The KE0FF ICOM Band Module Controller

A brief history and description (part 1)

By Joseph Haas, ke0ff
10/11/2012
joeh-*at*-rollanet-*dot*-org

Many years ago, I reverse engineered the control head signals for the ICOM IC-901A transceiver. I used what I discovered to create a remote controller that I interfaced to the University of MO-Rolla (now MS&T) W0EEE repeater system as a remote base. It later occurred to me that this information could also allow me to design a custom control system for a mobile radio that would use the ICOM band modules along with a distributed control system. Until recently, I have never seemed to find the time to put all the pieces together, but the idea remained and was something I thought of every so often. I'd sketch block diagrams, and collect ICOM modules, but never actually build anything.

All through the thread of this project, I have never really given much thought to "why?" It just wasn't pertinent to the tasks at hand. Now, however, looking back at the progression of the project, the why of it all tugs for recognition. I mean, there are a plethora of mobile radios out there, many with nearly all the features I might desire. Why use these radios as the basis for something I could buy ready-to-go?

I'm not sure I can give a clear answer. However, the simple answer is, well…simplicity. I wanted a radio that fit my sensibilities for mobile operation. One that was simple to use with distributed elements that can be placed in the vehicle where they make sense. The display goes up on the dash where it can be viewed while still having peripheral visual contact with the road in front of the vehicle. The PTT goes on the steering wheel, stick shift, or as a hand-held switch (no mic). Frequency and audio controls go near the center console and use large buttons and knobs. No radio I've seen or heard of has these features. Even the remote control head radios have all of the controls on the control head, which are typically small and out of reach or difficult to manipulate if you place the control head where it is easy to see while driving.

This drive for simplicity extends out of the ICOM modules themselves. It is the simplicity of the interface protocol that has kept this project alive over the years. A simple series of clocked bit fields control the module's configuration and frequency. 15 signal connections form a daisy-chained control bus for the IC-900 modules, with an additional 15 pin control bus for the expansion modules that were designed specifically for the IC-901A. With nothing more than a schematic and some test equipment, I was able to perform the initial reverse-engineering of the signal interface and decode the bit patterns of the control codes. With the aid of the service manuals and component datasheets, I was later able to take the investigation further to discover some of the "why" behind the "what".

This document is an historical presentation of the reverse engineering project as it moved from initial concept towards an ultimate (if sometimes fuzzy) goal. I have created a second document, "ICOM UX Module Design Guide", as a technical repository of the practical aspects of controlling the ICOM modules for any soul brave enough to attempt to design their own interface for these modules.

In the beginning, there was the 901 (circa 1991)

Actually, the IC-900 came first, but I have never seen one in person. When the W0EEE radio club purchased an ICOM IC-901A, I deduced (incorrectly) that it would be an easy task to design a remote control interface for the radio (I had already designed a repeater controller and HF remote interface). My limited experience with the IC-735 HF radio caused me to assume that all new radios would be remote-controllable (or, as it turned out, perhaps not).

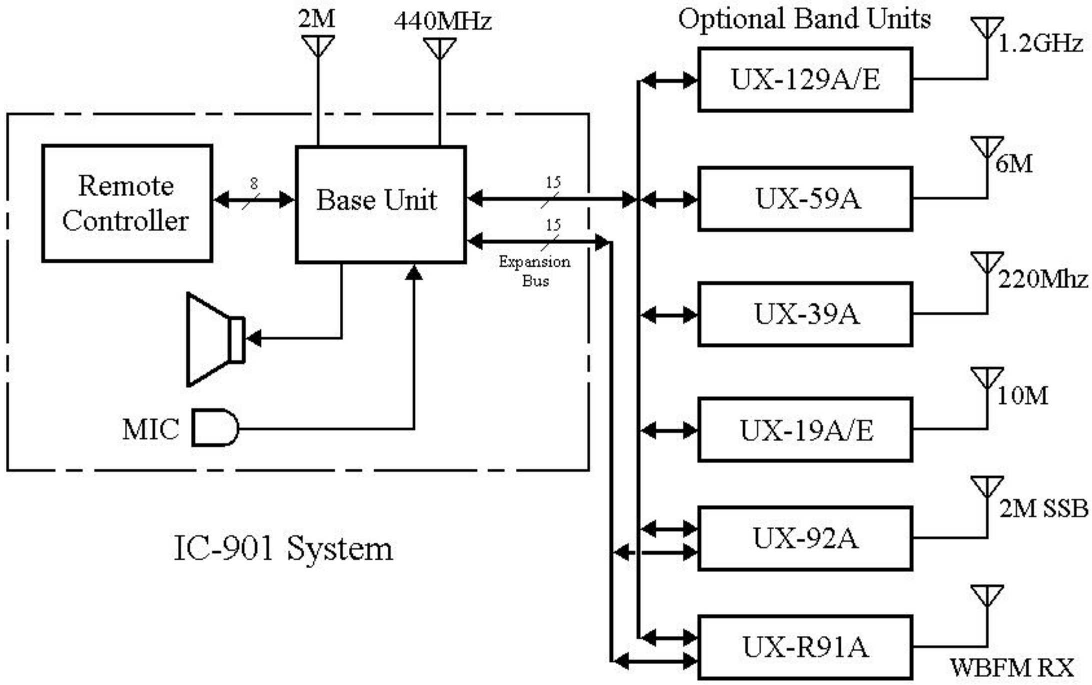

Figure 1. IC-901 simplified block diagram.

