

History

From "AllStarLink Wiki"

Contents

- 1 The History of App_Rpt
 - 1.1 Analog Radio Interface
 - 1.2 PCI Radio Card
 - 1.3 USB
- 2 Zapata Telephony and how it relates to Asterisk PBX

The History of App_Rpt

By Jim Dixon WB6NIL

Once upon a time, there was Asterisk PBX. I was heavily involved in its initial practical implementation if interested, see below (*Zapata Telephony Project as it relates to the Asterisk PBX*).

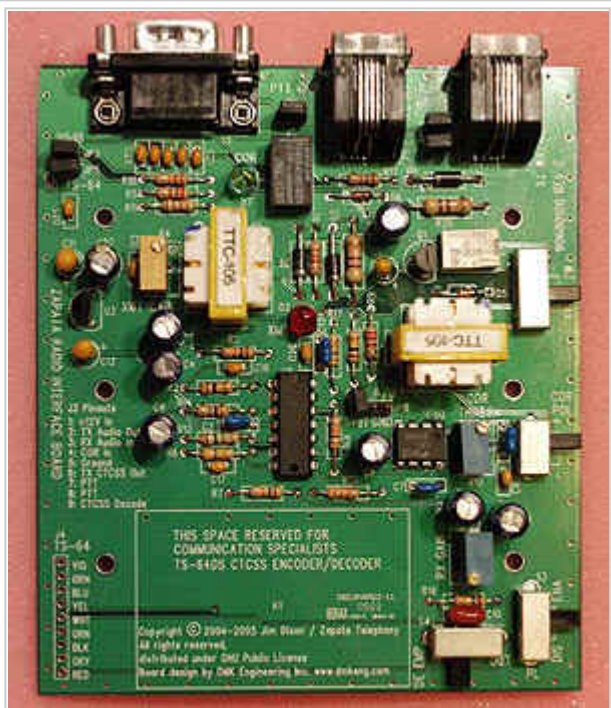
From its very onset, I saw that Asterisk was not only a good telephony switch, but also makes a good application implementation platform for virtually any telecommunication application that requires use of many of the things that Asterisk provides.

App_Rpt was started, not only to demonstrate this feature of Asterisk, but to essentially provide an outlet of nearly 30 years of experience and frustration with repeater and remote base systems, in the form of designing a system essentially "the way I think a system should be."

I learned from and was inspired by some of the best ideas and implementations and also from some of the worst, and tried to create a workable and desirable system that met as many needs as possible.

My first big hurdle was to design a radio interface that could interconnect with a PC internally and the typical interface signals of a two-way radio. In the interest of keeping the technology simple and inexpensive, I designed a board (the ARIB board) which interfaced 2 analog FXS ports (either on a multiport analog card, like the Digium TDM400P, or a channel bank connected to the PC through a T1 card, that is already part of an existing phone switch) to the Radio signals (and provide buffering, level control, etc).

Analog Radio Interface



Rev. C. Analog Radio Interface Board

This interface *did* work well to allow me to do the initial development of the software, and first half a dozen deployments, since most of us that had it first already had Asterisk phone switches already in operation, with spare ports on our channel banks.

The systems were originally deployed in Northern California (Oakhurst area) at a couple of locations, and in San Diego by Steve, WA6ZFT.

With Steve's greatly appreciated help, we were able to debug the thing and add many new and interesting features.

PCI Radio Card



Quad PCI Radio Card

After a few months of the system running stably (despite new additions all the time), we decided that better system integration was appropriate, so we (along with help from David Kramer) designed and produced the Quad PCI Radio Interface card. This allows interface between the PC and 4 separate radio systems

simultaneously, via 8 pin modular connectors.

In addition we also found an optimal PC in which to best use these PCI cards. Its a mini-ITX system that runs on 12 volts directly, 1 U rackmount and has no moving parts, and is comparatively inexpensive.

Together the PCI card and the PC make an ultra-stable, super high-end radio controller system, that, coupled with its open-source-ness gives the system implementer supreme flexibility and functionality.

For the next 2 years, we kept adding features, and refining the ones we had, and discovering that even though MANY people were QUITE interested in the technology, the price for entry was a bit high (for those who had no experience and wanted to just evaluate it). It became painfully obvious that something had to be done, somehow, to allow the hardware interface price to be lowered considerably without sacrifice in the quality of the technology.

USB



DMK Engineering URI

After looking at many possibilities, including usage of a PC's internal audio subsystem (yuch! yuch! yuch!), it sure seemed that the only realistic chance we had was to get something reliably working with a USB interface.

Experiments with USB several years previously show it to be utterly flaky and useless on Linux, and not much better even on Windows. But, since that's about all that's available any more (other than maybe Firewire, which is more or less the same thing) I figured we should re-visit the possibility of stable USB, and did so and found success. Its good that things change.

After some looking around, I found the C-Media CM-108 USB Sound IC. It is VERY simple, VERY high quality, VERY reliable and VERY inexpensive. Even already make USB sound cards (well, sound devices, they're awfully small to call a card) that could easily be modified for development and testing purposes.

