Caller ID on TMS320C2xx

Literature Number: BPRA056
Texas Instruments Europe
July 1997

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

List of Figures

List of Tables

Caller ID on TMS320C2xx

ABSTRACT

Caller ID is a method of transmitting the telephone number and/or name of the caller to the subscriber via a burst of V23 or BEL202 data during the ringing phase of a telephone call. Various methods of signaling the start of this message are used in different countries and this report attempts to identify a near universal core routine for handling the messages within the DSP environment. **ADSI** is an extension of Caller-ID used in some countries to deliver menu driven options to feature phones, the basic signaling is the same as Caller-ID except that the messages are sent during the call phase, and there is a full acknowledgment of the request to send data. This report covers the reception and decoding of either the Caller-ID or ADSI burst, it does not cover the interpretation of the burst or the recognition of non-DSP signaling, such as ringing or line reversal that may occur at the start, nor the need to switch a matching impedance onto the line. These are considered to be a part of the micro-controller function.

This application report is designed to be used with the Texas Instruments Caller-ID software package.

1. What is Caller-ID

Below is an example of a typical Caller-ID specification, some of the parameters do however vary between countries.

1.1 Physical Layer

Signaling may occur in either the idle state or loop state. All data transmitted by the physical layer consists of 8-bit characters transmitted asynchronously, preceded by one start bit and followed by a minimum of one stop bit. With the exception of the mark signal immediately following channel seizure, no more than 10 stop bits may be transmitted between characters.

Values for octets are given in the following format:

S2 M B7 B6 B5 B4 B3 B2 L S1

(Order of bits S1 first S2 last) where

- S1 = start bit
- S2 = stop bit
- M = most significant bit
- L = least significant bit
- B* = bit numbers 2 to 7

Octets are transmitted with most significant octet first.

1.2 Datalink Layer

The datalink layer provides framing of data into packets that can be distinguished from noise, and has error detection in the form of a checksum.

Channel Seizure	Mark Signal	Message Type	Message Length	Message (Presentation Layer)	Check -Sum

1.2.1 Channel Seizure

The channel seizure consists of a continuous sequence of alternate 0 and 1 bits at 1200 bits/s. The purpose of channel seizure is to minimize the possibility of noise mimicking a genuine carrier. The length of channel seizure as seen by terminal equipment is at least 96 bits (80 ms). It may be longer, up to 315 bits (262 ms)

1.2.2 Mark Signal

The mark signal seen by terminal equipment is at least 55 bits (45 ms) of continuous mark condition (equivalent to a series of stop bits, or no data being transmitted).

1.2.3 Message Type

The message type is a single binary byte. The value depends on the application.

1.2.4 Message Length

The message length is a single binary byte indicating the number of bytes in the message, excluding the message type, message length, and checksum bytes. This allows a message of between 0 and 255 bytes.

1.2.5 Message (Presentation Layer)

The message consists of between 0 and 255 bytes, according to the message length field. This is the presentation layer message (explained later). Any 8-bit value may be sent, depending on the requirements of the presentation layer and the application.

1.2.6 Checksum

The checksum consists of a single byte equal to the two’s complement sum of all bytes starting from the “message type” word up to the end of the message block. Carry from the most significant bit is ignored. The receiver must compute the 8-bit sum of all bytes starting from “message type” and including the checksum. The result must be zero or the message will be assumed to be corrupt.

1.3 Presentation Layer

Parameter Type	Parameter Length	Parameter Byte(s)	...	Parameter Type	Parameter Length	Parameter Byte(s)

The fields Parameter Type, Length, and Byte, together describe one presentation layer parameter, and may be repeated. Parameter Type will be discussed more fully in the next section. Parameter Length is a single binary byte of a value between 0 and 255. In Basic Mode a complete message must be contained within a single datalink packet, this means that the total length of presentation layer parameters must not exceed 255 bytes. Parameter Byte(s) contains zero or more bytes of application related information. The information contained in this parameter should be encoded in the local country specific character set. Most countries use some variant of ISO8859-1 (i.e., for the US this is ASCII, for the UK it is IA5). All of these coding variations are similar but have national characters for currency and accents. This display requirement is transparent to the DSP carrying out the reception

1.3.1 Parameter Type

There are currently eight parameter types associated with Caller-ID

Parameter Type Value	Parameter Name
00010001	Call Type
00000001	Time & Date
00000010	Calling line directory number (DN)
00000011	Called directory number
00000100	Reason for absence of DN
00000111	Caller name/text
00001000	Reason for absence of name
00010011	Network message system status

The calling line directory number is the number of the line from which the call was made, or a substitute presentation number. The called directory number is the number that was called. This is of significance when the call has been diverted.

