

Safe (malware preventative) file transfer modem software app for MSWindows and Linux through Software Defined Radio or audio hardware. Thoroughly proven schematics for TRRS/TRS adapters to RJ45 Cat6A telephone, and similar, included.

By transferring data in a discontinuous (ie. analog) format, none of the computer hardware interfaces will present either keyboard traffic that could transfer and install malware, nor any firmware reprogramming that could cause other devices to do this, nor any malformed network packets to buffer overflow.

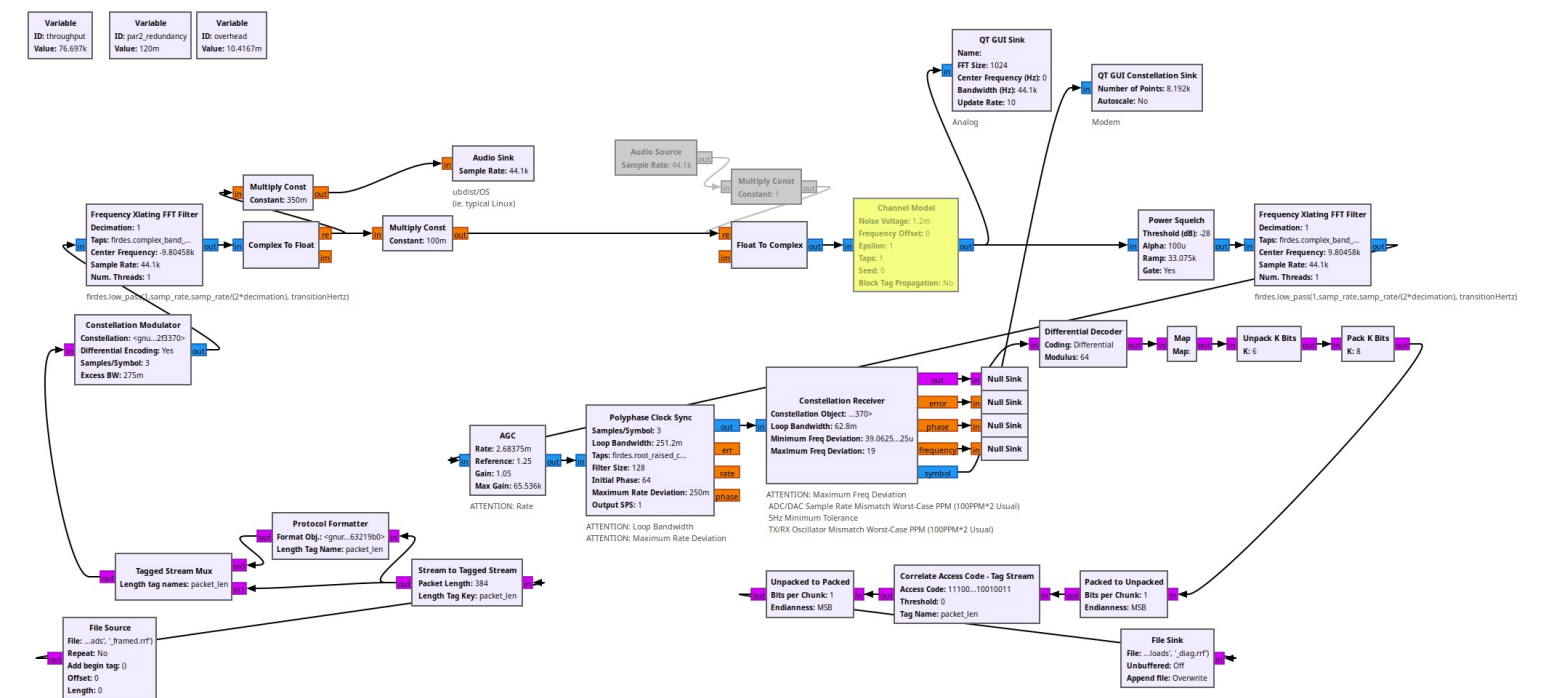
As such, this is a necessary Commercial-Off-The-Shelf workaround for the long history of malware abusing the insane proliferation of computer interfaces with completely inappropriate functionality (eg. reprogrammable SATA firmware wear leveling, BadUSB flash drive reprogramming, SD Card slot SDIO serial/keyboard/etc functionality, parallel port functionality, etc).

This system has been tested with real world audio hardware and does function as designed at decent throughput under challenging conditions.

Audio output must be just below the point of distortion (ie. 99% volume). Audio input volume MUST be kept lower, analog signal should appear at one quarter full input amplitude. Attenuation of more than 3dB to TRRS microphone ports is apparently unnecessary. Coupling impedance and DC load at TRRS microphone connection should be ~1.6kOhm. Higher DC impedance at TRRS microphone port will result in substantial amplifier current noise and EMI sensitivity. TRRS headphones connection DC load should be <<40kOhm. Heavy capacitive coupling >>250uF is apparently acceptable to bypass any DC impedance to improve signal strength, though any amplifier output without ~3000Ohm resistive coupling will impose a near short circuit to other outputs on the same line. Capacitor polarity should always place the negative side of the capacitor towards the line (ie. towards the load on the line to ground).

Auditing is a nice bonus for the audio interface - a mere continuous analog recording may be kept without possibility of tampering by the malware.

Optical Data Diodes may be formed by widely available ADC/SPDIF and DAC/SPDIF fiber optic adapters.



## Usage

```
./ubiquitous_bash.sh
```

Some computer systems may require audio jack retasking.

```
hdajackretask
```

# Standards

input.xrf  
framed.xrf  
output.xrf

Any automation of the preprocessing/postprocessing step by 'gr-pipe' may use a separate 'flowgraph' pointing to 'framed.xrf' . Similarly, shell/batch files may perform this step.



