A Digital Interface

By Pete Juliano, N6QW [n6qwham@gmail.com](mailto:n6qwham@gmail.com)

So, Ok, you are an entrenched homebrew, CW QRP, minimalist enthusiast and here is this guy from California trying to lead you to the digital dark side. Yes, that is exactly what I am doing but maybe only to share some of my experiences and information with you. Disclaimer: upfront you must have an SSB rig tuned to USB to have this work on the several digital modes that are mentioned.

I have been at this hobby a long time and aside from doing the WSPR thing several years ago with my Softrock V6.3, I am not deeply involved with digital operations. But seeing as I have a few SSB rigs available for experimentation coupled with some new digital modes I felt it time to once again test the waters. Coincidentally a new program from K1JT, Joe Taylor, called WSJT-X has many of the digital aspects rolled into a single program which are simply menu selected. You must download this free program to the computer you will use.

Two of the currently popular digital modes are WSPR (Weak Signal Propagation Reporting) and FT8. WSPR has been around a long time and essentially has you tune a specific frequency and listen for other stations. Periodically your transmitter, under computer control, is turned on and other stations listen for you. Most often stations run QRP power levels like 30 dBm signals (1 watt), some even less. I have been spotted 10,000 miles away running 500 MW.

The key feature to this program is the real time database where you can see who you have spotted and who spotted you. Today 40 Meters is a DX band and spots spanning great distances is often the common experience. WSPR uses a two-minute time block and so you can watch your lawn grow during the two-minute intervals. Thee 40 Meter WSPR frequency in 7.038600 MHz USB. Included on the WSPRNET webpage is a world map which captures the spots specific to your station. I learned a lot about my antenna as the spots are consistent NE SW

FT8 is a new addition to the digital array and this format is much quicker as each pass is 15 seconds and is more QSO like. A station sends an CQ call and the computer screen shows the station who sent it. Boom, double click on that call and the computer takes over by responding and then listening for a return. The computer will continue doing this until the person responds back OR another station picks him up. Once the connection is made the computers on each end exchange signal reports, confirm the reports and send 73’s. It is the latest craze; but some naysayers have likened it to remote quickie sex. A variant called FT8CALL now adds to the sequence by letting the operator add text and other information, so it is more like a QSO. The FT8 frequency is 7.074 MHz USB.

When I was using the Softrock V6.3 much of the interconnect to the rig and the computer program was done entirely in software within the computer. With a standalone rig, this requires some hardware to do the interconnect. A little time with the Internet resulted in a configuration that will do that task. Pre-made inexpensive circuit boards are readily available; but also, there had to be some adjustments made to work with modern computers.

The first board is a sound card interface kit from KF5INZ available on eBay for around $7 USD. This board has a couple of 600:600 Ohm modem transformers and some keying circuits that trigger and opto-isolator switch from either a DTR or RTS signal via computer Serial Port. That is the first issue as most modern computers don’t have a Serial port as everything has shifted to USB. That was one of the first nuts to crack. Although initially I did use the Serial port on an old Windows XP Pro computer; but then later shifted to the USB. Knowing what I know today I would not buy the kit, but just build the modem isolation part. Essentially, I have abandoned the Serial Port keying.

I also have abandoned the XP Pro machine and now the system is running on a small form factor Windows 10 NUC computer that is about the size of a CD case and 1 inch high. This computer has only an earphone output, so I added a plugin USB Sound card Dongle from Sabrent (about $7) as that provides the audio in/out capability. For the triggering of the PTT, Adafruit Industries makes a small USB to Serial board that is quite small (CP2140). One of the outputs is RTS –perfect. But the output logic is only 3.3 volts whereas the Serial Port is 10 Volts. So, I needed a way to trigger the PTT from 3.3 Volts. A simple 2N3904 transistor switch, a SPDT relay, isolating diode and one small resistor handles that chore. I have successfully loaded an earlier version of WSJT-X on to a Raspberry Pi3B but am having a bit of difficulty with the current version that has the WSPR and FT8. Just need a bit more time. But think of the possibilities.

The WSJT-X has a settings tab that requires you to enter your call sign, grid square and the method of keying (RTS) and the COM Port to be used. There are some other items that require checking the block, like double click to automatically respond to a CQ and to disengage the transmitter once your computer sends 73’s. Spend a little time with this setup and look at all of the tabs. I didn’t and then wondered why nothing worked.

