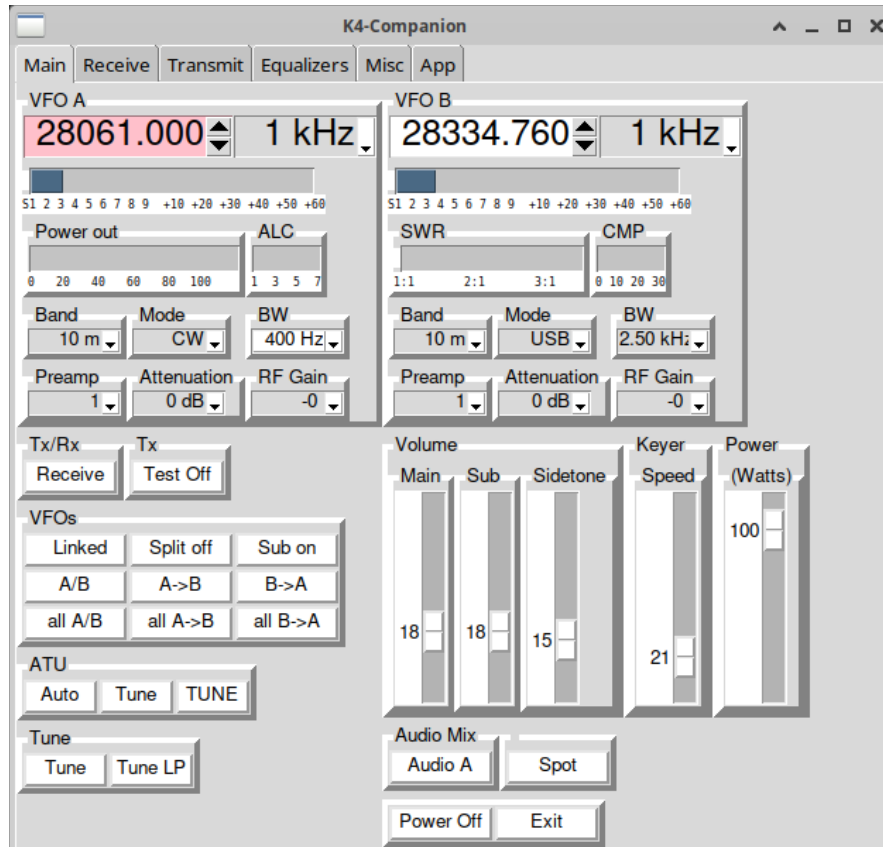


# *K4 Companion*

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 21<sup>st</sup>, 2025, based on version 1.2.4

# Table of Contents

Introduction:	Page	1
License:	Page	2
Installation on Linux	Page	3
Windows Installer	Page	4
Full Installation on Windows	Page	5
MacOS	Page	6
Running K4 Companion	Page	7
Configuration: App tab	Page	8
Operation and Controls	Page	9
Receive Functions	Page	12
Transmit Functions	Page	13
Equalizers	Page	14
Miscellaneous Tab	Page	15
Debugging	Page	16
User Notes	Page	17
Revision History	Page	18

# 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 now been successfully ported to both Mac OS, and Windows.

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 been tested on Mac OS and they work quite well.

**The version number is now displayed in the title bar.**

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

# Windows Installer & Binaries

Two options are available those who do not wish to create a full installation of Python3 or to examine the open source code. A Windows installer and Windows binary executable files are now available. The automated installer that places all items in the proper directory, creates a Start Menu entry, and it also will create a desktop shortcut if that option is selected. Double click the installer, then select the desired options. The user manual is also provided, found in %PROFILE%/k4companion/

NOTE: The automatic places opus.dll in the %PROFILE%/k4companion directory. Starting with version 1.2.4, it should no longer be necessary to move or copy opus.dll to C:\Windows\System32\ Windows users please replace your opus.dll if the size is < 2 MB. If it is ~500 kB, you probably have an older version of the file.

If preferred, a manual install is possible. There are three files that are needed to run under the Windows operating system. K4 Companion has been tested on Windows 11 and Windows 10.

The three files, available on the github download page are:

- k4companion.exe
- k4companion.yaml
- opus.dll

The three k4companion files should be placed in a directory (folder) of your choice. These need to be placed in the same directory (folder).

The configuration file, k4companion.yaml, remains fully editable with the same cautions noted above.

A note regarding Windows Defender: The Defender reaction will range from none at all, to identifying K4 Companion as a virus. If identified as a virus, Defender will quarantine the executable and ask if you want to delete it. Instructions regarding bypassing Windows Defender are found at <https://answers.microsoft.com/en-us/windows/forum/all/how-to-allow-an-ap-through-windows-defender/caab364c-9e65-448e-8635-98721be847ba>

# Full Python Installation on Windows

Installation on Windows follows the same principles as Linux. It is quite similar, once Python3 is installed. **NOTE:** Windows binaries are available. See Page 5.

