**Data Access Arrangement (DAA)**

A Data Access Arrangement (DAA) is an electronic interface within a computer and its modem to a public telephone line. The DAA isolates the electronic device from the higher voltage on the telephone line and act as a ring detection circuit.

There are two modes of operation for telephone equipment, On-hook and Off-hook. When CTE is in On-hook, the telephone line is available for call. Also the ring detector and CID amplifier circuits are active. When CTE is in Off-hook, the telephone line is engaged and loop current flows through the line.

After detecting the ring on RING pin of the DAA circuit, the snoop circuit becomes active. Ring can either be detected by RING going low or by polarity reversal. Depending on the countries standard the CID data burst is either sent between 1st and the 2nd ring (USA) or prior to the 1st ring.

**Caller ID**

Caller ID is a telephone service, available in analog and digital phone systems and most voice over Internet Protocol (VoIP) applications, that transmits a caller's number to the called party's telephone equipment during the ringing signal. The CID information is transmitted on the subscriber loop using frequency shift keyed (FSK) modem tones or DTMF (depending on countries). The information sent includes the date, time, and calling number.

Different countries often use different standards for transmitting caller ID information.

|  |  |
| --- | --- |
| Country | Caller ID standard |
| United States | Bellcore FSK |
| Canada | Bellcore FSK |
| China | Bellcore FSK |
| Hong Kong | Bellcore FSK |
| Ireland | ETSI FSK V23 (ETS 300 659-1) Ring Pulse Alert Signalling. Data sent after first short ring. |
| Taiwan | DTMF / ETSI FSK |
| United Kingdom | SIN227 (V23 FSK before first ring) |
| Japan | V23 FSK / DTMF |
| Spain | ETSI FSK |
| Brazil | Bellcore FSK / V23 FSK / DTMF |
| Norway | ETSI FSK |
| New Zealand | Bellcore FSK |

The CID information is sent on the destination subscriber loop between the first and second ring by means of two modem tones. The information is transmitted serially in FSK mode using one of the tones to represent a logic 1 (mark) and the other to represent a logic 0 (space). The message consists of a Channel Seizure string followed by a Mark string and then the caller information. The information is sent in one of two formats. The Single Data Message Format (SDMF) contains the date, time, and calling number. The Multiple Data Message Format (MDMF) contains the date, time, calling number, and the name associated with that number. Optionally, the number and name fields may contain data indicating that the information has been blocked by the caller or is unavailable.

**FSK implementation**

The data signalling interface has the following characteristics:

Link Type: 2-wire, simplex

Transmission Scheme: Analog, phase-coherent FSK

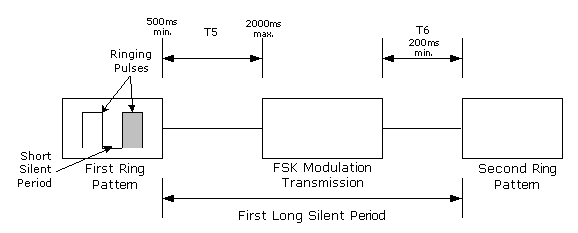
Logical 1 (mark) 1200 +/- 12 Hz

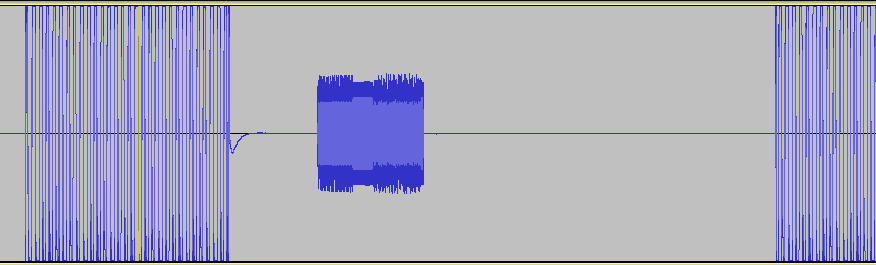
Logical 0 (space) 2200 +/- 22 Hz

Transmission Rate: 1200 bps

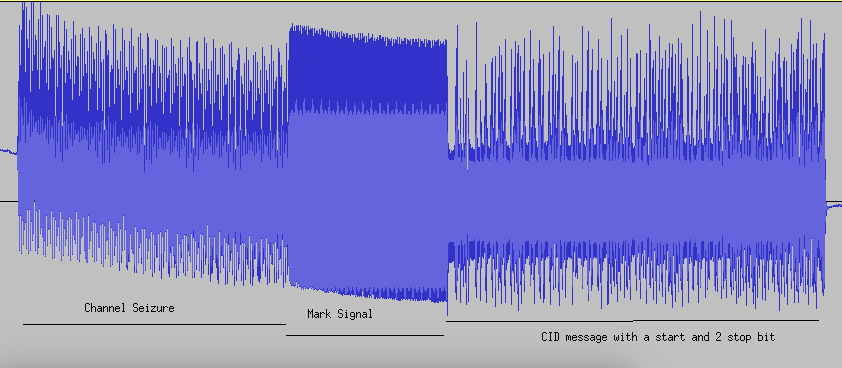
Transmission Level: 13.5 +/- dBm into 900 ohm load

Following a minimum of 500 ms after the end of the first ring, the sequence of transmission begins with a Channel Seizure. The Channel Seizure is a string of 300 continuous bits (250 ms) of alternating "0"s and "1"s. This string starts with a "0" and ends with a "1". A Mark Signal of 180 mark bits (150 ms) is sent immediately following the Channel Seizure Signal. The purpose of the Channel Seizure Signal and the Mark Signal is to prepare the data receiver in the Customer Premise Equipment (CPE) for the reception of the actual CID message. Once the Channel Seizure and Mark Signals have been sent the CID information is then transmitted starting with the Least Significant Bit (LSB) of the most significant character. This is true for both SDMF and MDMF[6]. Each character in the message consists of 8 bits. For displayable characters these bits represent a code defined by the American Standard Code for Information Interchange. When transmitted the character's 8 bits are preceded by a start bit (space) and followed by a stop bit (mark) giving a total of 10 bits sent for each character. The CID information is followed by a checksum for error detection.





There can be distortion in FSK modulation waveform, due to high voltage of the ring signal. But the DAA chip reduce that distortion. The fsk modulated caller ID looks like following ,



**FSK Demodulation**

To demodulate the fsk signal, we need to use IIR bandpass filters tuned to the respective frequencies. The IIR filter output depends on previous inputs as well as previous outputs. Depending on the sampling frequencies, the values of the IIR filter's input and output coefficient varies. The coefficient can be calculated using the [mkfilter program](http://www-users.cs.york.ac.uk/~fisher/mkfilter/).

