The µBITX is an inexpensive 80m through 10m low-power (QRP) SSB and CW transceiver from India. It is a descendent of the BITX single-band radio kits. It is sold as a partial kit requiring only mechanical assembly with no soldering. It includes a metal case. The project website is https://www.hfsignals.com/. Gigaparts is a US distributor. I paid $209 for mine last month.

The µBITX uses an Arduino Nano as its controller. The Nano is installed on a board with a 2.8" 320 by 24 color touchscreen display and a wave generator based on the SI5351 DDS chip. The SI5351 can generate three waveforms and they are used as first and second IF's and a BFO. In a twist on the name, the board is referred to as a Raduino. A more complete description of the operation is on the above website.

The rest of the radio hardware is on a second (analog) board. This includes mixers, filters, control relays and a RF and audio amplifiers.

The µBITX is designed to be modified and there is an active community on the BITX20 forum on GroupsIO. The µBITX has a built-in speaker, as well as front panel jacks for headphones and a microphone (which is included in the kit), but no connection for a computer for running soundcard digital mode programs such as WSJT-X (FT8) or FLDIGI.

On the analog board there is an 8-pin header labelled AUDIO1. It provides audio in which is the same as the front panel. It provides three pins for received audio, including one that is the fixed level audio fed to the audio amplifier for the speaker and headphones. This audio level is unaffected by the volume control.

As it is designed to be modified the µBITX came with a mating 8-pin connector with loose wires. PIN 1 has the mic audio in. This is the leftmost pin on the header looking at it from the front. On my unit, the wire for this pin is purple. PIN 2 (blue) is ground. PIN 4 (yellow) is the fixed level audio out.

I drilled a hole on the back panel and installed a panel mount 3.5mm TRS jack. I wired PIN 1 (mic audio) to the RING terminal, PIN 4 (audio out) to the TIP terminal and PIN 2 (ground) to the SLEEVE terminal.

I then made up a Y-cable with three TRS plugs to connect this jack to soundcard MIC IN and SPEAKER OUT jacks. On the radio end, all three conductors are used. On the soundcard end only the TIP and SLEEVE conductors are used. The SLEEVE is common to all three. On the MIC IN plug, the TIP is connected to the TIP of the radio plug. On the SPEAKER OUT plug, the TIP is connected to the RING of the radio plug. I placed a small section of red (mic) and green (speaker) shrink wrap over the cable near the corresponding jack.

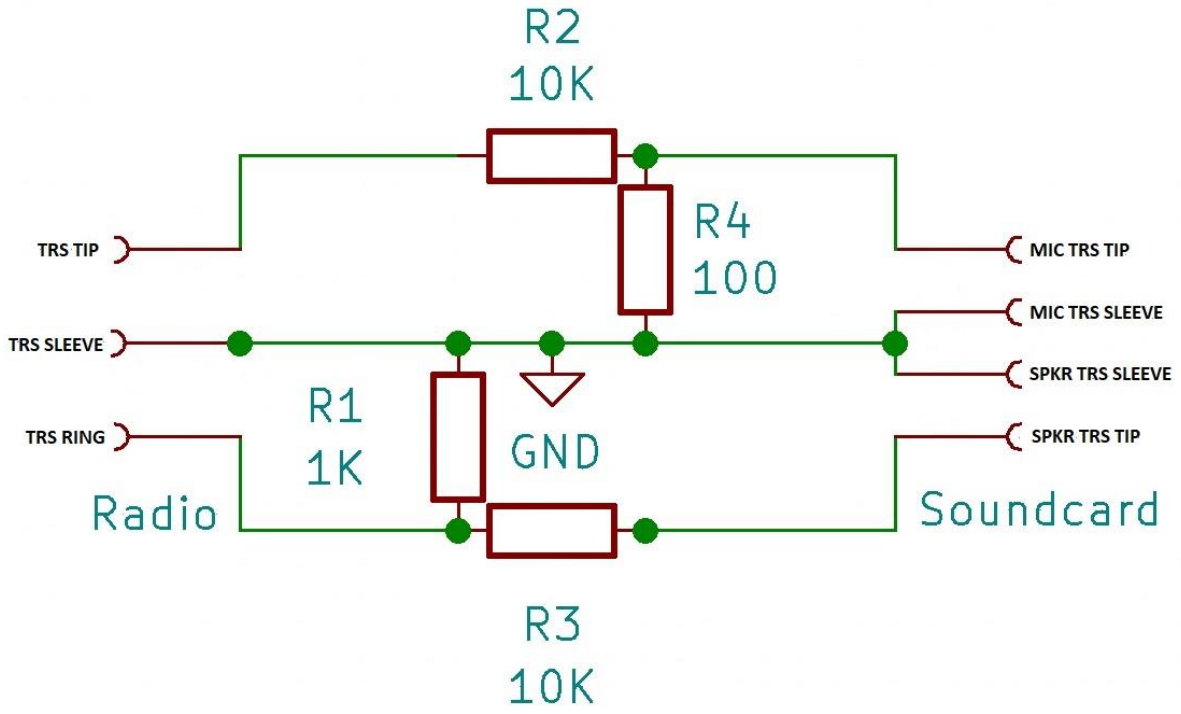
The Arduino Nano has a USB Mini-B connector used for programming. It can also be used by a program running on the Nano as a serial port. The µBITX program uses this connector to provide a CAT radio control interface. The CAT on the stock µBITX program emulates a Yaesu FT-817. I planned on using CAT commands to control Rx / Tx (PTT) switching.