There may be parameters of other types present. The call type parameter, if present, will always be sent first. Other parameters may be sent in any order. At least seven of these eight parameters must be recognized for the CLIP service (Called directory number is not necessary). Parameters may be sent with zero length. In such cases, parameter length will be zero and the checksum will be correct. Parameters are usually encoded in IA5. The version used is a 7-bit code and is sent in 8-bit bytes with the most significant bit set to zero. Non-displayable characters (codes 0-32 decimal) are not used. In the tables following, byte number 1 is sent first, followed by byte number 2 and so on.

1.3.1.1 Call Type Parameter

Byte Number	Contents	
1	Call Type Parameter Type Code(00010001)	
2	Parameter Length(1)	
3	Call Type	
	00000001	voice call
	00000010	ring-back-when-free call
	10000001	message waiting call

If the call type parameter is omitted then the call type is “voice call”. Additional Call Types may be defined later. Other call types (i.e., FAX, will be used when they are available). The “message waiting” call type is used to give an indication of a new message from a specific caller.

1.3.1.2 Time and Date Parameter

The Time parameter indicates the date and time (+/- 1 minute) of the event associated with the supplementary information message. Where the call type has a value 127 (01111111) or less, then the time is the current time and can be used to set internal terminal equipment clocks and calendars. For a “message waiting” call type the time and date refer to the time the message was left or recovered. For other call types with value 128 (10000000) or greater, the time and date may refer to some unspecified event and not necessarily to current time

Byte Number	Contents
1	Time & Date parameter type code(00000001)
2	Parameter length (8)
3,4	Month
5,6	Day
7,8	Hours
9,10	Minutes

1.3.1.3 Calling Line Directory Number Parameter

The maximum length of number sent is 18 characters. The first digit sent is in byte 3. The Calling Line Directory Number is a number that may be used to call back the caller, or the same service. It may not be the directory number of the originating call. For example, an 0800 may be associated with the caller. Where an alternative to the directory number of the caller is sent, this is known as a Presentation Number. There is no indication of which type of number is sent; this may change. If only a partial number is known then that partial number may be sent. This will be followed by a “-”. For instance, where a call comes from outside the digital network the area code may still be sent and shown as:

01604-66-or, for an international call from France; 00 33-

Byte Number	Contents
1	Calling Line Directory Number parameter type code(00000010)
2	Parameter length (n)
3	First digit
4	Second digit
.	.
.	.
n+2	nth digit

1.3.1.4 Reason for Absence of Directory Number Parameter

Byte Number	Contents
1	Reason for Absence of DN parameter type code(00000100)
2	Parameter length (1)
3	Reason

The reason will be one of the following IA5-encoded values
“P” = “Number Withheld” “O” = “Number Unavailable”

1.3.1.5 Called Directory Number Parameter

The Called Directory Number is the telephone number used by the caller when making the call. The maximum length of characters sent is 18, the first digit of the number is sent in byte 3, the second in byte 4 and so on.

Byte Number	Contents
1	Called Directory Number Parameter type code(00000011)
2	Parameter length (n)
3	First digit
4	Second digit
.	.
.	.
n+2	nth digit

1.3.1.6 Caller Name/Text parameter

At the launch of the service the Caller Name will not be available, the parameter will contain text only. The Name/Text consists of between 1 and 20 IA5 characters. The parameter may be used for other information when no name is available.

