DTMF Detection and Generation Virtual Peripheral Modules

Application Note 41

August 2000

1.0 Preface

This application note describes the methodology and the hardware/software needed to perform DTMF generation and detection with the SX communications controller. It also describes the DTMF operations and its implementation on the SX using the concept of Virtual Peripheral.

2.0 Introduction

In many cases, there is a need for specific hardware to perform data exchange via the telephone line. The technique to dial a number or detect a number is called Dual Tone Multi Frequencies (from now on only referred as DTMF). DTMF has two main functions, detection used to detect the dialed number (incoming), and generation to generate (dial) a number (outgoing).

DTMF utilizes two frequencies to symbolize each digit. One frequency symbolizes the row and one frequency the column of the digit. Four frequencies are used for the rows, and four for the columns, for a total of 8 frequencies and 16 possible combinations.

To generate a specific digit, the system needs to generate the frequencies corresponding to the row and column in which the digit resides. Each digit is substituted with a tone signal; the tone is a combination of the two frequencies modulated at the same time.

To detect a specific digit, the system needs to detect the frequencies, separate them in the tone signal, and then check them to see if they together qualifies as one of the

Table 2-1. DTMF Keyboard Matrix

		High Group Frequency (Hz)			
		1209	**1336**	**1477**	**1633**
Low Group Frequency (Hz)	**697**	1	ABC 2	DEF 3	A
	770	GHI 4	JKL 5	MNO 6	B
	852	PQRS 7	TUV 8	WXYZ 9	C
	941	*	0	#	D

3.0 General Description

Dual Tone Multi Frequency (DTMF) has been employed for the modern telephone systems worldwide with push-button or touchtone sets. The DTMF signaling (tone dialing) offers a higher speed of dialing than the pulse dialing systems, for use in voice mail systems, modems, and ATM machines.

DTMF signaling is a tone signaling method used to transmit address information and other information over voice frequency transmission facilities. A DTMF signal consists of two simultaneous sinusoid signals; one signal is selected from four high-group frequencies and one signal is selected from four low-group frequencies. These 8 frequencies can be combined in 16 possible combinations, as shown in Table 2-1, and represents the numbers from 0 to 9, # and * (and special characters A, B, C, and D, which are reserved for use in non-public telecommunication networks). The tone frequencies are designed to avoid problems from harmonics. For example, sending the frequencies 770Hz and 1336Hz simultaneously generates the digit 5.

As can be seen in Figure 3-1, the DFT absolute sum-of-product for DTMF digits separated from both voice and music. Thus, DTMF detection shall be able to detect encoded numbers and special characters) from music and voice.

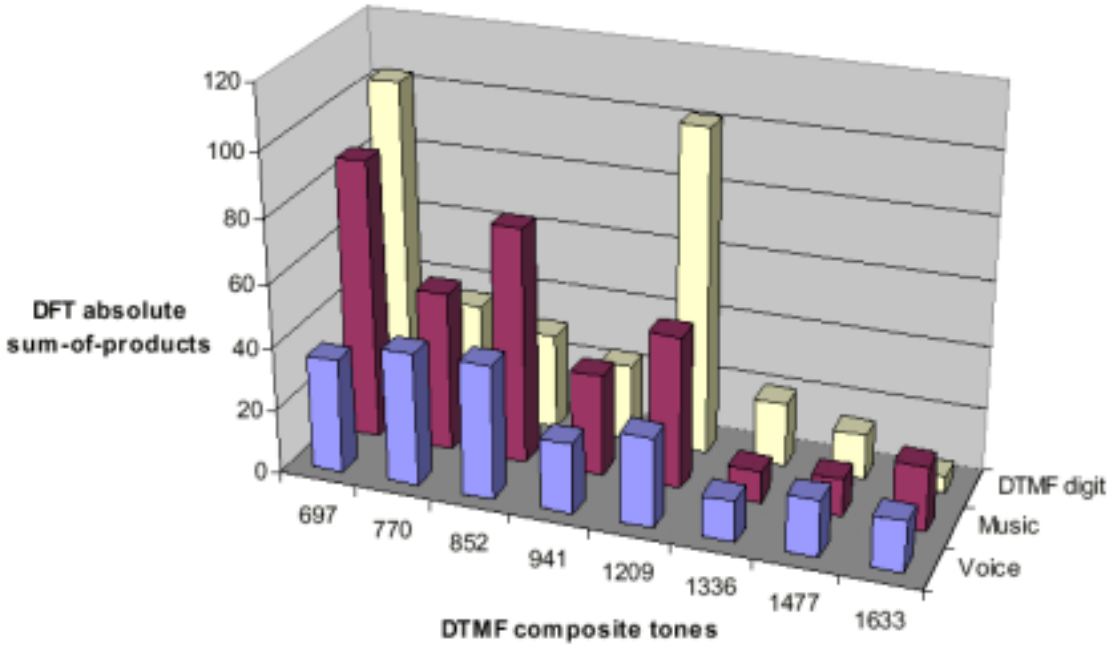

Figure 3-1. DFT Results for DTMF (Digit '1'), Music and Voice

A DTMF signal is defined by its timing, frequency, and power. The timing includes rise time, signal duration, fall time, signal-off time, interdigit interval, and cycle time. The signals nominal frequency, power and noise acceptance also have to be within the specifications for DTMF signals. The requirements that have to be met for valid DTMF signal, for both receiver (detection) and transmitter (generator), are described in the Bellcore specification (GR-506-CORE) [2].

This chapter describes the basics from Bellcore's specification and explains some of the requirements a DTMF receiver and a DTMF transmitter have to meet to ensure valid DTMF generation and DTMF detection.

3.1 DTMF Generation

To generate a specific digit, the system needs to generate the frequencies corresponding to the row and column in which the digit resides. Each digit is substituted with a tone signal; the tone is a combination of the two frequencies modulated at the same time.

The DTMF transmitter has to ensure that the generated DTMF signal is within the specifications for timing, frequency levels and power levels, in order for the DTMF receiver to detect the signal. For visualization of these terms described in the following sections see Appendix A. Figure 9-1 to Figure 9-6 shows the DFT (Discrete Fourier Transform) of the DTMF signals where frequency component, power, and twist can be seen. Figure 9-7 to Figure 9-11 shows the different timings for DTMF signals.

3.1.1 Frequency Requirements

The DTMF transmitter uses two sinusoid signals, one from the low-group frequencies that represents the row, and one from the high-group frequencies that represents the column. According to the specifications, each of these frequencies should be within ±1.5% of the nominal values specified in Table 2-1.

3.1.2 Power Requirements

The signal power for each of these frequencies shall according to the specifications be -7dBm0±1dB.

3.1.3 Timing Requirements

These two frequencies, that together represent a specific digit or special character, are added together, and the duration of this signal shall to be equal to or longer than 50 milliseconds for it to be a valid DTMF signal. An interdigit duration time has to come between DTMF signals. The interdigit duration time includes fall time, signal-off time and rise time, and shall be equal to or longer than 45 milliseconds. The signal duration and the interdigit duration together give the whole cycle time, and it shall be equal to or longer than 100 milliseconds. For more details about DTMF generation, see Bellcore specification.

3.2 DTMF Detection

To detect a specific digit, the system needs to detect the two frequencies by extracting them from within the input signal, and then checking them to see if they together qualify as one of the digits (one frequency for row and one for column). The DTMF receiver has to meet many requirements to ensure that only the DTMF signal is decoded, and not speech signals or other voice-band signals.

3.2.1 Code Validity Check

A DTMF receiver shall reject any received signal that does not simultaneous consist of exactly one low-group frequency and exactly one high-group frequency. This is the code validity check and it helps prevent speech and noise signals from being interpreted as DTMF codes.

3.2.2 Frequency Requirements

The DTMF receiver shall respond to signals where both DTMF frequencies are within ±1.5% of the nominal values specified in Table 1. And it shall reject DTMF signals where one or both frequencies are outside of ±3.5% of the nominal values. The DTMF receiver may accept or reject DTMF signals where one or both frequencies are between ±1.5% and ±3.5% of the nominal values.

3.2.3 Power Requirements

In addition to the frequency requirement, the DTMF receiver shall accept signals when the per frequency power of both frequencies is ≥-25dBm and ≤0dBm. And it shall reject signals when the per frequency power of one or both frequencies is ≤-55dBm. The DTMF receiver may accept or reject signals when the per frequency power of one or both frequencies is between –25dBm and

-55dBm.

3.2.4 Twist

The difference in power between the two frequencies of a DTMF signal is called twist, and the DTMF receiver shall accept signals when the power of the high-group frequencies is ≤(L+4) dBm and ≥(L-8) dBm, where L is the power of the frequency from the low-group frequencies, see Table 2-1.

3.2.5 Timing Requirements

A DTMF receiver shall accept DTMF signals when the signal duration is ≥40 milliseconds. It shall also reject DTMF signals when the signal duration is ≤23 milliseconds. The DTMF receiver may either accept or reject DTMF signals when the signal duration is between 23 and 40 milliseconds. A DTMF receiver shall accept and correctly interpret DTMF signals with interdigit intervals ≥40 milliseconds. This interdigit interval, that include fall time, signal-off time, and rise time, together with the DTMF signal duration gives the cycle time, and a DTMF receiver shall accept and correctly interpret DTMF signals with cycle time ≥93 milliseconds. A DTMF receiver may accept DTMF signals with a cycle time ≤93 milliseconds when the minimum DTMF signal duration time requirement is met.

3.2.6 Non-Linear Distortion

Non-linear distortion occurs in DTMF signals generated by network or end user equipment. Non-linear distortion produces extraneous frequencies in the DTMF signal that can cause degradation of DTMF receiver performance.

To ensure that these requirements are met, Bellcore has developed a DTMF detection test tape containing speech and voice-band signals to see if the receiver triggers when the requirements are not meet. The DTMF detection Virtual Peripheral is tested against to the Bellcore test tapes, for test result see section 7.2.5.

4.0 Ubicom DTMF Implementation

DTMF can be implemented on SX devices by using the available DTMF Virtual Peripheral modules at www.ubicom.com. The two modules available are:

Module	Source code	Template version
DTMF Generation	`dtmf_gen.src`	`vp_guide_1.04.src`
DTMF Detection	`dtmf_det.src`	`vp_guide_1.04.src`

The DTMF detection monitors an input port and uses a software Analog to Digital Converter (ADC), to convert the input signal to a digital value. The digital input value is then compared with each of the eight frequencies in the matrix in Table 2-1. The detection routine use the predefined frequencies as reference to compare the input signal over a time period; each time the two signals are equal a hit count is incremented, and when the number of hits reaches a specified number, it is a valid tone frequency.

To perform DTMF generation on the SX Communication controller a PWM Virtual Peripheral and an implementation of two artificial sine waves is required.

4.1 DTMF Generation Virtual Peripheral

The DTMF generation Virtual Peripheral uses two artificial sine generation routines, one for the low group frequency and one for the high group frequency. It multiplies the high frequency amplitude by 1.25 (twist, see section 3.2.4) and adds it with the low frequency signal. The sum of the two signals is sent onto the PWM pin (D/A conversion.). The mainline routine handles the digit to frequency decoding, and the dial and pause times. Detailed information about artificial sine generation and PWM is available at http://www.ubicom.com.

4.1.1 Software Theory

DTMF generation encodes the numbers 0-9, #, *, and the special characters A, B, C, and D using the frequency specter shown in Table 2-1; however, since the characters A through D are not used by public telecommunications services, these characters are not implemented in the DTMF generation Virtual Peripheral. The DTMF generation is implemented in software by sine generators that use 16-bit phase accumulators in addition to a table index, to produce a sine wave from a table stored in the EEPROM of the SX. A counter in the Interrupt Service Routine (ISR) determines when the index for the sine table shall be incremented. When the index is incremented it jumps to the next entry in the sine table. This is controlled by adding the two variables, `dtmfgFreqCnt` and `dtmfgFreqAcc`, and when the frequency is low the counter increments through the sine table at a slower rate, and when it is high at a higher rate. Only one artificial sine generator is used to generate FSK. To generate DTMF, two sine generators are used to create two tones with different frequencies. These two tones are summed to represent the DTMF signal and the result is located in the `dtmfgPFM0` register.

The Figure 4-1 show the PWM signal for DTMF #1 along with the filtered DTMF output.

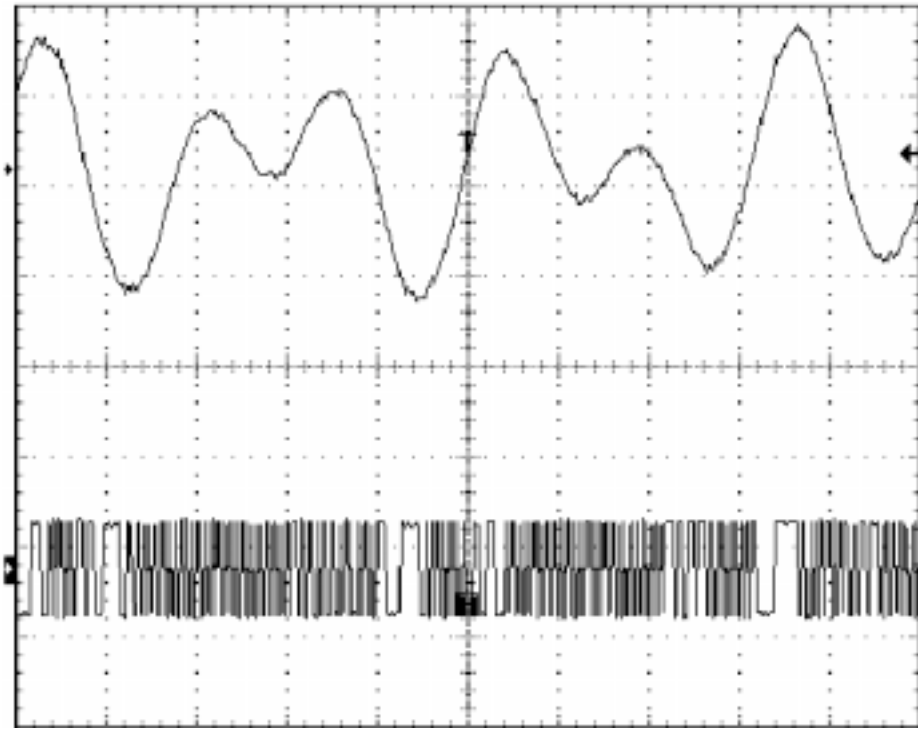

Figure 4-1. PWM Signal (lower part) and Filtered DTMF Output (upper part) for '1' (#1)

4.1.1.1 DTMF Signal Timing

When the DTMF generation dials a sequence of numbers the duration of each character must have a minimum time, for the receiver to detect the character. To meet the Bellcore specification, the signal duration must be ≥40ms.

The Figure 4-2 shows a dial sequence, where the signal duration is 100ms and the interdigit interval is 100ms.

This DTMF generation Virtual Peripheral implementation uses the following signal duration and interdigit interval:

Signal duration	60 ms
Inter-digit interval	50 ms

The timings implemented are within the criteria for the Bellcore specification; however, the signal duration and the interdigit interval can be changed in the implementation, but this can result in less efficient code.

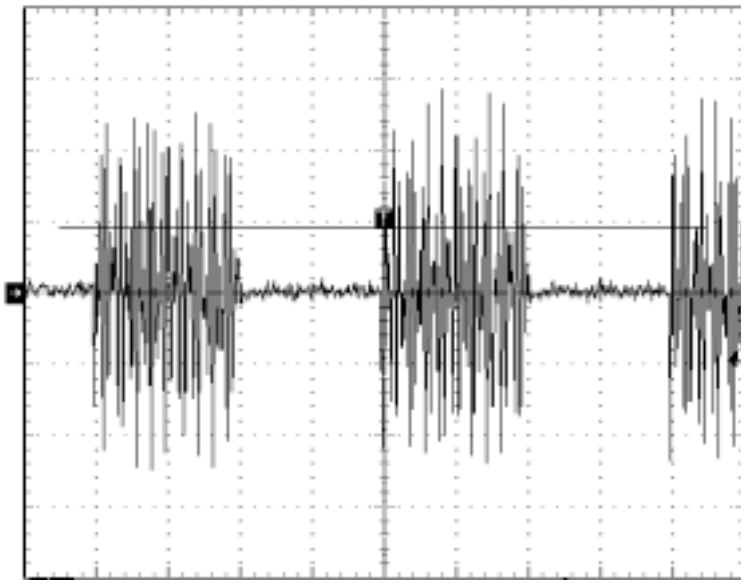

Figure 4-2. DTMF Output (when dialing a sequence of numbers and special characters)

4.1.1.2 DTMF Signal Frequencies

The DTMF generation virtual peripheral is implemented according to the frequencies specified in the Bellcore specifications; however, the generated PWM signal is depending on the frequency, interrupt period, etc., and if one of these settings are changed, the PWM output signal also changes. A change in the PWM signal results in not valid DTMF signals. The signal frequencies generated by the DTMF generation shall be within ±1.5% of the nominal values specified in Table 2-1. The DTMF signal frequencies are verified in section 7.0 (Test Description).

4.1.2 DTMF Signal Power

The steady-state power of DTMF signals during the signal duration interval shall be –7dBm0 ±1dB per frequency when measured with terminating impedance equal to the output impedance required in TR-NWT-000507 specification [3].

4.1.3 Software Implementation

Two artificial sine-generators are used in the DTMF generation Virtual Peripheral. Figure 4-3 shows the flowchart for the program. The sine-generators run in the interrupt service routine, transparent to the mainline routine.

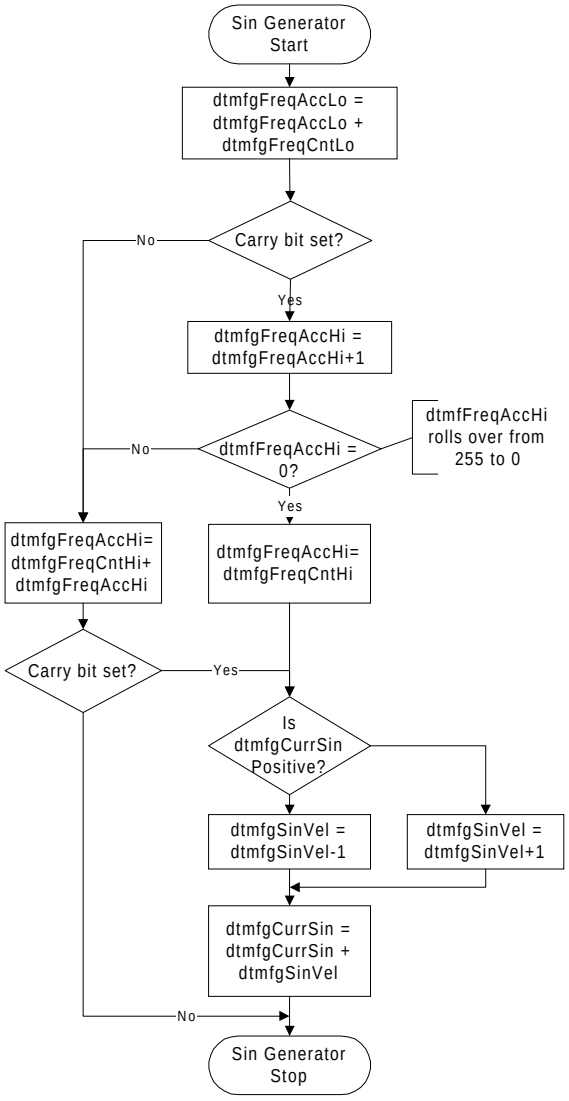

Figure 4-3. DTMF Sine Generation

Figure 4-4 shows the flowchart for the interrupt service routine used in the DTMF generation Virtual Peripheral. This flowchart is actually divided into two threads, one thread performs DTMF generation and the second thread performs both the PWM output and 16-Bit Timer. However, for easier understanding both threads are shown in one flowchart (below).

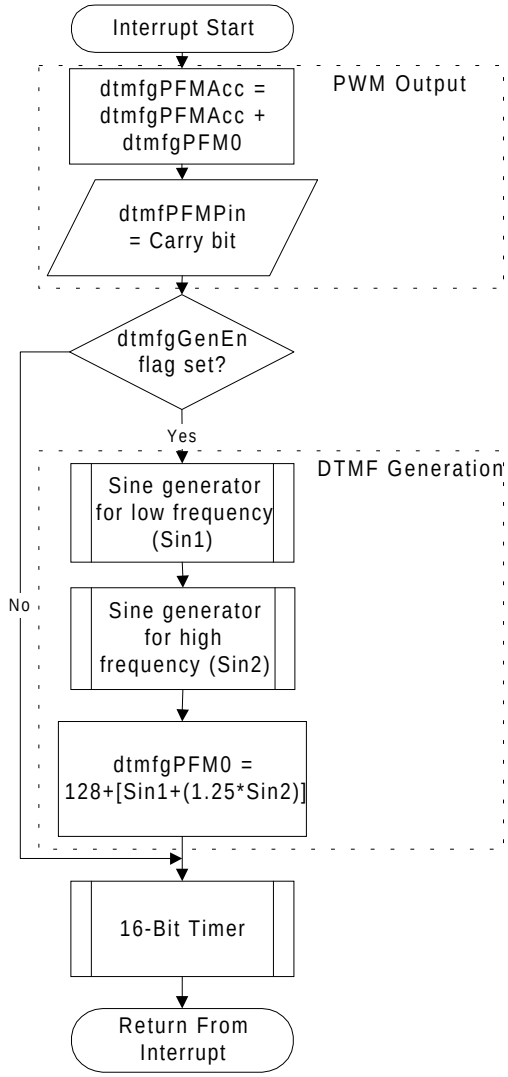

Figure 4-4. Interrupt Service Routine Used in DTMF Generation

DTMF Generation thread starts with a check if the `dtmfgGenEn` (DTMF Generation Enabled) flag is set. If the flag is set then the DTMF signal is generated. The execution of the threads are alternating, first DTMF generation and second PWM/Timer.

This implementation of DTMF generation does not include the character A through D (see Table 2-1. Only the digits 0 through 9 and the characters * and # are used in public telecommunications services. The characters A through D are reserved for use in non-public telecommunications network, and therefore not implemented in the DTMF generation Virtual Peripheral.

4.1.4 Functions

The DTMF generation Virtual Peripheral is implemented using several functions, these functions are described in the following sections:

dtmfDelay10nMs

This subroutine generate a delay in execution of the code, the delay is 10 millisecond and repeated n number of times (where n is the value of 'w' when the function is called).

Input: w – Number of times to delay 10 milliseconds.

Output: None

dtmfDisableOutput

Disables the outputs. Loads DC value into PFM and disable the output switch.

Inputs: None

Outputs: None

dtmfDigit2Index

This subroutine converts a digit from 0 through 9 or a '*' or a '#' to a table lookup index which can be used by the `dtmfLoadFreqs` subroutine. To use this routine, pass it a value in the '`dtmfgByte`' register. No invalid digits are used (A, B, C, or D).

Input: `dtmfgByte` - The ASCII value to be sent

Output: `dtmfgByte` - returns with ASCII `dtmfgByte` or $FF if an error has occurred

dtmfLoadFreqs

This subroutine loads the frequencies using a table lookup approach. The index into the table is passed in the `dtmfgByte` register. The DTMF table must be in the STRINGS_ORG page.

Input: `dtmfgByte` – the index to the lookup table.