So after some testing and verification of the USB hardware, I went to my friend Steve Henke, W9SH (who had already done some major work on the IAXRPT project for us), and asked him to use his amazing DSP skills to write code to do two-way audio processing right in the PC (going along with the Zapata Telephony motto of *"We don't need no stinkin' DSP!"*), which, happily he did.

We now have functional DSP code and channel interface code to allow USB devices to interface with App_Rpt and Asterisk, allowing high-quality, yet inexpensive interconnection with radio devices.

Zapata Telephony and how it relates to Asterisk PBX

By Jim Dixon, WB6NIL

About 20-25 or so years ago, AT&T started offering an API (well, one to an extent, at least) allowing users customize functionality of their Audix voicemail/attendant system which ran on an AT&T 3BX usually 3B10) Unix platform. This system cost thousands of dollars a port, and had very limited functionality.

In an attempt to make things more possible and attractive (especially to those who didnt have an AT&T PBX or Central Office switch to hook Audix up to) a couple of manufacturers came out with a card that you could put in your PC, which ran under MS-DOS, and answered one single POTS line (loopstart FXO only). These were rather low quality, compared with today's standards (not to mention the horrendously pessimal environment in which they had to run), and still cost upwards of \$1,000 each. Most of these cards ended up being really bad sounding and flaky personal answering machines.

In 1985 or so, a couple of companies came out with pretty-much decent 4 port cards, that cost about \$1,000 each (wow, brought the cost down to \$250 per port!). They worked MUCH more reliably then their single port predecessors, and actually sounded pretty decent, and you could actually put 6 or 8 of them in a fast 286 machine, so a 32 port system was easy to attain. As a result the age of practical Computer Telephony had begun.

As a consultant, I have been working heavily in the area of Computer Telephony ever since it existed. I very quickly became extremely well- versed in the hardware, software and system design aspects of it. This was not difficult, since I already had years of experience in non-computer based telephony.

After seeing my customers (who deployed the systems that I designed, in VERY big ways) spending literally millions of dollars every year (just one of my customers alone would spend over \$1M/year alone, not to mention several others that came close) on high density Computer Telecom hardware.

It really tore me apart to see these people spending \$5,000 or \$10,000 for a board that cost some manufacturer a few hundred dollars to make. And furthermore, the software and drivers would never work 100% properly. I think one of the many reasons that I got a lot of work in this area, was that I knew all the ways in which the stuff was broken, and knew how to work around it (or not).

In any case, the cards had to be at least somewhat expensive, because they had to contain a reasonable amount of processing power (and not just conventional processing, DSP functionality was necessary), because the PC's to which they were attached just didnt have much processing power at that time.

Very early on, I knew that someday in some "perfect" future out there over the horizon, it would be commonplace for computers to handle all of the necessary processing functionality internally, making the necessary external hardware to connect up to telecom interfaces VERY inexpensive and in some cases trivial.

Accordingly, I always sort of kept a corner of an eye out for what the "Put on your seatbelts, youve never seen one this fast before" processor throughput was becoming over time, and in about the 486-66DX2 era, it looked like things were pretty much progressing at a sort of fixed exponential rate. I knew, especially after the Pentium processors came out, that the time for internalization of Computer Telephony was going to be soon, so I kept a much more watchful eye out.

I figured that if I was looking for this out there, there **must** be others thinking the same thing, and doing something about it. I looked, and searched and waited, and along about the time of the PentiumIII-1000 (100 MHz Bus) I finally said, "gosh these processors CLEARLY have to be able to handle this".

But to my dismay, no one had done anything about this. What I hadn't realized was that my vision was 100% right on, I just didnt know that **I** was going to be one that implemented it.

In order to prove my initial concept I dug out an old Mitel MB89000C "ISDN Express Development" card (an ISA card that had more or less one-of-everything telecom on it for the purpose of designing with their telecom hardware) which contained a couple of T-1 interfaces and a cross-point matrix (Timeslot- Interchanger). This

would give me physical access from the PC's ISA bus to the data on the T-1 timeslots (albeit not efficiently, as it was in 8 bit I/O and the TSI chip required MUCHO wait states for access).

I wrote a driver for the kludge card (I had to make a couple of mods to it) for FreeBSD (which was my OS of choice at the time), and determined that I could actually reliably get 6 channels of I/O from the card. But, more importantly, the 6 channels of user-space processing (buffer movement, DTMF decoding, etc), barely took any CPU time at all, thoroughly proving that the 600MHZ PIII I had at the time could probably process 50-75 ports if the BUS I/O didnt take too much of it.

As a result of the success (the 'mie' driver as I called it) I went out and got stuff to wire wrap a new ISA card design that made efficient use of (as it turns out all of) the ISA bus in 16 bit mode with no wait states. I was successful in getting 2 entire T-1's (48 channels) of data transferred over the bus, and the PC was able to handle it without any problems.