Byte Number	Contents
1	Caller Name/Text Parameter type code(00000111)
2	Parameter length (n)
3	First character
4	Second character
.	.
.	.
n+2	nth character

1.3.1.7 Reason for Absence of Name Parameter

The reason will be one of the following IA5-encoded values
"P" = "Name Withheld"   "O" = "Name Unavailable"

Byte Number	Contents
1	Reason for Absence of Name type code(00001000)
2	Parameter length (1)
3	Reason

1.3.1.8 Network Message System Status Parameter

The value of the Network Message System Status parameter is a binary encoded value indicating the number of messages waiting in the message system. 0 means no messages, 1 means one or an unspecified number; other values up to 255 indicate that number of messages waiting.

This parameter is not necessarily associated with a normal phone call, and will probably be sent as a no ring call.

Byte Number	Contents
1	Network System Message Status Parameter (00010011)
2	Parameter length (1)
3	Network System Message Status

Unless a Call Type parameter is also set, then any time parameter sent with the Network System Status parameter will indicate current clock time. This is to enable the terminal equipment to assume the time is current time and to set its internal clock where no Call Type parameter is sent.

1.3.2 Length of Caller-ID message

The longest Caller-ID message, excluding datalink layer information is currently 64 bytes. This length is expected for call types "Voice", "Ring-back-when-free", "Message Waiting". In the future there may be additional parameters that could extend message length, these will be sent after the parameters Call Type, caller number, name/text, reason for absence of name or number, and Network Message System Status.

1.3.3 Received Characteristics of V.23 Signals

Modulation	FSK
Mark (Logic 1)	1300 Hz +/- 1.5%
Space (Logic 0)	2100 Hz +/- 1.5%
Received signal level for mark	-8dBV to -40dBV
Received signal level for space	-8dBV to -40dBV
Signal level differential	The received signal levels may differ by up to 6 dB
Unwanted signals	Total power of extraneous signals in the voice band is at least 20dB below the signal levels
Transmission rate	1200 baud +/- 1%
Data format	Serial binary asynchronous (1 start bit first, then 8 data bits with least significant bit first, followed by 1 stop bit minimum, up to 10 stop bits maximum. Start bit 0, stop bit 1

2. Software Performance

The Texas Instruments Caller ID and ADSI code implements the basic rate caller ID function (for both the transmitter and the receiver during call set-up) and, additionally, the receiver only for the more advanced full CPE alert/acknowledge signaling for off-hook data transmission. In all cases 1200 Baud V23/BEL202 is used for the basis signaling of the data message (Caller ID message). Please note that the code described in this documentation is evaluation code that has been generated to determine MIPs requirements, and has only been tested for basic self loop-back functionality; it has not been verified against published standards for V23, BEL202, Caller-ID or other standards. Coding has been based on draft specifications for both the British and French Caller-ID standards.

2.1 MIPs

Below are the MIPs and memory requirements for the Caller-ID software subroutines.

		Cycles	RAM	ROM
320C2x 58C80	Generate	52/69	12	294
	Receive 1st Ring	195/172	36	573
	Receive CPE	195/172	84	784
	Receive Auto	281/172	84	784
320c2xx	Generate	57/76	12	280
	Receive 1st Ring	200/169	36	568
	Receive CPE	200174	84	775
	Receive Auto	287/169	84	775
320c5x	Generate	47/59	12	256
	Receive 1st Ring	196/167	36	569
	Receive CPE	196/172	84	770
	Receive Auto	271/167	84	770

The First figure in the cycles table is the maximum number and the second is typical, the maximum can occur for a maximum of 2 cycles every 6 $^{2}/_{3}$ cycles. The worst case scenario is 2 max., 4 nom, 2 max., 5 nom 2 max. 5 nom,…

3. Generation Routines (Fskgen25/50.asm)

3.1 Caller ID Generation

This software takes a message pointed to in data memory by AR0 and generates a sequence of serial modem output based on a complete Caller-ID transmission. The message pointed to must have the following structure.

Word	Meaning
SeizCnt=x	The length of the line seizure burst will be 10*x+6 bits
MarkCnt=y	The length of the mark burst will be 10*y bits
DataCnt=z	The next z words contain the Caller-ID message in the lower byte of each word.
z Data Words	Each word contained one Byte of Caller-ID data.

