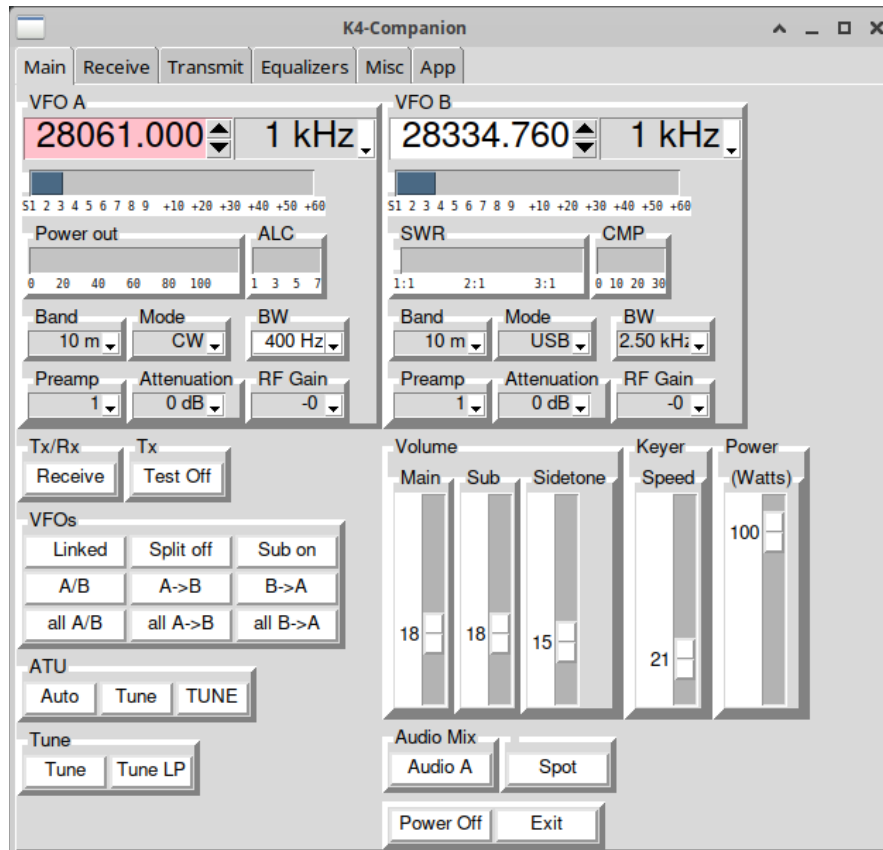


# K4 Companion

An application for remote control of Elecraft the K4 series SDR transceiver



Developed by Dale Farnsworth, W7DA

Based on a simple utility originally devised by Charles Powell, NK8O

Revision date: April 7<sup>th</sup>, 2025, based on version 1.1.33

# Table of Contents

Introduction:	Page	1
License:	Page	2
Prerequisites	Page	3
Running K4 Companion	Page	4
Configuration	Page	5
Operation & Controls	Page	6
Receive Functions	Page	9
Transmit Functions	Page	10
Equalizers	Page	11
Miscellaneous Tab	Page	12
Reserved	Page	13
Debugging	Page	14
User Notes	Page	15

# Introduction

K4 Companion is an application written in python3 that can remotely control an Elecraft K4 transceiver via TCP/IP. It currently controls the main K4 features and is very usable as is, but new features are being added all the time. K4 Companion is very configurable.

K4 Companion began life as a simple macro-sending program called K4Macro-Python, created by Charles Powell, NK8O. It has now grown far beyond a simple macro-sending program into a full-fledged remote control program for the K4.

Please send problem reports either: by sending an email, by entering an issue on github, or by making a pull request. Problem reports and suggestions are greatly appreciated.

Configuration information is maintained in a separate YAML file named, by default, k4companion.yaml. Custom configurations can be loaded with the *-config* option between the python executable and the desired configuration file.

Dale Farnsworth, W7DA  
[dale@farnsworth.org](mailto:dale@farnsworth.org)

# License

Copyright (C) 2025 Dale Farnsworth

#

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

#

# This program 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 General Public License for more details.

#

# You should have received a copy of the GNU General Public License  
# along with this program. If not, see <https://www.gnu.org/licenses/>

# Installation on Linux

One of the beauties of open source software is that it is typically cross-platform. Using Python3, the application is available on any platform that has Python3 available. It was developed on Linux, but it has not been tested on Mac OS, and Windows is under development.

