Inexpensive Homebrew Soundcard/PTT for TNC

Gordon Gibby KX4Z
Updated: October 3, 2016
Updated: October 12, 2016

Purpose: This paper is written to help you put together an extremely inexpensive terminal node controller that will work for modes such as AX.25 packet, PSK31, MT63, and WINLINK WINMOR.

An updated blow-by-blow construction article with more detail than this overview paper is available at: http://www.qsl.net/kx4z/TNCConstruction.pdf

Introduction

You can put together an extremely inexpensive terminal node controller using a cheap soundcard dongle. In fact, for modes that don't require fast back-and-forth, you can actually do "digital" by holding your radio near your computer and simply using acoustic coupling with modern soundcard-based programs such as FLDIGI.

That doesn't work so well for modes that require back and forth for error correction, such as connected AX.25 packet, or winmor WINLINK. With those modes, you need electronic control of the push-to-talk on the radio. With a soundcard, this can often be provided easily with a simple transistor or transistor/relay circuit.

Many people get hung up on *where* the processing of the signals occurs. Whether in a "hardware" purchased TNC, or in freebie software on a personal computer really makes relatively little different -- in both cases, there is a microprocessor *somewhere* executing some sort software to make or decipher the audio signals. Either system may be better than the other, depending on the smarts of the particular software and the signal involved.

Many of these protocols are not that complicated. For example, AX.25 1200 baud packet involves two audio frequencies that are WAY apart. Software should be able to easily discern the "high" versus "low" frequency even if they aren't perfect. RTTY also involves two different frequencies quite well separated. In engineering, there is the concept of "good enough" and of "diminishing returns" for systems that are linearly better, but exponentially more complicated or expensive. With that in mind, I present an extremely inexpensive (on the order of $10) soundcard-based TNC using a cheap soundcard dongle and a homebrew push to talk circuit. The circuit isn't original with me, I took the idea from another web page (which I can't immediately re-find).

Component acquisition:

Below are the components, as well as suggested sources. Feel free to obtain them from other sources; these are just examples.

Adafruit sound card dongle $6 https://www.adafruit.com/products/1475?

gclid=Cj0KEQjw6uO-BRDbzujwtuzAzfkBEiQAAnhJ0IUJGeu1nRxC7ReKMqXn3GcSvaFREAY5aXYbMq_EoGgaAic08P8HAQ

Transistors: 3 NPN transistors, preferably 2N3904 or 2N2222
Electrolytic capacitors:
To smooth the power voltage, 25-100 microfarad, rated 15VDC or higher, qty 1
Audio coupling: 1 microfarad, >= 15 VDC; qty 2.
Capacitors, ceramic or mylar, 35VDC or higher
0.01 microfarad, qty 7
0.1 microfarad qty 1
Resistors, 1/4watt or 1/2 watt
4700 ohm
2200 ohm
10K ohm, qty 2
47K ohm qty 1
100K ohm qty 1
Diode, any style or voltage or current, 1N4148 or 1N4004/5/6/7 useful, qty 2
Trimmer potentiometers, anywhere from 600 to 10K ohms, solder leads/terminals, qty. 2
LED diode, any color, rated in the range of 10-20 mA
Perforated hobbyist circuit board, with or without plated contacts.
Homemade RFI inductors: wind 10-12 turns of fine wire on any non-metallic form such as a small dowel rol or outer plastic portion of a bic pen. Drilling a hole through and through at each end and passing the wire through it will help to secure it. This will form approximately a 0.6 microhenry rf choke.

Optional Reed Relay: https://www.digikey.com/product-detail/en/coto-technology/9007-05-00/306-1062-ND/301696

Various colors of #22 or #24 stranded wire; need about 10" of each for ground, signal in, PTT out, +V in.

Schematic: http://www.qsl.net/kx4z/Schematic4.pdf

NOTE: Schematic now includes series inductors on the lines from the transformer -- which makes it much more immune to RFI.

Circuit description
An audio frequency amplifier takes the full output of the soundcard, with some RF filtering, and drives a simple diode rectifier, with a small filter capacitor chosen to give a very small

turn-on and turn-off delay. This drives the bases of one transistor switch to drive an indicator LED (which is optional) and another to drive an open collector transistor switch that can close positive-voltage push-to-talk circuits. If you wish, you can replace that with a relay, which will increase the RFI protection as well. A suitable relay is: https://www.digikey.com/product-detail/en/coto-technology/9007-05-00/306-1062-ND/301696 Don't forget to put the backwards-diode across the relay to protect the connected transistor from the reverse voltage spike on relay turn-off.

Simple RFI filters consisting of a series inductor and a shunt capacitor (on the side of the inductor AWAY from the transsceiver) on the mic, speaker and ptt wires coming from the transceiver dramatically reduce the impact of RF on the circuit.

Construction

Split open the $5 Adafruit sound dongle and the color coded USB wires will provide you with +5 V. On each of the 1/8" sockets, the "ground" is at the far end, towards the edge of the board, and the speaker or mic signal will be the contact closest to the center of the board. You can easily solder small stranded wires to both the mic and speaker/headphone connections and pick up the ground connection also. I found it was easiest to get the +5 Vby gently stripping the red wire of the USB wires. Use good technique building the circuit with relatively short leads to minimize RF pickup and use the bypass capacitors (47mfd and .01 mfd) to reduce RF pickup and any hum.

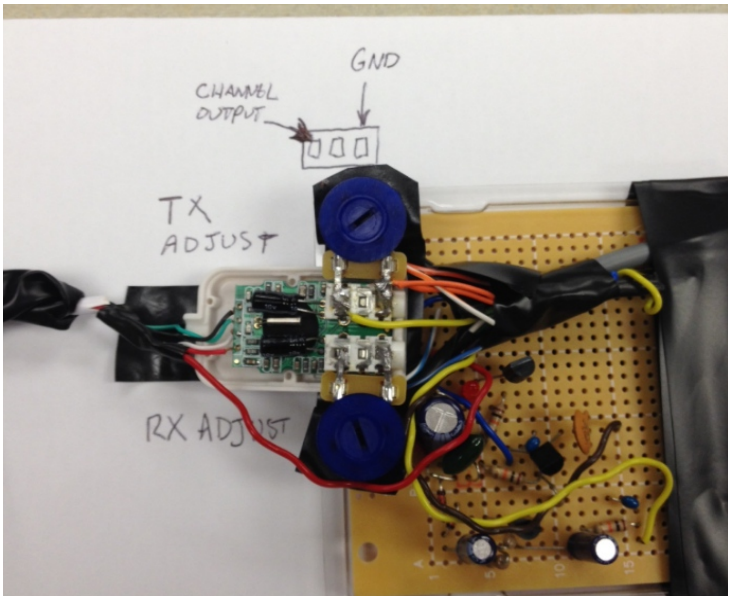


Photo of the opened USB sound card dongle and PTT circuitry.

USB color code
RED = Vcc (5VDC)
BLACK = Ground
White and Green are data wires (don't use)

Jacks for earbuds and microphone have very simply made terminals. Ground on both is to the right, and channels (L/R) are to the left. In this photo I soldered the volume trimmers directly to the jacks, but it might be better to put them on the perfboard and connect to the

*jacks with stranded wire. When you are done, cover all exposed wiring and shield either with aluminum foil or a metal enclosure. Put ferrites on cable*s.