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/>

# MacOS

Development on MacOS has been challenging. Binaries are not practical, due to Apple's security policies. Every library that is compiled into an executable throws an error. Development of a MacOS binary has been abandoned, at least for the time being.

There are many quirks using Python3 and MacOS. Background colors and active colors do not work on the widgets. All python references must be *'python3'* and all pip references must be *'pip3'*

More to come.



# Running K4 Companion

There are several ways to run K4 Companion, depending on the operating system. To execute the script directly, it requires the file to be marked as executable. Under Linux and Mac OS, this can be done immediately 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. No modifications of the Windows executable files are needed.

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
- To start the Windows binary navigate to the file in the chosen folder, then either click on `k4companion.exe` or highlight and press ‘enter’
- If the Windows installer has been employed, K4 Companion can be run from the Start Menu or a Desktop icon if this option was selected at the time of installation

Linux users: 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.

Windows users: 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’. If the installer was employed, K4 Companion will be on the Start Menu. If selected there will also be a desktop shortcut.

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

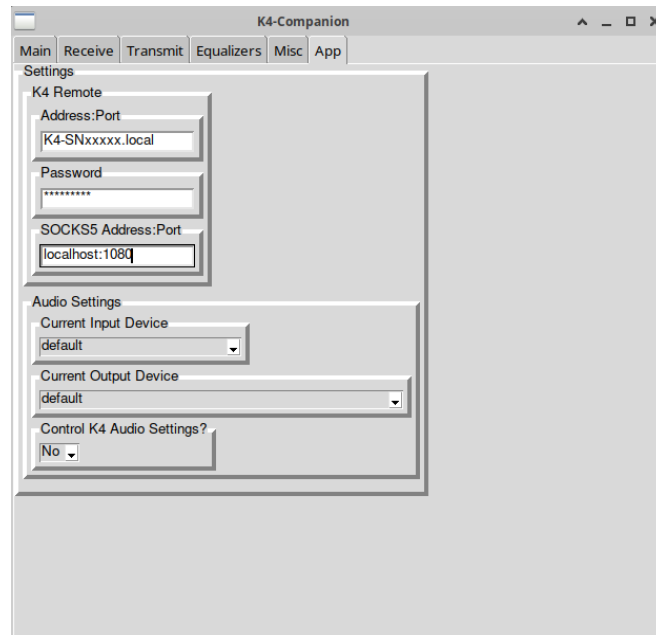
Mac users: A shortcut is possible using a shell script to start K4 Companion. Create a blank file on the desktop. Using Terminal, `cd Desktop ; touch k4companion ; chmod 755 k4companion`. Then use nano to edit: `nano k4companion`.

The file should contain:

```
#!/bin/sh
~/path/to/k4companion
# where path/to/k4companion reflects your actual file path
exit 1
# EOF
```

**The version number is now displayed in the title bar.**

# 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.

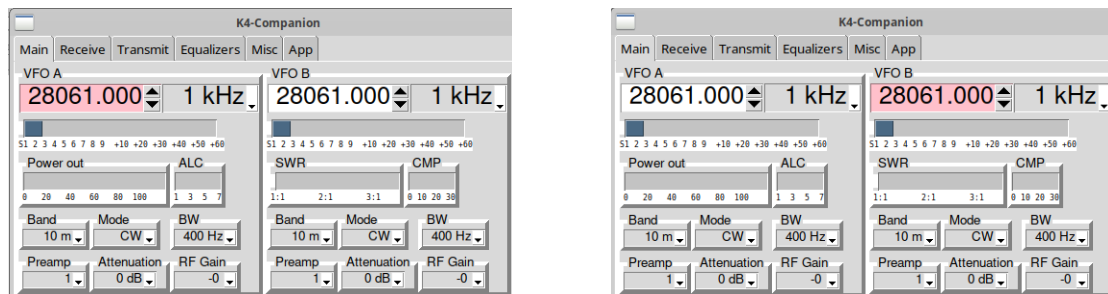
**Control K4 Audio Settings?** If you are using K4 Companion to control the radio while sitting in the shack, this can be set to Yes. Otherwise for remote operations, set it to No. See **Notes on Operation** below.

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.