Output: `dtmfgByte` – returns with index into DTMF frequency constant lookup table

dtmfDialIt

The subroutine puts out whatever frequencies were loaded for DTMF_TONE_TIME x 10ms, and ends with DTMF_QUIET_TIME x10ms silence.

Input: `dtmfgByte`

Output: `dtmfgByte` – returns with index into DTMF frequency constant lookup table

dtmfInit

Initializes the sinusoid frequencies with correct values (Initialize variables prior to sending a new DTMF character).

Input: None

Output: None

4.1.5 Adjustable Parameters

Except for the frequency constants, there are two other important parameters to adjust for the DTMF Generation Virtual Peripheral.

DTMF_TONE_TIME

This parameter determines the signal duration time for the generated signal. The signal duration can be adjusted in steps of 10ms:

$$T_s = \text{DTMF\_TONE\_TIME} \times 10\text{ms}.$$

DTMF_QUIET_TIME

The interdigit interval can also be adjusted in steps of 10ms:

$$T_i = \text{DTMF\_QUIET\_TIME} \times 10\text{ms}$$

4.2 DTMF Detection Virtual Peripheral

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. In this algorithm, the input signal is digitized and compared to the generated sine and cosine of the 8 DTMF composite frequencies. For each composite frequency, a sine and cosine sum-of-products accumulator register is incremented or decremented depending on whether the input signal matches or does not match its corresponding reference sine or cosine. If a sine or cosine reference signal is found to be present in the input signal, its corresponding accumulator will increase in magnitude. If not, each accumulator will hover around zero.

4.2.1 Software Implementation

The DTMF detection algorithm requires both a sine and a cosine reference signal to be generated for each target frequency. This is because, if only a sine reference was provided, it would be possible to end up with a small magnitude in the accumulator even if the two signals are the same frequency and the input signal was phase-shifted relative to the sine reference. By providing an additional cosine reference and summing the absolute magnitudes in the sine and cosine accumulators, detection of a phase-shifted signal is still possible.

These references are generated or updated on every pass of the DTMF thread in the ISR, and the sampling is done in another thread. To make the comparison between the actual and the reference samples relevant, the sample rate from the A/D must be equal to the reference thread-rate. As the DTMF detection can be setup for any target frequency, one must ensure that the sample frequency is always above Nyquist frequency of $>2f_{in}$.

Sampling:

An analog sample is taken at a sample rate equal to the thread rate (10kHz). Each sample is obtained from 36 (ADNOSAMPL) samples of sigma-delta Analog to Digital conversion on the analog input. These samples are taken with only 10 clock cycles between each sample, which means that it takes 360 cycles (7.2us @50MHz) to obtain a valid sample.

Correlation Algorithm:

The correlation algorithm has the advantage of using a minimal amount of RAM to perform DTMF detection. 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, 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:

$$Amount\_of\_signal = \sqrt{Sin\_Accumulator^2 + Cos\_Accumulator^2}$$

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, only the low (-band) frequencies or the high frequencies are correlated 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 present, 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 generated using 16-bit phase accumulators as an index into a reference table. Each reference wave is multiplied with the input signal, and the result of the multiplication is added to an accumulator. For each frequency, two 16-bits accumulators are required for sine and cosine accumulation.

For easy multiplication, the reference waves only consists of the values 0, 1, 2, -1, and –2 as shown in Figure 4-5. A value of 64_d is added to the high byte of the phase accumulator to obtain the cosine reference, 90° out of phase of the sine reference (360° x 64/256)

Example:

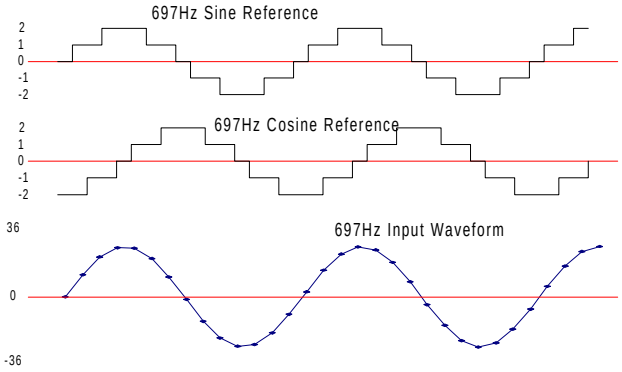

Figure 4-5. Reference Wave Comparison with Input Signal

In this example, the input waveform is the same frequency and phase as the 697Hz sine reference. The result will be a large value in the 697Hz Sine Accumulator, 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.

When a specific number of samples are compared with the input (115 samples for high frequencies, 140 samples for low frequencies), the amount of signal present is calculated using the formula described under Correlation Algorithm.

4.2.1.1 Interrupt Service Routine

The ISR for the DTMF detection program performs these virtual tasks: A/D Converter, UART transmitter, DTMF Detection, LED flasher and a 5ms timer. Because of the Nyquist sample rate and the number of cycles required to perform each DTMF sample, the interrupt rate for the DTMF detection source code (thread rate) has been set to a relatively low 10 kHz. The total length of all the tasks in the DTMF interrupt routine makes it impractical to implement it as one single thread. The Sigma Delta A/D is one of the biggest consumers of program cycles, and it is not possible to divide this routine in smaller separate threads. Thus, this routine will be the one to determine the minimum length of the interrupt period. The next higher value of the interrupt period that makes the ISR run at a multiple of the thread rate, determines the maximum length of the interrupt period. Thus, the interrupt routine for the DTMF is separated into two parts (threads) where the dashed line is inserted in the flow chart. The two threads are approximately equal in length.

UART Transmitter

The detected DTMF digits are sent via the UART. If there are any data present in the UART's transmitter buffer, one bit will be sent on every pass of the ISR.

5ms Timer

The 5ms timer decrements the *Timer5msDiv* register on every pass. When *Timer5msDiv* becomes zero, 5ms has passed, and the *timer5msFlag* is reset the *Timer5msDiv* is set to `TIMER_COUNTER` again ($=50_d$ because the thread-rate is 10.000Hz)

Led Flasher

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

A/D Conversion

Analog to Digital conversion is performed in software, within the interrupt service routine. The analog circuitry required to perform the conversion is shown in Figure 4-7.

A counter adds the number of negative and subtracts the number of positive pulses fed back for a specific number of pulses. The flow chart for the Sigma Delta A/D is shown Figure 4-8.

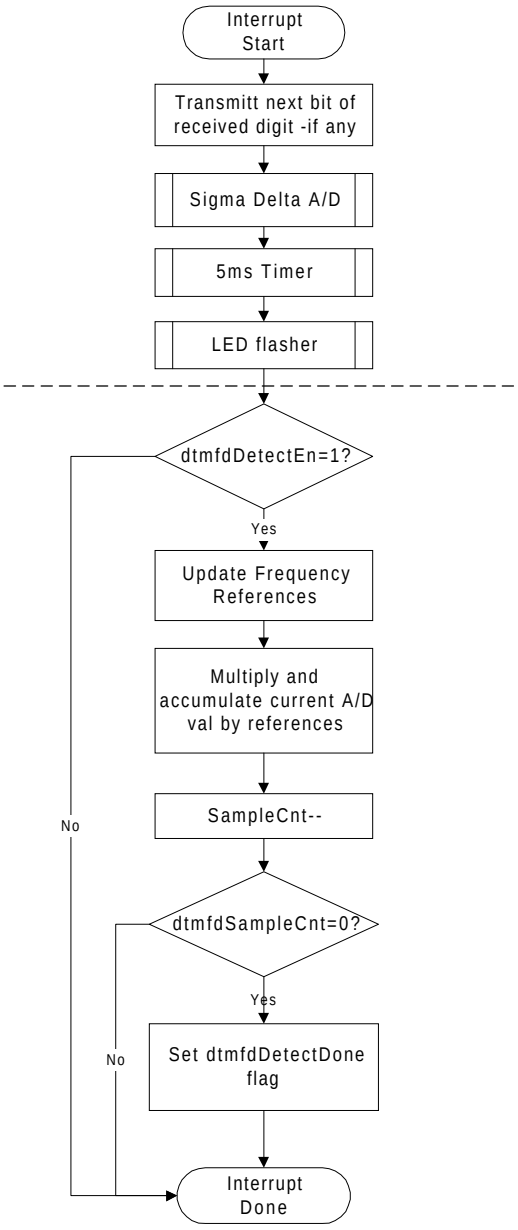

Figure 4-6. ISR Flow Chart

A/D Conversion

Analog to Digital conversion is performed in software, within the interrupt service routine. The analog circuitry required to perform the conversion is shown in Figure 4-7. A counter adds the number of negative and subtracts the number of positive pulses fed back for a specific number of pulses. The flow chart for the Sigma Delta A/D is shown Figure 4-8.

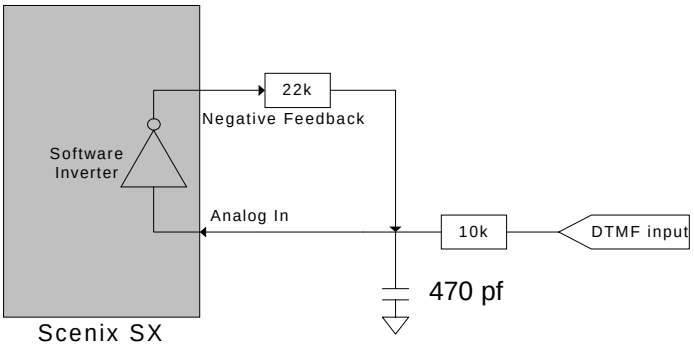

Figure 4-7. Software Sigma Delta A/D Converter

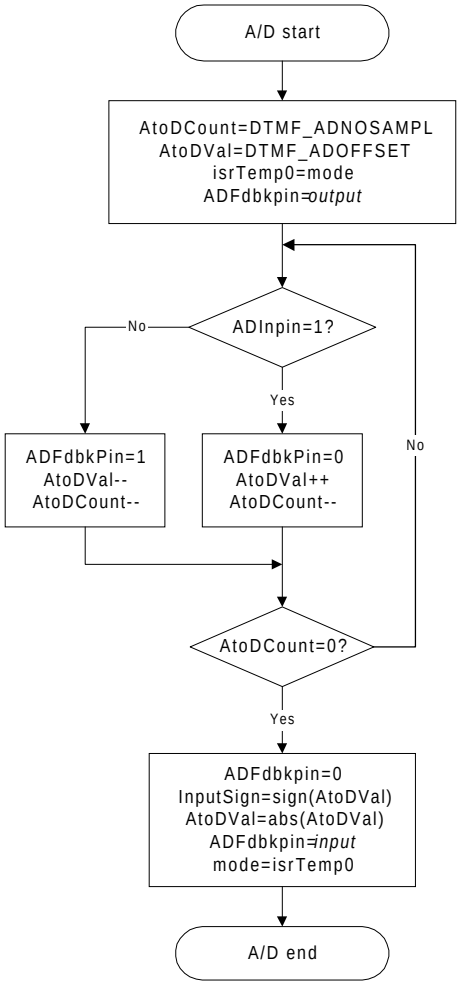

Figure 4-8. A/D Sample Routine in Interrupt

Updating the Frequency References

The interrupt service routine generates the reference samples of four simultaneous frequencies internally to multiply them with the input signal samples. 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°.

Multiplying the Input by the References

The value in the reference accumulators is used to generate both a sine and cosine signal. For the cosine reference, we only need to phase shift the sine reference 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, adding 0x40 to the MSB of the reference sine phase accumulator, results in a cosine phase value.

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.