I soon discovered that there was no remote control capability built-in to the IC-901 in spite of the fact that the control head offered that exact behavior. Undaunted, I investigated the control head to see if I could decipher the control protocols – there were only 6 wires (including power and ground), how hard could it be? From the schematic, it was clear that there were two basic control paths, one from the control head to the IC-901 base unit which used synchronous clock and data signals, and another from the IC-901 base unit to the control head that used an asynchronous protocol. My initial assumption wass that the bulk of the radio intelligence was in the base unit, and the display unit was simply a "terminal" used to provide a display and inputs. I was surprised to discover that even though the IC-901 base unit featured a microprocessor (as did the control head) it did not feature any intelligence about the radio configuration. All of the radio's personality, the operating frequency, memories, CTCSS settings, and every other user defined setting were contained in the control head. In addition, all of the control operations were sourced from the control head. Without the control head, the IC-901 wouldn't even receive at some default frequency (it wouldn't even turn on).

After probing the signals with a scope to determine their time-domain particulars, I designed a microprocessor based appliance to intercept the serial signals and present them to a terminal interface where they could be captured on a PC and analyzed. This was a project all unto itself, but the effort made it easy to capture large amounts of data for analysis. I took the "particle accelerator" approach to the decoding effort. I'd "disturb" the IC-901, and then capture the resulting data stream(s), noting the nature of the disturbance (such as, changing the RX frequency by 5Khz and recording the new frequency and captured data). I'd repeat this many times to build up a pattern of data, and then examine the data to isolate changes in the bit fields in an attempt to isolate the relevant bits. The trick was to only change one thing at a time. The focus of the analysis was then to isolate the pertinent data fields without deciphering every single bit (it would turn out that the IC-901 produced a LOT of bits).

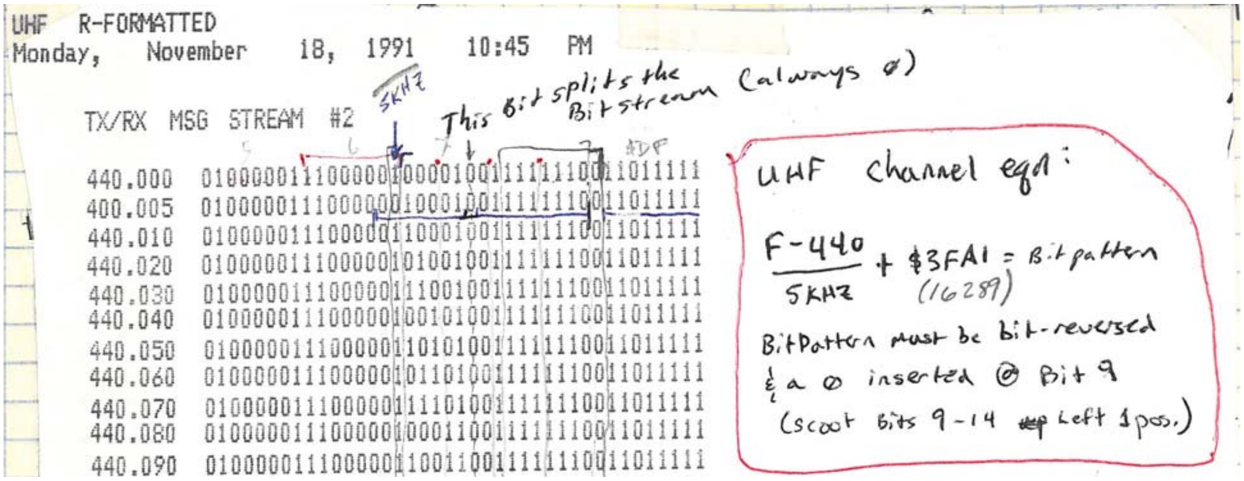

Figure 2. IC-901 data streams for the UHF unit. This print-out was edited to isolate the PLL bit fields.

The synchronous stream was sent as 1 or more groups of data, each 40 bits in length. The base unit processor would monitor the streams and toggle various strobe signals at different points in the data stream. Of course, this I surmised from the schematics since I was only interested in the black-box behavior of the system. The asynchronous stream consisted of 12 bits of data sent with a 200us bit period. Every 16th asynchronous packet conveyed microphone button status, while the other 15 packets conveyed S/RF data and squelch open/closed status.

By now, I had devised a control scheme that would essentially switch the base unit between the control head and the remote control interface using CD4066 switches for the analog signals, and logic gates for the digital ones. At the time, the W0EEE repeater was located in the club radio shack and this configuration would allow the IC-901 to be used as a local radio when not in remote control mode. The system worked remarkably well, although I was never able to quite get the CTCSS decoder figured out (no one ever seemed to miss it). The W0EEE shack was eventually moved off-campus for a short time before finding a home in the new EE wing. This move necessitated that the repeater be relocated to the remote TX site at the top of the tallest dorm hall, which severed the remote base connections for the HF rig and IC-901. By the time W0EEE was in its new home, I had since left the area and no one ever re-visited the remote base idea (that I know of). For the time it was functional, that remote base was the most awesome (remote) radio shack I have ever operated.

The Saga of the UX Module (circa 1996)

The IC-900, predecessor to the IC-901, featured separate modules for all of the supported bands: 10M, 6M, 2M, 220 MHz, 440 MHz, and 1.2GHz. The IC-901 used these same "optional" modules except for the 2M and 440 which were built in to its base unit. I had known about the ACC FC-900 remote control interface, and got the opportunity to trace the data directly at the module interface courtesy of N4NEQ and his FC-900/RC-85 combination. This required a slight modification to my trace/capture tool since the band module protocol included a strobe signal. Interfacing to the FC-900 was attractive since there were a lot of FC-900 systems in play and it would be easier to sell my controllers to users who could re-use some of their previous investment.