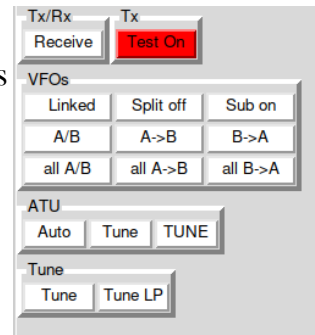
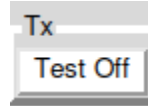
The VFO boxes support **direct entry**, **scrolling** with the Up←→Down arrows, or if there is a single click in the VFO box, the keyboard Up/Down buttons also allow frequency scrolling

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

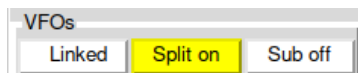
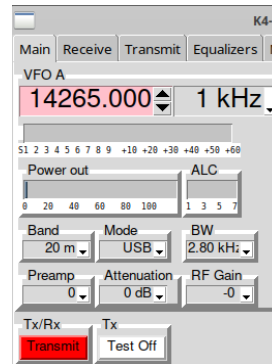
When the application opens, it sets the K4 to Test Mode, as shown. This button lights up red when engaged but turns white when off. This prevents unintended transmission when setting up. **This can be disabled at the top of the YAML file**, but it can be a useful feature. The Test button turns white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



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

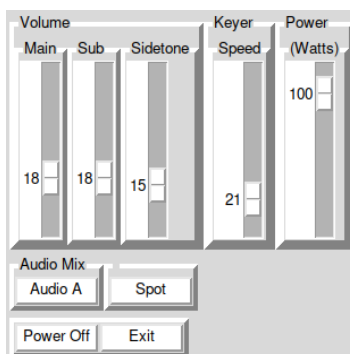
ATU controls have the following functions:

- 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

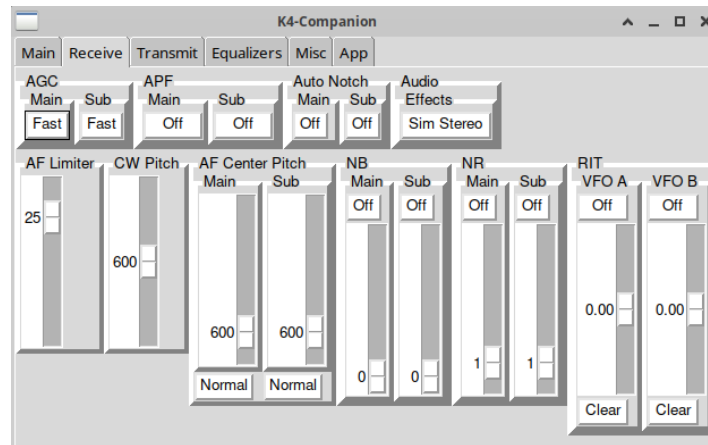
- Main volume – equivalent of the small knob on the lower left of the K4
- Sub volume – equivalent of the concentric inner knob on the lower left of the K4
- Sidetone level – useful especially if keying is done with macros or locally. Set to zero if keying is produced locally
- Power – sets power output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

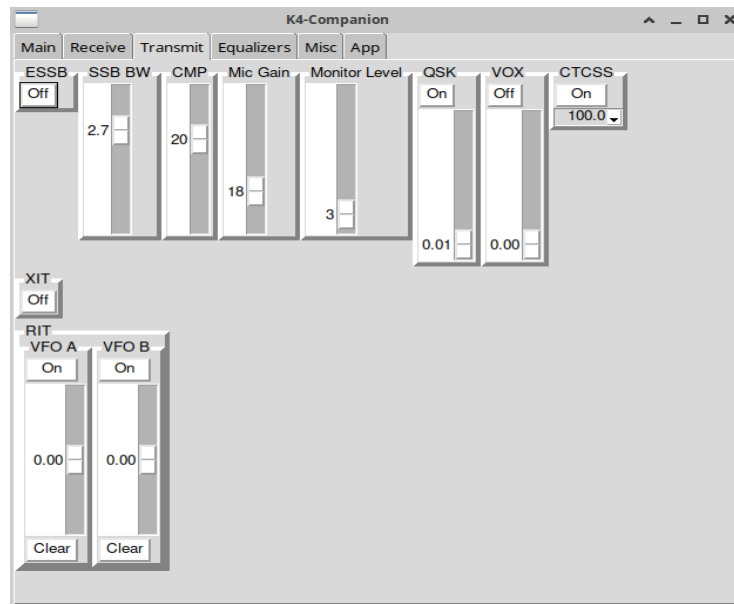


Controls for AGC, Audio Peaking filter, Auto Notch (SSB) and Audio Effects are in the top row. The function of each is explained in the K4 User Manual. Options for Audio effects include