I used this cable to connect the radio to a $7.99 Sabrent USB soundcard from Amazon Amazon.com: Sabrent Aluminum USB External Stereo Sound Adapter for Windows and Mac. Plug and Play No Drivers Needed. [Silver] (AU-EMAC): Computers & Accessories). When the USB cable was connected to the computer, Windows added a microphone and speaker audio devices. I specified the appropriate audio devices and rig (CAT) interface in the WSJT-X program. WSJT-X uses the Hamlib library to provide radio control. Unfortunately, the FT-817 emulation with the stock µBITX program was not sufficient for Hamlib

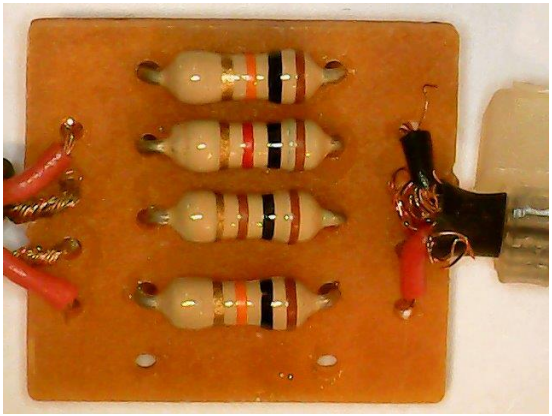
and WSJT-X could not completely connect to the radio. It would initially connect, but then fail whenever I tried to activate PTT.

As I mentioned, there is an active community on GroupsIO. At least one of the members, Reed, has developed an alternative µBITX program for the Arduino Nano. Reed's firmware also implemented FT-817 CAT, but WSJT-X was able to connect to and control the µBITX with his program.

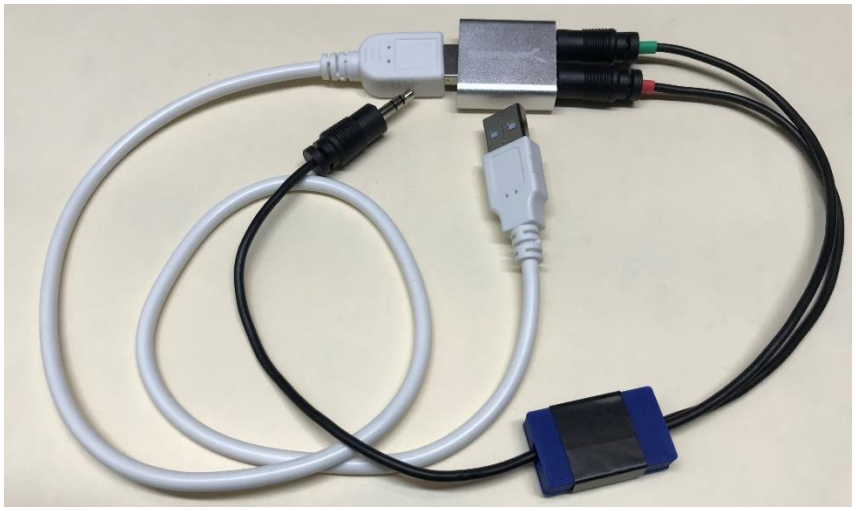
I was able to make FT8 contacts, but the audio levels to and from the radio were too high. I had to adjust the levels to almost minimum. I needed to pad the levels. With some experimenting, I found some suitable resistor values.




I initially soldered the resistors at the junction of the Y cables, but while it worked well, the result was ugly and fragile looking. I made a small PCB.



I then 3D printed a small enclosure for the PCB and connected the cable to the USB soundcard.



Everything works well and I have made several FT8 contacts.