

AN-201 Producing DTMF using FreqOut()

Introduction

Dual-tone multi-frequency, or DTMF, is the signaling system used to communicate with telephone system switching equipment, e.g. for dialing a phone number. With the proper encoder and decoder, however, it can be used for any number of control purposes.

The DTMF system, also called TouchTone, uses 8 unique frequencies divided into two groups to make 16 possible coding combinations. The frequency combinations corresponding to the layout of the TouchTone keypad are shown in the table below.

	1209 Hz	1336 Hz	1477 Hz	1633Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

This table indicates, for example, that the frequencies 770Hz and 1336Hz are combined to represent a 5.

Implementation

The ZBasic `FreqOut()` subroutine is designed to be able to generate a signal that represents the combination of two frequencies and is thus well-suited to generating DTMF. The subroutine `DTMF()` below is called giving the pin number on which the signal should be generated and a byte value giving the ASCII character code of the desired signal (corresponding to the entries in the table above).

The `DTMF()` subroutine is fairly straightforward. It uses three data tables to determine the pair of frequencies to be used. The tables `rowFreq` and `colFreq` give the row and column frequencies from the matrix above and the `freqIndices` table specifies the index for the row table and the column table to use for each supported character. The row/column indices are combined into a byte value to reduce the space needed for the table. The index table has entries for the lowest and highest character values supported with the entries corresponding to invalid characters set to zero. Once the combined index value is determined and validated, it is a simple matter to extract the separate indices and apply them to the frequency tables to obtain the proper frequency pair.

```
' This table gives the four row frequencies (aka low group).
Private rowFreq As IntegerVectorData({ 697, 770, 852, 941 })

' This table gives the four column frequencies (aka high group).
Private colFreq As IntegerVectorData({ 1209, 1336, 1477, 1633 })

' This table gives the indices into the row and column
' tables above for each supported character. To save space,
' the indices are packed into a byte with the row index in
' the high nibble and the column index in the low nibble.
Private freqIndices As ByteVectorData({
' indices character/hexadecimal code
  &H43      ' # &H23
  0         ' invalid
  0         ' invalid
  0         ' invalid
  0         ' invalid
```

AN-201 Producing DTMF Using FreqOut()

```
0      ' invalid
0      ' invalid
&H41  ' *  &H2A
0      ' invalid
0      ' invalid
0      ' invalid
0      ' invalid
0      ' invalid
&H42  ' 0  &H30
&H11  ' 1  &H31
&H12  ' 2  &H32
&H13  ' 3  &H33
&H21  ' 4  &H34
&H22  ' 5  &H35
&H23  ' 6  &H36
&H31  ' 7  &H37
&H32  ' 8  &H38
&H33  ' 9  &H39
0      ' invalid
0      ' invalid
0      ' invalid
0      ' invalid
0      ' invalid
0      ' invalid
0      ' invalid
&H14  ' A  &H41
&H24  ' B  &H42
&H34  ' C  &H43
&H44  ' D  &H44
}))
```

```
Sub DTMF(ByVal pin As Byte, ByVal key As Byte)
    ' ignore codes outside the table range
    If ((key >= &H23) And (key <= &H44)) Then
        Dim index As Byte

        ' fetch the row/column indices from the table
        index = freqIndices(key - &H23 + 1)

        If (index <> 0) Then
            ' generate tones and pause
            Call FreqOut(pin, rowFreq(Shr(index, 4)), colFreq(index And &H0f), 100)
            Call Delay(0.050)
        End If
    End If
End Sub
```

Using this subroutine as a building block, we can construct another subroutine that will generate a sequence of DTMF tones corresponding to a sequence of characters. The routine below, `Dial()` takes a parameter giving the signal pin and a second parameter giving the sequence of characters specified as a string.

In addition to the usual TouchTone characters, this subroutine also accepts a comma to represent a half-second delay. Other characters are simply ignored so it is possible to include a dash, for example, for readability purposes.

```
Sub Dial(ByVal pin As Byte, ByVal dialStr As String)
    Dim strLen As Integer
    Dim i As Integer

    strLen = Len(dialStr)
```