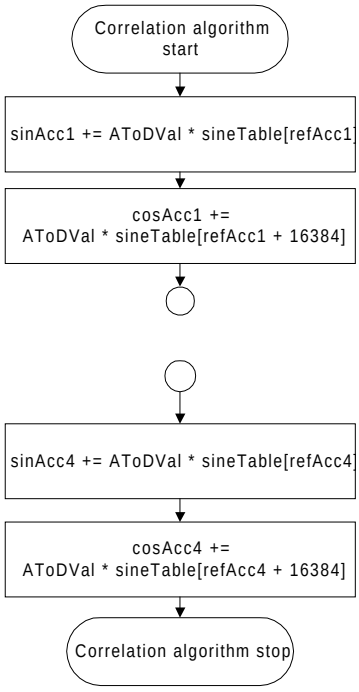

Figure 4-9. Correlation Algorithm

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 phase accumulator is = 1, the reference value is negative. The routine also checks the sign of the sampled value to make sure that the product of the multiplication gets the correct sign.

When the routine has done this correlation for the number of times defined by `DTMF_SAMPL_LO` (for low frequency group) or `DTMF_SAMPL_HI` (for high frequency group), the `dtmfdDetectDone` flag is set to tell the main program that a correlation value is calculated.

4.2.1.2 Main Program

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 to the terminal screen.

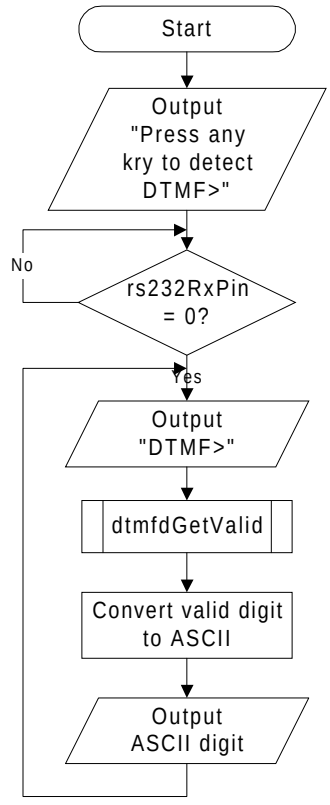

Figure 4-10. Main Program

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 theoretically minimum time to detect a valid DTMF digit is 37ms + 34ms of silence, but other combinations may also succeed. I.e. 43ms + 28ms of silence (7ms of a tone can be detected as silence). See flowchart (Figure 4-11) for timing.

As can be seen from the flowchart, the only way to get a valid digit is to detect one valid high frequency twice, then a low frequency followed by silence. Also, if the difference in power level between the high and the low frequency differs too much, the detection will fail and start all over from top.

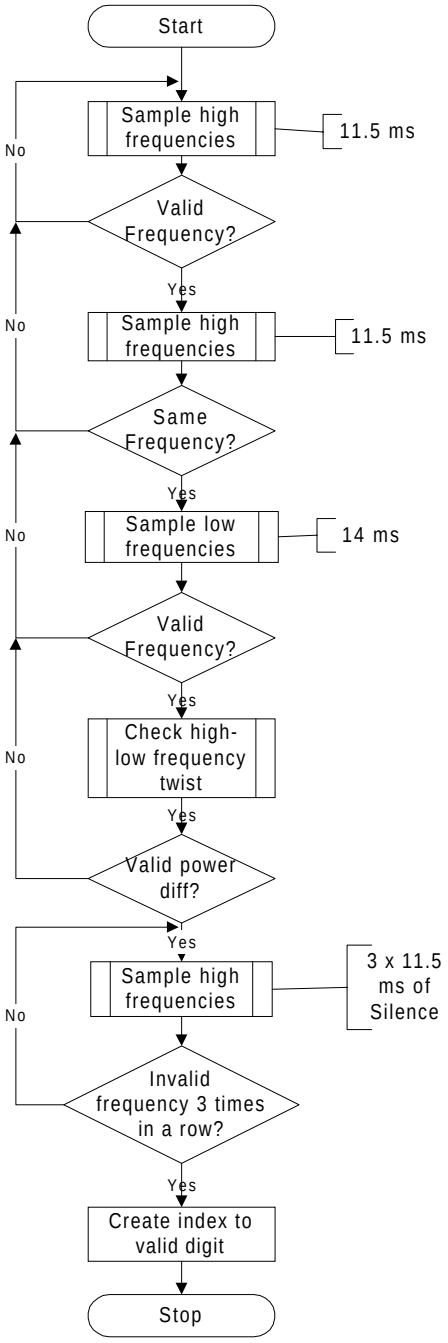

Figure 4-11. Flowchart of the dtmfdGetValid Subroutine

dtmfdSample:

The `dtmfdSample` routine clears all of the sine and cosine accumulators, enable the DTMF detection and waits for a DTMF sample to take place. Before calling `dtmfdSample`, the `dtmfdSampleLows` bit must be set up. A '1' in `dtmfdSampleLows` will force it to sample low frequencies, and a '0' will force it to sample high frequencies. After an accumulation has finished, `dtmfdSample` 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,#dtmfdSinAcc1Lo    ; do the cal-
                             culations on acc1
page   dtmfDetectCalcs
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 `DTMF_SCALING` number of LSB's from the value, to reduce it to 8 bits value. `DTMF_SCALING` 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.

4.2.2 Adjustable Parameters

Except for the frequency constants, there are some other important parameters to adjust for the DTMF Detection Virtual Peripheral.

DTMF_SCALING

This constant determines how many bits the accumulated correlation results (16-bits) shall be shifted to the right to squeeze them into 8 bits. Decreasing this value increases the sensitivity to smaller signals. Be aware that this value scales the correlation result that the DTMF_MIN_SCORE is compared with. I.e. increasing the DTMF_SCALING by one is “approximately” the same as doubling the DTMF_MIN_SCORE, and visa versa.

DTMF_ADOFFSET

This constant is a correction value for ADC and is meant to adjust for DC-offset on the input pin (offset from 2.5volt); when there is no input signal, the sampled A/D value (dtmfdAtoDVal) measured is should be zero.

DTMF_ADNOSAMPL

This constant determines how many samples to do of the input signal. Higher values take more time to process, and the minimum time to detect a valid digit will increase. On the other hand, if this value is decreased further, the dynamic range of the A/D will decrease (see section 4.2.3).

DTMF_MIN_SCORE

This is the minimum score required to detect a digit. However, a valid digit requires the measured value in ResultN (accumulated correlation value) to be twice this value;

$$ResultN > 2 \times DTMF\_MIN\_SCORE$$

DTMF_SAMPL_LO

This is the number of samples to take to detect a low frequency. Increasing this value increases the minimum time to detect a digit.

DTMF_SAMPL_HI

This is the number of samples to take to detect a high frequency. Increasing this value increases the minimum time to detect a digit.

4.2.3 Special Considerations

The A/D network on the Ubicom Modem board, shown in Figure 4-7, is not optimal for the Sigma Delta A/D implemented in the DTMF Detection @50MHz. The theoretically maximum input range for the A/D is 2.5 ±2.5V. If the input voltage is at the limits, the feedback voltage will never be able to force the capacitor (input) voltage to 2.5V because of the differences between the input and the feedback resistor.

The time available for 36 samples to reach the max/min input value is 7.2us @50MHz, and the smoothing capacitor is 470pF (also, additional capacitance between the traces on the board increases the total capacitance). The time constant (t = RC, time constant where 63% of the voltage is achieved) for the feedback network alone is 10.3us, much more than the time available. This means that the feedback cannot fully charge/discharge this capacitor during the sample time. A smaller feedback resistor will increase the performance (input range and speed), but this should only be combined with a high quality capacitor (SMD and high quality material for high frequencies) and improved board layout.

4.3 Hardware

Certain basic isolation circuitry is required to properly interface to the telephone network. There are many possible variations in requirements based on area and telephone network providers, so check with your network provider first. The ADC used in DTMF detection requires some basic filter and feedback circuitry outside of the SX, and we have used the Ubicom Modem board rev. 1.2 for to simplify the testing, and this is shown in Figure 4-7.

The output from the DTMF generation Virtual Peripheral is a Pulse Width Modulated signal (PWM), but with a minimum of additional hardware this signal can be transformed into a DTMF signal.

4.4 Summary of The Ubicom DTMF Solution

Proven in several production designs, this DTMF detection algorithm is ideal for low-cost embedded systems where the addition of a separate DTMF generation or detection IC is not feasible from a price standpoint, and the algorithm must be embedded in the host SX. ATM, vending machines, answering machines and security systems are a good example for 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 (www.ubicom.com).

5.0 Specifications

The DTMF detection and generation has completely different operations and therefore the specification for the different DTMF functions (detection and generation) are different.

5.1 DTMF Generation

Operational mode:	Generation of DTMF tones
Clock speed:	50 MHz
MIPS usage:	34MIPS
Max cycles in each thread:	73
Flash usage:	
RAM usage:	16 bytes in bank 1, 3 byte in bank 2, 2 bit global temp variable
Pin usage:	2 pins, DTMF PWM and LED output
RTCC setting:	Timer interrupt running every 1.62us ‡ ISR rate = 617kHz.
Ubicom mnemonics:	Yes
Multithreaded:	Yes
Thread Rate:	2/2
Bellcore Compliance:	Yes
Generator Frequency Tolerance:	≤ 0.6% of nominal DTMF tone specification
Signal duration:	58 ms
Inter digit interval:	50 ms

5.2 DTMF Detection

Operational mode:	Detection of DTMF tones,
Clock speed:	50 MHz o
MIPS usage:	10 MIPSo
Max cycles in each thread:	483 (510 for SX 48/52),
Sample rate:	10kHz
Resolution on A/D:	72 values
Flash usage:	
RAM usage:	7 bytes in bank 1 14 bytes in bank 5 16 bytes in bank 6 10 bytes in bank 7 6 bits global flags variable Total: 47 bytes and 6 bits SX
Pin usage:	2 pins, A/D feedback and A/D output. Additional pins in use for demo
RTCC setting:	Timer interrupt running every 10ìs for 100.000kHz speed
Ubicom mnemonics:	Yes
Multithreaded:	Yes
Thread Rate:	2/10
Bellcore Compliance:	Yes, with exceptions: - Frequency rejection - Twist
Frequency Tolerance – Operation:	≤ 1.5% of nominal DTMF tone specification
Frequency Tolerance – No Operation:	≥ 4% of nominal DTMF tone specification
Power Level per Frequency – Operation:	0 dBm to –25 dBm (see Note 1)

Power Level per Frequency – No Operation:	≤ -55 dBm
Power Level Difference (Twist):	+3 dB
Signal Duration Limit – Operation:	> 45 ms
Signal Duration Limit – No Operation:	< 28 ms
Interdigit Interval:	> 29 ms
Cycle Time:	> 74 ms/digit
Two Buttons simultaneously:	No Detect if < 45 ms apart
Single DTMF Tone	No Detect

Note 1: How, this level was determined, see section 7.2.4.2

6.0 DTMF Demo Description

The (original) source code for DTMF detection and generation is set to match the settings on the Ubicom modem (demo board). To run a demo of either the detection or generation Virtual Peripheral you need this equipment:

- Ubicom Modem demo board
- PC with programming equipment, and a HyperTerminal (9600bps, N, 8, 1) for DTMF Detection.
- Analog telephone line and a telephone (or using the DTMF generator to generate DTMF tones)

Note! If you use the DTMF generation instead of an analog telephone you need two Ubicom modem demo board.

Before any demo can take place, the line “DEMO” in the beginning of the code must be uncommented. Correct part (SX18–SX52) and assembler (SX_key) must also be uncommented.

The DTMF demos can be run on your own analog telephone line, or on a phone line simulator (“Ring It”).

6.1 DTMF Generation Demo

This demo describes how you can test the DTMF Generation “at home” on an analog telephone line.

1. Replace the default number in the code (210-1500) with the phone number you want to dial up for the test (_dialString), for instance your own cellular phone. Make sure that the string ends with a zero (‘numbers’,0) to terminate the dial string (null = string termination).
2. Connect the debugger/programmer and power up the modem board.
3. Connect the modem demo board to your analog phone line and run the code.
4. Run the generation code on the modem board, and verify that the number you put in the dial string actually are called.