General equation for all the filters is,

**y[n] = c0\*x[n] + c1\*x[n-1] + ... + cM\*x[n-M] - ( d1\*y[n-1] + d2\*y[n-2] + ... + dN\*y[n-N])**

where N = no. of previous outputs

M = no. of previous inputs

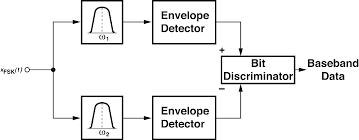
c1,c2,...,cM = input coefficients

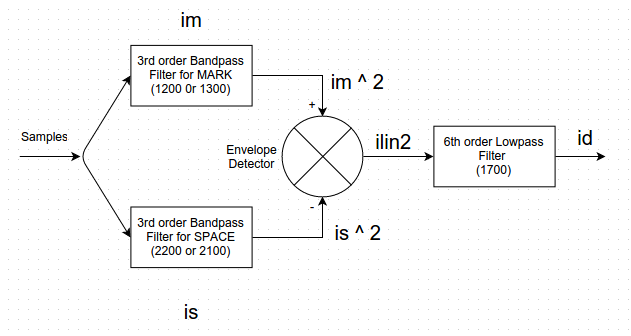
d1,d2,...,dN = output coefficients

x[n] = input values

y[n] = output values

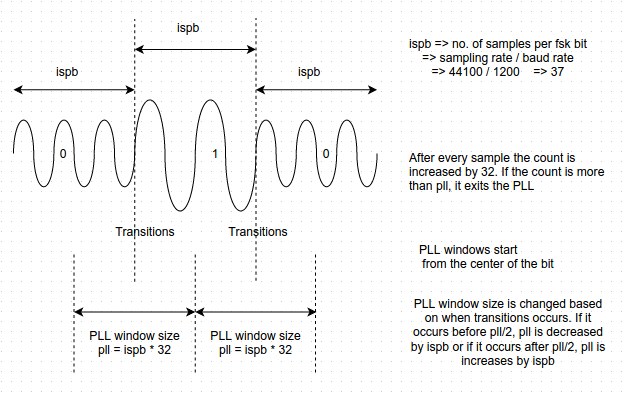
To detect a particular frequency, we can either use a FM discriminator or Phase Locked Loop (PLL). But for Binary FSK, since there is only two frequency, Mark and Space, we can also use Matched-filter detector. In matched-filter, the output of the mark filter and the space filter are compared and the decision is made. If the output of the mark filter is more than the space filter, then it is a Mark bit or else its Space.





For the general case of non-coherent FSK with non-white noise interference, the problem of optimum filter design is more complex. To minimize the effect of non-white interference, it is desirable to use a bandpass filter with relatively steep attenuation skirts and without the side lobes that are characteristic of the sinc-function filter. It is also desirable that the filter perform well in white noise. For each filter shape, there is an optimum bandwidth. In general, if the filter bandwidth is too wide, excess noise energy will be included. If the filter bandwidth is too narrow, consecutive signal elements will interfere with each other. To specify filter characteristics, it is convenient to talk in terms of the filter’s 3 dB bandwidth.

Even if we are using matched filter detector, we still need to use PLL for synchronizing the number of samples required to detect a bit. We can get samples per bit by dividing sampling rate with the baud rate. The digital PLL looks for the transition between 1s and 0s. If the transition occurs before the half point, it reduces the number of samples required to detect the bit and if the transition occurs after the half point, it increases the count.



So for the Caller ID program – **(ciddeco)** , we first need some bandpass filter for filtering respective mark and space frequencies. After passing the stream of samples of the signal to these filters we get a mark and a space wavforms. By using the matched filter we differentiate between the two waveform's value and pass it to the low pass filter with center frequency as 1700 hz. 1700 Hz is center for both 1200-2200 and 1300-2100 ranges. We get a stream of single digit values as output of this filter, -ve values for mark and non -ve values for space..

**Signal Handling**

To simulate the RING interrupt from DAA chip, we can use Linux signal handling concept. The Caller ID message is sent between the 1st and the 2nd ring. So when we get the first ring we can use SIGINT to send the interrupt to the ciddeco program to start capturing samples from the sound card and decode the message. The duration between 1st and 2nd ring is usually around 4 sec. So as soon as we get the SIGINT, we can use alarm() system call in its signal handler, so that after 4 sec we get another signal interrupt SIGALRM to stop capturing the samples. Both the processing of capturing samples from the jack and decoding of the Caller ID happens simultaneously and are continuous running like a daemon process. To terminate the program we can either use ctrl-C twice or ctrl-Z to send SIGTSTP.

**PCM sound card**

To integrate the sound card with ciddeco, we can use the tinyalsa library which provide simple application likes pcm\_capture to capture the samples into wav files and pcm\_play to play the pcm wav files. As we cannot change the hardware parameters of the sound card, we use the pcm\_capture to capture the samples with 2 channels i.e. stereo with interleaved mode. So in interleaved mode samples are stored in alternate fashion. For example if we use 16 bits per sample, the samples are stored in the form of R R L L R R L L..... where R is sample from right channel and L is sample from the left channel. If bits per sample is 24 the format will be R R R L L L R R R L L L.... because it will require 3 bytes to store the sample. As we require the ciddeco program to run as a daemon, we open the pcm sound card as soon as we get the interrupt for the 1st ring and close the pcm sound card when we get the 2nd ring interrupt.

When we initialize the PCM sound card, we set the period\_size and period\_count size enough to avoid buffer overrun. So when the sound card's buffer get full, the pcm\_read function copies the samples into the user buffer. This process takes time, but the decoding of caller id id very fast. Pcm\_read takes around 93 ms to read 16384 bytes of data from the sound card, of which 8192 bytes are from each channel. On the other hand decoding 8192 bytes of data takes around 2-3 ms. So for synchronizing both these process, we use semaphores.