- 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

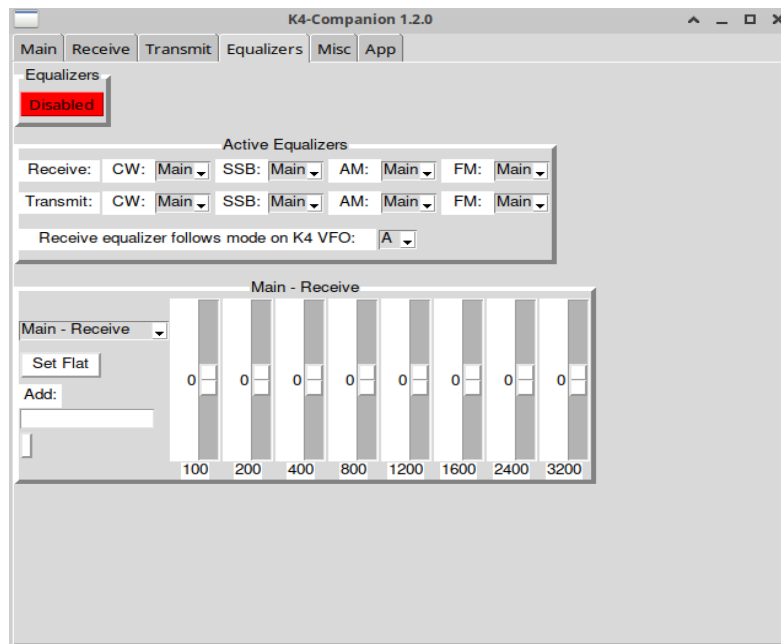
Note that the AF limiter only functions if the AGC is “Off”. This prevents extremely loud signals from blasting through at full-volume if the AGC is absent. AF Center Pitch functions by mode, and returns to the last user settings when changing modes. Noise Blanker, Noise Reduction, and RIT are explained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



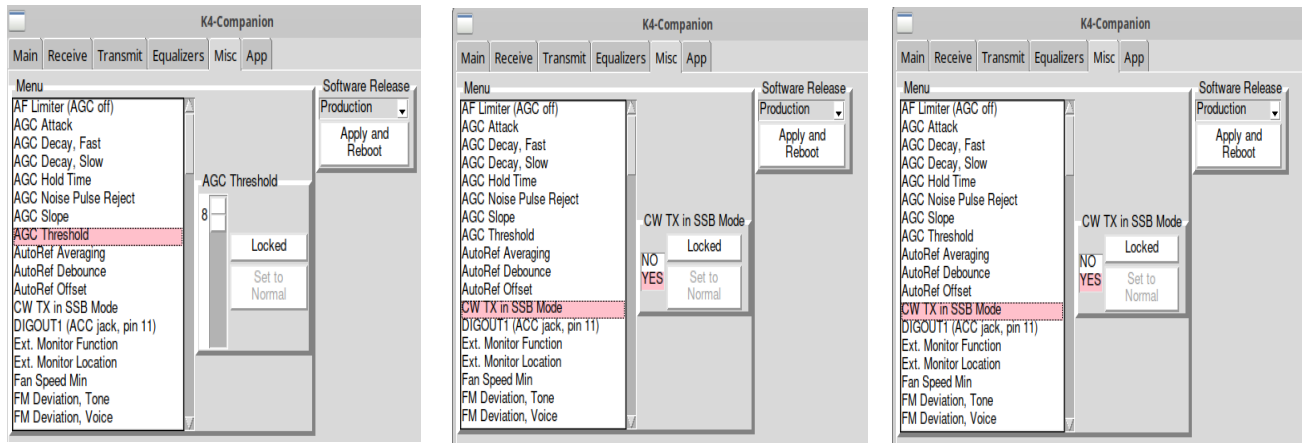
Transmit and Receive equalization can be enabled or disabled, and be set by mode. These are personal preferences. When enabled **the settings on the K4 are altered**. The K4 settings do not revert to previous settings if the Equalizers on K4 Companion are disabled.

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 Update Windows/k4companion.exe to 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

# 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. Unfortunately the debugging option does not work with the compiled Windows EXE file. Report the problem in as much detail as possible to the developer, and copy any error messages that do appear.

# 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. In Windows, the App settings are likely to be “Microphone Array” for the “Current Input Device” and “Speakers/Headphones” for the “Current Output Device”.

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\* In the current versions this is very unlikely to be necessary.

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

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. Use *mmsys.cpl* to access the Sound App. Control panel is often not very enlightening and the Sound App gives more information. As always, YMMV – your mileage may vary!

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.

Enjoy!

Charles  
NK8O  
VE3ISD  
5H3DX  
[nk8o@arrl.net](mailto:nk8o@arrl.net)

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

# Revision History

version 1.2.4 Windows installer update

version 1.2.4, runs with opus.dll in the same folder/directory with EXE file

v1.2.4:

Update changelog for v1.2.4

Don't terminate the GUI on close until exit

We were terminating the GUI while queued commands were still being processed, which sometimes resulted in tkinter exceptions.

Fix Linked/Unlinked togglebutton

Split mode and Panadapter modes should be independent of the VFO B Band Independence setting. Also, the K4 does not automatically send the Band Independence state in response to a RDY; command (as I believe it should), so we'll specifically request it at startup.

update Windows installer to version 1.2.3

add Windows executable version 1.2.3

updated user manual to version 1.2.3

v1.2.3:

Update changelog for v1.2.3

Call subscribe\_responses only when appropriate

In several cases we called subscribe\_responses(), when it was clear that we were subscribing to a single response. In those cases, we now call subscribe\_response() instead.