Here are the items one needs to take full advantage of K4 Companion and all its features:

1. You must have a way to turn your radio on remotely. This is detailed in the K4 user manual (pages 17 & 18) and requires the momentary closure and grounding of pin 8 to pin 5 on the DB15 ACC connector. Unfortunately the Wake On LAN feature is not available for the K4, at least not at this time. Other details are beyond the scope of this document. Nonetheless, there are many operators who simply leave the radio on at all times.
2. A computer capable of running a full installation of Python is needed. This has been tested with Xubuntu, Debian 12, Chromebook, and various iterations of the Dell XPS series running a variety of installations.
3. Install portaudio (*sudo apt install portaudio19-dev* on Debian based distributions)
4. Install Python3. This is pre-installed on most Linux distributions. Please refer to readily available documentation for installation on other operating systems.
5. Install 'python-tk' (*sudo apt install python3-tk* on Debian based distributions)
6. Install 'pip3', required for the remaining dependencies (*sudo apt install python3-pip*)
7. Install 'pyyaml' using pip
8. Install 'opuslib' for audio support using pip
9. Install 'numpy', and 'pyaudio' with using (Some of these may be pre-installed) portaudio is required for pyaudio to install correctly
10. Install 'PySocks' with pip. This allows SOCKS 5 proxy connections

Notes: the typical installation of Python dependencies is with the command *pip install xxxxx* however security consciousness has caused the need for override with a number of operating systems. If your OS complains about installing the extras, use the command *pip install --break-system-packages xxxxx* . It doesn't really break anything. It installs the package in the individual user's profile. I have done many such installs and it has never been a problem. Incidentally the *pip* command can be used to install '.whl' Python files. Chirp, the well known open source HT programming software is provided this way, and using the .whl file is an easy way to keep up with changes and updates.

Although it has not been tested on Mac OS, if you are experimenting you **must** install Python 3 because the default version is Python 2, and it is deprecated. When installing the pip dependencies you **must** specify *pip3* or you will get Python 2 files that won't work. **Background colors** do not work under Mac OS. Buttons that have background colors unfortunately do not change on Mac OS. The most recent versions have not been tested on Mac OS.

The latest version is available **here:** <https://github.com/DaleFarnsworth/K4-Companion/>

# Installation on Windows

Installation on Windows follows the same principles as Linux. It is very similar, once Python3 is installed.

Here are the basic step:

1. As with Linux, you must have a way to turn your radio on remotely. This is detailed in the K4 user manual (pages 17 & 18) and requires the momentary closure and grounding of pin 8 to pin 5 on the DB15 ACC connector. Unfortunately the Wake On LAN feature is not available for the K4, at least not at this time. Again, there are many operators who simply leave the radio on at all times.
2. A computer capable of running a full installation of Python is needed. Windows 11 is the platform upon which testing has been done. With Windows 10, things may or may not work well.
3. Install Python3 <https://www.python.org/downloads>
4. Update pip *python -m pip install --upgrade pip*
5. Install numpy *pip install numpy*
6. Install opuslib *pip install opuslib*
7. Install pyaudio *pip install pyaudio*
8. Install socks support *pip install pysocks*
9. Install YAML support *pip install pyyaml*
10. From Contributions on the K4 Companion github site, download opus.dll and move it to the folder C:\Windows\System32

Download k4companion and k4companion.yaml from the github site. Place them in a convenient folder. I suggest using PowerShell to use to open k4companion. It allows the use of the Ampersand, like Linux, to fork a process into the background and gives a little more flexibility. One caveat: if you close PowerShell or your cmd program on a running version of K4 Companion, it will exit the program.

The utility is being tested on Windows and it is under development. Some known quirks are that the extensions are incorrectly assigned. The main file may download with the '.txt' extension and it should be '.py', if anything. Windows also wants to drop the '.yaml' extension. It is best in both instances to specify the extension when the downloads are saved. If you have missed any of the dependencies, Python is very good at telling you what it needs.