I also started to employ schematic analysis as the schematics and service manuals for the IC-90x series had started to show up on the internet, as well as the data sheets for the components. This allowed me to augment the "particle-physics" method of reverse-engineering the modules behavior. It sometimes raised more questions than it would answer, so I tended to use it more as a planning tool when trying to figure out a new module, or troubleshoot why it wasn't working they way my initial analysis had predicted.

Without the overhead necessary for the IC-901 base unit, the relevant bit streams for the UX modules were much shorter and simpler. By now, I had studied the modules sufficiently enough that I understood how the serial data clocked into the various parts of the module. The clocking was in two parts: the first part was 10 bits to address the module and set up the steering logic, and the 2^{nd} part was a 20 bit (nominal) stream that fed into the PLL chip. A simple, single chip processor interface with a logic IC (to invert the COS signals) and voltage regulator was all that was needed to interface to the UX modules via the FC-900.

About this time, someone put a bug in my ear about creating a Doug Hall version of the FC-900 controller. I sketched up some diagrams, but never got enough time to pull it off the paper.

The Austin Motorola Marathon (circa 1998)

After moving to Austin, TX, I got interested for a brief time in APRS. I built my own tracker using a 68HC11 to gather data from a GPS, and send it to a TNC. For the radio, I used a UX-29H (this is the 45W version of the ICOM 2-meter modules). The 68HC11 controlled the frequency of the UX-29 and I could send configuration commands via an RS-232 interface to control the radio. The package was pretty compact for the day and actually featured a good deal of empty space. I used a Bud chassis and was somewhat limited in the available sizes; a custom enclosure could have been significantly smaller. With the addition of a small speaker/microphone interface module, I could use the setup as a voice radio. This was not all that far afield from what I had in mind for a mobile controller. The tracker was used as the lead tracker (a small vehicle used to pace the lead runner) in the Austin Motorola Marathon for a couple of years. The 45W output of the module and small size made it an attractive choice to deal with the artificial canyons of downtown Austin.

The Dark Times (circa 2001)

For many reasons, my interest in ham radio, both as a business and hobby, waned around the turn of the century. From a business standpoint, I hadn't been able to get to critical mass with my repeater controllers, so the return on investment never exceeded unity. I had the supreme luck to have chosen one of the most niche markets (repeater controllers) in one of the most niche fields (ham radio) one could imagine, where all of the customers wanted mil-spec features at swap-meet prices. There was a lot of effort and not enough return to justify the endeavor. Doing electronics day (I had a day job throughout) and night also took a toll and I found that I was just no longer very interested in things ham radio. I'd go through short spurts of interest, but they didn't last more than a few weeks.

In 2004, I got a new (used) car and set out to finally build my super mobile rig. I even ran some 4 AWG power cables (this was simply a 20ft jumper cable with the battery grips removed) and built a 100W PA for 6M using a MastrII PA module (this was to be used with my UX-59 module). The PA module was an interesting project (see "PIN T-R Switch for 6M") that I actually managed to finish (tho have yet to use) but that was the extent of this burst of activity. Subsequent activity spurts would generally consist of a couple of weeks of driving around with my HT and mag-mount, but little more.

The Resurrection (circa 2012)

Even though I stopped marketing my controllers, I still supported the fielded units, even if they changed hands. Every so often, I'd get a query or a unit that needed repair (usually lightning related). Around May of 2012, I received a query from a new owner of a controller that had been recently purchased through ebay. They were interested in the FF-8900, which was the FC-900 version of my remote control interface. Based on inputs from this new customer, I added several features to the 8900 and, in the process, cleaned up the software.

This re-sparked my interest in the ICOM mobile radio idea, and I put the HT and mag-mount back in the car. It was about this time that I ran into K5UUT on-air and he asked (probably not for the first time) if I could build a Doug Hall interface for the UX modules. I was already thinking about my mobile interface and this provided an additional spark to produce some actual hardware.

The UX modules need little more than a few control signals and power to function. There needs to be a squelch control, and some audio conditioning for RX and modulation signals. I used the IC-901 and FC-900 schematics as a basis and put together a simple audio chain for the RX and TX sides – nothing too elaborate, but enough to get the job done.

The FC-900 used module addresses 0 and 7 to control on-board logic (including the MX-315P CTCSS encoder chip). However, I wasn't limited by a narrow path to the processor like the FC-900 had. My processor was on-board, and so I didn't need to use any band-module addresses for on-board controls. I could also use a digital pot for squelch control, which would be needed for the mobile rig version. The mobile version also needed to support the IC-901A expansion modules (I have a 2M SSB module, and a wideband RX module), so there were extra control lines needed for those features. The DHE version used individual squelch pots (one for each band module) like the FC-900 because I wasn't sure that the DHE squelch functions had been implemented by any of the controller companies (I hadn't implemented this either).