3.1.1 Init_V23/Init_BEL202

These subroutines initialize the FSK modulator to generate either V23 or BEL202 modem tones, extract the data lengths from the message passed in AR0 and put the Caller-ID generator into state “Seizure”

3.1.2 GetCID

This subroutine generates one 8kHz sample of the output modem wave-form. It first checks the serial buffer variable “InMask” to determine if there is room in the parallel/serial data stream for another 10 bits (start, byte, stop). If not, it goes straight to **Sample** to generate the next modem sample. If there is room for more data it branches to the state in “State” an inserts the next byte of serial data. Operation is not defined if neither Init_V23 nor Init_BEL202 has been called first.

3.1.3 Seizure

Inserts 10 bits of seizure data in the serial stream, decrements the number of samples left, when no more samples sets state to **Mark**.

3.1.4 Mark

Inserts 10 bits of mark data in the serial stream, decrements the number of samples left, when no more samples sets state to **Message**.

3.1.5 RunOff

Performs run off the any remaining serial data once all message data has been put in the serial buffer. When no more samples, sets state to **Finished**.

3.1.6 Finished

Outputs a zero value as the modem value to indicate the whole message has been transmitted, also outputs zero if GetCID is called additionally after completion.

3.1.7 Message

Inserts into the serial stream 10 bits of data consisting of a start bit, 8 serial data bits (one byte) and a stop bit. When no more samples, sets state to **RunOff.**

3.1.8 Sample

Generates a modem sample based on the current bit transmission, advances phase by a frequency dependent amount, adjusting for 24kHz common denominator of 1200 baud data and 8kHz sample data on boundary bits to generate an output “phase”.

3.1.9 Symmetrical Compression

Compresses the calculate “phase” to 9 bits by truncation, then 7 by symmetry and look up in 1/4 wave table, then re-expands out to regenerate negative half cycle.

3.1.10 Variables

State	->	Address of routine for present state of message generator.
BytePtr	->	Parallel Data Pointer.
SeizCnt	->	No of 10 bit samples of Seizure left to serialize.
MarkCnt	->	No of 10 bit samples of Mark left to serialize.
DataCnt	->	No of bytes of Caller-ID data left to serialize.
DigitOne	->	8kHz Phase Change for a logic One
DigitZero	->	24kHz Phase Change difference between logic Zero and logic One.
Input	->	Serial Data Stream.
InMask	->	Valid bits in “Input”.
BitTime	->	No of 24kHz samples remaining for current bit.
Phase	->	Output Phase pointer for modem data.
Out	->	The serial data output sample.

4. Caller ID Receive Routines (Fskrcv.asm)

4.1 Caller ID Receiver

4.1.1 Standard(s) Background

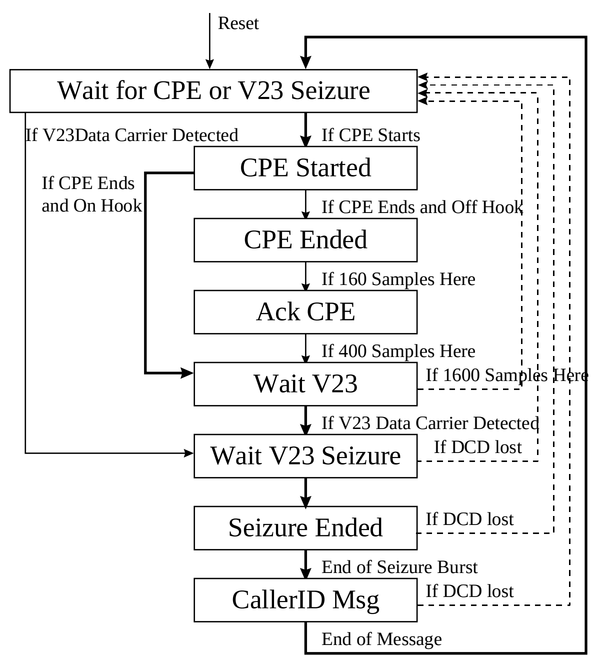
There are two alternatives for signaling the start of FSK Caller-ID in common usage. These are commonly described as “before the first ring” or “after the first ring”.