The latest version is available **here**: <https://github.com/DaleFarnsworth/K4-Companion/>

# Running K4 Companion

There are essentially two ways to run a Python script. To execute the script directly, it requires the file to be marked as executable. Under Linux and Mac OS, this is done after downloading the script. In a terminal window and using the command line, issue the command `chmod 755 k4companion`. It is not necessary to make any modifications or change permissions of the YAML file. The YAML file should be in the same directory or folder as the `k4companion` executable. Otherwise, the command line becomes complicated.

Methods for starting K4 Companion:

- Assuming the file has been made executable, from a Linux terminal window issue the command `./k4companion &` (enter) from the directory or folder where the script, as well as the configuration file are located. This will work with the “stock” configuration YAML. To use a custom configuration, use `./k4companion --config my_k4companionfile.yaml &` (enter)
- The alternative method, which requires no modifications nor marking files as executable is as follows: `python k4companion &` (enter). The same approach can be used here for a custom YAML file

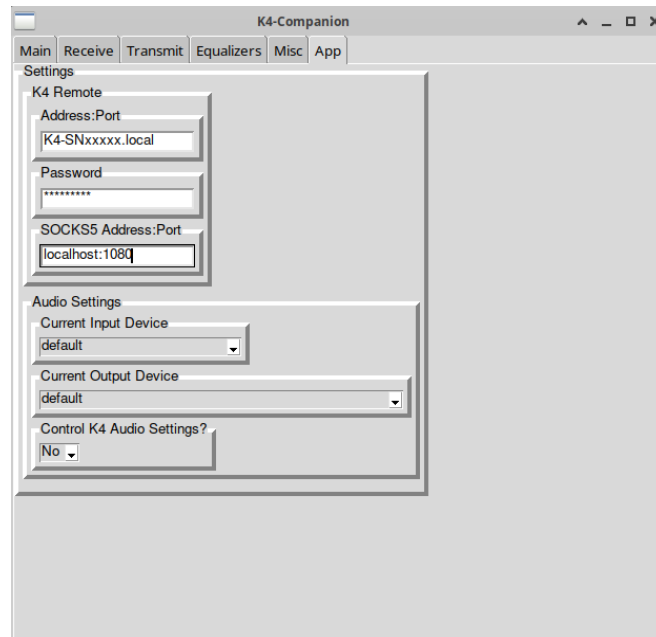
If you create a shortcut or desktop launcher K4 Companion, it must contain the full path to the executable in order to find the configuration file. When using Windows, versions of PowerShell > 6.0 allow the use of the ampersand (&) to fork processes into the background. Windows requires calling `python k4companion &` (enter) and does not recognize ‘python3’.

```
cwpowell@newXPS:~/bin$ ./k4companion &
```

```
cwpowell@newXPS:~/bin$ python3 k4companion &
```

```
C:\Users\cwpowell\Dropbox\bin> python .\k4companion&
```

# Configuration: App tab



Configuration is done through the ‘App’ tab. Configuration parameters **must** be entered but K4 Companion will save these items and open after going back to the Main tab. Items needed are:

- The remote address of the K4 to be controlled. This can either replace the ‘xxxxx’ portion with the serial number of the radio, or it can be the actual IP address of the K4
- Operation outside your LAN will require either port forwarding, an SSH “tunnel,” a VPN, or SOCKS 5 proxy. This will require a change in the IP address, depending on the configuration used or specifying the proxy server information. If you operate from a single remote address, it is possible to “open” your router to accept only that address
- The password to access your K4. For remote control this **must** be set on the radio. In addition, the number of connections to your K4 must also be specified. This can be set to one, and up to four connections allowed. No connections will be permitted if the parameter is set to zero
- K4 Companion is fully configured for SOCKS 5 proxy. Enter the proxy address and port as shown above, but only if needed. The default port is 1080 if not specified, but it can be set as above if another port is desired

Audio connections can generally be set to **default** in a typical Linux installation. Choices vary according to the Linux flavor and sound systems installed. In Windows there are a variety of settings, but generally these mention Speaker and Microphone. See **Notes on Operation** below.

**Control K4 Audio Settings?** If you are using K4 Companion to control the radio while sitting in the shack, this can be set to Yes. Otherwise for remote operations, set it to No. See **Notes on Operation** below.