I recently purchased a quantity of six MX-315P CTCSS chips (at no small cost) so that I could have a stock for repairs for the FC-900 (I don't see many of these, but they do show up wounded once in a while). My FF-8900 customer had a completely dead FC-900, and I was down to my last MX-315P. This chip is a simple CTCSS encoder that produced the most popular 39 CTCSS tones. However, it is no longer manufactured (I had to buy the six pieces from a Chinese surplus company). The successor, the MX-465, does encode and decode, but is very expensive, and isn't readily interchangeable with the MX-315P. The MX-465 also does not do simultaneous encode/decode. This throws a wrench in the works when you start to think about sub-band TX and cross-band repeat. In addition, most folks do not use CTCSS decode for remote base operations. So, for this project I simply included the capability for the processor to generate the CTCSS encode tone using DDS (direct digital synthesis) for the DHE version. This was a nifty cost reduction and was easy to do since I have already executed a number DDS project.

While I included the MX-465s in the design, and while they do audio filtering for CTCSS, I chose to do the filtering separately to allow the DHE version to have the same audio chain as the mobile version. This meant that there would be 4-poles of high pass active-filtering on the receive chains (Fc = 330 Hz) and 2-poles on the transmit chain (the transmit filter was omitted in the initial design and had to be scabbed-on later).

All of my previous controller work had used Motorola (now Freescale) processors, in particular, the 68HC711E9 with the software in assembly. However, most of my processor work over the last 6 years has been with 8051-style processors using C. While staying with the HC11 meant lots of potential code re-use, it also meant staying with a 20+ year old processor and development tools that were not far behind. Switching to C involved some re-coding, but this was a relatively benign exercise compared with the thought of having to write more assembly code on an old processor. There was a lot of new code to write anyway, so the re-use benefit would have been minor compared to the new code.

I decided on the SiLabs C8051F360 since it had a DAC (digital to analog converter) for the DDS tone generation, had sufficient memory and I/O, and was reasonably priced. I also have a personal library of C code for the SiLabs family, so that improved the odds of code re-use for drivers and low-level interface code. PWB size was an issue due to cost. I had initially postulated that I could hand-wire the interface, but the prospect of building at least two copies cured me of that affliction and I decided to gin up a PWB. I settled on a 3"x 6" card size as my best compromise between area and cost.

A two-layer design would have been the least cost, but I did not believe that I could accomplish this in a reasonable footprint. So, I settled on a 4-layer design with one ground-plane layer, and three routing layers. This would increase the cost, but I was able to find a vendor that offered a special for 4-layer designs that would give me 4 PWBs for about $300 plus shipping. This was about half the best price I could find anywhere else, so that was a real shot-in-the-arm for cost reduction.

For power, I went with linear regulators. I didn't expect the card to draw much power (less than 100mA), so the inefficient nature of linear regulators wasn't a huge drawback. I also decided to use a -5V DC-DC converter to provide a negative supply for the op-amps. I've used uni-polar op-amp supplies in the past, but the performance and implementation always seems to be better when the op-amps are powered with bi-polar supplies. This also allowed me to direct couple the internal stages which kept a bunch of coupling caps out of the mix. I filtered the op-amp supplies with a 10Ω/1,000µF RC filter to help keep out any audible ripple (this gives a low-pass F_c of 16 Hz).

I also filtered all of the off-card I/O signals (except the user function outputs) with π-filters that were arranged using a 2.2nH chip ferrite and two 100pF capacitors (for each off-card signal). The F360 processor can run at up to 100 MHz (I usually like to the processor pretty hot since I have it well de-coupled from the power supply) so I didn't want to take any chances with conducted chirpies. I also filtered the +12V input to reduce the likelihood of conducted emissions from the power supply.

I copied the FC-900 de-emphasis and squelch configuration for this design. It was simple, and I didn't see any benefit to adding complexity. The FC-900 used N-FETs to gate the receiver audio by shunting the DET audio path to ground. Since my audio chain was biased to 0V, that wouldn't work since the N-FET would shunt to ground on negative audio voltage transitions. I decided to use a CD4053 analog MUX to gate the audio. In a TSSOP package, it wouldn't take up much space, and would give good attenuation in the "off" state.

I decided to use TSSOP packages for all of the parts except the op-amps to save space. I typically used SOIC parts because they were the first of the surface mount devices, and I usually had them on-hand in a variety of flavors. They are also easier for me to see with my ageing eyes. However, for this design, I wanted to keep the size down, so I took the plunge and kept the devices as small as I could (without going to chip-scale, of course). For the op-amps, the resistors and caps were all 0603 sized, so going TSSOP wouldn't have shrunk things much without going to 0402 Rs and Cs, and I didn't want to go there since I have a couple of design kits for 0603 parts available, which makes design tweaks much more manageable as I have a nearly full collection of values for Rs and Cs in 0603 packages.

The CTCSS band-guard filters follow the squelch gate and provide RX audio to the system connector. The Main and Sub outputs are provided to support the dual-band options for the assembly. The Doug Hall interface version only supports a single module, but the sub-band was populated on this build to validate the circuits during development (I ended up using the sub-band filter for the modulator chain). The squelch gate supports a sub-enable option that can be driven by the processor to allow for software configuration of the output. A solder jumper is also supported to enable the sub-band output without intervention from the processor. Figure 3 shows a photo of the completed control board (and yes, there were a few design changes).