This change should result in no behavioral changes.

Rename subscribe\_cmds to subscribe\_responses

This better reflects its meaning.  
This change should not affect behavior.

Subscribe to all responses before connecting to K4

There was a race condition where sometimes the K4 sent a parameter value before the app was ready to receive it. This was noted when the VFO linked button had the incorrect initial value.

upload Windows installer for 1.2.2

manual edited to reflect current changes, known issues 2025-04-18

Update Windows executable file to 1.2.2

v1.2.2:

Update changelog for v1.2.2

Try again to fix winfo error on closing

commit b97d7eabc24355a21e29b2276afdcdb24b580bb76 doesn't appear to fix the problem, so we'll try a more drastic workaround.

Add TLS support and port 9204

This code is only used on python version 3.13 or later.

Fix wininfo error after closing

The error rarely occurs because it results from a race condition. We sometimes (rarely) try to query information from tkinter after it has been closed. We'll try to avoid the error by checking if the server connection has been closed.

remove stale installer

/Windows/k4companion.exe updated to 1.2.1

updated manual 2025-04-17 version 1.2.1

updated /Windows/k4companion-installer.exe to version 1.2.1

v1.2.1:

Update changelog for v1.2.1

Windows installer 1.2.0 update

Removed stale 1.1.40 Windows installer

Update User Manual 2025-04-17

update Window/k4companion.exe 1.2.0

Update README.md 2025-04-17

Update opus.dll to version 1.5.2

Fix lost sub-receiver audio with 2 panadapters

When dual panadapter mode is selected, the K4 sends an SB3; response. However the Programmer's Manual says the only valid responses are SB0; (sub-receiver off) and SB1; (sub-receiver on). We treated an SB3; response as meaning sub-receiver off. Now, we treat it as sub-receiver on.

equalizers: Show active equalizer when disabled

When the equalizers are disabled, we'll indicate the (pending) active equalizers in pink, after enabling, they are indicated in light green.

update Documentation/K4 Companion User Manual.pdf to version 1.2.0

updates to Documentation/K4 Companion User Manual.pdf

add Windows/k4companion-1.2.0-installer.exe

Add Windows/k4companion.exe 1.2.0

Update changelog for v1.2.0

v1.2.0:

Fix bugs in equalizer code

There were a few. In particular, it would previously unconditionally set the K4's equalizer settings on startup. Now, we've added an

Enable/Disable button. When disabled (default), no equalizer changes are sent to the K4. When enabled, changes are still unconditionally sent at startup according to the equalizer settings in k4companion.

This is a substantial rewrite of the equalizer code.

Update To Do list

Control app Mic Gain locally

Add Windows/k4companion-1.1.40-installer.exe

Update K4 Companion User's Manual.pdf to v1.1.40 (revised)

Update Windows/k4companion.exe to 1.1.40 revised

Update Windows/k4companion.exe to v1.1.40  
Apparently, this didn't work last time.

Add version number to the displayed title

Update K4 Companion User's Manual.pdf to v1.1.40

Update Windows/k4companion.exe to v1.1.40

Update changelog for v1.1.40

v1.1.40:

Request that the K4 send opus-float encoded audio  
This is the default, but other software may change the format.  
If so, we reset it to the preferred format.

Revert "Be more liberal in accepted audio encodings"  
This reverts commit 4c734154999d1606813edf87ae308609e61b61ca.

Instead, we'll request that the K4 send us opus-encoded audio  
via the EM3; command

Update changelog for v1.1.39

v1.1.39:

Be more liberal in accepted audio encodings  
Accept encode\_mode\_opus as well as encode\_mode\_int16.

Encode\_mode\_opus has been tested. Encode\_mode\_int16 has not.

Update Documentation/K4 Companion User Manual.pdf

Update k4companion.exe to v1.1.38

Update changelog for v1.1.38

v1.1.38:

Add version display to App tab

Update To Do list  
Add mic gain control to local audio setting.

Add version display in k4companion on the App tab.

Update k4companion.exe to v1.1.37

I tried to do this earlier, but failed.

Restore WIndows/k4companion.exe

I don't know how the deletion happened.

Revert "Update Windows/k4companion.exe to v1.1.37"

This reverts commit 1b3b6b02aabaec66cf496e0088dfd72f0e2885b.

Update Windows/k4companion.exe to v1.1.37

v1.1.37:

Workaround for no pyaudio default audio device

This is a problem reported by Margaret, K4ZMA, on Windows 11.

v1.1.36:

Ensure we have a config dir on all platforms

If the config dir doesn't exist, create it on darwin, win32,  
and Linux platforms.

