[CEN 584]

Summer 2015

42854

CPT Internship Report

Insensi Inc.

Jenil Jain

ASU ID: 1207917824

Embedded Software Intern

Jonathan Megevand

jonathan@insensi.com

+1 9168487477

--------------------------- ----------------

Mentor’s Signature Date

--------------------------- ----------------

Faculty Chair/Program Chair Signature Date

**Company information:**

Insensi Inc. is a hardware startup based in New York City, founded by CEO Ilan Abehassera. In order to have a safe and hassle free way of staying connected with all family members, from a 3 year old kid to 93 year old grand parents, a device named ILY was designed by Insensi. ILY is a dedicated device to make video calls or just audio calls to other ILY's as well as any other smartphones containing the ILY application, all over the world. The device is also land-line compatible. This is the company's first product and is expected to launch this fall.

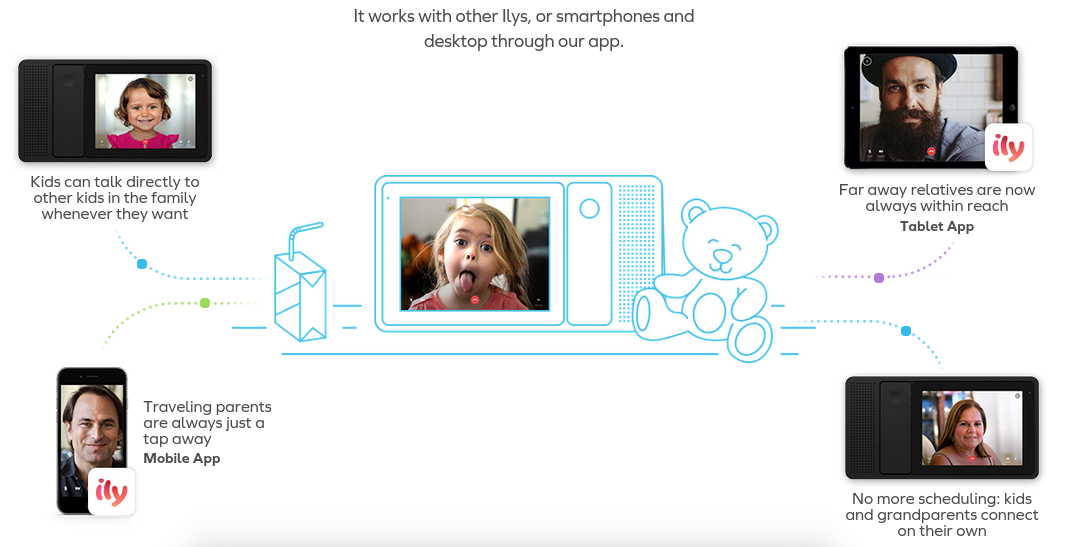


Fig. 1 ILY device and its scope of usage

I completed my internship as Embedded Software Intern working with the Embedded hardware team. The team comprised of a Senior Embedded Software engineer, a Electronics & Embedded engineer and other interns. The team is responsible for designing the Insensi development board based on iMX6 Dual lite processor containing dual ARM Cortex A9 core and porting Android/Linux board support packages on it. Apart from that, integrating the board with other sensors and peripherals like speakers, wireless handset, temperature sensor, etc; kernel configuration, memory management, managing device drivers were some of the other responsibilities of the Embedded hardware team.

**Background about the problem:**

One of the aim of the product is to make it compatible with the current land-line device, so that user don't need to have two communication devices at home. A land-line telephone uses a metal copper wire as the mode of transmission. The connection consist of a jack which is made of a Tip and a Ring to carry the DC signal over the network. To ring the telephone to alert a subscriber to an incoming call, about 90V of 20 Hz AC current is superimposed over the DC voltage already present on the idle line. To protect the device from such high voltage network a phone line interface DAA (Data Access Arrangement) is used. The DAA chip also gives an interrupt when there is an alert for the incoming call to decode the Caller ID (caller identification, CID).