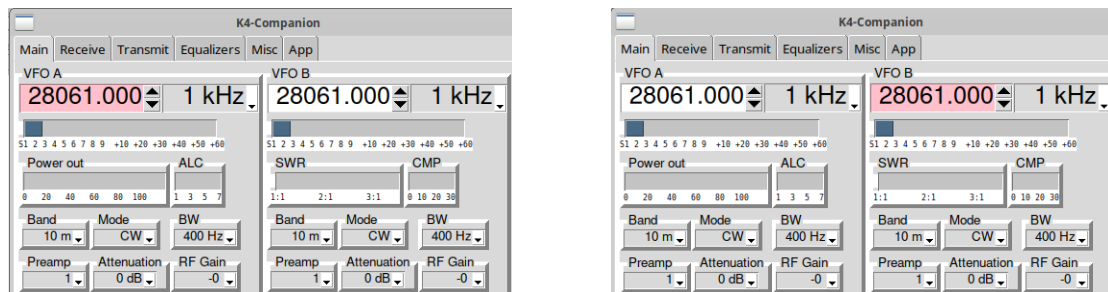
Once this tab is configured, the settings are saved automatically. No changes are needed unless something in your operational configuration changes.



# Operation and Controls

K4 Companion is logically organized with largely self-explanatory labels and functions. Each VFO can be linked, operated independently, swapped, placed into Split mode, Sub-receiver mode, or both simultaneously. Various functions are controlled in the main tab, including preamps, attenuators, RF gain, ATU functions, audio controls, keyer speed, plus sidetone volume as well as power output, and the rig can be shut down from the K4 Companion.

The active transmit window is shown in pastel pink, so it is possible to see at a glance which VFO will transmit when the radio is keyed. This prevents confusion when using Split or Sub modes.



In the image on the left, VFO A is active for transmitting, and on the right, VFO B is ready for transmit. Note on the left that the K4 is actually transmitting at 5 watts, the ALC is normal at 5, and the SWR is slightly greater than 1:1. The details available at a glance are the frequency of each VFO, the S-meter reading for each VFO (if both are active), power out either 0-10 watts or 0-110 watts, SWR, and for phone operation, compression.

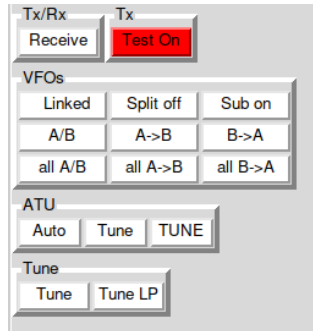
The scroll rate for each VFO can be set independently from 1 Hz to 10 kHz, and these can be modified, if desired, in the YAML file.

The VFO boxes support **direct entry**, **scrolling** with the Up←→Down arrows, or if there is a single click in the VFO box, the keyboard Up/Down buttons also allow frequency scrolling

Pull-down menus are available for each VFO to set the following parameters:

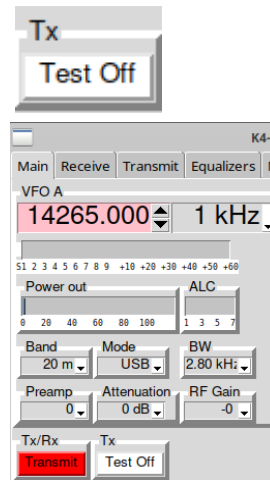
- Band selection, 160 through 6 meters
- Mode selection, including all available modes for the K4 (does not include digital sub-modes)
- Bandwidth selection from 50 Hz to 5 kHz (for AM reception)
- Preamp selection – none, 1, 2, or 3 (depending on the band)
- RF attenuation in 3 dB steps up to -21 dB
- RF gain (user configurable in the YAML file but only used occasionally)

The following controls are fairly self-explanatory, at least once you are familiar with K4 Companion



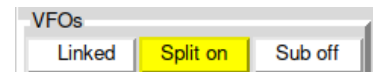
When the application opens, it sets the K4 to Test Mode, as shown above. This button lights up red when engaged but turns white when off. This prevents unintended transmission when setting up. This can be disabled at the top of the YAML file, but it is a useful feature.

Tx/Rx button has two features. If selected, the button warns that the K4 is in transmit mode, and at it turns red. Note that pressing this button does not actually put out any power. It is the equivalent of using a foot switch or pressing the PTT on the mic without actually speaking. One could use it to suppress QSK in CW mode but there is very little utility in that. QSK parameters are adjustable and the results are automatic. The button also turns red when sending CW as the transmitter is keyed.



