DTMF stands for Dual-Tone Multiple Frequency… It is the standard method of dialing for North America. DTMF utilizes two frequencies to symbolize each digit. One frequency symbolizes the row of the digit and one frequency symbolizes the column. Four frequencies are used for the rows, and four for the columns, for a total of 8 frequencies and 16 possible combinations.

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

DTMF Detection is the function of decoding a valid DTMF digit, made up of two distinct frequencies, back to the original digit. This document describes how to implement DTMF detection using a correlation algorithm on a Scenix SX microcontroller.

Correlation Algorithm Summary:

The DTMF detection algorithm this document describes is a *correlation* algorithm, meaning the algorithm measures the extent to which an input signal matches an internally generated reference signal. The correlation algorithm has the advantage of using a minimal amount of RAM to perform DTMF detection. One implementation on an SX18 was able to perform 6 channels of simultaneous DTMF detection, meeting MITEL specs, using only 136 bytes of RAM. The algorithm uses a running sum of the input signal multiplied by each reference signal. The correlation algorithm for each frequency utilizes both a sine-wave reference signal and a cosine-wave reference signal. Using the two reference waves, 90 degrees out of phase, accommodates any phase of input signal. At the end of the running sum period (≅ 13ms), the accumulator for the sine wave is squared and added to the squared accumulator for the cosine wave. The square root of the new sum is taken, and the resultant value is indicative of the amount of signal present for that particular reference frequency:

$$\textit{Amount of Signal} = (\textit{Sine Accumulator}^{\,2} + \textit{Cosine Accumulator}^{\,2})^{1/2}$$

The result for each frequency is compared to the other frequency detectors to determine if an input frequency is present. An accumulator has to be at a level sufficiently above the levels for the other detectors to trigger a valid frequency. Because of RAM constraints, either the low frequencies or the high frequencies are sampled at any time. When a valid frequency is detected in the high-band, the algorithm runs the detectors again for the high band. If the same valid frequency is re-detected, the detectors are run once again for the low-frequency band. If a valid low frequency is detected after the same valid high frequency has been detected twice, the DTMF tone is considered valid. If two frequencies are detected as valid at the same time in the same band, then it is considered an invalid result and the detection is reset.

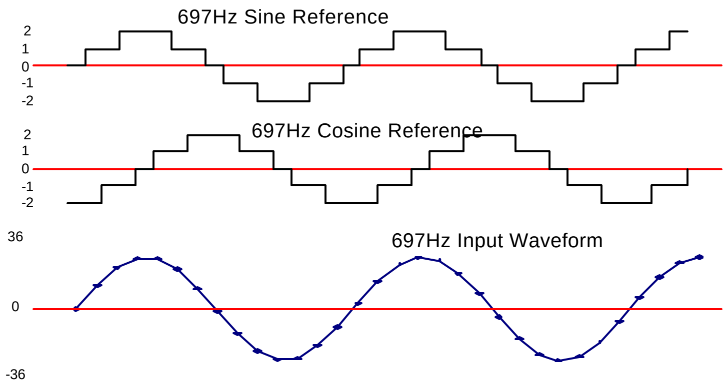
Reference waves:

The reference waves are multiplied by the input signal at the sampling rate of 9600Hz and the result of the multiplication is added to an accumulator. For easy multiplication, a table consisting of only 0, 1, 2, -1, and -2 values generates the reference wave. The reference waves are generated using 16-bit phase accumulators as the index into the reference table. Since only 4 reference frequencies are generated at a time, the reference waves use 4*2words of RAM, or 8 words. A value of 64 is added to the high byte of a phase accumulator to obtain the 90-degree phase shifted value, or cosine reference.

Sampling:

An analog sample is taken at a sampling rate of 9600Hz. Each sample is converted using 36 cycles of sigma-delta Analog to Digital conversion on the analog input. Each sample is passed into a routine that multiplies the references by the input value and adds the result to the corresponding reference's accumulator.