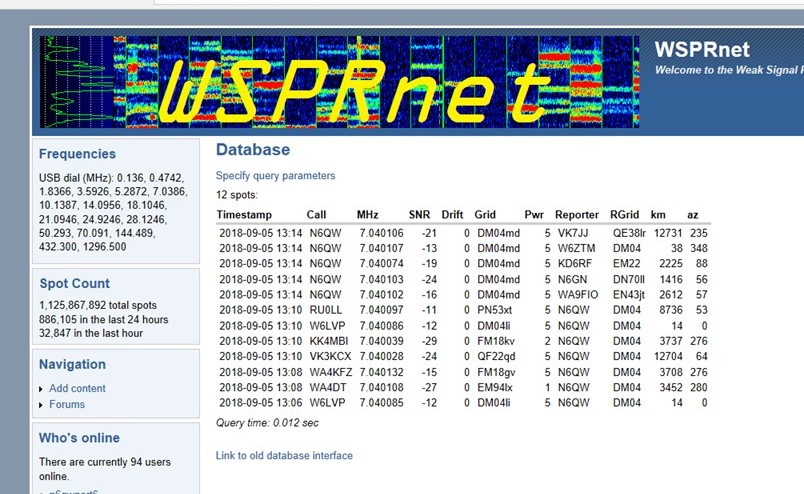
Oh, a cool feature of the WSJT-X –it is linked with hamlib. When you call a station, it tells you the azimuth (like if you have a beam) and the distance from you. Pretty cool

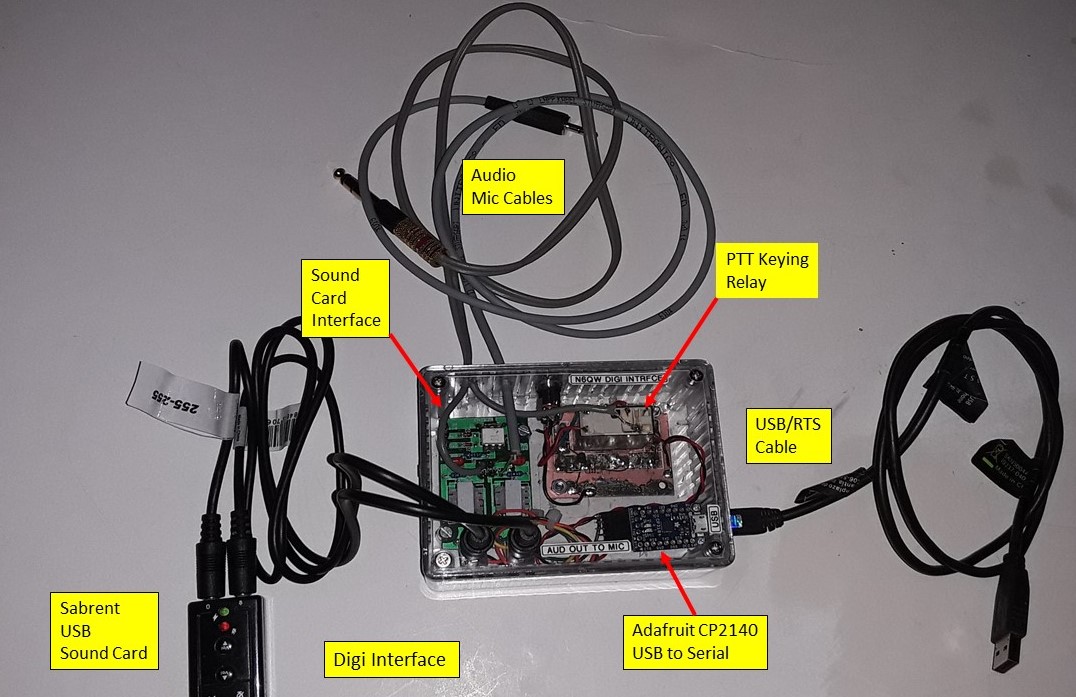
I am happy to report that my interface box has been in operation for about a week now and works perfectly.

73’s

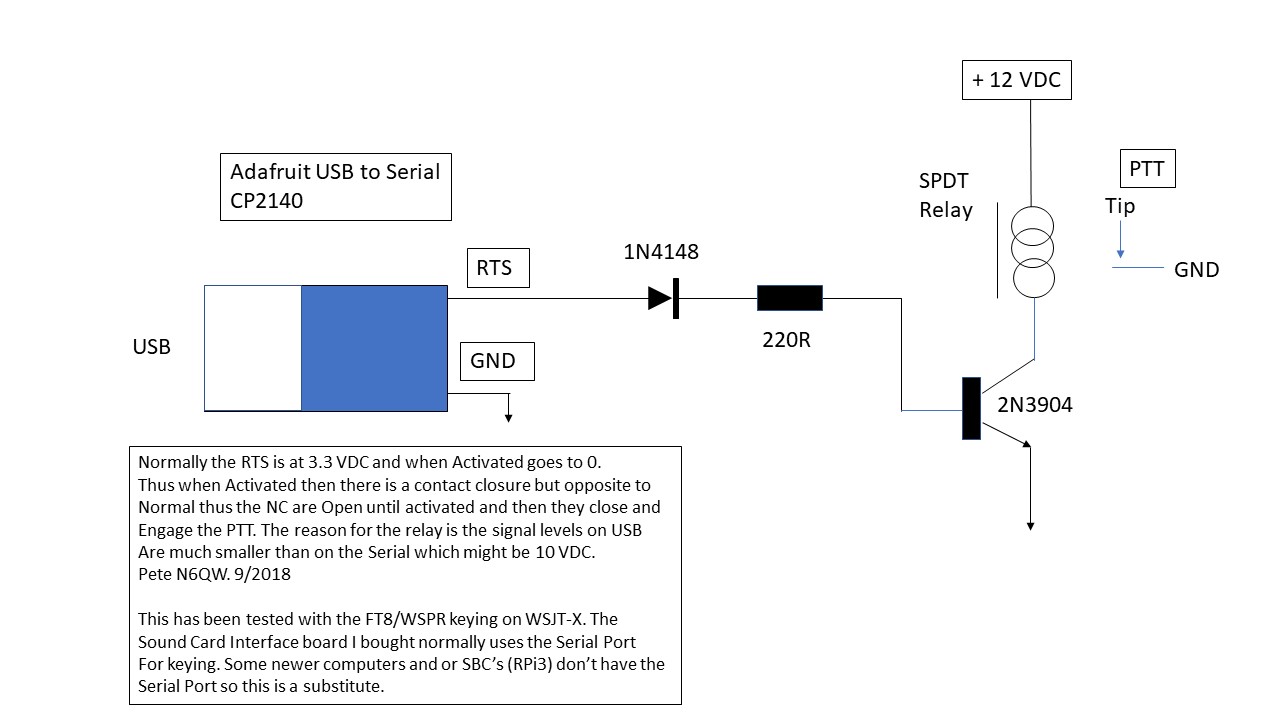
Pete N6QW

Below is a page from the WSPR database. I was running 5 watts (37 dBm) with a droopy dipole. Imagine being heard 12000 km away with that kind of lash up. But that is a significant bonus to digital operations, especially FT8, as it levels the playing fields running low power and modest antennas. So that may be a real appeal to those with QRP rigs who have antenna restrictions or live in apartments. Following that is a photo of the actual interface hardware.