If you also have a telephone connected to the same line, it is possible to hear the DTMF signals, dialed by the modem board, in the speaker. This makes it possible to let the modem board (generation) dial the number, and the phone (i.e. your cellular phone) to verify that the call was a success.

6.2 DTMF Detection Demo

The DTMF detection needs some minor modifications to bypass the filters on the Modem Demo board. To bypass the LP-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. To disable the HP-filter, the jumper “JP6” (upper one of the four to the right of the SX in the figure below) must be inserted. This jumper connects the output from the filter, to the “cntrl_3” pin on the SX that pulls the signal to ground.

The consequence of bypassing these filters is an increased gain from the input to the SX chip of approximately 13dB. This means that we cannot assume that the required signal levels on the input are valid for this modified board.

This demo describes how you can run DTMF Detection on your analog telephone line, together with a telephone connected to the same line or on another line (number). It is possible to use a cellular phone to call the line the modem board is connected to.

1. Setup your HyperTerminal (9600bps, N, 8, 1) and connect the modem board to your PC with a serial cable.
2. Connect the debugger/programmer and power up the modem board.
3. Connect the modem demo board to your analog phone line and run the code.
4. From another telephone (line), call the number the modem board is connected to.
5. Run the detection code on the modem board.
6. Once the modem board is being called (you can hear the dial tone in the calling phone), press a key in the HyperTerminal window to start the demo.
7. From the calling phone, verify that the modem board has “answered” the call (the ring tone is replaced by silence).
8. From the caller phone, press some digits and see that the digits pressed are transferred to the HyperTerminal window.

If you don’t have two telephone lines available, you may connect both the modem board and a phone to the same line. Follow the steps 1-3 and 5 above:

1. Press a key in the terminal window.
2. From the phone connected to the line, press some digits and verify that the digits pressed are displayed in the terminal window. Note that you actually dial the number you press on the phone. Thus, stop dialing and the program before you have completed a valid phone number, or use a number where the recipient is aware of the call.

Figure 6-1. DTMF HW Modification for the Modem Board

7.0 Test Description

The Virtual Peripheral is tested and verified. This section describes the test environment, a description of how the test was performed and the results. For the user of the Virtual Peripheral this section is an analysis of the Virtual Peripheral's functionality and integration.

7.1 Test Environment

The equipment used when testing both the DTMF detection and generation Virtual Peripherals are listed below:

Oscilloscope	:	Tektronix TDS 3014 (100MHz)
Demo board(s)ý_	:	Ubicom Modem (version 1.2)
Devices:	:	Ring-It! (Telephone line simulator) 1 Analog telephone PC with HyperTerminal software PC with "Cool Edit 2000" software
SX Key assembler	:	Tested with SX 18/28 Key version 1.09 rev E/F Tested with SX 48/52 Key version 1.19 (only programmed and run, no functional test)
SASM assembler	:	Tested with SASM version 1.45.4
Debugger	:	Parallax SX Key rev. E
External Components	:	N/A

7.2 Functional Test Description

Both Virtual Peripherals are tested on the Ubicom modem demo board. However, before testing DTMF detection, some minor changes were applied to the board. The hardware changes to the demo board are shown in Figure 6-1. In addition to this strapping, jumper 'JP 6' was also inserted (see section 6.0 for more information).

Before starting to test either of the Virtual Peripheral, we defined a set of scenarios to perform:

1. DTMF generation — Frequency analysis (using oscilloscope)
2. DTMF generation — Tester (Ring-It!)
3. DTMF detection — Analog telephone HyperTerminal
4. DTMF detection — PC (Cool Edit 2000) HyperTerminal
5. DTMF detection according to the Bellcore specification
6. Test pin and port configuration for the DTMF Virtual Peripheral.

7.2.1 DTMF Generation --- Frequency Analysis (using oscilloscope)

The purpose of this test was to analyze the frequencies generated by the DTMF Virtual Peripheral.

1. DTMF generation code modified for an "infinite" single digit tone to simplify the analysis.
2. Inserted one single digit in the dial-string, loaded and ran the program on the Ubicom modem demo board (SX 28 part only)
3. Analyzed the frequencies (FFT) in the generated signal measured after the OP-amp in the PPM Filtering.
4. Steps 2 and 3 above were repeated for the digits '0'-'9' and '*' and '#'.

The Bellcore specification [2] also contains some criteria that the DTMF generation has to fulfill to generate valid DTMF signals. These criteria's are frequencies, signal duration and interdigit interval.

The oscilloscope pictures captured during this test are shown in Appendix A.

7.2.1.1 Frequencies

The Frequency components are approximately 0.6% higher than the frequencies listed in Table 2-1; however, this is within the frequency range that the Bellcore specification defines as a valid DTMF signal, and also within the accuracy of FFT module in the oscilloscope.

7.2.1.2 Signal Timing

The timing measured during the test was compared to the Bellcore specification; the result is shown in the Table 7-1.

The timing listed in the table is visualized in the oscilloscope pictures shown in Appendix A.

Table 7-1. Test Result - Timing with Default Settings

Description	Test result	Bellcore
Signal duration Duration of a DTMF signal generated.	58 ms	≥ 40ms
Signal rise time	< 3 ms	≤ 5 ms
Signal fall time	< 3 ms	≤ 5 ms
Inter-digit interval	» 50 ms	≥ 40 ms
Cycle time	» 109 ms	≥ 93 ms

7.2.1.3 Summary

The result of this test shows that the DTMF generation frequencies deviates from the ideal values specified, but are within the criteria's of the Bellcore specification. The signal timing is also in accordance to the Bellcore specification.

7.2.2 DTMF Generation --- Tester (Ring-It!)

When the signal was analyzed to be correct, the phone line simulator "Ring-It" tester was used to verify correct DTMF generation. The signal duration was extended to make it possible to see the digits flashing by on the "Ring-It" tester.

1. All the valid digits (0-9, * and #) were inserted in the dial string.
2. The modem board was connected to the "Ring-It" tester, and the modified code was loaded into the SX28.
3. With the "Ring-It!" the inserted dial string were detected correct.

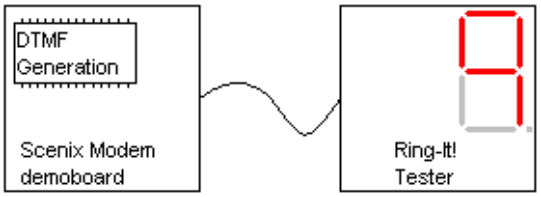

Figure 7-1. DTMF Generation - "Ring-It!"

7.2.3 DTMF Detection --- Analog Telephone --- HyperTerminal

The DTMF Detection contains an RS232 interface that makes it possible to send (echo) the digit detected to the Hyper Terminal via the COM port.

1. The Ubicom modem demo board and an analog telephone were connected via the "Ring-It" telephone line simulator.
2. HyperTerminal was connected and set up to communicate to the DTMF Detection program (running on Ubicom modem demo board).
3. The DTMF generation code was loaded and run on the SX 28 part (Ubicom modem demo board).
4. Valid telephone network numbers and special characters were dialed with the telephone.
5. The DTMF detection detects the numbers and special characters, and echoes them to the HyperTerminal window.

First all the numbers on the telephone keypad were pressed, and all the digits were echoed to the terminal window. Then, one key at the time was held down to verify that the DTMF detection did not detect any digit before the key were released. Both tests were performed with correct behavior and result.

7.2.4 DTMF Detection --- PC (Cool Edit 2000) --- HyperTerminal

The "Cool Edit 2000" software can generate DTMF signals, with several parameters such as signal duration, interdigit interval, twist, amplitude and frequency deviation. In the following sections there are referred to parts of the Bellcore specification [2]. This is only meant to see the test result in accordance to the specification.

7.2.4.1 Values to Detect a Valid Frequency

The following test was performed to get a picture of what basis the dtmfGetWinner routine determines if a frequency is valid or not. Table 7-2 and Figure 4-4 show the results available for determining the low and the high frequency group. The R1 - R4 are the dtmfdResult1-dtmfdResult4 values for the frequencies in the current group. These were obtained by stopping the program at the dtmfdWinnIndex routine when the dtmfdSampleLows flag are set. The input signal was generated in the Cool Edit 2000 program for each digit in the table. The level of the input signal was set to the maximum value where clipping occurs (see Figure 7-2)

The values at the left shows the results when the frequencies applied are exact equal to the specification, and the ones to the right are measured at +3.5% deviation of the original frequency. The numbers in the tables are approximately values obtained by averaging 3-5 samples for each frequency. The result for the actual frequency is highlighted in the table.

The same test was performed both for the low group and the high group frequencies and is shown in Table 7-3.

I.e., when the digit '1' is pressed and the correlated values are calculated, we can see from the correlation results for the high frequencies (133, 30, 4 and 4 from Table 7-3) that the 1209Hz frequency (R1) has much more energy than the other frequencies.

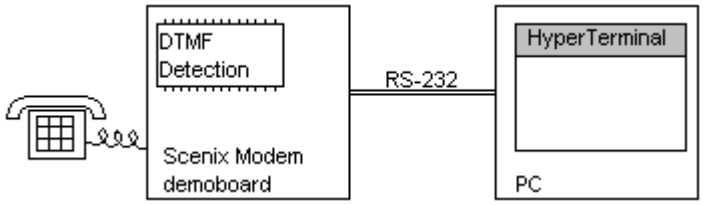

Figure 7-2. DTMP Detection

Table 7-2. Results used to Determine if the Measured Lo-Frequencies are Valid or Not

Frequency deviation: 0%						Frequency deviation: 3.5%				
Digit	R1	R2	R3	R4		Digit	R1	R2	R3	R4
1	123	6	7	11		1	97	43	17	13
2	115	7	10	13		2	85	51	16	7
3	118	14	6	12		3	110	42	7	10
4	8	127	14	17		4	29	103	26	10
5	4	124	16	20		5	30	103	31	9
6	16	128	16	9		6	22	102	35	12
7	8	12	124	4		7	19	28	99	19
8	9	14	128	26		8	18	26	94	27
9	7	18	138	35		9	13	26	95	30
0	7	11	17	130		0	10	14	30	95
*	31	24	24	121		*	14	7	9	74
#	14	21	21	132		#	6	14	23	94
A	130	3	10	18		A	105	45	14	13
B	6	132	28	14		B	29	99	28	3
C	17	31	137	21		C	12	23	106	33
D	29	17	22	139		D	11	7	19	95

Table 7-3. Results used to Determine if the Measured Hi-Frequencies are Valid or Not

Frequency deviation: 0%						Frequency deviation: 3.5%				
Digit	R1	R2	R3	R4		Digit	R1	R2	R3	R4
1	133	30	4	4		1	90	5	15	10
2	28	130	21	10		2	5	68	10	10
3	5	20	127	16		3	15	10	65	20
4	135	27	6	15		4	90	8	12	10
5	27	128	22	10		5	8	73	8	8
6	5	22	123	18		6	15	10	62	15
7	131	25	5	6		7	84	7	18	17
8	22	128	22	11		8	10	72	8	9
9	10	20	120	12		9	17	12	61	20
0	22	128	25	15		0	10	75	10	13
*	134	27	17	3		*	86	5	17	10
#	10	20	123	15		#	13	12	59	15
A	5	10	12	116		A	5	5	17	48
B	5	9	13	117		B	7	5	14	49
C	8	10	12	116		C	14	6	14	49
D	6	7	13	115		D	7	7	13	48

7.2.4.2 Input Signal Level Range

The DTMF detection Virtual Peripheral shall detect DTMF signals with signal level between 0dB and -25dB. Between -25dB and -55dB DTMF signals can either be accepted or rejected, but below -55dB it shall be rejected.