So I had ISA cards made, and offered them for sale (I sold about 50 of them) and put the full design (including board photo plot files) on the Net for public consumption.

Since this concept was so revolutionary, and was certain to make a lot of waves in the industry, I decided on the Mexican revolutionary motif, and named the technology and organization after the famous Mexican revolutionary Emiliano Zapata. I decided to call the card the "tormenta" which, in Spanish, means "storm", but contextually is usually used to imply a " **BIG** storm", like a hurricane or such.

That's how Zapata Telephony started.

I wrote a complete driver for the Tormenta ISA card for *BSD, and put it out on the Net. The response I got, with little exception was "well that's great for BSD, but what do you have for Linux?"

Personally, I'd never even seen Linux run before. But, I can take a hint, so I went down to the local store (Fry's in Woodland Hills) and bought a copy of RedHat Linux 6.0 off the shelf (I think 7.0 had JUST been released but was not available on shelf yet). I loaded it into a PC, (including full development stuff including Kernel sources). I poked around in the driver sources until I found a VERY simple driver that had all the basics, entry points, interfaces, etc (I used the Video Spigot driver for the most part), and used it to show me how to format (well at least to be functional) a minimal Linux driver. So, I ported the BSD driver over to Linux (actually wasn't **that** difficult, since most of the general concepts are roughly the same). It didnt have support for loadable kernel modules (heck what was that? in BSD 3.X you have to re-compile the Kernel to change configurations. The last system I used with loadable drivers was VAX/VMS.) but it did function (after you re-compiled a kernel with it included). Since my whole entire experience with Linux consisted of installation and writing a kernel module, I **knew** that it **had** to be just wrong, wrong, wrong, full of bad, obnoxious, things, faux pauses, and things that would curl even a happy Penguin's nose hairs.

With this in mind, I announced/released it on the Net, with the full knowledge that some Linux Kernel dude would come along, laugh, then barf, then laugh again, then take pity on me and offer to re-format it into "proper Linuxness".

Within 48 hours of its posting I got an email from some dude in Alabama (Mark Spencer), who offered to do exactly that. Not only that he said that he had something that would be perfect for this whole thing (Asterisk).

At the time, Asterisk was a functional concept, but had no real way of becoming a practical useful thing, since it didn't, at that time, have a concept of being able to talk directly (or very well indirectly for that matter, being that there wasn't much, if any, in the way of practical VOIP hardware available) to any Telecom hardware (phones, lines, etc). Its marriage with the Zapata Telephony system concept and hardware/driver/ library design and interface allowed it to grow to be a real switch, that could talk to real telephones, lines, etc.

Additionally Mark has nothing short of brilliant insight into VOIP, networking, system internals, etc., and at the beginning of all this had a great interest in Telephones and Telephony. But he had limited experience in Telephone systems, and how they work, particularly in the area of telecom hardware interfaces. From the

beginning I was and always have been there, to help him in these areas, both providing information, and implementing code in both the drivers and the switch for various things related to this. We, and now more recently others have made a good team (heck I ask him stuff about kernels, VOIP, and other really esoteric Linux stuff all the time), working for the common goal of bringing the ultimate in Telecom technology to the public at a realistic and affordable price.

Since the ISA card, I designed the "Tormenta 2 PCI Quad T1/E1" card, which Mark marketed as the Digium T400P and E400P, and now Varion is marketing as the V400P (both T1 and E1). All of the design files (including photo plot files) are available on the Zapatatelephony.org website for public consumption.

We have more, higher-density designs on the way.

As anyone can see, with Mark's dedicated work (and a lot of Mine and other people's) on the Zaptel drivers and the Asterisk software, the technologies have come a long, long way, and continue to grow and improve every day.

Footnote:

Has anyone ever taken a moment to sit back and consider the ENORMOUS responsibility that Mark has taken upon himself by doing this project? Have you ever thought of how incredibly many things that he has to concern himself with, and that it just **NEVER ENDS!** At this point, I believe that I have worked with him on this project longer than just about anyone, including some of his employees, and believe me, I have a good vantage point to see at least some of the stuff that he has to go through to accomplish this.

Personally, I would have **NEVER** taken on such a task, being that I am and was quite aware of the level of responsibility required to do so.

Yes, the task that I took on was and is quite a task, and quite a responsibility, but I did what I knew I could accomplish. Mark's part is way larger than mine, and all I can say that I know what it takes for him to do what he is doing, and I seriously appreciate the time and dedication that he has put into all the incredibly wonderful things that he has done for it and all of us.

Furthermore, I'd like to seriously thank all of the project contributors and everyone else that has done some part to help with this project. Thank you for demonstrating that you believe in it, and that you believe in us.

Editors note: Jim became a SK in December of 2016. My best guess is these articles were written around 2006. I've applied minor formatting changes for readability.

Retrieved from "<http://wiki.allstarlink.org/w/index.php?title=History&oldid=879>"

-
- This page was last modified on 15 March 2018, at 01:27.