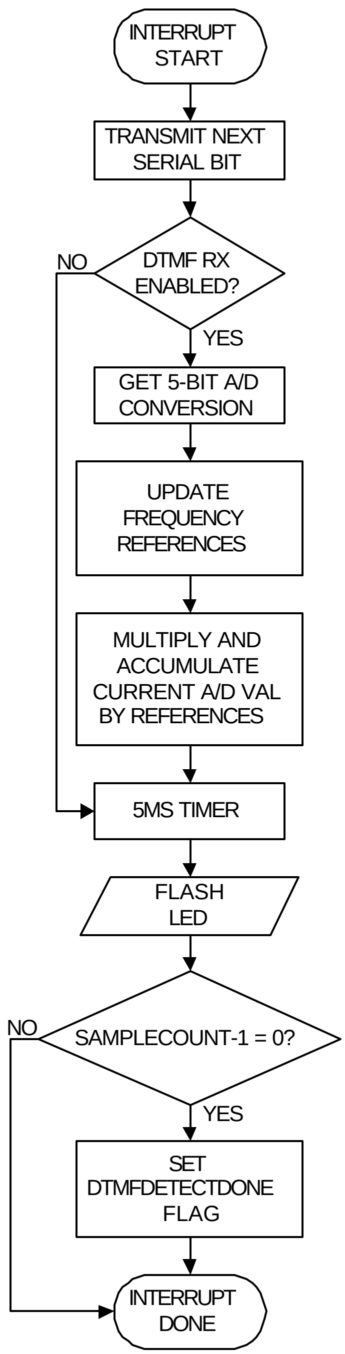
For each frequency, two accumulators are required, a sine and a cosine accumulator. Each accumulator is 16-bits wide. Since there are 4 simultaneous frequencies being sampled at any time, each channel of DTMF detection requires its own 2 * 4 * 16 bits of space, or 16 bytes of RAM.



In this example, the input waveform is the same frequency and same phase as the 697Hz sine reference… A large value in the 697Hz Sine Accumulator will result, and an insignificant value will be present in the 697Hz Cosine Accumulator. If the input waveform's phase were to shift, the value in the cosine accumulator would increase at a greater rate, and the value in the sine accumulator would increase at a slower rate. If the input signal did not have a frequency component that is similar to the reference signal, then the resultant sum for that reference signal would be insignificant. After a specific number of samples are compared with the input (105 samples for high frequencies, 140 samples for low frequencies), the amount of signal present is calculated using this formula:

$$\textit{Amount of Signal} = (\textit{Sine Accumulator}^{\,2} + \textit{Cosine Accumulator}^{\,2})^{1/2}$$

The Scenix DTMF Detection Interrupt Service Routine



The interrupt service routine for the DTMF detection program performs these virtual tasks: A/D Conversion, 9600bps serial transmitter, DTMF Detection, and a 5ms timer. Because of the number of cycles required to perform each DTMF sample, the interrupt rate for the DTMF detection source code has been set to a relatively low 9600bps. This insures that the Interrupt Service Routine does not overflow the 104us allowed for one interrupt.

9600bps Serial Transmitter

If there is any data to be sent, the 9600bps serial transmitter outputs the next bit of data on each pass of the interrupt service routine.

5ms Timer

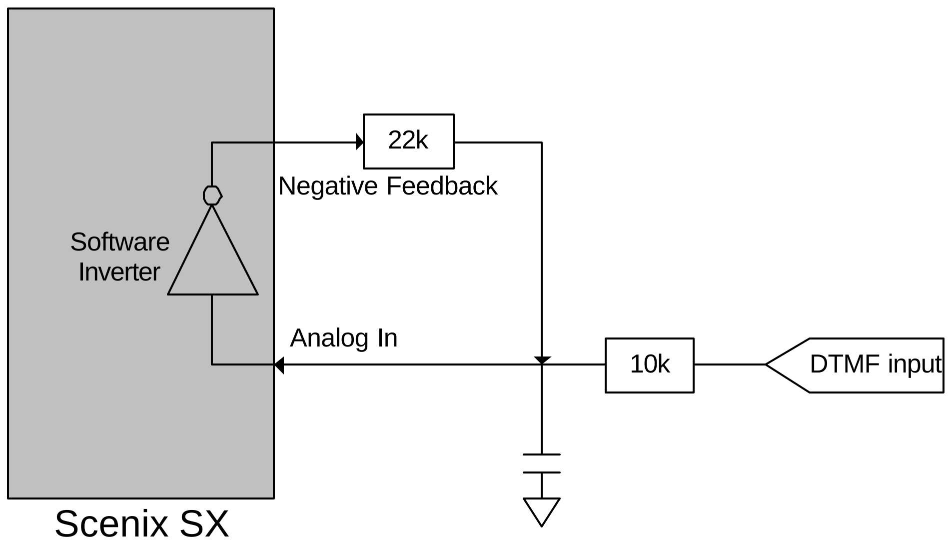
The 5ms timer increments the *timer5ms* register every 5ms. If *timer5ms* rolls over from 255 to zero, the 5ms timer VP sets the *timerFlag* flag.