Update k4companion.exe to v1.1.35

Remove ESSB startup bug from To Do list

v1.1.35:

Fix ESSB togglebutton and SSB BW slider

Update user manual to changes in v1.1.34

Add a Windows .exe file in the Windows directory

Update the Users Manual including Windows version

Add Contributions/opus.dll and associated README

This is a Windows dll file needed for running k4companion.

v1.1.34:

Fix Exit button operation on lost connection

There were two problems. First the directive, QUIT!, was missing the exclamation point. The second was that after losing connectins, we correctly didn't send commands to the K4. Incorrectly, we also didn't act on internal directives like QUIT!.

For good measure, instead of calling the quit() function, we instead call the root.destroy() if tkinter is running, and we call sys.exit() if tkinter is not running.

Update the To Do list

Improve Poll class implementation

Make vfo scanning smoother

We used to bunch up sending frequency changes every 50 mS.

That worked well before we added audio support. Now, we send them without delay for smoother audio reception.

v1.1.33:

- Handle lost connection to the K4 more gracefully
- Popup a an error message on socket send or receive error and mark the connection as closed.

- Create .ini file directory on Windows if needed

v1.1.32:

- Only popup not-connected warning once

- Fix bug in direct frequency entry
- The cursor would remain in the right side of the frequency spinbox and wouldn't permit the entry to be changed. Fixed.

- Fix diversity audio and sub off audio

- Remove Sub AGC control
- Contrary to the documentation, the K4 AGC control affects both main and sub receivers.

- Remove mic gain fom to do list, already done

v1.1.31:

- Show a popup for connected, but no response.
- This occurs when the password is wrong. Suggest that the user check the password in the App tab.

- Avoid exceptions on controls when not connected
- Some controls would generate an exception when changed while not connected to a radio. We now warn instead.

- Add widgets for VOX gain and anti-vox

- Update To Do List
- Add panadapter/waterfall

- Update the user manual with changes from v1.1.30

v1.1.30:

- Update to-do list

- Restart connection when password is changed

- Remove ":Port" from label for K4 address entry box
- Generally, there is no need to enter the port number, as the default port of 9205 is preferred. It will still be accepted though.

- Update nk8o\_k4companion.yaml

- Call socket.send on socket.sendmsg failure

v1.1.29:

- Reconnect to server when address/port changes



Fix exception for non-existent audio device(s)  
This exception occurred when the name of the audio device in the .ini file doesn't exist on the current machine.

Add a to-do list

Restore cat-only operation on port 9200  
Commit b6be0efd, which introduced handling audio receive packets in the server receive thread, broke cat-only operation. This commit fixes it.

Introduce Server.cat\_only variable - internal only  
Replace the various checks for port == 9200 or 9205 with checks for a Server.cat\_only variable. No externally visible changes.

Read up to 10 frames before deciding: no audio  
After opening the input device, read up to 10 frames of silence before deciding that there is no input.

Update README

Update README

Add the User Manual to the Documentation directory  
Also move screenshot.png into an images sub-subdirectory.

v1.1.28:

Fix pyaudio open code retry code  
In commit a0bcbb, I added two retries on opening the audio devices. Unfortunately, I even retried opening the input device when the previous opens succeeded. This commit should fix it.

Add FW versions widget

Add SOCKS5 proxy support

v1.1.27:

Audio Mixer toggle button now displays all options  
It still only allows selecting some audio mixer combinations, but will correctly display other combinations if they are selected via the menu setting.

Implement all audio mixer combinations

Retry opening PyAudio a couple of times  
PyAudio sometimes fails to open. We'll retry it twice. If this doesn't work, we'll try to delay a second before retrying.

Always send Audio MX commands to the K4  
Previously, they weren't send if "Control K4 Audio Settings" was set to No.

Add QSK and VOX widgets

Limit range of af-center-pitch slider when in CW  
In CW the range is 300-2000, otherwise it's 300-3000.

v1.1.26:

- Add k4 address and port to App settings

- Change to Opus-encoded transmit audio

  - Thanks to Tom Blahovici, VA2FSQ, for a hint on setting the audio packet version number.

v1.1.25:

- Attempt to make server startup more reliable

  - There were race conditions where responses were sometimes received from the K4 before we were prepared to receive them.
  - I hope this commit addresses the problem.

- Update nk8o.k4companion.yaml

- Add RIT and XIT widgets

- Minor change to audio-effects toggle button

- Add Software Release Widgits

- Add "Normal" buttons for AF Center Pitch

v1.1.24:

- Reorganize widgets on receive and transmit tabs

- Add equalizer tab

- Standardize calls to server.subscribe\_cmds

  - Internal change only, no externally visible changes.

- Widen AF Center-Pitch settings

- Add SI parameters to the list

  - Include SIDA, SIDD, SIDU, SIFP, SIRF, and SIRC. These are not fully documented in the Programmer's Reference Manual.