In the next row of buttons the options are:

- Linked VFOs or Unlinked. Elecraft calls this “Band Independence”
- Split on, Split off. This button lights up yellow as a secondary warning for the change of transmit VFO, and VFO B turns pink to warn that it is now active for transmit
- Sub activates the K4 sub-receiver. This can be used either with Split mode to move the transmit VFO to B, or without. The main and sub-receivers have separate volume slider controls (see below)



The middle row of the VFO section is self-explanatory.

- Swap VFOs A/B (may be repeated consecutively)
- VFO A to B
- VFO B to A

The finals row “All” swaps or copies all parameters, equivalent to a “double tap” of the buttons on the K4.

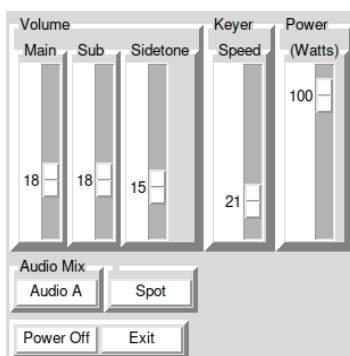
ATU controls have the following functions:

- Auto = ATU active, or Bypass
- Tune = normal ATU function (single tap)
- TUNE = extended tune (double tap) The Tune selection function in the same way the buttons on the front panel of the K4

The “Tune” buttons on the bottom row produce a continuous wave carrier

- Tune produces a carrier at the full (selected) power carrier
- Tune LP produces a low power carrier at 5 watts

Next is the bottom right hand panel segment.



The sliders control

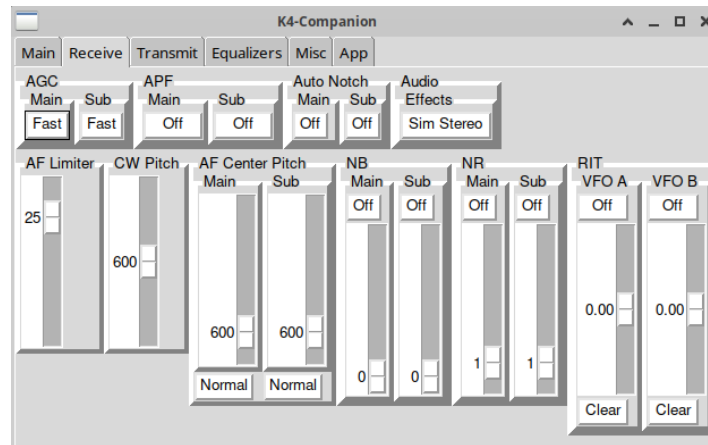
- Main volume – equivalent of the small knob on the lower left of the K4
- Sub volume – equivalent of the concentric inner knob on the lower left of the K4
- Sidetone level – useful especially if keying is done with macros or locally. Set to zero if keying is produced locally
- Power – sets power output level of the K4 from 0-110 watts

The remaining four buttons

- Audio mix selects A+B, useful in Sub mode when chasing DX, A+A = main audio only, or finally B+B = sub-receiver audio only.
- Spot will tune a reasonably strong CW signal to the chosen center frequency
- Power Off shuts down the K4 (PS0;)
- Exit closes K4 Companion but does not turn off the radio

This completes the tour of the main tab window of K4 Companion.

# Receive Tab

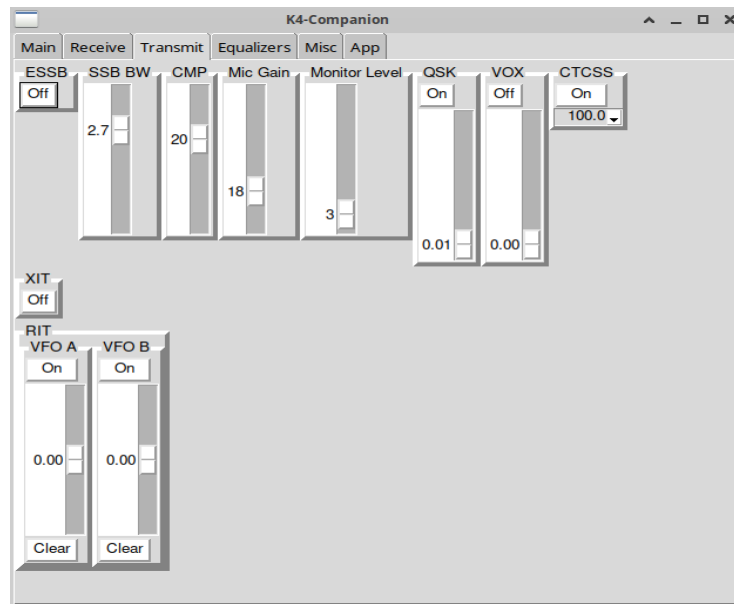


Controls for AGC, Audio Peaking filter, Auto Notch (SSB) and Audio Effects are in the top row. The function of each is explained in the K4 User Manual. Options for Audio effects include