Led Flasher

The LED flasher copies the state of bit 3 of the *timer5ms* register to *ledPin*.

DTMF Detection ISR: A/D Conversion

Analog to Digital conversion is performed in software, within the interrupt service routine. This is the analog circuitry required to perform the conversion:



The software of the A/D converter counts the number of negative/positive pulses fed back over a specific number of pulses. The pseudocode for the conversion is as follows:

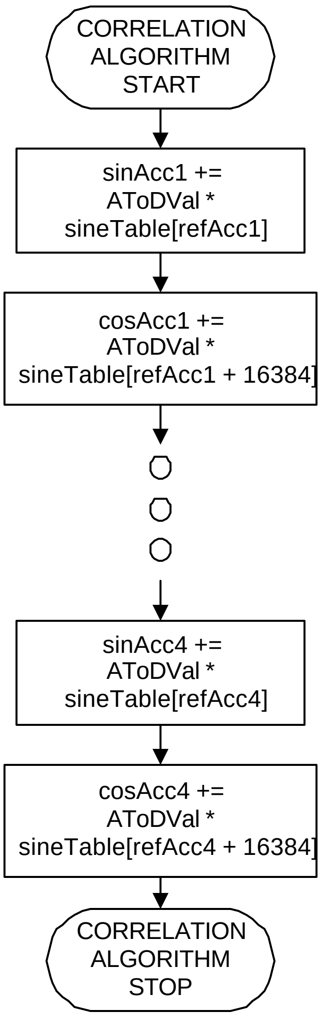
1.0 Perform Analog to Digital Conversion on input signal
- 1.1 Load loop counter with number of desired samples
  - 1.11 Invert current state of the analog input pin and output to negative feedback pin
  - 1.12 If Negative Feedback Pin == 0, increment A/D Value, else decrement A/D value
- 1.2 Decrement loop counter.
  - If loop counter != 0, do loop again, else exit

DTMF Detection ISR: Updating the Frequency References

The interrupt service routine generates 4 simultaneous frequencies internally, used as references to multiply with the input signal. The frequencies are generated using 16-bit phase accumulators. The phase accumulators are updated by adding a 16-bit constant to them. This 16-bit constant corresponds to the phase that each accumulator must increment for each sample, with 0 corresponding to a phase of 0° and 2^{16} corresponding to a phase of 360°.
The value in the reference accumulators is used to generate both a sine and cosine signal. For the cosine reference, we need a phase shift of 90°. As we are only concerned with the highest nibble, we can see that adding a value of 16384, or

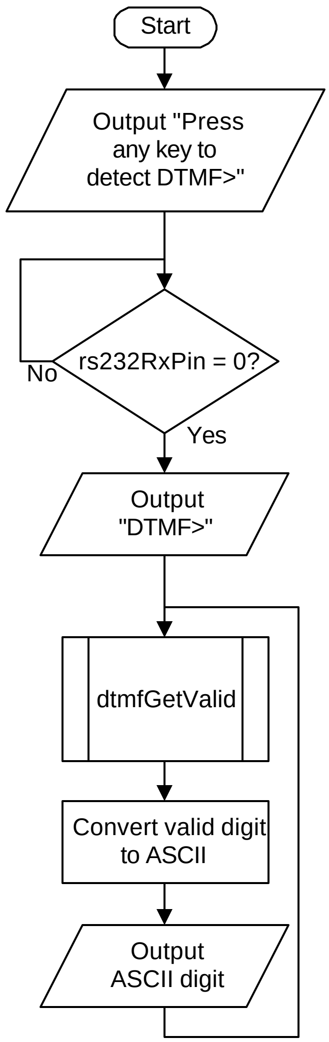
0x4000 to the reference adds a phase shift of 90°, so we add 0x40 to the MSB of the reference accumulator.