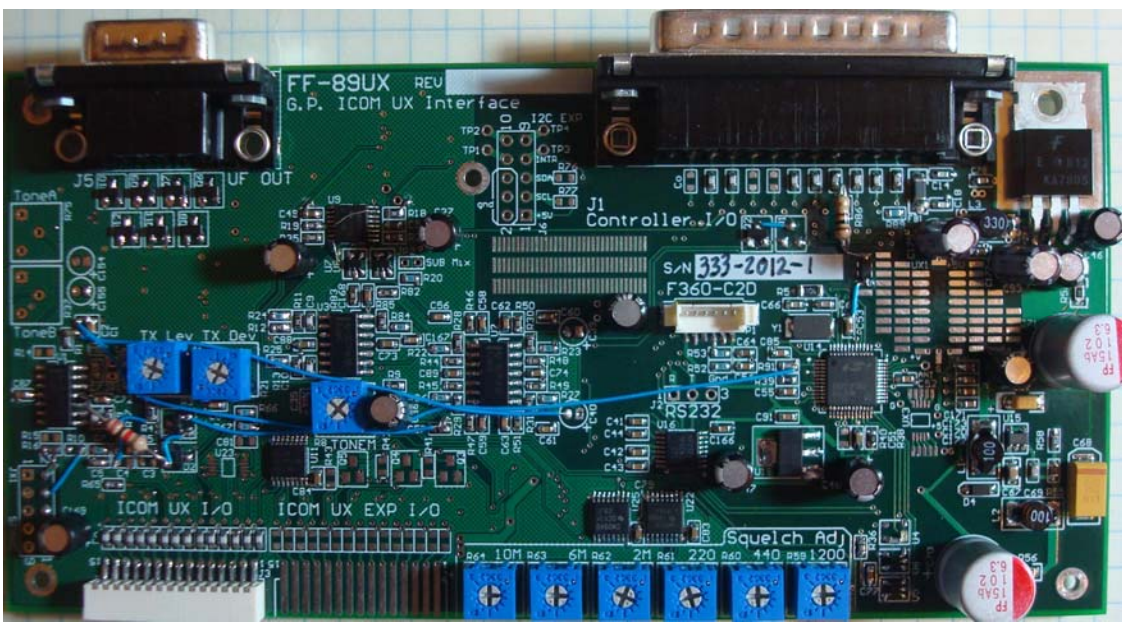

Figure 3. DHE version of the UX control module

The initial modulator circuit was also reminiscent of the FC-900. It features a level adjustment to drive a pre-emphasis circuit and a CTCSS band-guard filter to keep voice artifacts out of the CTCSS band. A mixer combines the CTCSS encode source(s) with a deviation adjustment to drive the MOD input of the UX modules. The FX-465 circuits are encode/decode only configurations (none of the filter pathways are used) and are left un-populated for the DHE version. For the mobile radio, the DDS tone generator will be used for DTMF generation, so the FX-465 parts would be used for the CTCSS encode feature.

Initial testing of the modulator was disappointing. The audio sounded “mushy”. Now, I define “mushy” to be the opposite of “tinny” (too much high frequency response), so “mushy” is not enough high frequency response. I spent a lot of time trying to get the modulator shaping right (or at least close enough to right), more on that later.

Let There Be Light (Emitting Diodes)

When the PWB layout was complete, I realized that I could squeeze in a display deign and save some cost for a part of the project that would normally be difficult to justify a PWB (I will only need one or two at most). I had some 0.8” 7-segment LEDs that I had purchased some time ago for this project, but I needed some additional devices for the sub band and CTCSS code displays. I only had 2”x 6” to work with, which was really about right (much bigger and the display would be ungainly) but it would take some effort to get all the pieces crammed in. At one point, I thought I might have to add a couple of wire jumpers, but managed to avoid that embarrassment.

More about why: I had originally envisioned an LCD display for the system display. I may still add one for configuration functions, but an LED display is really hard to beat. What it lacks in flexibility it makes up for in readability. This is a big factor for me, so I decided some time back that LED was the way to go for the main display.

I wanted the following parameters to be displayed: Main frequency covering 0 to 1999.999999 MHz with 1Hz resolution with +/-/S repeater offset indication; Sub frequency from 0 to 1999.995 MHz with at least 5 KHz resolution; Main and Sub S/RF bar-graphs (minimum 8 segments); Main and Sub RX & TX indicators; CTCSS frequency readout (4 digits, Main only) and Encode/Decode status.

I decided to use the MC14489 LED driver IC from Freescale. This is the same device I’ve used in the past for my repeater controller display. It was a known quantity, and the cost was not prohibitive, but 5 devices would be needed to accomplish the desired display formats. Using a FET switch, I was able to add two brightness levels to the MC14489 for a total of 4 levels. This would allow a graduated auto-dim feature if I decided to go to that extreme.

Figure 4. Display module, assembled

The display turned out well, but I used a linear 5V regulator without considering the power draw of up to 40+ LED segments (one of the many perils of a short design cycle). The almost 0.5A current consumption with all LEDs on is considerable and lends itself more to a switching regulator than a linear. Thus, I plan to add a simple switcher circuit to replace the LM7805 and eliminate the need for a substantial heatsink.

It is…ALIVE! (mostly)

During the initial hardware testing, I found several glitches in the design (to say the least). Some were inevitable consequences of a first spin, but others were just plain stupid. The saving grace was that all were relatively simple to fix, so the initial fiberglass was still viable.

Perhaps the greatest concern and effort was directed towards the modulation circuit. This was largely copied from the FC-900, but the audio just didn't sound right (not that I am a good judge). This led me to confront a basic shortcoming of my lab: it lacked sufficient receiver/transmitter test equipment. I was able to perform some spectral measurements (crudely) using a signal generator and an oscilloscope with FFT capability, but this only offered a relative measure of performance and it was nothing I could take to the bank.

Of greater help was a pspice demo package I had. It was limited in the degree of circuit complexity that could be modeled, but this was generally not a significant hindrance since the area of interest is often just a few components, as was the case here. I focused on the limiter/filter that was the heart of the FC-900 modulation chain. However, I ended up modeling most of the circuit before long, and was able to see the evidence of what I was hearing. This was important since it led credibility to the model and allowed me to then tweak the circuit to try to get the result I was looking for.