In “before the first ring”, the start of the Caller-ID message is signaled by an alert tone commonly called either “Idle state tone alert” or “Customer Premises Equipment alert” (CPE), whereas, in “after the first ring”, it is the ring itself that signals the burst of data. The former systems are more user-friendly in that, if the user answers too quickly, they don’t get a burst of modem data and/or lose the Caller-ID data - but they do require more complex software to detect. Some countries have opted to use both systems, indeed in some countries, such as the United Kingdom, both systems are in common use, with British Telecom using a “before” system and some combined cable TV/phone companies using an “after” system.

This software takes serial FSK data in either V23 or BEL202 format and regenerates the transmitted Caller-ID message. The code follows the state machine shown below right, each of these blocks is described in more detail below.

4.1.2 resV23

This subroutine resets the receiver.



4.1.3 Wait for CPE or V23 Seizure

Label: RX_CID

This subroutine takes a received 8kHz line value in the accumulator calls standard V23 receive SW and then examines the data to try and extract a Caller-ID message. The local variable “CIDstat” mirrors the message states of the Caller-ID protocol. The subroutine first calls V23rx (see later). Three situations can occur

1. If DCD has already been detected, it continues to decode the received data.
2. If DCD is detected but has not already been detected, the state machine is forced into the state “Wait V23 Seizure” by forcing “CIDstat” to be **Wseiz**

3. Continues with normal state machine processing.

4.1.4 CPE Started

Label: StCPE

Waits for the end of the CPE burst then, if on hook, sets state to **Wseiz** or, if off hook, sets state to **EndCPE**.

4.1.5 CPE Ended

Label: EndCPE

Waits 160 samples then sets state to **AckCPE**.

4.1.6 Ack CPE

Label: AckCPE

Sends a 400 sample burst of DTMF tone “D” then sets state to **WaitV23**.

4.1.7 Wait V23

Label: WaitV23

Waits up to 1600 samples for a V23 carrier detect before abandoning the CPE sequence. When DCD goes active sets state to **WSeiz**.

4.1.8 Wait V23 Seizure

Label: WSeiz

Waits for a 10 bit burst of alternating marks and spaces to be received. This will be received as 0x55. When received, sets state to **Seiz**.

4.1.9 Seizure Ended

Label: Seiz

Waits for the end of the seizure burst by throwing away any additional 0x55’s received. At the end of the burst of 0x55’s, during the switch to mark, an erroneous character may be received (0x57,0x5f,0x7f or 0xff) or one may get the first byte of the Caller-ID message, 0x80. This routine ignores the false seizure-mark byte or passes the valid message byte to **CIDmsg;** in either case the “CIDstat” is set to **CIDmsg**.

4.1.10 Caller ID Msg

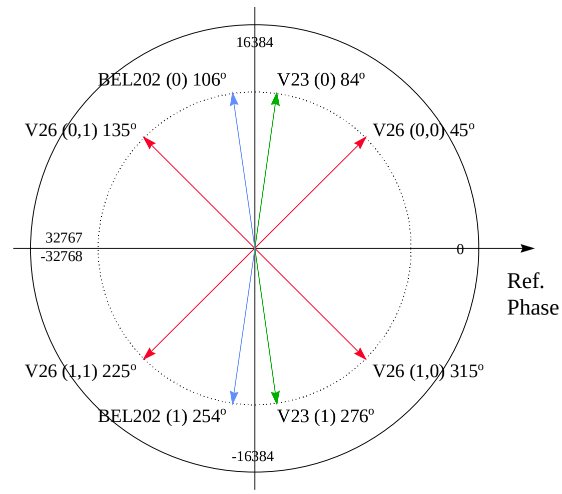
Label: CIDmsg

Outputs valid bytes to the DSP parallel output port PA3 until DCD loss indicates end of message.

4.2 FSK Demodulation Background

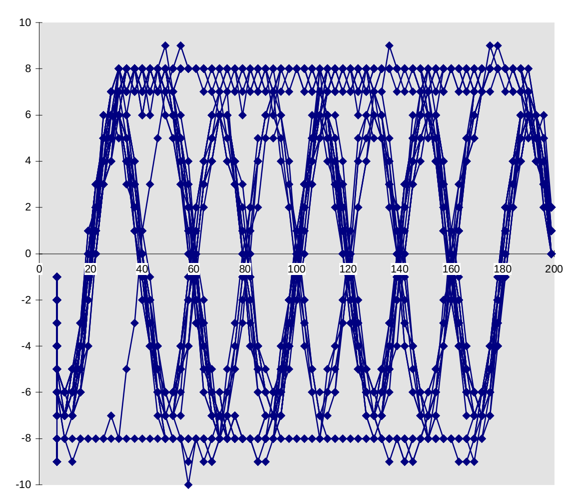
The V23 decoder software is capable of decoding data transmitted in either the V23 or Bell 202 1200 baud standards with 1 start, 8 data and 1 stop bit per symbol. The FSK modem data is decoded using digital quadrature demodulation techniques.