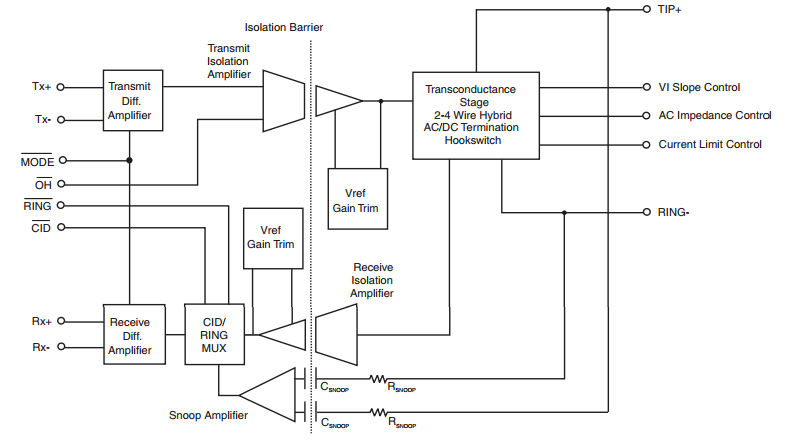
Caller ID is a telephone service, available in analog and digital phone systems and most voice over Internet Protocol (VoIP) applications, that transmits a [caller](https://en.wikipedia.org/wiki/Caller_(telecommunications))'s number to the [called party](https://en.wikipedia.org/wiki/Called_party)'s telephone equipment during the ringing signal. Common standards for Caller ID transmission is either FSK (Frequency Shift Keying) modulation or DTMF (Dual Tone Multi-Frequency). In USA, the standard used is Bellcore FSK while the standard used in other countries like India is DTMF.

So to make the product land-line compatible, Decoding the caller ID from the subscriber line is a very important and necessary feature. In a land-line telephone there are some dedicated chip to decode the Caller ID, for both DTMF and FSK modulation which can be used along with the micro-controller. Using a dedicated chip can reduce the complexity of the problem but it increases the final cost of the product. To reduce the cost and avoid adding any specific chip for decoding Caller ID, we decided to use the software implementation using digital filters to decode CID without any help of dedicated chip.

As different countries have different standards for Caller ID transmission like USA uses Bellcore FSK with mark frequency as 1200 Hz and space as 2200 Hz while UK uses v23 FSK with mark as 1300 Hz and space as 2100 Hz. On the other hand countries like India uses DTMF while some countries like Japan uses both v23 FSK and DTMF. So it was really important to make the decoding scalable and independent of any specific standard to make the device compatible with all different standards of different countries. So using the software implementation has another benefit as compared to dedicated chip.

**Focus of the effort**

To approach the problem I started with reading datasheets of the DAA chip (CPC 5620) and the audio codec (CS 4245). In [public switched telephone networks](https://en.wikipedia.org/wiki/Public_switched_telephone_network), DAA chip is an interface responsible for [data transmission](https://en.wikipedia.org/wiki/Data_transmission) at the customer side of the network interface. The codec on the other hand performs stereo analog to digital (A/D) and digital to analog (D/A) conversion. It also provide different serial audio interface formats like I2S, left-justified as well as right-justified to transfer samples from codec to processor and vice versa. After understanding datasheets of both the components, I was able to read the existing drivers and write the user application in C for them. The CSE 498 Embedded System Programming course covered both these topics, device drivers and writing user application in C. Also it cleared many topics of C programming language which helped me to write a good software application.