RFI Protection

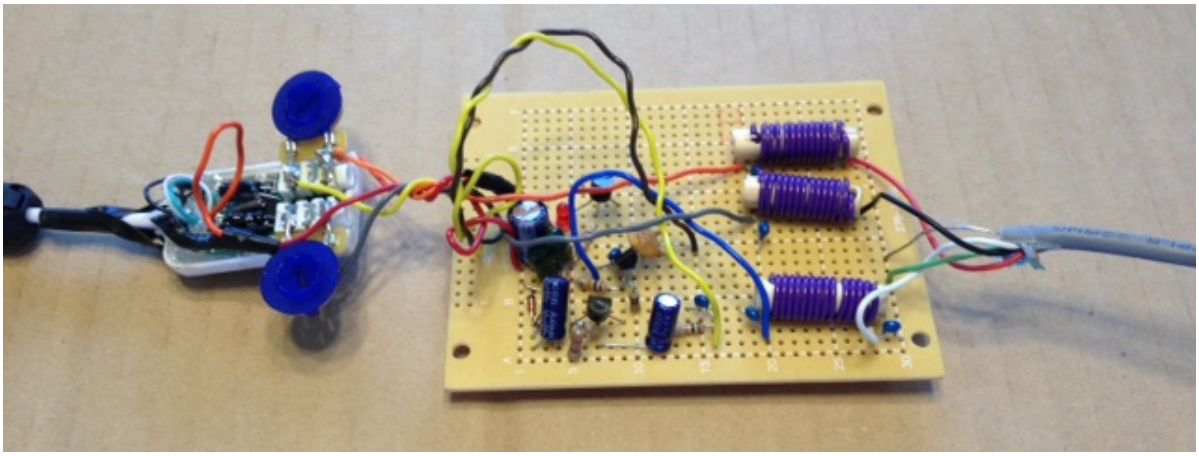
Even a small 2-meter handitalkie can produce surprising radio frequency interference. I've watched a Baofeng UV5RA peg an analog VOM on the 10volt dc scale with the leads connected to nothing. The leads were great antennas!

If RF voltages picked up by audio leads exceed the logic voltages or semiconductor turn-on voltages --- circuits are going to do unexpected things. So the goal is to reduce RF currents flowing in your audio leads, and to shield them where ever possible.

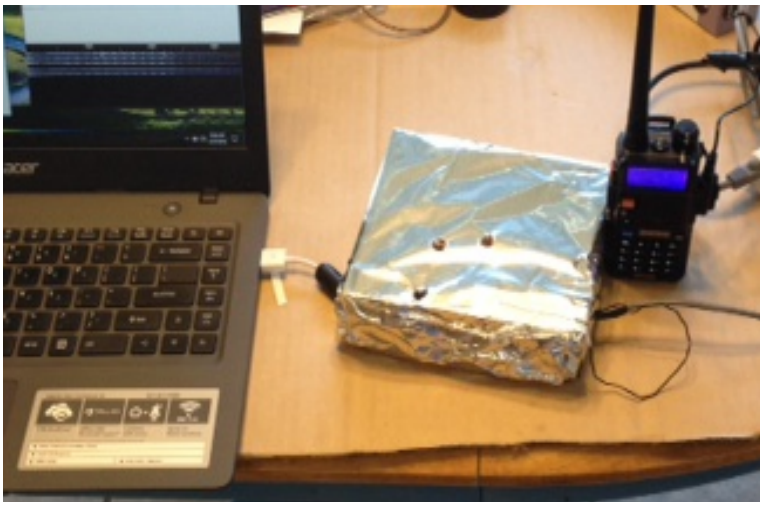
Many inexpensive cables for ham radio products are using unshielded wires....even for microphone circuits. I prefer to use shielded wires where ever possible, with the shield connected to ground at least at the "computer" end. 4-conductor shielded wire isn't that hard to find -- here is some meant for guitars on Amazon that would work very well to get signals from this TNC to your 2 meter radio's mic/speaker connectors:

https://www.amazon.com/Shielded-4-Conductor-Guitar-Circuit-hookup/dp/B00UB29MRO/ref=sr_1_4?ie=UTF8&qid=1474044543&sr=8-4&keywords=shielded+4+conductor+wire

Almost all TNC's are built inside metal (shielding) enclosures. When you finish making your homebrew soundcard TNC, shield it inside a metal enclosure. A very easy way to do this is to put it inside a small cardboard box (e.g., that granola bars come in) and then cover with aluminum foil.



Circuitry before being shielded.



Holes punched to allow access to mic and speaker trimmer pots, and to see the push to talk LED

Setting the Volume

1. Set the RX volume control in the middle of the range where your software properly decodes some normal-sounding beacon or packet station.
2. St the TX volume control (with your computer or Raspberry PI set to 100% output volume) by monitoring your transmitted signal, and setting the TX gain just below the point where the received audio no longer gets any louder.