- Simulated Stereo – introduces a slight delay in L – R audio to reduce listening fatigue
- Pitch map – moves audio from left to right as the deviation from the center frequency decreases or increases
- Off – removes all tuning and listening effects

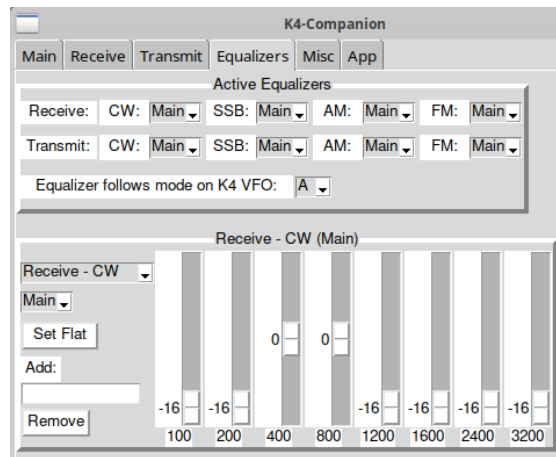
Note that the AF limiter only functions if the AGC is “Off”. This prevents extremely loud signals from blasting through at full-volume if the AGC is absent. AF Center Pitch functions by mode, and returns to the last user settings when changing modes. Noise Blanker, Noise Reduction, and RIT are explained in the K4 user manual.

# Transmit Tab



- Transmit controls include ESSB (wide-band, high fidelity SSB). When selected, the SSB bandwidth varies between 3 kHz and 4.5 kHz
- SSB bandwidth for standard or CESSB (Controlled Envelope SSB) ranges from 2.4 to 2.8 kHz
- CMP, when set > 0, specifies the level of transmit audio compression
- Mic Gain – set to optimize SSB drive
- Monitor level – useful primarily if using K4 Companion for ancillary control of the radio while operating in the shack. No audio is returned when operating remotely. This control is independent of Sidetone level on the Main tab.
- QSK control On/Off and delay settings
- VOX control
- CTCSS – sets access “PL™” tones for FM repeater operation
- XIT implementation

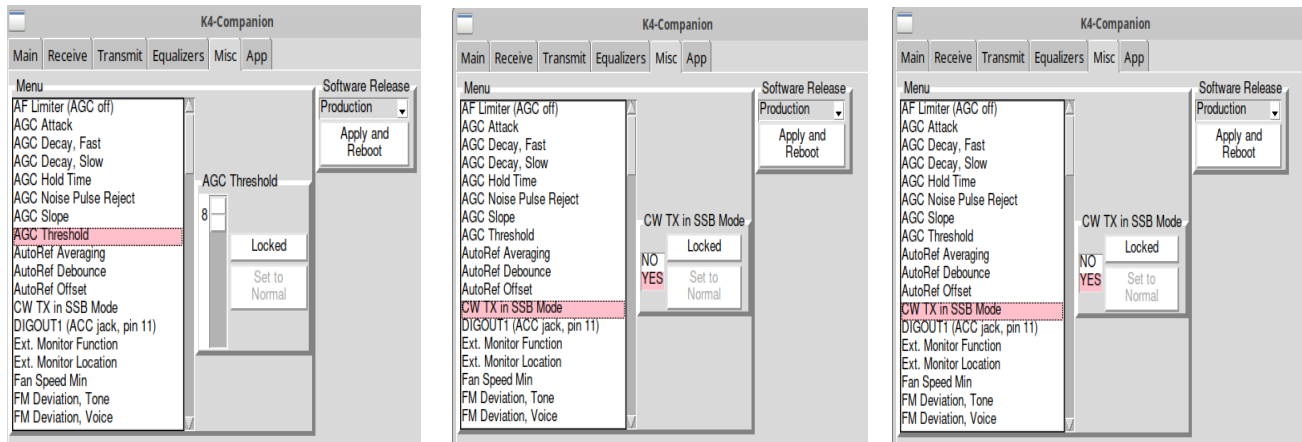
# Equalizers



Transmit and Receive equalization can be set by mode. These are personal preferences. Please refer to K4 forums or the K4 User Manual for recommendations.