The pre-emphasis was provided by a capacitor and resistor at the input to a high-gain (adjustable) op-amp stage. The RC network formed a 48 KHz high-pass response that gave the +6 dB/octave shaping that was needed for pre-emphasis. The high-gain op-amp was needed to scale up the signal to compensate for the attenuation introduced by the depth of the pre-emphasis. In my opinion, the depth was greater than necessary since the slope of the RC filter is effectively stable below about 0.8 F_c. This meant that a lower F_c could be used which would increase the level of audio available to the modulator. Cutting the F_c in half still provided the shaping needed in the 0.3-3 KHz bandwidth (and then some) while boosting the available modulation level by 20 dB. Half again still left plenty of margin above the 3 KHz audio roll-off point, so I settled on 12 KHz for the pre-emphasis filter (the IC-901A pre-emphasis rolled-off at around 7.2 KHz).

The limiter was simply a pair of back-to-back diodes followed by a 3-pole RC filter. The FC-900 design used 3 identical RC stages in series, which distorts the F_c calculation due to inter-stage loading. I modeled this circuit and found that the real F_c was about 1/3 that calculated for a single stage, which put the -3dB F_c at around 3 KHz. I tweaked the resistor values in the model so that I could get the response of the pre-emphasis network to extend to at least 3 KHz and gave the new circuit values a spin on the hardware. The result was an improvement. There still seemed to be some high-frequency roll-off, but it was much better than before.

With the help of K5UUT, I used his IC-901 to characterize the mic-to-mod path through the IC-901 and establish the frequency response, and absolute signal level at the mod input to the modules. This at least provided a base-line for these parameters that I could (re)design against.

About this time, my focus settled on the limiter. It was not behaving like I expected, nor like the limiter used in the IC-901. Observing the modulation signal on a scope and slowly increasing the input level, I could see the limiter action as a flattening of the sine peaks. However, the limit wasn't hard by any stretch, further increases to input level increased the flattening, but also increased the peak level considerably. The whole point of the limiter here was to provide a peak deviation demarcation so that the transmitter could not be over-deviated and the limiter I had was not meeting this crucial design criteria at all. So, it was time for a series of science experiments. But first, I had to close my eyes and slowly count to ten. This was just two diodes, it was supposed to be the easy part!!!

The IC-901A limiter behaved as I expected based on what I saw on the oscilloscope – increasing the input signal would eventually result in flattening of the modulator output at a fixed value. Using the spice tool and actual circuit modifications, I tried several variations of the limiter design. Actually, there were essentially two versions: The first was the back-to-back, shunt diode pair version, and the second was a back-to-back diode pair in the feedback path of an op-amp. All of the circuits attempted provided the same result.

To try to get my self properly aligned, I analyzed the modulator circuit for the IC-901A. Their limiter was the voltage rails of an op-amp. Saturating an op-amp usually causes problems in that the response coming out of saturation can be delayed. However, for 3 KHz bandwidth signals, this is probably not a big issue. If need be, I decided that this could be a fall-back option, but I hadn't given up just yet.

The problem with a diode limiter is that diodes are real, not ideal. I actually wanted the non-ideal forward voltage drop to set my 1.4Vpp limit, but I didn't want the forward diode resistance to be real, I wanted it to be the ideal of zero. Even though a typical Rd for a Si diode was on the order of 30Ω, this was too much for my simple shunt limiter as it allowed the relatively "soft" limit I was seeing (the Rd isn't linear in the knee region either, which adds to the conundrum). After much consternation and gnashing of teeth, it occurred to me that the way to get what I needed was to raise the limit level to something on the order of 5 Vpp, and then attenuate the result. Attenuation of the signal after limiting in this way would then reduce the degree of backlash. I used a pair of 2.5 V zener diodes to produce +/-2.5 V bias points to apply separately to each of the limter diodes. This resulted in a nice, solid 5.6 Vpp limit that behaved as advertised. I then passed the limited signal through a resistive divider to get 1.44 Vpp which was my original design point.

It turns out that this is essentially what the IC-901 does, but it didn't occur to me until after I tried the diode offset idea. The IC-901 limits the modulation in an op-amp stage, then the op-amp is passed thorough a 22:1 resistive divider. I had been looking at the 901 circuit for a long time, it would have been nice to have seen this a couple of days earlier.

The output of the post-limiter filter still showed some distortion, but this indicates that the Q of the 4 pole filter is probably a too high. Since it isn't supposed to operate in full limit under normal circumstances, the distortion wasn't bad enough to try to eliminate. Finally, I could get back on track.