The purpose of this test was to determine the input voltage range the DTMF detection accepts. A signal duration of 115ms (Ts=75ms and Ti=40ms) and twist=3dB was used. The reason for doing this is because of the increase in signal gain after bypassing of the LP- and HP-filters described in section 6.2.

	$V_{min\ p\text{-}p}$	$V_{max\ p\text{-}p}$
Input voltage*)	102 mV	1.5V

*) The input voltage was measured at the transformer (T1) pin 4 on the "inside".

The signal into the A/D converter (measured at "P1" on U7C pin 8) and the signal out of the transformer (pin 4 on T1) is shown in Figure 7-3 and Figure 7-4 (channel 1 measured at pin 4 on T1, and channel 2 is measured at U7C pin 8). As can bee seen, the signal into the A/D is at a maximum even for the minimum input signal for detection (102mV). The reason for this high gain is explained in section 6.2.

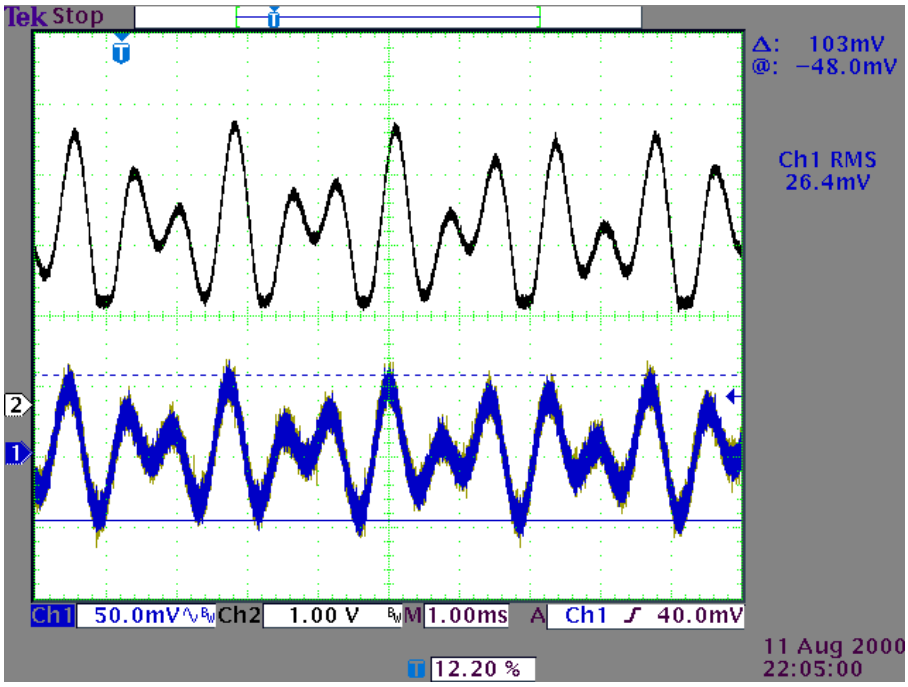

Figure 7-3. Input Signal (ch1) = 102mV p-p When the Input Signal to the A/D (ch2) Starts Clipping

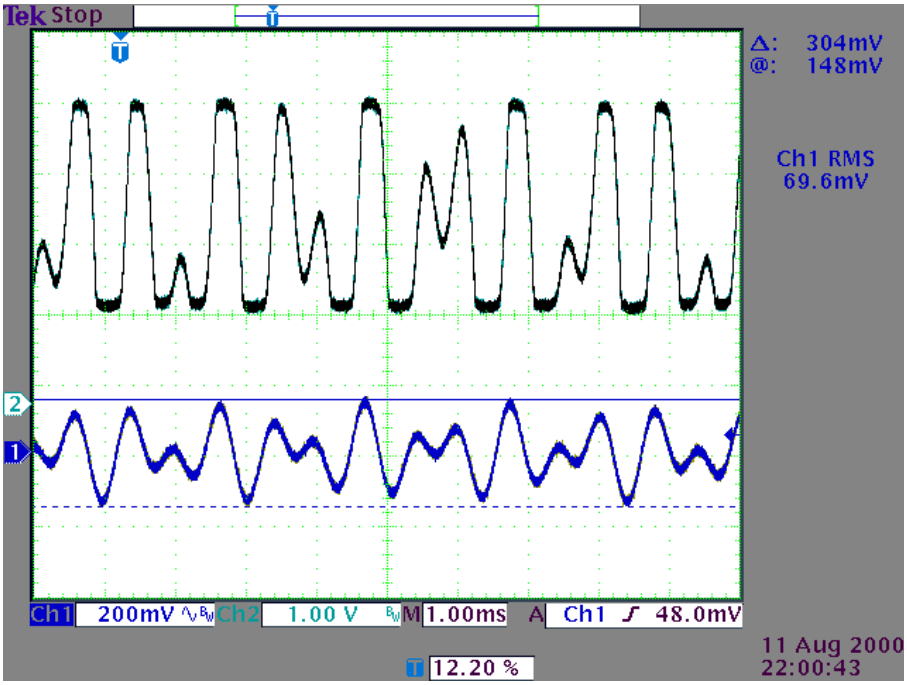

Figure 7-4. Input Signal (ch1)= 300mV p-p Results in a Clipped Signal to the A/D (ch2)

The measured signal levels needed to be calculated for a complete verification of the signal level:

- Minimum signal level in for detection with no error was measured to 102mV p-p. Thus, this level is defined as -25dBm (minimum level according to specification).
- Maximum signal level in for detection with no error was measured to 1.5V p-p, this was calculated to -1.6dBm.

7.2.4.3 Signal Timing

The signal timing affects how accurate the DTMF detection Virtual Peripheral detects the DTMF signals. With perfect frequencies (no frequency deviation), signal level of 300mV p-p (-15dBm according to minimum level detected) and 3dB twist the minimum signal timing is:

Signal duration Ts = 46 ms @ Ti=40ms.

Inter-digit interval Ti = 29 ms @ Ts=50ms.

When both variables was set to this minimum length, the following result was obtained:

Digit generated	Errors
12.000	1

7.2.4.4 Frequencies

The DTMF detection shall detect all DTMF signals where the frequencies are within -1.5% and +1.5% from the ideal values (see Table 2-1). Frequencies between the ±1.5% and ±3.5% limit are uncertain, they can be detected or rejected, but all frequencies outside the -3.5% and +3.5% interval shall be rejected.

However, some measurement showed that these limits were not met completely, so these tests show the limits for detection and rejection:

± 1.5% deviation

Ts = 110 ms

Ti = 40 ms

Twist = 3 dB

All DTMF signals shall be detected:

#	Generated DTMF digits	Errors
1	12.000	1

The requirement for rejecting digits when the frequency deviated more than 3.5% was not met. Instead, a deviation of 4% gave the following results:

± 4% deviation

Ts = 110 ms

Ti = 40 ms

Twist = 3 dB

The input voltage is set to 300mV p-p as measured in section 7.2.4.2.

#	Generated DTMF digits	Errors
1	2500	1
2	2500	1
3	2500	1

The digits detected were the '4' (2) and the '7' (1).

7.2.4.5 Summary

Before making a summary of the test it must be mentioned that the hardware (modified demo board) may have had a great influence on the test results as non-linear distortion.

Non-linear distortion produces extraneous frequencies in DTMF signals that can cause degradation of the DTMF receiver (detection) performance.

The DTMF detection algorithm, dtmfGetWinner, is very level dependent. Also, the op-amp on the modem board seems to cut-off the input signal when the voltage on the output when the voltage goes below 1.4V or above 4.2V. This will result in a non-linear input to the ADC used for DTMF detection. The reason why we chose to use this non-linear operation area of the op-amp, it was because of the greater input signal range without influence of the dtmfGetWinner level dependency (also see section 4.2.3).

7.2.5 DTMF Detection According to the Bellcore Specification

Testing DTMF detection according to the Bellcore specifications requires play back of the Series-1 Simulation Test Tapes for DTMF Receivers that consists of six half-hour sequences of speech samples. These six parts are used as a standard test source for measuring the speech-rejection capabilities of DTMF receivers in telecommunication system. None of the 29-minute sequences are duplicated. There are over 50.000 speech samples, including some music on the tapes. The samples range from about 50 to 500ms in duration and are separated from each other by quiet intervals of at least 50ms. It is estimated that the six parts of the Series-1 tapes are equivalent to the exposure of one million customer-dialing attempts in a local central office.

When all the digits are used, the DTMF detector shall not detect more than 666 valid DTMF digits.

1. The test tapes were sampled and stored as wave-files (16bit, 44.1kHz).
2. The sound card on the PC was connected to the microphone input on a telephone.
3. The HyperTerminal were set up to communicate to the DTMF Detection program.
4. The modem board was programmed with the latest code and run.
5. First sound track was played and adjusted for correct tone level (-20dBm set to 180mV p-p measured pin 4 T1).
6. All six sound tracks were played repeatedly only separated by a known string to get the results for each run.
7. After three runs, the tapes are stopped and the number of hits is counted.

#	Detected DTMF digits
1	1580
2	1594
3	1545

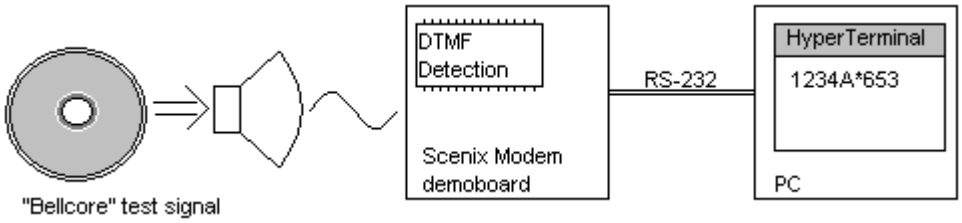

Figure 7-5. Testing DTMF Detection According to Bellcore Specification

7.2.6 Summary

The most important test for these Virtual Peripherals was to test in accordance to the Bellcore specification. The result shows that the DTMF detection does not fulfill all the requirements in the Bellcore specification; however, the result may be affected by the hardware.

These results reveals that the DTMF detector does not fulfill all the requirements in the Bellcore specification; due to deviation from the frequency rejection requirements, the DTMF detection Virtual Peripheral exceeds the limit for detected DTMF digits when testing against the Bellcore tapes. We believe further improvements in the dtmfGetWinner routine can solve this problem.

Both the DTMF generation and DTMF detection are tested on the "Ubicom modem demo board" and therefore not on the SX 52 part.

8.0 References

This section includes the list of several reference documents.

