The IC-901 Tone Unit Clone 5/08/21

Joseph Haas, KE0FF

I recently obtained a couple of IC-901 radios. They came with a fair compliment of accessories, but one of them was missing the "Tone Unit" which is a CTCSS encoder. This "option" wasn't really an option, it came standard with the radio when new. Thus, there are few of them to be had in the after-marketplace. There are many encoder options available, but some assembly would be required to allow the serial protocol of the IC-901 to control the CTCSS selection.

I've done CTCSS (and tone generation in general) on microcontrollers for decades using DDS (Direct Digital Synthesis) techniques. While I wasn't instrumental in their development, I have put them to good use on a number of occasions. DDS allows for precision tone generation with relatively simple hardware and software.

My go-to tiny MCU is the SiLabs C8051F531 (the 530 is also compatible). I've used it for several of my projects. One of its many advantages is that it's GPIOs are 5V compatible. I've recently been introduced to the ATtiny series from Microchip and have been rather impressed with its capabilities. However, the clock system of the 3217 (which I have some infrastructure to support) didn't appear to readily support a crystal source for its timer systems. I was all geared up to go with the new processor, but decided to stick with what I knew would work.

To this end, I copied the code from my HM133 microphone interface project as the seed for this one. There, I used the MCU to generate DTMF tones from the serial data codes produced by the ICOM HM133 (and HM151) microphone. This application would be a simplified subset of that project.

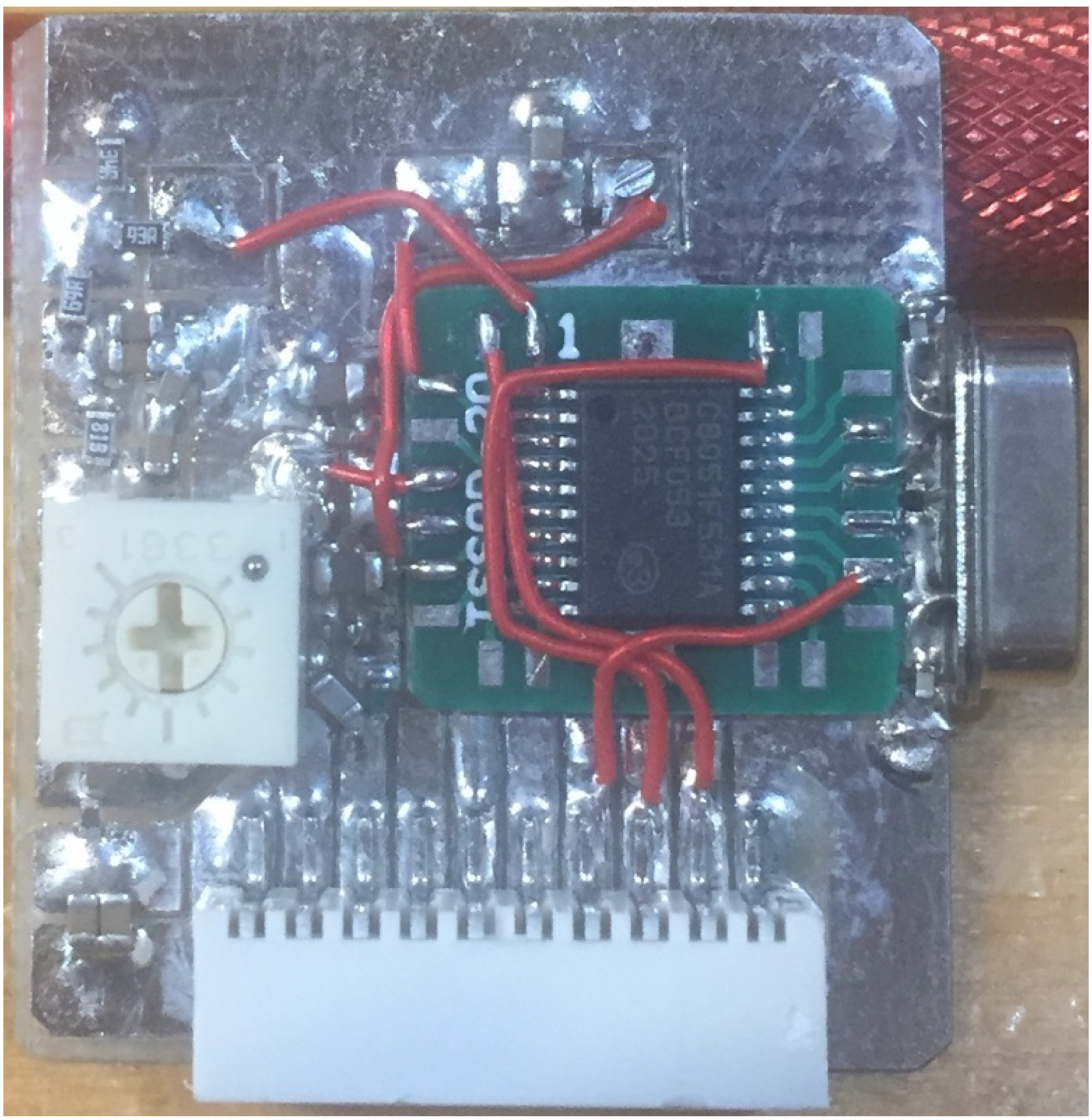
Since the F531 doesn't have an on-chip DAC, I used one of the PWM features to generate the analog output. The result is a pretty decent pure tone adjustable by very small frequency increments. By oversampling the DDS clock (the DTMF application runs at about 23KHz, while the CTCSS app will run at 6991 Hz) a 2-pole RC filter does a great job of removing the PWM carrier artifacts.