AN-201 Producing DTMF Using FreqOut()

```
For i = 1 to strlen
  Dim c As Byte

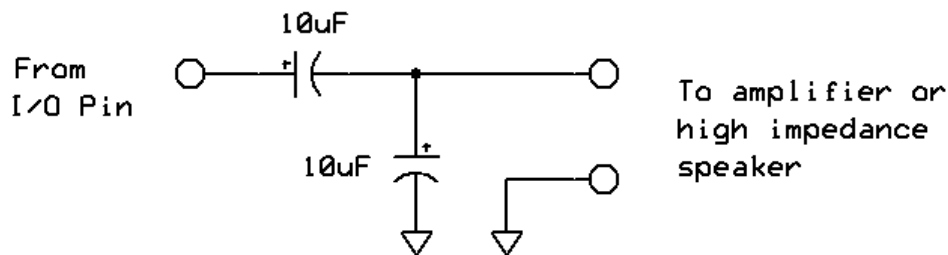
  ' get the next character of the string
  c = Asc(dialStr, i)
  If (c = &H2c) Then
    ' comma detected, generate a delay
    Call Delay(0.5)
  Else
    Call DTMF(pin, c)
  End If
Next i
End Sub
```

Finally, here is a main routine that invokes the Dial() subroutine to dial a phone number.

```
Sub Main()
  ' Using the routines above, this call will dial
  ' the number for toll-free information using pin 20.
  ' Note that a 9 is first dialed to get an outside line.
  Call Dial(20, "9,,800-555-1212")
End Sub
```

Filtering

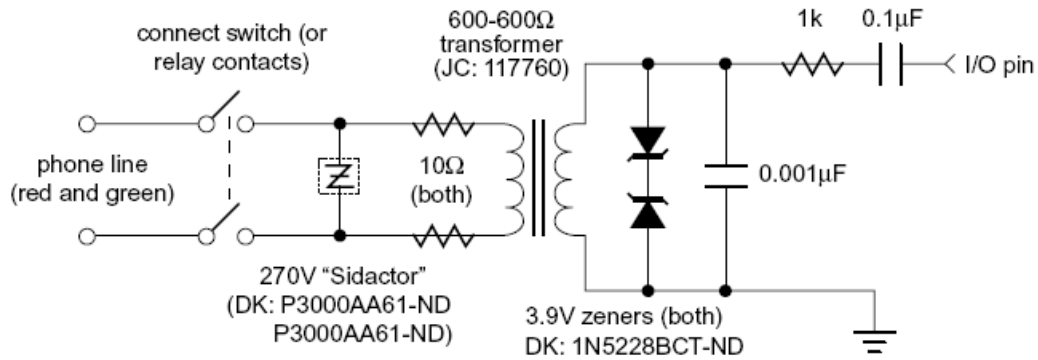
Because of the high frequency characteristics of the pulse train that FreqOut() uses to synthesize the waveform some filtering is required. The example circuit below may be used to both filter the signal and couple the output to a high impedance speaker ($> 40\Omega$) or an amplifier. Note, however, that the signal is too large to be fed to the microphone input of an amplifier. Instead, the Auxiliary or Line input should be used. This simple filter has been used to successfully dial a telephone number using the output from the program given. Depending on your application you may need a more sophisticated filter.



Telephone Interface

The generated signal may be acoustically coupled to the telephone network simply by placing the speaker near or against the telephone's microphone. Alternately, the signal may be introduced to the telephone network using a special interface that protects both the network and the attached equipment. Such a device is usually called a DAA. The telephone company has strict rules regarding the characteristics of the signals introduced into their network and the load presented by equipment connected to their lines. There are many books available on the subject that give suggested interface circuits and other resources may be found online. An example of a telephone line interface circuit, excerpted from the Parallax Basic Stamp manual, is given below. The author has not built and tested this circuit.

AN-201 Producing DTMF Using FreqOut()



Parts Sources

Digi-Key (DK), 1-800-344-4539
or 218-681-6674

Jameco (JC), 1-800-831-4242
or 415-592-8097

Schematic of the phone-line interface.

Caution

Although the DC voltage present on the telephone line is typically under 50 volts, it can have voltages in excess of 100VAC during ringing and transients can occur on the line due to accidents, lightning strikes, etc. Appropriate safety practices should be implemented when working on or around telephone equipment.

Author

Don Kinzer is the founder and CEO of Elba Corporation. He has extensive experience in both hardware and software aspects of microprocessors, microcontrollers and general-purpose computers. Don can be contacted via email at dkinzer@zbasic.net

e-mail: support@zbasic.net

Web Site: <http://www.zbasic.net>

Disclaimer: Elba Corp. makes no warranty regarding the accuracy of or the fitness for any particular purpose of the information in this document or the techniques described herein. The reader assumes the entire responsibility for the evaluation of and use of the information presented. The Company reserves the right to change the information described herein at any time without notice and does not make any commitment to update the information contained herein. No license to use proprietary information belonging to the Company or other parties is expressed or implied.

Copyright © 2005 Elba Corporation. All rights reserved. ZBasic, ZX-24, ZX-40 and combinations thereof are trademarks of Elba Corp. or its subsidiaries. Other terms and product names may be trademarks of other parties.