Table 8-1. Reference Table

Ref	Title	Author(s)	Date
1	Virtual Peripheral guide v. 1.04 (Template for VP development)	Ubicom	June 2000
2	GR-506-CORE (Generic Requirement)	Bell Communication Research, Inc. Further information: Ralph Battista 331 Newman Springs Road Red Bank, New Jersey 07701 (201) 758-5075	1987
3	TR-NWT-000507 (Part of the LSSGR)	(LSSGR) LATA Switching Systems Generic Requirements, FR-NWT-000064.	1987

9.0 Appendix A

9.0.1 Test Results DTMF Generation

9.0.1.1 DTMF Generation Spectrum Analysis (15.4.3-15.4.5)

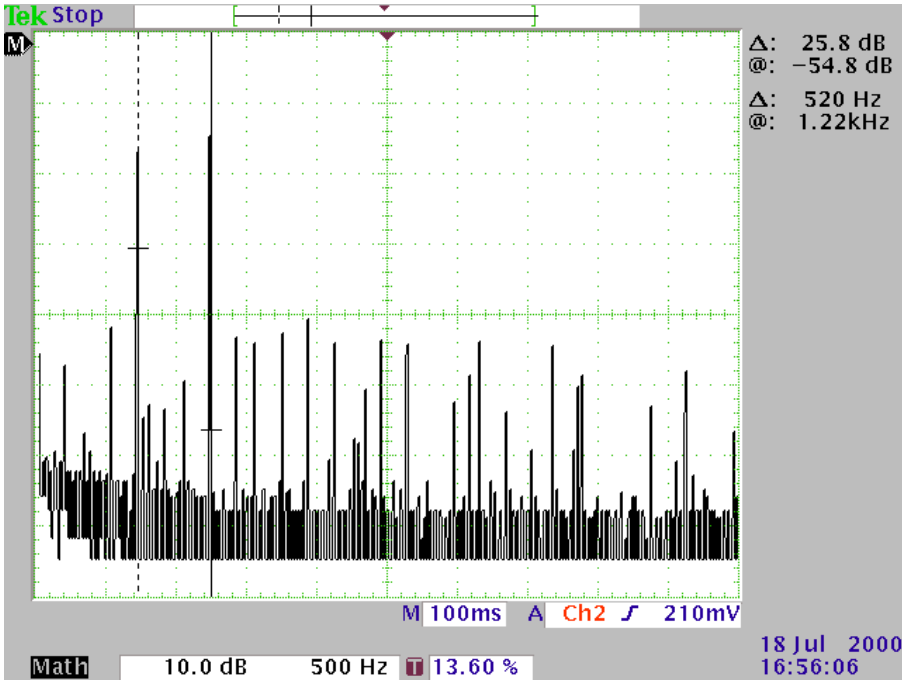

Figure 9-1. The '1' Digit with fL=701Hz and fH=1216Hz. Level Difference to Nearest Noise Component is 24dB (f=1.9kHz)

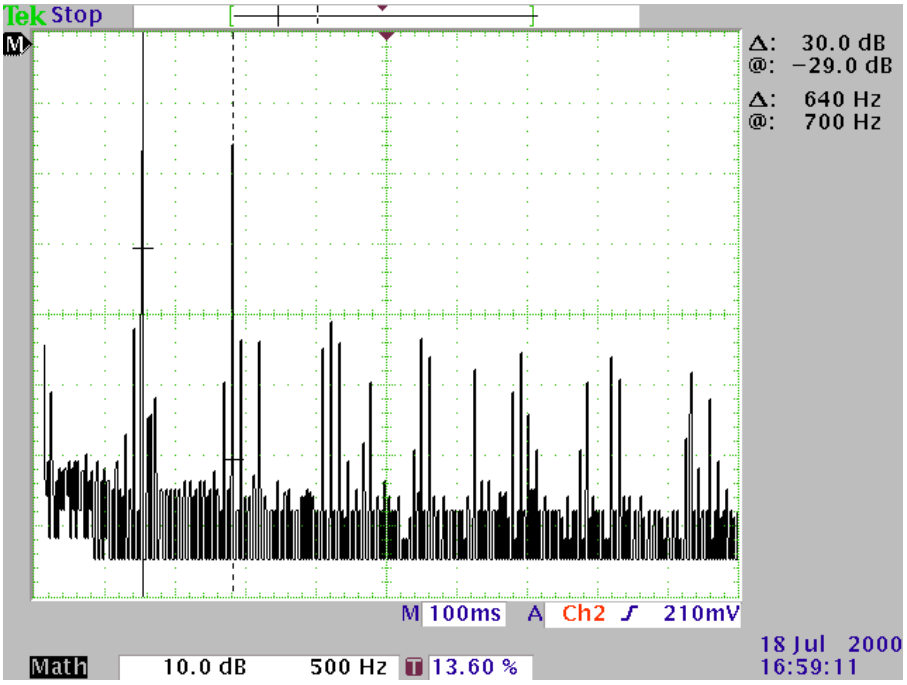

Figure 9-2. The '2' Digit with fL=701Hz and fH=1344Hz. Level Difference to Nearest Noise |Component is 24dB (f=2.05kHz)

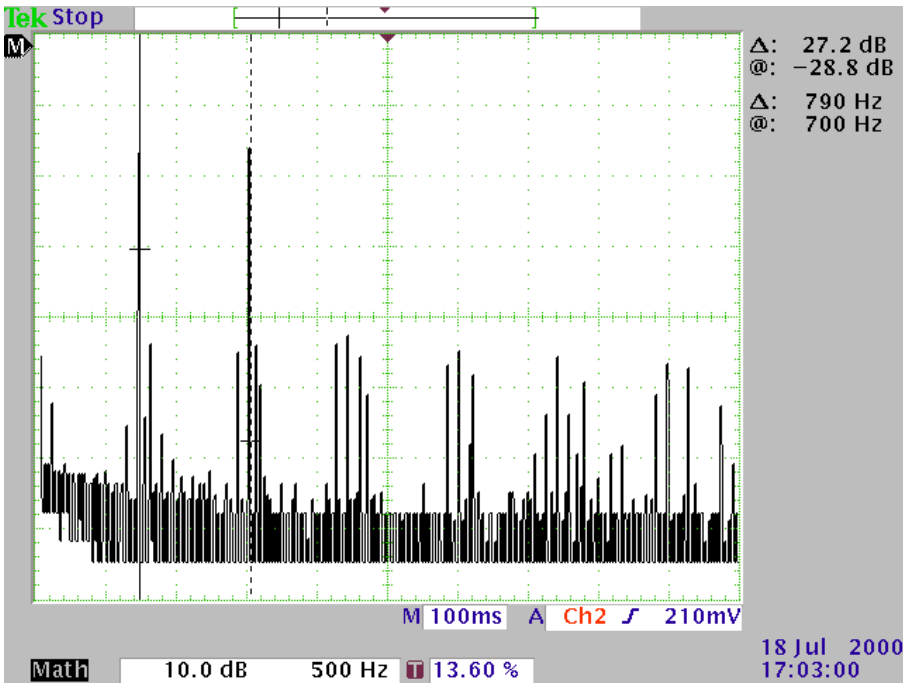

Figure 9-3. The '3' Digit with fL=701Hz and fH=1486Hz. Level Difference to Nearest Noise Component is 26dB (f=2.19kHz)

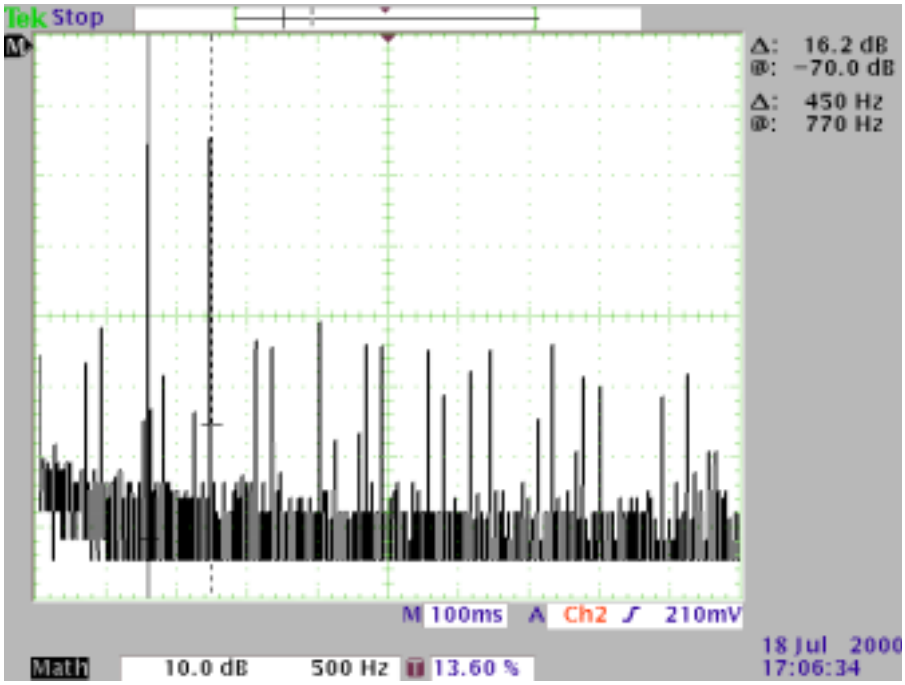

Figure 9-4. The '4' Digit with fL=775Hz and fH=1217Hz. Level Difference to Nearest Noise Component is 25dB (f=1.99kHz)

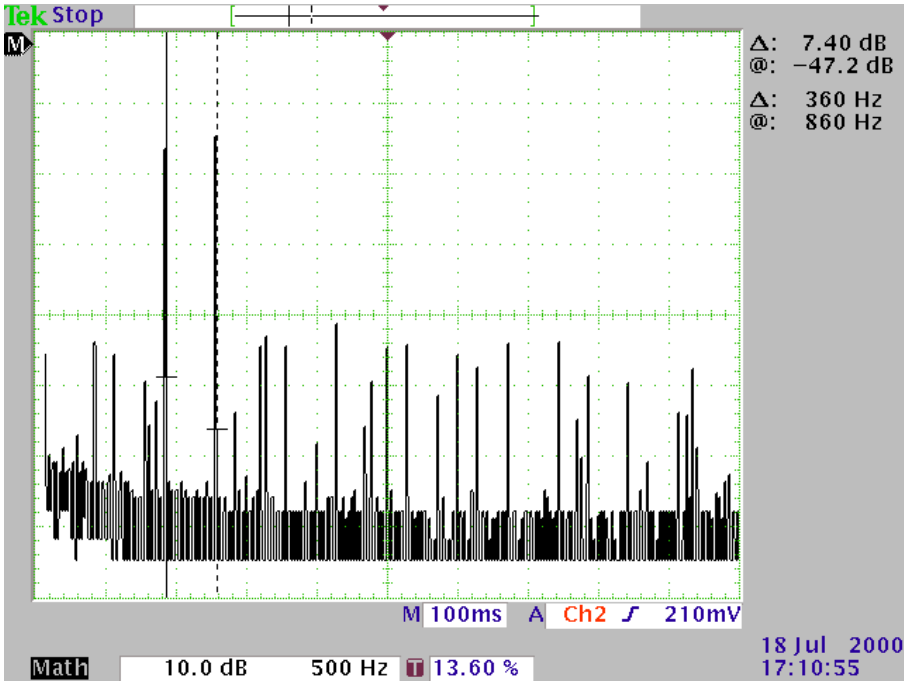

Figure 9-5. The '7' Digit with fL=857Hz and fH=1217Hz. Level Difference to Nearest Noise Component is 26dB (f=2.07kHz)

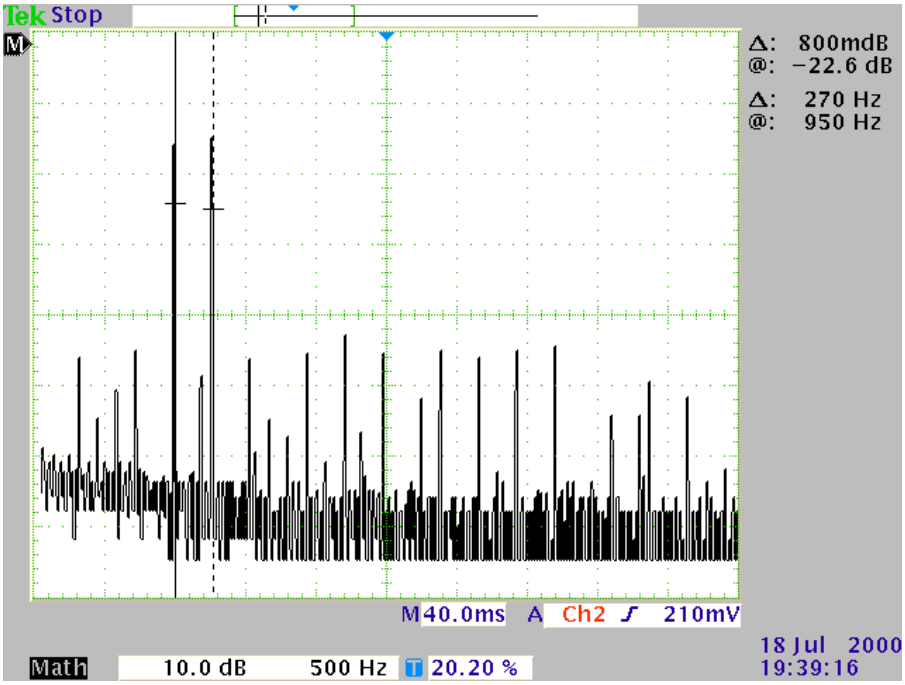

Figure 9-6. The '*' Digit (star) with fL=946Hz and fH=1216Hz. Level Difference to Nearest Noise Component is 26dB (f=2.16kHz).