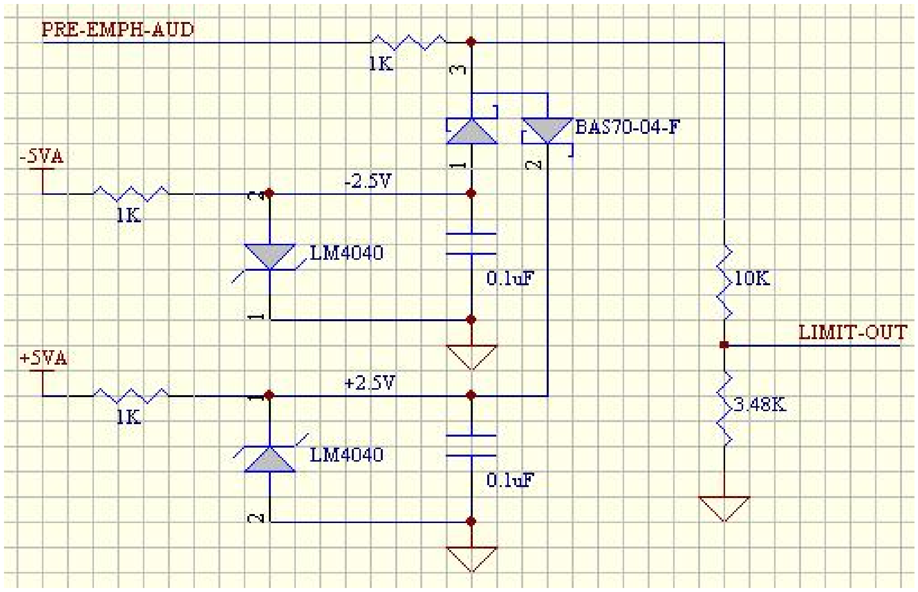

Figure 5. Revised limiter schematic

The limiter fix still left other issues regarding the pre-emphasis response. Early checks of the response indicated that there was some early roll-off on the high end. Also, the high-Q of these filters was producing some overshoot which results in some increase to the output beyond the limiter level.

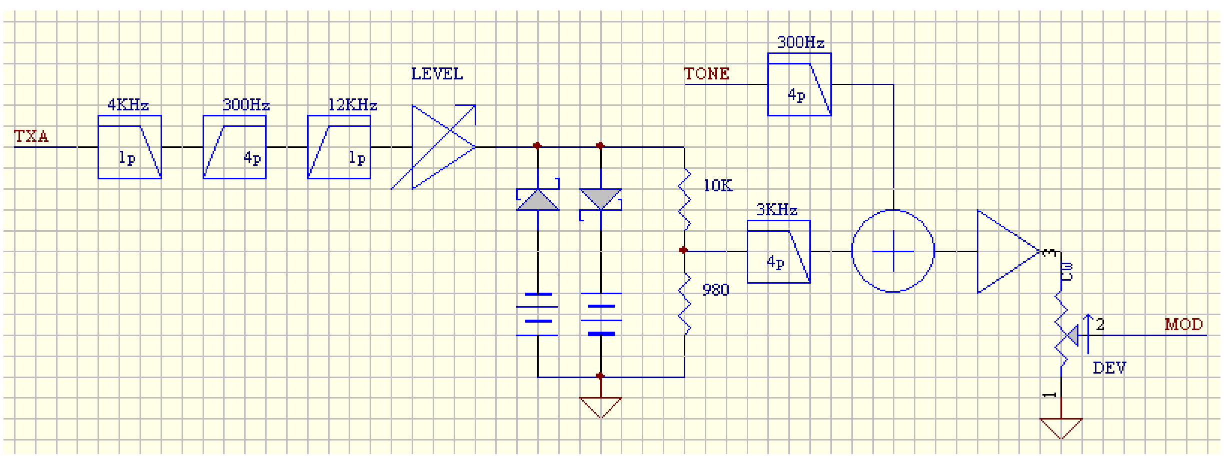

Figure 6. Final FM MOD filter chain for ICOM UX module

In order to try to reduce the overshoot effects, I re-arranged the filters to move the CTCSS clearing filter (a 4 pole, 300Hz high-pass response) in front of the limiter. A single-pole low-pass filter was also added to the front end to reduce any high-end noise (which might drive an op-amp stage into saturation). The diagram of Figure 6 illustrates the filter chain for the modulator. The resulting response, plotted in Figure 7 shows a curve that closely matches what I was hoping for. The low-end has a bit more roll-off than I wanted, but this wasn't a game-ender (depending on the on-air sound of the whole set-up, of course).