The DDS code requires processing at each interval of the sample rate. On the F531, the PWM cannot initiate an ISR or easily trap the condition of the start of period. Thus, a timer ISR was needed to accomplish the DDS processing. The timer maintains synch with the PWM simply by using the exact rate that the PWM is set to generate.

After removing the un-needed code from the seed application, the CTCSS code occupies less than 900 bytes, with about 300 of that allocated to a sine table and a CTCSS frequency lookup table. Everything happens in interrupts, so the main loop is empty except for a line of code to put the MCU into IDLE mode (basically, a wait for interrupt condition).

While the F531 has an SPI peripheral, it was not suitable for this application. Here, there is no chip select, only a strobe signal. We need the last 8 bits sent before the strobe went active and the SPI peripheral would not accomplish that task. Thus, it was necessary to design a “bit-bang” SPI. An edge interrupt grabs the serial clock from the radio and shifts the data into a memory register while another edge interrupt grabs the strobe line. The strobe interrupt triggers the data register to be acted upon (lookup the CTCSS tone and enable or disable the tone output).

The Prototype:



The chicklets (a technical term describing any two or three-terminal SMD device) above the pot are level adjust and filters for the tone signal. The three wires attached to the connector are the data, clock, and strobe lines. The colorburst crystal is to the right helps keep the frequencies on target as opposed to using the on-chip oscillator. The remaining chicklets are for power supply bypassing and reset. The back side is bare save for a small patch of copper to which the crystal case is soldered.

The MCU is soldered to a “SCAB” (Supplemental Circuit Addition Board), a term that ends up being appropriate if not savory. Over the years, I have developed several of these for different packages that I can deploy for prototyping. They literally solder to the base PCB (which is bare FR-4, 0.030" thick, 0.5oz Cu) which was etched with an Xacto knife and a particular pair of tweezers that I happen to have which are very good at grabbing copper foil and tugging without letting go (the 0.5oz thickness helps a lot with this... carving 1oz Cu is doable, but with increased difficulty). The breakout pads are small, but just big enough that they can be managed without much effort.

This is an image with the temporary programming cable attached: