         bcf     INTCON,GIE    ; Disable interrupts to protect ISR from...
              00142                               ; corrupting OUTLOW & OUTHIGH values
002C 00A4     00143         movwf   OUTLOW        ; Set PWM Duty Cycle
002D 0924     00144         comf    OUTLOW,W
              00145
002E 3E0A     00146         addlw   IntEndHi-IntVector ; Adjust for Int Service time
002F 00A5     00147         movwf   OUTHIGH
0030 0824     00148         movf    OUTLOW,W
0031 3E0A     00149         addlw   IntEndHi-IntVector ; Adjust for Int Service time
0032 00A4     00150         movwf   OUTLOW
              00151
0033 0EA4     00152         swapf   OUTLOW,F        ; Swap nibbles so that interrupt service...
0034 0EA5     00153         swapf   OUTHIGH,F       ; will not corrupt STATUS
0035 178B     00154         bsf     INTCON,GIE      ; Re-enable interrupts
0036 0008     00155         return
              00156
              00157 ;*************************************************************************
              00158 ;                       Lookup Table for Sine Wave
              00159 ;       This 32 entry table was generated to produce a 0.1*Vdd to
              00160 ;       0.9*Vdd (typicaly 0.5 to 4.5 volt) sine function.
              00161 ;*************************************************************************
0037          00162 SineTable
0037 0782     00163         addwf   PCL,F           ; Increment into table
0038 3400     00164         retlw   .0              ; Dummy table value
0039 3480     00165         retlw   .128            ; 0 degree, 2.5 volt
003A 3494     00166         retlw   .148
003B 34A7     00167         retlw   .167
003C 34B9     00168         retlw   .185
003D 34C8     00169         retlw   .200
003E 34D5     00170         retlw   .213
003F 34DE     00171         retlw   .222
0040 34E4     00172         retlw   .228
0041 34E6     00173         retlw   .230            ; 90 degree, 4.5 volt
0042 34E4     00174         retlw   .228
0043 34DE     00175         retlw   .222
0044 34D5     00176         retlw   .213
0045 34C8     00177         retlw   .200
0046 34B9     00178         retlw   .185
0047 34A7     00179         retlw   .167
```

```
0048 3494     00180         retlw   .148
0049 3480     00181         retlw   .128            ; 180 degree, 2.5 volt
004A 346C     00182         retlw   .108
004B 3459     00183         retlw   .89
004C 3447     00184         retlw   .71
004D 3438     00185         retlw   .56
004E 342B     00186         retlw   .43
004F 3422     00187         retlw   .34
0050 341C     00188         retlw   .28
0051 341A     00189         retlw   .26             ; 270 degree, 0.5 volt
0052 341C     00190         retlw   .28
0053 3422     00191         retlw   .34
0054 342B     00192         retlw   .43
0055 3438     00193         retlw   .56
0056 3447     00194         retlw   .71
0057 3459     00195         retlw   .89
0058 346C     00196         retlw   .108
              00197
              00198 ;***************************************************************************
              00199 ;                       Time Delay Sub-routine                             *
              00200 ;       The time delay is used to create the precision 32 steps.  The      *
              00201 ;       32 step times totaled together add up to a 60 Hz rate. Note that   *
              00202 ;       constants DELAYCNT# are used so that other frequencies can easily  *
              00203 ;       be generated (example: FSINE equ .50 for a 50 Hz  sinewave).       *
              00204 ;***************************************************************************
  00000A2C    00205 TDELAY          EQU     FINST/FSTEP     ; # of delay count cycles
  0000032D    00206 ADJTDELAY       EQU     TDELAY/3 - 55   ; Adjust for main routine cycles
  00000003    00207 TDELAYHI        EQU     high ADJTDELAY  ; Most Significant Byte of TDELAY
  0000002D    00208 TDELAYLO        EQU     low ADJTDELAY   ; Least Sig. Byte of TDELAY
              00209
0059          00210 Delay
0059 3003     00211         movlw   TDELAYHI
005A 00A2     00212         movwf   DELAYCNT2               ; Load high byte delay counter
005B 01A1     00213         clrf    DELAYCNT1
005C          00214 LoopD1
005C 0BA1     00215         decfsz  DELAYCNT1,F     ; Finished with 256 loops ?
005D 285C     00216         goto    LoopD1          ; No ... keep going
005E 0BA2     00217         decfsz  DELAYCNT2,F     ; Yes... Done with TDELAYHI loops ?
005F 285C     00218         goto    LoopD1          ; No ...
              00219
0060 302D     00220         movlw   TDELAYLO        ; Yes... Load low byte with adjust for...
0061 00A1     00221         movwf   DELAYCNT1       ; main routine cycles.
0062          00222 LoopD2
0062 0BA1     00223         decfsz  DELAYCNT1,F     ; Finished with TDELAYLO loops ?
0063 2862     00224         goto    LoopD2          ; No ... keep going
0064 0008     00225         return                  ; Yes... Finished
              00226
              00227         END                     ; That's all Folks !

MEMORY USAGE MAP ('X' = Used,  '-' = Unused)


0000 : X---XXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXX----------- ----------------
2000 : -------X-------- ---------------- ---------------- ----------------

All other memory blocks unused.

Program Memory Words Used:    98
Program Memory Words Free:   414


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     3 reported,     0 suppressed
```