9.0.1.2 DTMF Generation Timing Measurements

The same timing measurement is used in all the figures in this section, but different parts are zoomed.

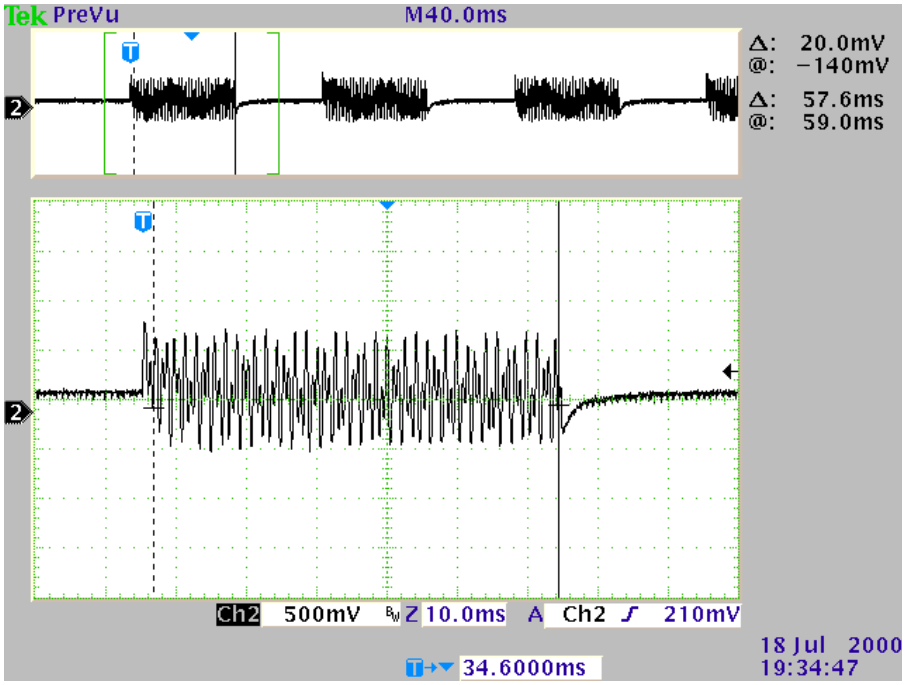

Figure 9-7. Signal Duration Measured to 58ms

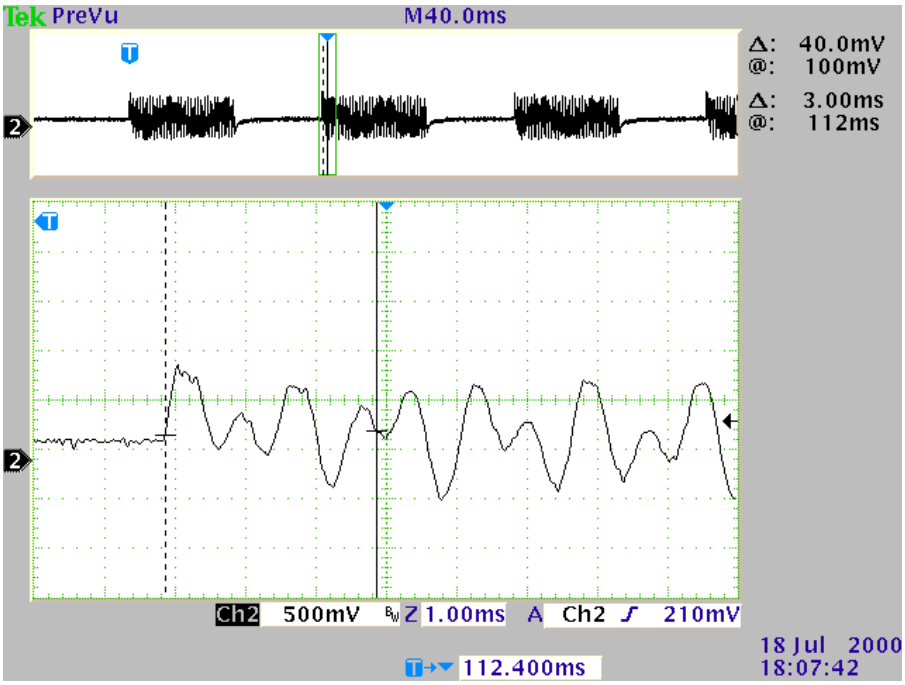

Figure 9-8. Signal Rise Sequence Showing that the Rise Time is <3ms

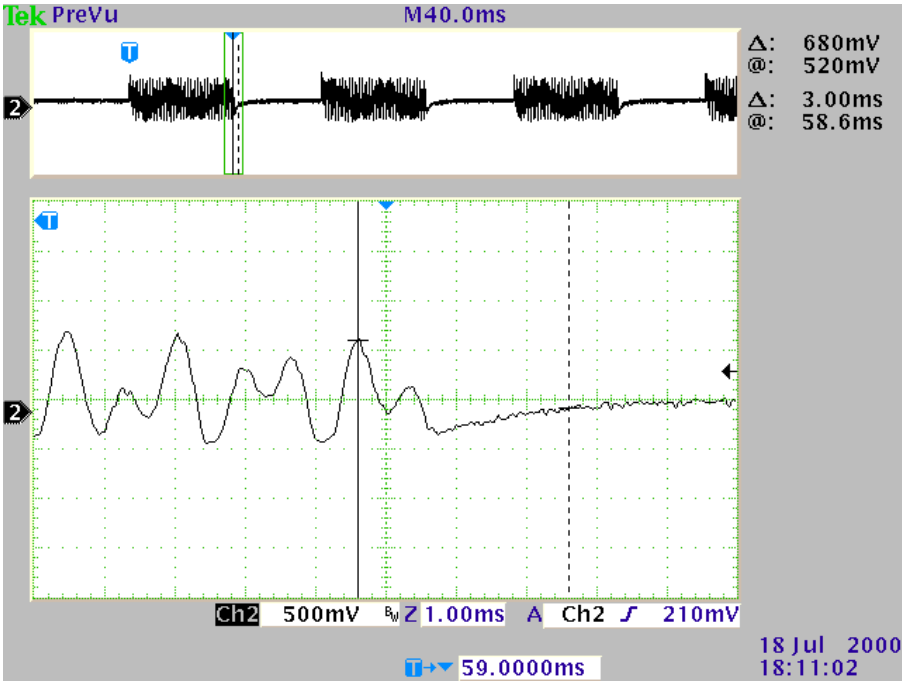

Figure 9-9. Signal Fall Sequence Showing that the Fall Time is <3ms

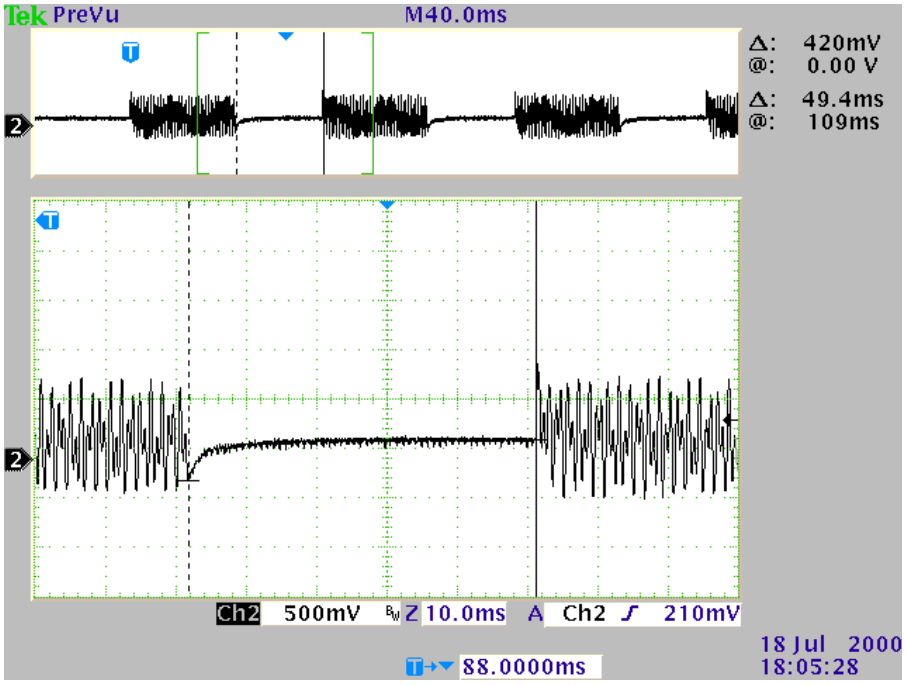

Figure 9-10. Inter-Digit Interval Measured to Approximately 50ms

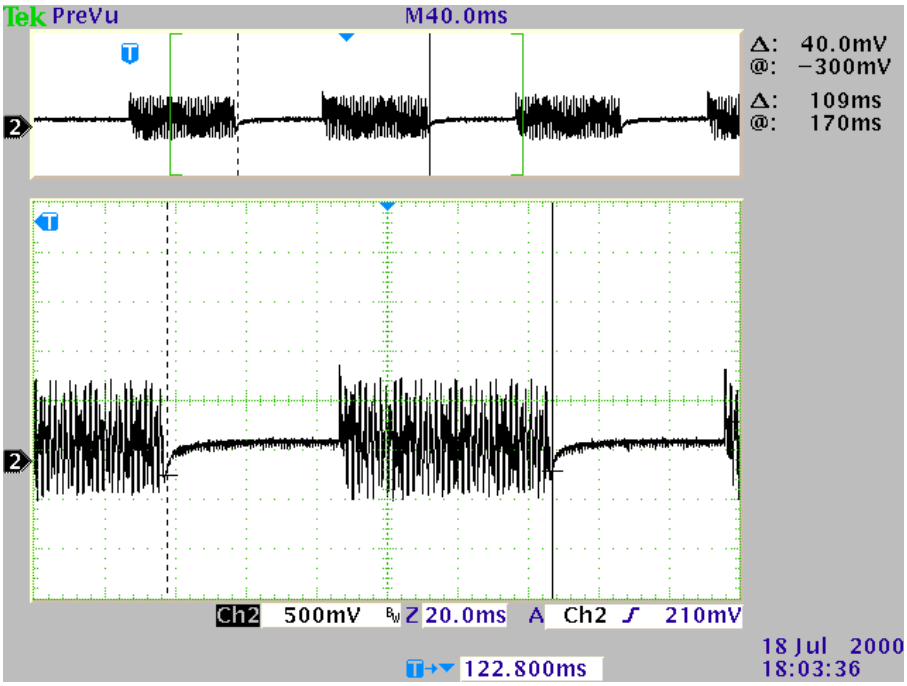

Figure 9-11. Cycle Time Measured to 109ms

Lit #: SXL-AN41-01

Sales and Tech Support Contact Information

For the latest contact and support information on SX devices, please visit the Ubicom website at www.ubicom.com. The site contains technical literature, local sales contacts, tech support and many other features.

1330 Charleston Road
Mountain View, CA 94043
Contact: Sales@ubicom.com
http://www.ubicom.com
Tel.: (650) 210-1500
Fax: (650) 210-8715