# Miscellaneous Tab

The Misc tab reproduces **all** of the menu functions of the K4. These functions are covered in the user manual. The configuration menu is a long, scrollable list. When an item is selected, it appears in pink. Then a secondary widget opens that offers the menu choices. Some of these are simple YES/NO selections, others present a list, and some have a slider to select a value for the menu item.



The images show a few of the selections possible through the menu system. Please be sure to read the K4 user manual carefully. Changing menu settings without due consideration can have unexpected results.

Finally, software “Firmware” releases from Elecraft can be applied through the menu on the far right. Selections are:

- Production
- Beta
- Previous Production

**Reserved**



# Debugging

No guarantee is made that this software will run on a particular machine or distributions. However, debugging information may be useful to the developer. Information for submissions is listed on github.com by opening an problem on the “Issues” tab of the K4 Companion site. To initiate debugging, use the ‘-d’ option: `./k4companion -d -d` . This will show a running list of actions by the script onscreen. To save the debugging information to a file, use `./k4companion -d -d >> mydebugfile 2<&1&` . The debug file can be named anything you like and the file will be created if it doesn’t exist already. To append an existing log file, use `./k4companion -d -d >> mydebugfile 2<&1&`. The file can then be posted or e-mailed to Dale, W7DA.

Information from dmesg may be helpful as well: `sudo dmesg` , although more likely if there is a system failure somewhere and not necessarily with K4 Companion.

When running on Windows, the most common error seen has to do with opus.dll . The supplied DLL is 64 bit and it works on both Windows 10 and Windows 11. The issue seems to be with the permission on the file. Navigate to the folder C:\Windows\System32\opus.dll with the file utility (not a shell or cmd prompt) and right click on the file. Be sure the permissions include your profile, especially if you are not running as a System Administrator. The procedure and information regarding file permissions is available online.

# Notes on Operation

Audio functions on Linux can sometimes be challenging. A working knowledge of ALSA, alsamixer, and alsactl is useful. Pavucontrol can assist with selecting the correct audio input and output. Be sure to check the Configuration tab in pavucontrol, and if audio is not working, try another combinations. These may include “Stereo Duplex,” “Play HiFi quality Music,” and “Pro Audio,” among others.

In the **App** settings, **Audio Settings** will generally this will be left at “default,” but there may be circumstances where other selections are needed. These may include “OSS” or various settings found in the sound system. Most of the audio issues appear to revolve around audio selection outside K4 Companion.

If there is INVALID information in the configuration file, *settings.ini*, you might need to edit the file directly. The path to the file is *\$USER/.config/k4companion/* The following lines are where you might have problems. These can be edited directly, or the incorrect information removed and entered correctly on the next startup of the program. Under Windows, the INI file is found at *\$USER\AppData\Roaming\k4companion\*

```
[Elecraft K4]
insecure_password_hash =
address = 192.168.1.128
port = 9205
proxy_address =
```

An experiment recently proved that K4 Companion will run on WSL – Windows Subsystem for Linux, but it is not easy to set up, and there are quite a few stumbling blocks. Choppy audio is probably the greatest impediment, probably due to adding extra layers of complexity to the audio subsystems. It will like run under Parallels or other emulators, but with the same audio delay issues.

There is currently a bug that will sometimes not allow the program to exit normally. Two ways around this in both Linux and Windows: 1) hit the ‘X’ on the frame and often that will close the program, or close the terminal window you used to open K4 Companion. This is generally problematic if you shutdown the radio.

Another note about Windows: I have noted it is prone to random audio failures, including disabling the audio for unknown reasons. You might have to dig into the subsystem a bit. I recommend using the Sound App, not the control panel. Control panel is often not very enlightening.

As always, YMMV – your mileage may vary!

Enjoy!

Charles  
NK8O  
VE3ISD  
5H3DX

**This manual was produced using FOSS, Free Open Source Software**