DTMF Detection ISR: Multiplying the input by the references

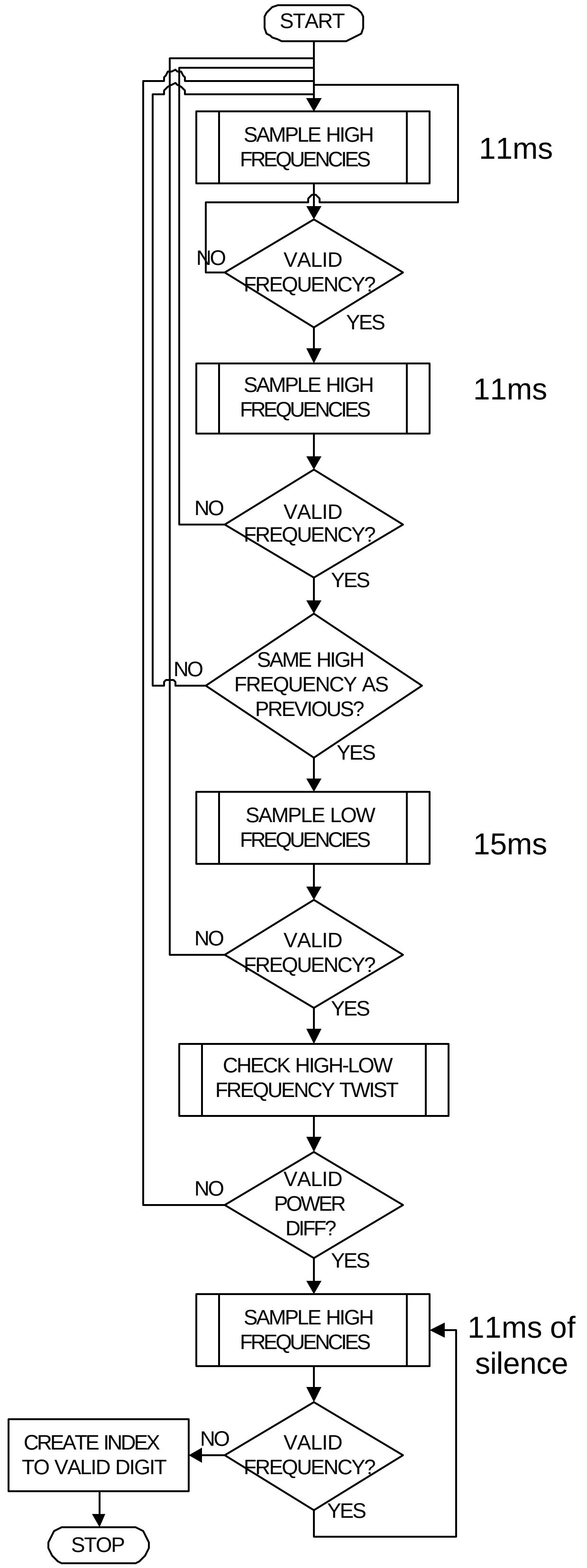


The correlation algorithm requires that the input signal be multiplied by both a sine and cosine version of each reference. The results of these calculations are stored in the sinAcc and cosAcc accumulators, each of which is 16-bits wide.

The subroutines *dtmfDetectSine* and *dtmfDetectCosine* perform the multiplications and accumulations within the Interrupt Service Routine.
Multiplication is made easy by the small values used in the reference wave. The table of magnitudes for the references contains these 8 values: 0, 1, 1, 2, 2, 2, 1, and 1. When the reference wave is zero, the sampling routine simply exits, since no change will be made to the accumulator. If the magnitude of the reference wave = 1, the sampling routine adds the A/D value to the accumulator. If the magnitude of the reference wave = 2, the sampling routine shifts the A/D value left once before adding it to the accumulator. When the MSB of the reference accumulator is = 1, the routine negates the A/D value before performing the addition.



The main loop of the DTMF detection program begins and awaits a key-press from the user (A low voltage on the rs232RxPin.) Once the low rs232RxPin is pulled low, the program begins detecting DTMF. As valid DTMF digits are detected, they are output, in ASCII and comma delimited, to the terminal screen at a rate of 9600bps, N, 8, 1.



dtmfGetValid

The dtmfGetValid subroutine performs all the functions required to get one valid DTMF digit. It will return when a valid DTMF digit, followed by silence, is detected. On return, an index to the valid digit is stored in the *dtmfDigitIndex* register. The minimum time to detect a valid DTMF digit is 37ms + 10ms of silence, and the worst case time is approximately 44ms + 10ms of silence. See flowchart for timing.

