October 27, 2021

DTMF Input for the TI 99/4A:

Controlling a matrix switch from the TI 99/4A was simple enough, as was generating MF and DTMF tones for communication with other switching equipment. Unfortunately, that's only half the battle. My first thought was to use an off-the-shelf DTMF tone decoder and a simple microcontroller to "listen" for tones on the audio line and communicate a digital representation of those tones to the TI over a serial line. At the present time, I've run into a problem where I'm already using the entire I/O capacity of the TI (I've only got two serial / PIO controllers in the expansion chassis for a total of 16 bits I/O.)

The second problem is that the MT8870 DTMF decoder that I bought doesn't work.

The application notes for the 8870 are written with the idea of interfacing to an Arduino microcontroller and can be found at:

https://microcontrollerslab.com/mt8870-dtmf-decoder-module-pinout-interfacing-with-arduino-features/

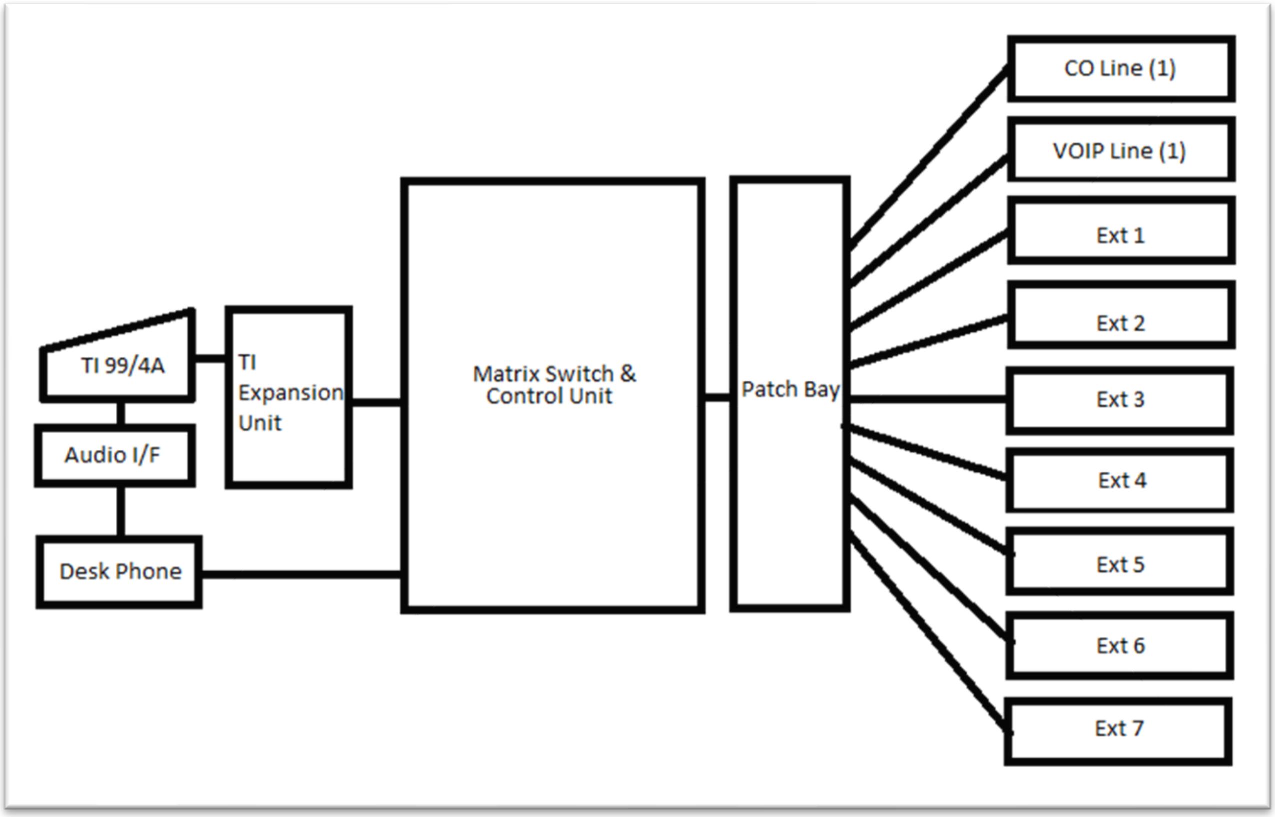
The 4-bit BCD output I'm getting with my Raspberry Pi test-machine seems to be more of a random number generator. At the present, I'm getting proper codes for digits 4, 5, 6, 7, 8, 9 and nothing reasonable for 1, 2, 3, *, #, and A, B, C, D (the "fourth column" of tones widely used in control and switching applications.) I've ordered a board from another vendor, but it seems unlikely to get here before the end of the month.

Another problem I've discovered is that, while my matrix switch works very well for a single connection, adding multiple connections can create additional spurious connections between unintended stations. The solution I've come up with is to use a "T" type flipflop to control a latching relay on the output of the matrix switch. The software has been modified to clear the X & Y control values in the A & B registers after a set or reset command – the previous behavior was to keep the X & Y values and simply add the subsequent connections to the retained set values between connection cycles.

I've also made a serious dent in my stock of general-purpose logic gates, and can only come up with a total of 10 T-style flip-flops. Reducing my ambitions for the PBX somewhat, I'm temporarily changing the wiring to reflect a subset of the original array – a total of two CO lines, eight extensions, and three simultaneous connections.

This is more like a "key" telephone system than a true PBX, but supply chain logistics dictate that I can only accomplish so much in the next few days.

Revised (Current) System Architecture:



"Local Call" (Extension to Extension) – Dial 60 followed by extension number (601-607)

"Outside Call" – Dial 9 to get an outside line. Wait for phone co. dial-tone. Dial the complete 10-digit telephone number.

"Incoming Calls" – Automatically ring into the desk phone (extension 0) and extension #1. Call may be transferred using transfer option on TI console or by dialing a "feature code" on extension #1 (to be determined.)

-- Paleoferrosaurus