- Add CTCSS toggle and dropdown buttons

- Add noise reduction toggles and sliders

v1.1.23:

- Add noise blanker toggles and sliders

- Add auto-notch toggle buttons

- Add transmit voice monitor level slider

- Add mic gain slider

- Add center pitch sliders

- Add AGC toggle buttons

- Add audio effects toggle button

- Add CW pitch slider

Add Receive tab with AF-Limiter and APF sliders

nk8o\_k4companion.yaml: Update to latest

Rename the Settings tab to be App

Since all the parameters we can configure can be considered settings, label the k4companion-specific settings "App".

Add transmit tab with compression and ssb bw

To the transmit tab, we add a compression slider, an ESSB on/off togglebutton, and an SSB bandwidth slider.

v1.1.22:

Queue received audio in the receive thread

Previously, the main thread would dequeue audio packets from the receive queue and then queue them on the audio queue to be picked up by the audio input thread. Now, the received thread places audio packets directly on the audio queue, reducing audio latency.

I attempted to drop received audio packets if \*any\* were already queued. This resulted in too many dropped packets and audible artifacts.

Remove duplicate set\_only suffixes

Internal change only, no visible impact.

Change mic attenuation to a factor of 2000000

It was 5000000

v1.1.21:

Add initial audio transmit capability

Fix TX Monitor Remote Menu initial setting

Add timer functions - useful for debugging

Clean up socket sendmsg errors

Clean up error handling on socket connect failure

v1.1.20:

Queue popup msgs, making them thread-safe

Popup messages would not display properly when called from threads other than the main thread.

Move except clause after queue.get\_nowait()

Code cleanup only. No change in behaviour

Fix the sidetone slider

The sidetone slider now ignores "ML;" responses not representing CW sidetone

Update nk8o\_k4companion.yaml

With Charles' latest.

Improve reliability of slider controls

If a slider was moved rapidly, it was possible, though difficult,

to get the slider value out of sync with the value sent to the K4.  
For example, the power slider might read 84 Watts, but the radio  
would actually be set to 79 Watts. This commit should fix that problem.

bugfix: If audio input device is in settings  
This is part of the preparation for handling transmit audio

v1.1.19:

bugfix: Fix Control K4 Audio setting

Fix the Transmit/Receive togglebutton  
Before, it would work when clicked, but did not track the K4's state  
properly when the XMIT button on the K4 was pressed.

v1.1.18:

Fix audio on system without 12000 Hz sample rate  
I obviously insufficiently tested commit e43a09ae  
"Resample to least multiple of K4 sample rate"

v1.1.17:

Resample to least multiple of K4 sample rate  
Instead of resampling only to 48000 Hz, resample to the least integer  
multiple of the K4's sample rate. I.e, the first supported rate of:  
12000, 24000, 36000 or 48000 Hz. This will save CPU time on devices that  
support one of the lower output sampling rates.

Integrate "directives" into the cmd structure  
Previously, buttons could have an option, "directive", indicating  
actions to be performed by the application. Now, directives are commands  
whose name ends in '!'. Examples:  
"directive: connect" becomes "cmd: CONNECT!;"  
"directive: quit" becomes "cmd: QUIT!;"  
"directive: show\_previous\_page" becomes "cmd: SHOW!PREVIOUS;"  
"directive: show\_page XXX" becomes "cmd: SHOW!XXX;"

Require debug\_level >= 2 to print sent/received msgs  
Then uncommon debug messages won't become lost in a sea of common  
debug messages.

No longer specifically request menu definitions  
Now that we get a dump of all K4 parameters at startup, we no longer  
need to later request the menu definitions.

Add 5 Khz receive bandwidth option to config file

v1.1.16:

Fix initial state of the Test Mode togglebutton  
We do this by delaying sending the RDY; command requesting the  
state of the transceiver until after we have sent our startup  
commands.

Always control K4's audio on port 9200  
When receiving audio, we may not want to control the actual radio's  
audio, but when not receiving audio in the app, we always want to  
control the radio's audio.

Fix debug msgs "sent: ..." on port 9200

No longer request state for specific parameters  
Because we now get a complete dump of the K4 state, we no longer need to request the state of each parameter we're interested in.

v1.1.15:

Finish startup audio settings

Commit 409ffac was incomplete. This commit attempts to finish the job and also keeps the audio subsystem notified of changes to the diversity setting.

v1.1.14:

Fix broken setting of audio\_output\_device

Commit [b2c91fb4 Control PC audio mix and volumes] broke the ability to set the audio output device. This commit fixes it again.

v1.1.13:

Fix connections on port 9200 (unauthenticated)

Commit aaf589ad7a02322bdd9249cb549fb1942258147a broke connecting on port 9200. This commit fixes it again.