SAMPLE

The SAMPLE routine clears all of the sine and cosine accumulators, enables the DTMF detectors, and waits for a DTMF sample to take place. Before calling SAMPLE, the sample_lows bit must be set up. A '1' in sample_lows will force it to sample low frequencies, and a '0' will force it to sample high frequencies. After an accumulation has finished, SAMPLE utilizes the dtmfDetectCalcs routine to calculate the amount that each reference signal resembled the input signal. It also calls the dtmfGetWinner routine, which compares all four accumulated results and finds the highest score and the second-highest score.

dtmfDetectCalcs

The dtmfDetectCalcs subroutine is passed a pointer to the first byte in of four that makes up a set of accumulators. Example:

```
mov  w,#sinAcc1Lo       ; do the calculations on acc1
call @dtmfDetectCalcs
```

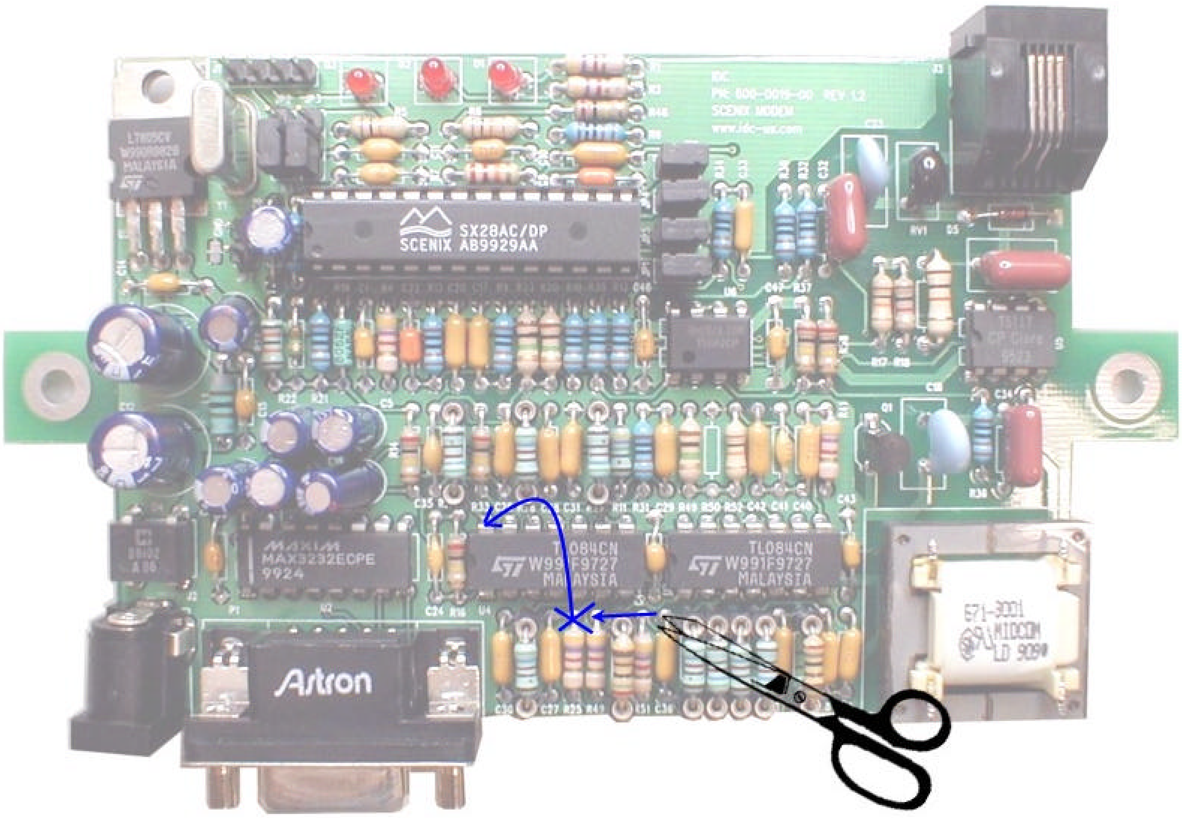
After calling the absoluteScaleAndSquare twice to scale and square the sine and cosine accumulators, dtmfDetectCalcs adds the squared sine and cosine accumulators together and takes the square root of the result. It returns the result of the calculation in the W register: $W = \sqrt{(sinAcc^2 + cosineAcc^2)}$

absoluteScaleAndSquare

The absoluteScaleAndSquare routine is passed a pointer to a 16-bit signed value in the W register. The routine first converts the 16-bit signed value to a positive 16-bit number, and then calls the scaleByN routine to truncate *scaling_factor* LSB's from the value to cut it to 8 bits. *Scaling_factor* is a constant defined at the top of the source code. Depending on the method used for performing the A/D conversion, scaling factor will have to be adjusted to allow for differing sized A/D values and, hence, a variety of accumulated values. Using 36 cycles of the sigma-delta A/D requires a scaling factor of 3 or 4, since it generates maximum values of 36 or -36.