# Conclusions

\*) Framing arbitrary size files with stock GNURadio is not possible, due to 'Stream to Tagged Stream', at least with the stock version, blocking output until a multiple of the specified length for tagging.

\*) Cygwin installation of GNURadio may be possible. Possibly also gr-pipe .

## Safety

# Reference

<https://lists.gnu.org/archive/html/discuss-gnuradio/2017-07/msg00249.html>

[https://wiki.gnuradio.org/index.php/Correlation\\_Estimator#:~:text=Correlation%20Estimator%20The%20Correlation%20Estimator%20block%20correlates%20the,to%20get%20a%20time%20and%20phase%20offset%20estimate.](https://wiki.gnuradio.org/index.php/Correlation_Estimator#:~:text=Correlation%20Estimator%20The%20Correlation%20Estimator%20block%20correlates%20the,to%20get%20a%20time%20and%20phase%20offset%20estimate.)

<https://dsp.stackexchange.com/questions/68306/16-qam-gnu-radio>

[https://wiki.gnuradio.org/index.php?title=QPSK\\_Mod\\_and\\_Demod](https://wiki.gnuradio.org/index.php?title=QPSK_Mod_and_Demod)  
'might have an ambiguity of 90 degrees in the constellation. Luckily, we avoided this problem by transmitting differential symbols.'

<https://discuss-gnuradio.gnu.narkive.com/KrFwQ9Fz/why-no-phase-ambiguity-in-digital-bert>  
'scrambler/descrambler pair is insensitive to the phase ambiguity'

[https://www.gnuradio.org/grcon/grcon17/presentations/building\\_a\\_moderately\\_complex\\_mode\\_with\\_spare\\_parts/Dan-CaJacob-Building-a-Moderately-Complex-Modem-with-Spare-Parts.pdf](https://www.gnuradio.org/grcon/grcon17/presentations/building_a_moderately_complex_mode_with_spare_parts/Dan-CaJacob-Building-a-Moderately-Complex-Modem-with-Spare-Parts.pdf)  
'Correlation estimator and 2nd Costas Loop clean up the ambiguity'

<https://github.com/greatscottgadgets/hackrf/issues/1159>  
MAJOR - SEVERE - 'Possible solution: at TX startup, have the M4 not run baseband\_streaming\_enable until the first two 16KB transfers have arrived from the host, meaning that the M0 has a full buffer ready to transmit.'

[https://www.reddit.com/r/RTLSDR/comments/o7owrl/hackrf\\_frequency\\_drift/](https://www.reddit.com/r/RTLSDR/comments/o7owrl/hackrf_frequency_drift/)  
<https://imgur.com/a/ggsuPTm>

<https://stackoverflow.com/questions/54946638/punctured-convolutional-codes-in-gnu-radio>  
'Gnu Radio Puncture pattern' 'Puncture size' 'Delay values'

[https://aaronscher.com/GNU\\_Radio\\_Companion\\_Collection/Audio\\_modem.html](https://aaronscher.com/GNU_Radio_Companion_Collection/Audio_modem.html)

<https://wiki.gnuradio.org/index.php?title=CygwinInstallMain>

[https://wiki.gnuradio.org/index.php/Packet\\_Communications](https://wiki.gnuradio.org/index.php/Packet_Communications)  
MAJOR - 'Functionally it replaces a 'File Source' block and a 'Stream to Tagged Stream' block. The advantage of this block is that when the input file size is not an exact multiple of the selected packet length, the remainder at the end of the file is not lost in the 'Stream to Tagged Stream' buffer. This precludes the need for a pre-processor such as the text padding program above.'

[https://raw.githubusercontent.com/gnuradio/gnuradio/master/gr-digital/examples/ofdm/ofdm\\_loopback.grc](https://raw.githubusercontent.com/gnuradio/gnuradio/master/gr-digital/examples/ofdm/ofdm_loopback.grc)

[https://wiki.gnuradio.org/index.php/Packet\\_Communications](https://wiki.gnuradio.org/index.php/Packet_Communications)  
MAJOR - 'Header Format Object' .  
MAJOR - 'Using Header Format Default and Correlate Access Code' .

[https://wiki.gnuradio.org/images/f/fd/Pkt\\_7\\_base\\_fg.png](https://wiki.gnuradio.org/images/f/fd/Pkt_7_base_fg.png)  
'Protocol Formatter' 'Format Obj'  
'Header/Payload Demux'

[https://wiki.gnuradio.org/images/8/89/Pkt\\_7\\_base.grc](https://wiki.gnuradio.org/images/8/89/Pkt_7_base.grc)  
Format Obj.: hdr\_format

[https://github.com/gnuradio/gnuradio/blob/master/gr-digital/examples/ofdm/tx\\_ofdm.grc](https://github.com/gnuradio/gnuradio/blob/master/gr-digital/examples/ofdm/tx_ofdm.grc)  
Format Obj.: header\_formatter.base()

[https://www.gnuradio.org/doc/doxygen/classgr\\_1\\_1digital\\_1\\_1packet\\_\\_header\\_\\_default.html](https://www.gnuradio.org/doc/doxygen/classgr_1_1digital_1_1packet__header__default.html)  
'packet\_header\_default'

<https://www.youtube.com/watch?v=RnAgqGR-D-8>  
<https://www.youtube.com/watch?v=VR0mej2o-SM>  
Windows... HackRF...

# Copyright

This file is part of pumpCompanion.

pumpCompanion is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

pumpCompanion is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with pumpCompanion. If not, see <<http://www.gnu.org/licenses/>>.