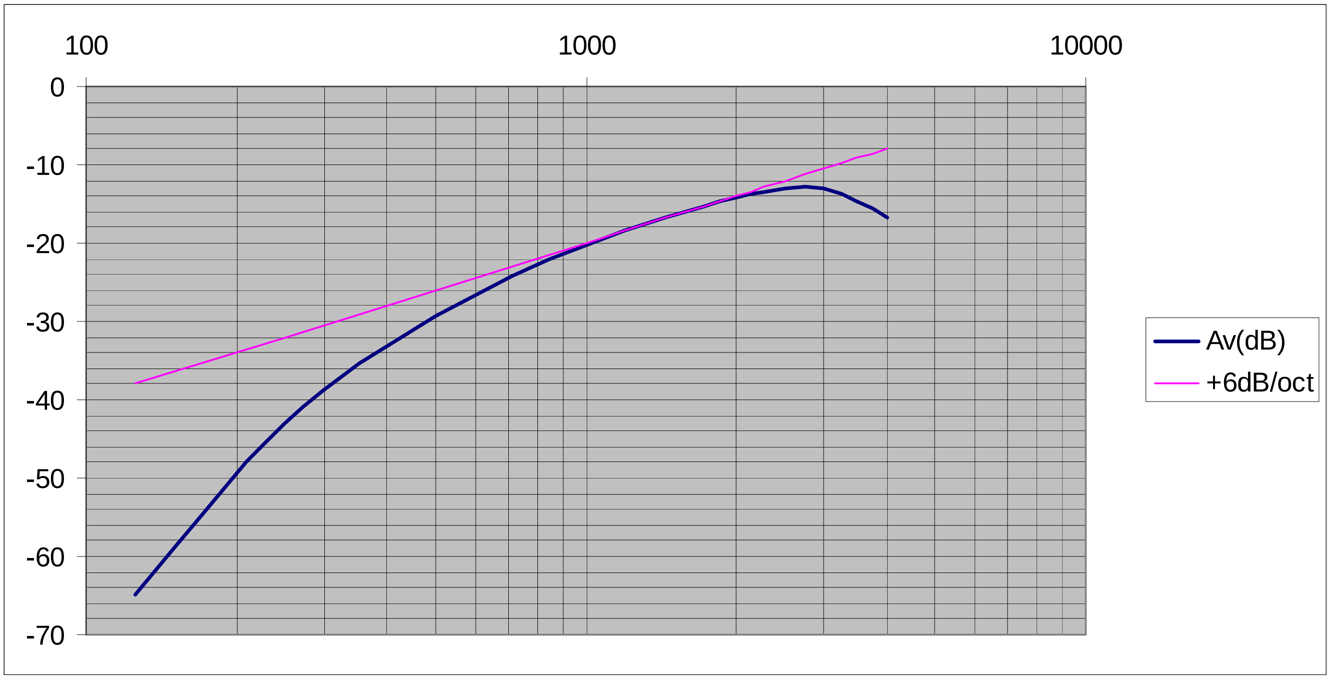

Figure 7. Measured frequency response of FM modulator circuit.

Quick Filter ™

While thumbing through one of the EE magazines I get at work, I read one of those quick-blurbs about a company called Quick Filter Technologies that has a series of FIR digital filter devices that are targeted at audio applications. Their devices have the filter operations hard coded, with the filter coefficients loaded at run-time (from a host processor). They also have design software available to produce virtually any filter response you could imagine. It was a DSP without the overhead of DSP hardware and software development.

In short order, I was able to produce a filter (see Figure 8) that had sharp roll-off at the high (3.5KHz) and low (300Hz) ends of the typical, narrow-band FM passband. In addition, the +6db/octave pre-emphasis response was also included. The advantage of the QF parts was multi-fold: 1) Ease of design and implementation, 2) complex frequency response with steep skirts, and 3) programmable. This last advantage was key because the 2M SSB module required a completely different modulator shaping scheme (no pre-emphasis and no CTCSS band-guard). Having a filter that was re-configurable would be just what I needed to reduce the complexity of the modulator shaping circuit for my mobile version.

Even though QF has a newer device that could be used to limit the deviation, the simpler part (QF1Da512) was more attractive. Since I now know how to make a limiter that is workable, that part of the design is much more manageable now. At the moment, this is still an idea that I'd like to implement, I'll cover it in more detail in part 2 of this document.

Figure 8. Quick Filter response for narrow-band FM shaping.

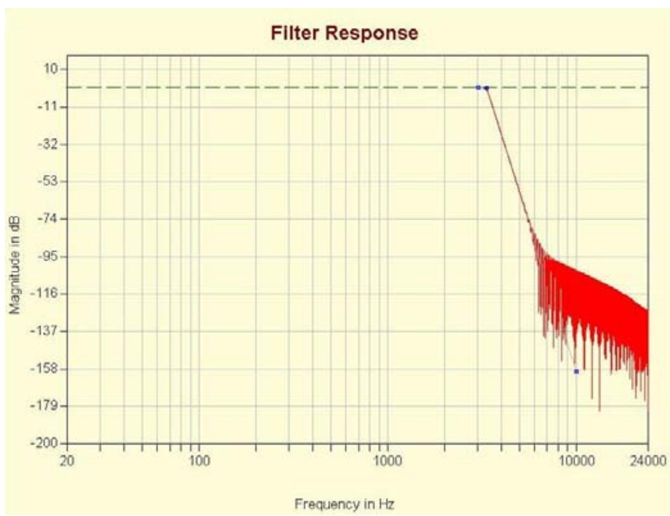

Figure 9. Quick Filter response for SSB shaping.

Closing Remarks

The ICOM UX series of modules for the IC-900 have been on the market for well over 20 years yet these devices still command a respectful exchange rate on the used market. Some of the rare modules, such as the UX-19 and UX-129, are difficult to find at any price. Most of the components are still available either in their original part# or as modern replacements. Still, there are a few critical components (most notably, the PLL chips) that are very difficult to find, at least in small quantities which is the primary threat to the longevity of these modules. Their features and small size make them attractive nonetheless and, in my opinion, they continue to be ahead of their time.

As I close this document, the Doug Hall version of the FF-89UX is ready for field trials and I am ready to begin work in earnest on the mobile version of the ICOM UX controller. Part 2 will chronicle the mobile version and discuss some transverter projects that I am considering to complement the UX controller.

References

Datasheet, PLL2001, Nippon Precision Circuits, Tokyo, Japan (no date)

Datasheet, MB87001A, Fujitsu Microelectronics, September 1995 (Edition 8.0a)

Datasheet, uPD2834C, NEC, dated 1980

Datasheet, TC9181P, Toshiba, dated May 19, 1999

ICOM Service Manual, IC-900A/E

ICOM Instruction Manual, IC-900A/E

ICOM Service Manual, IC-901A/E

ICOM Instruction Manual, IC-901A/E

ACC FC-900 Interface Schematic, 3/1/1989

RBI-1 Generic Data Manual, Doug Hall Electronics, Columbus, OH, 2/23/1995

QuickFilter QF1Da512 datasheet (Rev A18, 4/22/2010) and QFPro software (www.quickfiltertech.com)