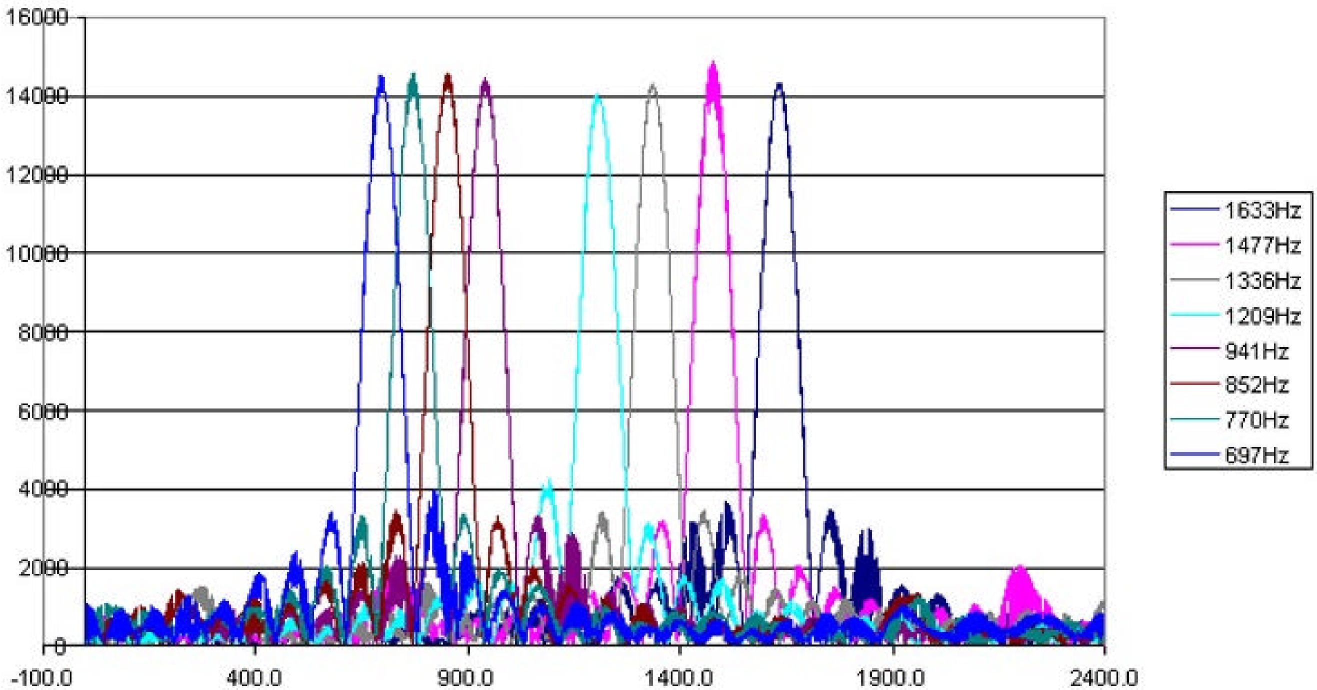
The Scenix DTMF Detection Hardware

The DTMF detection source code can be run on the Scenix Modem Demonstration board, with a few minor modifications to bypass the filters. To bypass the filter circuitry, simply cut the top lead of R25, and jumper the newly chopped lead of R25 to pin 14 of U4. A diagram of the modifications to the Modem Demo Board is shown below:



Testing

Feeding the algorithm digitally generated sine waves of a ramping frequency generated this response curve for each filter. Each of these curves was based on 115 samples on the input, and the input wave was then re-started. Increased selectivity is accomplished by increasing the number of input samples taken.



In customer tests, this algorithm, linked to a digital codec, outperformed a TI DSP in talk-off tests using the Bellcore test tapes, but required small tweaks to decrease its frequency selectivity.

Summary of The Scenix DTMF Detection Solution

Proven in several production designs, this DTMF detection algorithm is ideal for low-cost embedded systems where the addition of a separate DTMF detection IC is not feasible from a price standpoint, and the algorithm must be embedded in the host SX. Answering machines and security systems are a good example of this type of application.

Since each channel's 8 accumulators are stored in their own bank, this algorithm can also be modified to perform up to 6-channels of simultaneous DTMF detection in an SX28 or SX18. Decreased MIPS usage and additional accuracy would be realized by replacing the software A/D with a hardware analog to digital converter or CODEC.

For more information on implementing this algorithm using a CODEC, see Application Note 19 - *Implementing DTMF Detection using the Silicon Laboratories DAA (Data Access Arrangement)* by Abraham Si