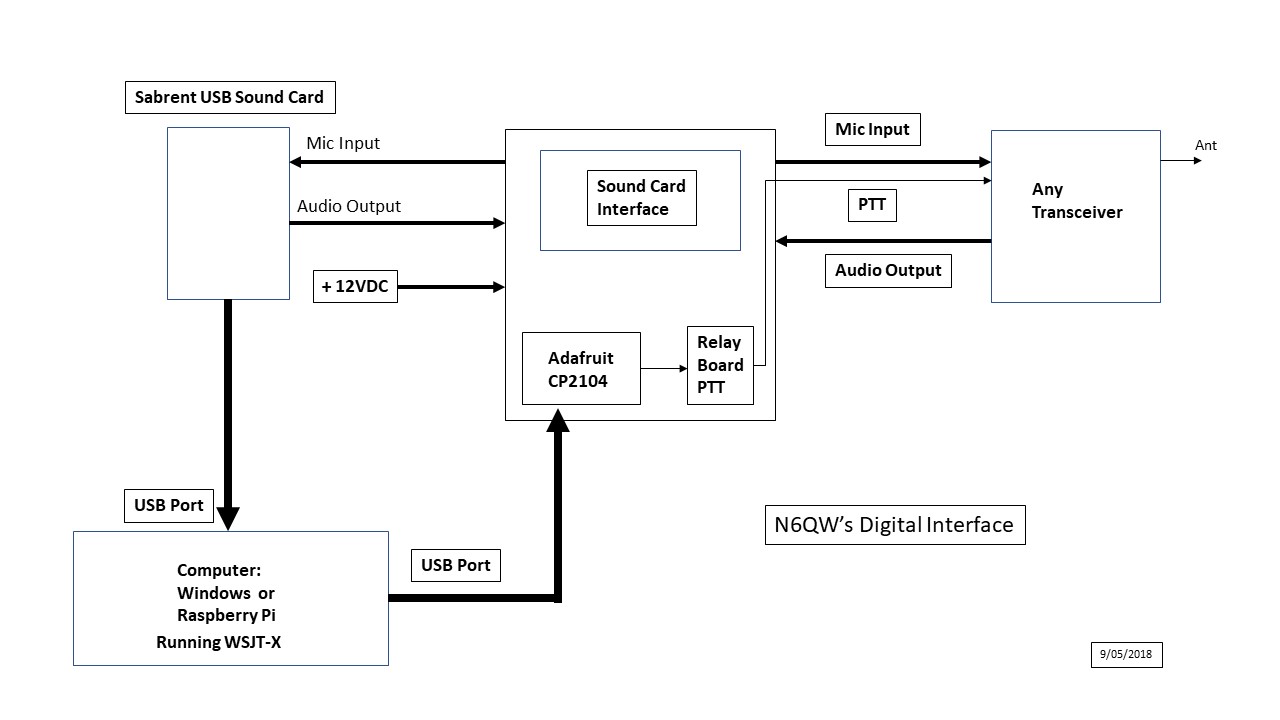




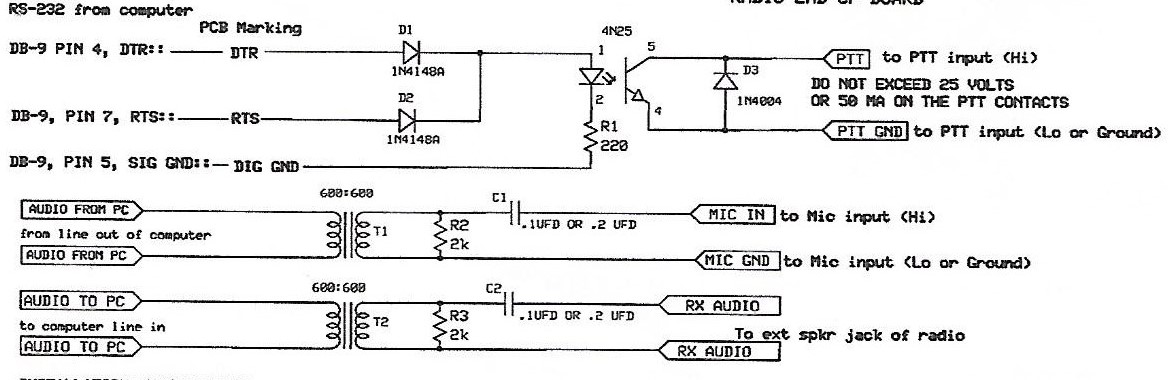
This is the modification to the Adafruit CP2140 to key the PTT on the transceiver



Below is a pictorial of the component/board interconnects.



Finally, this is the schematic of the Sound Card Interface board. Based on my current experience I would just build the modem interface which is two transformers, two resistors and two capacitors. A friend in the UK was going to purchase the board from the US and the shipping cost was $20 for a $7 board. So, skip the Serial Port Interconnect and go with the USB.



A couple of cautions. Upon initial setup with the Adafruit board connected to your computer using the control panel, find the COM port associated with the USB as you will need that to enter information into the settings page. The next caution involves the computer, which by the way had me baffled for about an hour or two. I could see the signals on the waterfall; but they were not being registered on the panel and neither was anyone copying me when I transmitted. Like a ton of bricks, it hit me. My automatic update of the computer clock was not updating automatically. A synchronization with Internet Time and all was good.

Unlike other programs and since my rigs are homebrew I must physically tune the rig to the WSPR or FT8 frequencies. With other software auto tuning of the rigs to the frequencies is done automatically. Appliance operators would consider this an inconvenience but as a home brew guy it is just another day in the shack.

I should note that you will need a super stable VFO to work the digital modes. My homebrew rigs (nearly two dozen) all use the Si5351 PLL or AD9850 DDS. It may be possible to use an analog VFO but unless it is using a X Lock 3 frequency stabilizer (from G4GXO) your results may be marginal.

I will be looking for your signals on WSPR or FT8.