Fig. 2 Data Access Arrangement Circuit Fig. 3 Caller ID simulator

For the caller ID decoding, I had to understand all the caller ID standards and their specifications as well as their protocols. US uses Bellcore FSK with mark as 1200 Hz and space as 2200 Hz. FSK modulation is very popular technique for transmitting digital data because of its good signal-to-noise ratio as compared to other modulation techniques. DTMF is another popular modulation technique which allows telephone to indicate which number is pressed by its operator. To understand both these techniques, I needed to have knowledge of Digital signal processing. Courses like principle of communication engineering in undergraduate studies and CS 501 Computer Systems in graduate studies were really helpful for understanding modulation techniques. While course like Signals and System and Discrete-time Signal processing helped me with concepts like designing Finite Impulse Response filters, Butter-worth and Chebyshev bandpass and low pass filters.

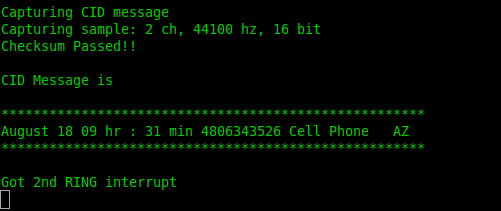
For the FSK demodulation of the Caller ID, as the specifications varied with different standards, I had to design the digital bandpass and low pass filter for various frequencies and then use a digital demodulation technique like a frequency discriminator, Digital phase-locked loop (DPLL), envelope detectors. Of which the Digital PLL is a better option because it works well even with less SNR. So for writing source code for such filters and DPLL, I had to study concepts of DSP programming. To understand the caller ID decoding better, I took the help of Asterisk library, which is an open source library for Private Branch Exchange (PBX) by reverse engineering the fsk demodulation process.

To test the application, I used a CID simulator, which contains many different programmable CID message. As the integration of audio codec with the processor was not complete, I needed to use the PCM sound card of the computer to record the signal into a WAV file and then integrate the sound card with the application to decode the Caller ID signal at real time. So for dealing with the sound card and capturing signal I had to get acquainted with the Advanced Linux Sound Architecture (ALSA) library and Audacity tool which helps in analyzing audio files. Also since, the product have only one sound card, we needed to find a way to share it with other programs as well by using direct memory access with mmap. As the DAA driver was also not yet integrated with the board, so to simulate the RING interrupts when the device receives a call, I used various signal handling concepts which was again taught in CSE 438 ESP course.

For compiling various source files and other libraries, I was required to write complex makefiles which was introduced to us in CSE 438 ESP course. In all, the embedded system programming course was very helpful during this internship. Apart from the programming and developing part, I also had experienced with GIT, which is a software revision control tool and documentation tool like Doxygen.

**Results**

The Caller ID decoding application was tested with several land-line connection from different service provider across US. It was successful in decoding the time of the call, name and number of the caller in real time using the sound card of the device. It even performed well in the presence of noise. To check the scalability, the program was tried with different sampling rate as well as different specifications and it worked every time. This application was used in the ILY device, to decode the caller ID message from the land-line connection and display the caller's information on the screen. It was an important feature to meet all the requirements of a land-line device.



The application used the sound card of the PC with the hardware parameters as stereo channel, 44100 Hz sampling rate and 16 bit per sample to capture the signal from the land-line connection. The Limitation of this application was that no two instance of this application can run simultaneously. Further development have to be made in sharing the pcm sound card using either dsnoop plugin or using direct memory access and sharing the mmaped area between two application.

**Learning experience**

This internship dramatically improved my C programming and software development skills. After this internship I was able to develop complex software which could be used in real time environment. The Embedded System Programming course was very beneficial at every point of this internship. Skills like writing device driver, knowledge of Linux kernel and operating systems proved to be very handy. I could easily read the existing drivers for the codec and write an user application for them. I was introduced to new topics like DSP and DSP programming and got acquainted with ALSA library for audio programming. In addition to that I learned small but very essential things for a software developer like writing Makefiles, documenting source code using Doxygen, Use SVN like github to collaborate with others in a project.

The ways in which this internship helped me:

* Improved my technical and programming skills.
* Acquired many soft skills like teamwork, problem solving and decision making and focused work ethics.
* Gained insight into working of a hardware company, specifically a startup.
* Made be more responsible, committed and dedicated.