Although V23 and BEL202 use FSK modulation, this demodulator uses a $\pi/2$ BPSK (binary phase shift key) core. It is beyond the scope of this report but it can be shown that $\pi/2$ BPSK=FSK with a modulation index of 0.25, with a carrier frequency of 1700Hz as, per both V23 and BEL202, this corresponds to FSK frequencies of 1275Hz and 2125Hz. Both the V23 and BEL202 standards are close enough to these frequencies to be demodulated transparently to whichever standard is actually being used.



The relative phase difference for adjacent bits for the different V23/V26 and BEL202 standards is shown in the I/Q plot alongside. As can be seen, both BEL202 and V23 logic 0's are represented by a positive phase change and logic 1's by a negative phase change. The actual phase demodulator is capable of handling more complex modem standards such as QPSK. The V26 points, also shown on the diagram, are the phase positions for a QPSK V26 decoder. The reference frequency for this is1800Hz while for V23/BEL202 the reference frequency is 1700Hz.

To the right is an eye plot of phase difference within the BPSK demodulator for a typical V23 received byte sequence at the minimum signal level at which detection is required. As can be seen, detection is quite clean even at this poor signal level. From left to right the bits are stop, msb data … lsb data, start. (only one msb is a one in this message).



4.2.1 *V23rx (Demodulation)*

This subroutine demodulates the data received in "V23Sig". First, it updates the phase in the I and Q oscillators by advancing the 1700Hz (carrier), then mixes each to produce a phase quadrature base-band signal.

These are then low-pass filtered with a raised cosine FIR to remove the higher frequency components from the mixing. The two resultant phase and quadrature amplitudes are squared and added to give the real square amplitude.

4.2.2 Data Carrier Detect (DCD)

This takes the square amplitude of the incoming data, averages it over time, and then compares the instantaneous amplitude with the long term average amplitude to check for a stable amplitude FSK type signal. The instantaneous amplitude variation is again low-pass filtered to ignore spikes and produce the data carrier detect. Overall, the value in DCD will be negative when the signal has <6dB amplitude variation over approximately 10 bit periods.

4.2.3 Phase Detect

The measured amplitude is used to generate an Automatic Gain Control for the I and Q signals, which are normalized to a 0dBm0 signal, the MSB's of each of the I and Q signals are then merged into a byte and the arc-tangent taken to determine the phase of the signal. In order to keep the mathematics easier later, the circle is divided into 256 "degrees" rather than usual 360, this means that, when the phase comparison is performed, module 256 arithmetic can be used and overflows ignored.

4.2.4 Bit Extract

By comparing the phase of the current sample with the phase 5 samples previously, the frequency can be obtained over the last bit period relative to the 1700Hz mixing signal. If the phase difference is negative, then a frequency below 1700Hz (or a one) was sent. Conversely, if positive, a frequency above 1700Hz (or a zero) was sent. A simple +/- comparison was found to give a sufficiently low error rate though margin could be introduced for further improvement. The transmitted bit stream has now been reconstructed and it is thus possible to extract the data.

4.2.5 Clock/Byte Extract

First check for a data carrier. If one was not detected, then signal No Carrier to calling routine. Once a carrier has been detected, change the state to Carrier Idle and start looking for a negative phase to indicate a logic 1 start bit. As soon as it is detected, this indicates the beginning of the byte so set the bit clock to 1 1/2 bit times. The first and subsequent data bits will then be sampled in the middle (hopefully); continue to sample at bit intervals until 8 data bits have been received and then check for a stop bit. If DCD is lost, then throw away the data. When the stop bit is successfully sampled, set the state to Carrier Stop. For the calling software to extract the data from "Data", and start looking for another start bit, the extraction clock is re-synchronized for each byte in true "asynchronous" manner.

