

This information captured from Helen & Rudi Wiedeman's Website at http://home.comcast.net/~wiedemann/software_tnc.htm

Software-based Sound Card TNC for Packet, PSK31, Wefax, etc.

BACKGROUND

Digital communications over radio always needs a way to convert the sound signals over the air into bytes for computers to understand. This is usually done using a box called a 'Terminal Node Controller' (TNC). This device performs at least 2 stages of conversion: 1) change audio signals to a bit-stream; and 2) interpret the bitstream into a particular 'protocol' such as AX.25 (packet radio standard) or TCP/IP (internet standard).

The advantage of performing these functions in a piece of dedicated hardware is that, once 'programmed', you can disconnect the computer and it will continue to do it's thing. This is great for an APRS 'tracker box' which usually does not need a dedicated computer. However, this dedicated hardware can be a bit expensive. For example, the popular Kantronics KPC3+ is \$179 plus tax.

But if you're more interested in 'real-time' digital communications modes like Packet or PSK31, you'll almost certainly be using a computer as part of the process. Since most modern PCs have 'sound cards' for audio input and output, there is a way to use this internal hardware along with special Windows software to perform the same functions as a dedicated TNC. The only task remaining is to build a suitable interface system between the radio and the computer. This can be done nicely for under \$30! It is also much smaller than a hardware TNC and requires no external power which is great for portable operation.

INTERFACE CONCEPT

Any connection between a radio and a computer must achieve three goals: 1) isolate the radio and computer 'grounds' to prevent noise corruption of the signals; 2) get the signal (volume) levels in the right range to be useful; and 3) control the Push-To-Talk (PTT) signal to trigger the radio to transmit the audio sent from the PC.

This project assumes you will be using a 'Handy-Talkie' (HT) portable VHF radio for packet communications. Many HTs already have a 'dual-prong' remote mike socket on the radio. If your radio doesn't have this, then you will need to purchase a small conversion cable compatible with your HT's special socket. It's a good investment anyway, since most remote 'speaker/mikes' are configured for the dual-prong style of connection.

Once you understand the principles, you can easily convert your interface to work with VHF mobiles and HF base stations (PSK31 and Wefax). It also assumes you'll be using a PC with Win98, Win2K, WinME or WinXP (Win 3.1, WinWG, WinNT are not supported). Win95 **may** work, but is not recommended.

PARTS LIST

All the components you need to build this interface can be purchased from your local Fry's and Radio Shack stores. The cost (not including enclosure) is about \$25.

Note: This interface has been optimized for the Yaesu VX-5R HT. If you are using a different radio, please adjust the values of the 2.2K resistor and the 10uF capacitor which is connected to

the **tip** of the 2.5mm **stereo** plug (some manufacturers wire 5VDC to the center ring on the plug). For example, the Alinco DJ-191T uses a 10K resistor and a 1uF capacitor. Check the "packet connection" section of your manual for the appropriate component values for your interface.

CONSTRUCTION TOOLS

You will need the following items to build your interface:

- ✖ Soldering pencil (15W-25W) with a 'fine' tip
- ✖ Solder and tip-wetting sponge
- ✖ Small needle-nosed pliers
- ✖ Small diagonal cutters
- ✖ Wire strippers
- ✖ Panavise or other way to hold board and components for soldering

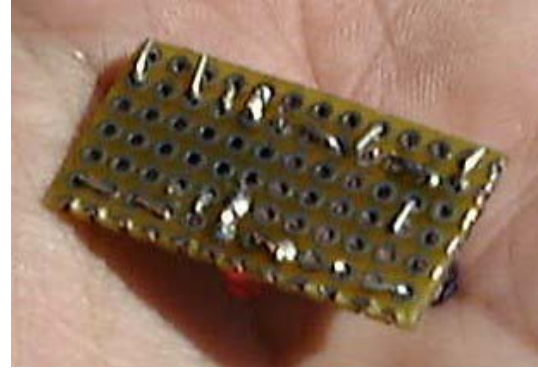
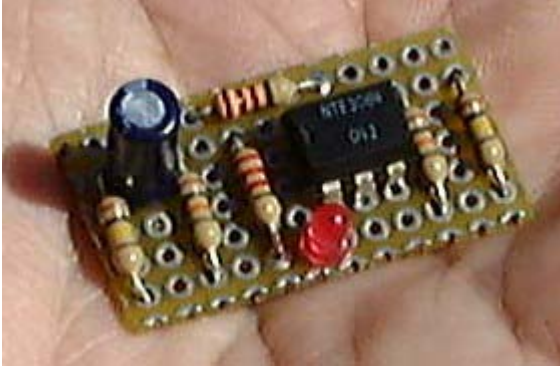
PC PREPARATIONS

You will need to download and install two free software packages: 1) [AGWPE](#) controls the sound card and comm port; and 2) [AGWterm](#) is used to communicate with other packet stations. Once installed, you will need to 'configure' AGWPE for a sound card style interface. You can download an excellent [document](#) (self-extracting pdf file) which details how to do all this. However, you'll find that the interface circuitry discussions are somewhat scattered and the component values are not specified, which is why I pulled together the hardware information for this project. Just focus on all the settings in AGWPE for sound card operation.

Note: Although most sound cards will work fine with AGWPE, it's always good to check it for compatibility. One simple test can be done by connecting only the receive side of the interface. Simply start AGWPE, right click on its icon in the system tray and select 'Sound Card Tuning Aid. Notice how smooth the waves are when packets come in. Now beacon and look again at the waveform. If it has turned 'jagged' and fails to decode further packets, then either you need new drivers or this sound card won't work with AGWPE.

HARDWARE CONSTRUCTION

The circuit is built on a 'perf-board' style card which allows a great deal of flexibility in component placement and connection (see [layout](#)). It also facilitates seeing how the parts are connected to ensure it matches the [schematics](#). The components are placed so that their 'tails' can stick through the board and be bent over to connect to the next component wherever possible, minimizing the number of additional wires on the board. Solder these 'on-board' connections first. Then lead the the 'tails' of the cable wires through the component side of the board and make the connections to the rest of the components on the underside of the board. Make the final on-board connections using a bit of 22-24 gauge stranded wire. Now solder the leads from the two isolation transformers to the board. Check all solder joints and verify against the schematic that all circuit connections are made.



RECEIVE-SIDE DEBUG

Plug the 'speaker out' into the radio and the 'mike in' to the PC's 'line-in' or 'microphone' connector. Right-click on the little speaker symbol in the task bar in your system tray (in the lower-right corner of your Windows desktop). Set all volume levels to midpoint and un-mute the mike input. Pull down the 'file' menu and click on 'advanced'. Navigate to the advanced controls for 'recording' and be sure to uncheck the box for any 'input level boosting'. Click OK until you're back to the desktop. Set your radio to the APRS frequency of 144.390 which has a steady stream of packets. Set your radio volume to about 25% of maximum. You should now hear the packets on your PC. If not, check your connections and sound card settings. Right-click on the AGWPE icon in your system tray and select the 'Sound Card visualization tool'. Click on 'sine wave' view. You should see a modulated sine wave pattern every time a packet comes in. Adjust your radio's volume setting so that the tallest waves fill about 1/3 the height of the black window. You should now see the little green light flash on the 'modem' symbol in your system tray every time a packet comes in. This means AGWPE is decoding these packets. You should now notice messages appearing in AGWterm. You are now successfully receiving packets. Some older HTs will have a slow responding squelch to incoming packets. If you find you're decoding only the loudest packets, then try lowering the squelch level till it's always open.

TRANSMIT-SIDE DEBUG

Plug the 'mike in' to the radio and the 'line out' connector on the PC. Start AGWPE and plug the DB9 connector into your Comm port. Adjust the radio to use an open simplex frequency such as 145.550 MHz. Go to Properties in AGW Term to set up your beacon text and then browse to its location. That will let you send a stream of packet data by using the yellow whistle icon on the toolbar. Right click on your speaker in your tray to open the volume controls. Use the 'wave' output slider to adjust the sound level when you 'beacon' to be about the same as the level of incoming packets. You should notice a little red light on the 'modem' icon in the system tray every time the radio transmits. If your radio fails to transmit when you beacon, check the polarity on your LED. The short lead is the cathode which goes to pin 1 of the IC. Then check value of the resistor connected to pin 5. A high value may keep your radio from transmitting. A low value may keep it in transmit mode all the time. Typical values are from 2.2K to 10K.

Once you've got both receive and transmit working, then plug in a good outdoor antenna and fire up WinPack which is a full-function packet program. (be sure to set up your personal info in the Options menu, and set the Comms host to AGW.) Tune your radio to the local packet frequency of 145.050 and start 'connecting' with other operators on the channel. You should now be able to do packet just as well as when using a hardware TNC.

ADVANCED CONCEPTS

If you plan to use the SoundCard TNC exclusively with one radio and PC, you may want to shorten the cables to suit your wiring needs and use the optimal R & C values. You can easily modify your circuit to be able to switch between two pre-set conditions by using a 'Berg strip' style of single pin PC mount connectors. Use a piece with 3 pins and wire the outside pins to your alternate components. Then use a computer board jumper to bridge either side to the center pin and wire that to the rest of your circuit. So for example, you could select either a 10K resistor (reduction to 1/11 of source level) or a 50K resistor (reduction only to 1/3 of source level) to work in combination with the original 100K resistor. You could also use the same approach with the 1uF and 10uF capacitors.

If you want to play with different radios and computers which use significantly different volume levels, you could replace the 10K/100K resistor pairs with a 100K single turn PCB mount potentiometer (Radio Shack P/N 271-284, \$1.19) Wire the outside pins of the pot to the where the transformer leads connect to ends of the 2 resistors, and the center pin to where the shared resistor connection was. You could also replace the resistor from pin 5 of the IC with a 10K potentiometer (Radio Shack P/N 271-282, \$1.19) and just tie the center pin to one of the end pins to make a variable resistor. Set the pots (and your PC volume control sliders) to about midpoint and start experimenting with the pots to optimize the receive and transmit levels.

For the more ambitions among you, AGW does in fact support two *simultaneous* interfaces because the PC interface is of course stereo. The channel you wired above uses the left channel. Just use the other channel (center ring on the plug) to a second circuit which is connected to a second radio. Remember to configure AGWPE to use dual port operation. That way you could easily be running APRS on one radio while doing packet on a second radio (different frequencies of course).

HAPPY INTERFACING !