4.3 Additional Function for CPE Based Systems.

4.3.1 CPE Alert

The Software can detect the CPE alert signal which consists of a dual tone at 2130Hz and 2750Hz and respond by transmitting the required DTMF ‘D’ back to the exchange. The CPE alerting signal will be detected 32-40ms after the start of CPE alert transmission. After the end of the CPE alert tone has been detected, the code will return a mute request code, and 20ms later will transmit a 50ms CPE Acknowledge signal (DTMF ‘D’); the code will then wait 200ms for V23 data to appear on the line. If data appears, it will be demodulated. If no data appears, then it will revert to waiting for the CPE alert signal. When in the on-hook state, the CPE alert/acknowledge cycle is bypassed and the code will decode any V23 data present on the line.

The filter response curves for the individual tone detection filters on the CPE alerting signal are shown in the figures 1 and 2 below. Each filter is a 6th order IIR resonator made up of 3 bi-quads. Due to the closeness of the 2 frequencies and the bandwidths within which the signals must be recognized, the filters incorporate a notch at the mid-way point between the two frequencies (2440Hz) to prevent a single frequency at this level falsely triggering both resonators as would occur if simpler filter designs were used. In order to ensure recognition of the signal, which may be 38dB below the speech level though present for a relatively long time, filters with a stop-band rejection of at least 40dB are required. These filters have a stop band rejection of 50 - 60dB.

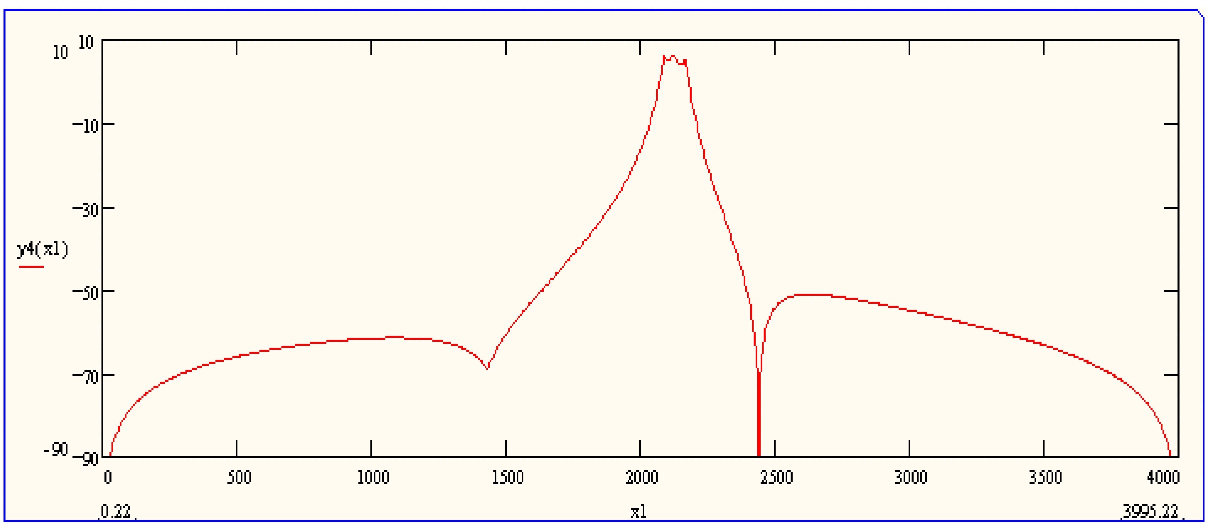


Figure 1: Low Frequency Filter for ADSI CPE Alert

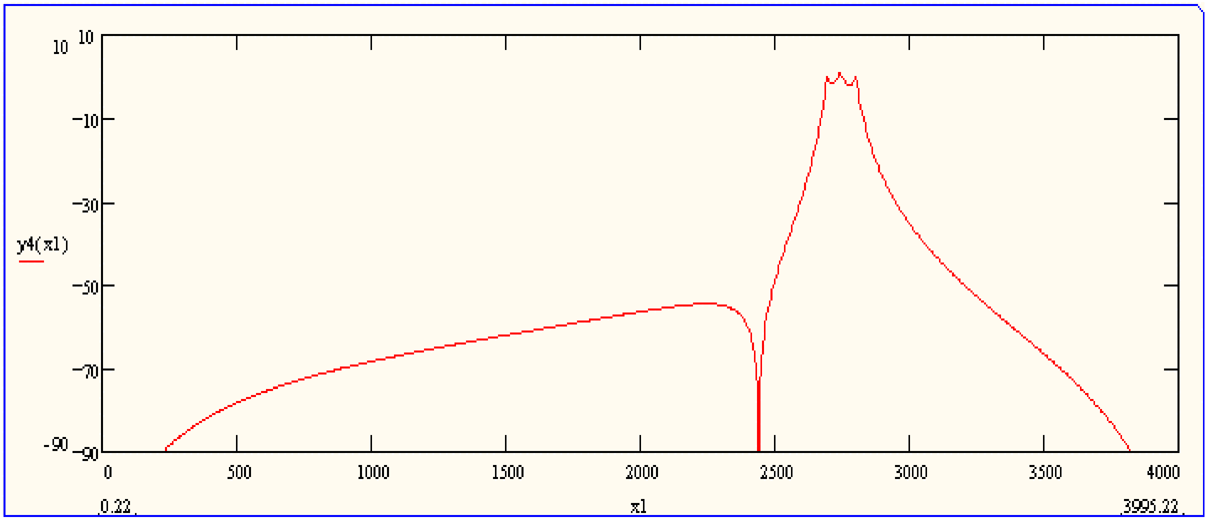


Figure 2: High Frequency Filter for ADSI CPE Alert

4.3.2 ADSI Main Code

The following global variables should be initialized before calling the interrupt service routine.

V23Sig	The latest sample from the incoming line in Q15 format
OnHook	Should contain 1 if the phone is currently on hook or 0 if the phone is currently off hook.

After calling, information is available in the following 3 variables:

CPEstate	NoCPE	No CPE Alert not detected
	StCPE	Start of CPE Alert detected
	EndCPE	End of CPE Alert detected
	AckCPE	Data register contains CPE Acknowledge signal in Q15 format
	WaitV23	Waiting for start of V23 data
	Wseiz	Waiting for V23 Seizure burst
	Seiz	Waiting for end of Seizure burst
	CIDmsg	Processing CID message
State	0x00	Carrier not Detected
	0x01	Carrier Detected, No data transmission
	0x03	Start of byte detected
	0xy3	Bit y of byte received
	0x05	Stop bit detected, Data register contains received byte in 8 ISB’s

Data This location contains either the CPE acknowledge signal in Q15 format or the received data byte depending on the state indicated in the status words.

If “CPEstate” is StCPE or higher, then the main routine should mute the handset and keypad of the telephone. The V23 routine is only called when either “OnHook” indicates the phone is on hook or the “CPEstate” indicates that V23 data is expected.

5. Tone Generation Routines (Tone.asm)

5.1 INITDTMF

This subroutine takes the digit passed in the Accumulator and resets the tone generation variables to generate samples of this DTMF tone. The digits 0-9 should be passed as 0-9; for *,10; #,11; A-D, 12-15. It must be called before the first call to TONE_SAMP. This subroutine may be called directly for dialing purposes during call set-up, reducing the need for a separate dialer chip.

5.2 TONE_SAMP

This subroutine requires no parameters and returns in the accumulator the next sample of the DTMF tone in Q15 format for output to the line. This subroutine may be called directly for dialing purposes during call set-up, reducing the need for a separate dialer chip.

The values for programming as DTMF dial tones are stored internally and are loaded via INITDTMF. The actual frequencies generated are shown in the table below.

Table 1: Actual DTMF Frequencies Generated

Required Frequency	Actual Frequency	Error	Required Frequency	Actual Frequency	Error
697	696.97	+0.0045%	1209	1208.98	-0.0013%
770	770.02	+0.0025%	1336	1335.98	-0.0016%
852	852.02	-0.0027%	1477	1476.98	-0.0013%
941	940.99	-0.0006%	1633	1632.99	-0.0004%