Fix startup issue with 'blank interface'

Commit aaf589ad7a02322bdd9249cb549fb1942258147a didn't fix the problem it was intended to address. After sending the password to the K4, we need to wait a bit before sending the first command. This commit has us wait until the K4 has sent some data before we send the first command.

v1.1.12:

Resample to 48000 Hz by repeating samples

We have seen artifacts (clicks or ringing) when receiving CW with a couple of resamplers. Let's try a simple resampling by just repeating samples. The K4 provides 12000 Hz samples. Repeat them 4 times to yield a sample rate of 48000 Hz.

We can only hope that (almost) all sound device support 48000 Hz.

v1.1.11:

Delay initial dump of K4 parameters until ready

We had a race condition where sometime the initial dump of K4 parameters was received before our widgets had subscribed to them. This caused the vfo widgets to sometimes not be initialized. We now delay telling the K4 that we're ready for the dump until we're actually ready.

I hope this fixes the issue we were seeing.

Fix the audio mx togglebutton cmds

Control PC audio mix and volumes

Also add a new setting to optionally control the K4's audio settings as well.

Update README.md to mention audio receive

v1.1.10:

Optionally, save K4 password in settings

Enable turning on diversity

Diversity is particularly useful since we currently hear audio from both receivers all the time. Diversity keeps both receivers in sync.

Fix the connect togglebutton

It used to say 'connected' even on connection failure.

Now it says 'connecting' until the K4 actually responds.

Add an autospot button

Don't print PING/PONG cmds unless debug  $\geq 3$

Disable automatically, repeatedly, sent reports

v1.1.9:

Handle pyaudio errors during playback

Add saving/loading of settings file

Avoid trying to send commands until server connect

Add Setting tab including audio device settings

Drop audio packets if we're behind processing them

Use pydub to resample to 44100 Hz

Move audio processing to a its own class

Reduce audio delay by reducing audio write size

Set frame rate to 48000 by duplicating samples

Raise volume by multiplying pcm audio by 20

Resample audio to 44100 Hz using resampy

First working audio at 12kHz sample rate

v1.1.8:

combobox: Allow setting colors for comboboxes

Previously, the same color was applied to all comboboxes.

Now the colors are applied to individual combobox instances.

Decrease vfo spinner repeat rate

v1.1.7:

Fix but in text widget

v1.1.6:

Document some additional options in the yaml file

Document is too strong of a word. A brief description is more accurate.

Add the fact that widgets now support the leading option for additional vertical space before the widget and a label to surround the widget with a labeled frame.

Move power-off and Exit buttons to column 2

Improve password popup  
It's now a modal window, forced to remain on top.

v1.1.5:

Add authenticated connection support on port 9205  
If the K4's port is specified as 9205, a password-protected connection is made on that port. This is a step toward supporting audio and PAN data on this connection.

v1.1.4:

k4companion.yaml: Simplify yaml  
No externally visible changes in behavior.

Remove groups that wrapped widgets for a label  
We instead put the label as an option of the widget.

Add support for labels & leading to all widgets  
This avoids the need to wrap the widget in a group just for a label and/or leading.

Simplify font inheritance  
No outwardly visible changes except for a minor speedup.

Permit a label to be added to widget items

k4companion.yaml: Clean up the config file

Improve validation of required item options

Allow dash characters '-' in yaml file definitions

Fix initial state of xvtr band selection settings

Add a current screenshot to the README

v1.1.3:

Rename program to K4-Companion  
This program has grown far beyond its initial ability to send macros to the K4. The name change better reflects its capabilities.

Code cleanup

k4macro.yaml: Move power slider  
This puts the audio sliders above the audio mixer selector.

v1.1.2:

bugfix: Restore display of group labels  
Commit ceee3ba6a, which added tabs to pages, stopped displaying labels on groups. This commit restores them.

v1.1.1:

menu: Defer loading menu definitions until needed  
We now don't request menu definitions from the K4 until the menu is to be first displayed.

v1.1.0:

Add K4 menu support

It may not be pretty but it appears functional

Add tab support to pages

Avoid requiring a notebook item being added to pages in order to get tabs.

nk80\_k4macro.yaml: Add power\_dropdown widget

connect\_button: Detect when k4 is connected

nk80\_k4macro.yaml: Update from Charles, NK80

config: fix power\_dropdown widget

toggle\_button: Allow some selections to have a cmd

Previously, if one selection specified a cmd, all selections were required to specify a cmd. Now only some of the selections may specify a cmd.

Dropdown: Fix font setting

v1.0.43:

Fix the range of the main and sub volume sliders

The K4 maintains volume in a range of 0-60. In a misguided attempt to go from 0-100%, I scaled that to 0-100. Unfortunately, that does not map well. We now show the actual volume range of 0 to 60 for the volume sliders.

bugfix: Avoid recursion on power slider

The recursion was introduced when we permitted responses from the K4 that match current values to be forwarded to subscribers. That change was necessary.

Simplify K4 error code handling

v1.0.42:

Validate item options in config file

Check for required as well as optional options.

v1.0.41:

Reformat K4 error messages

v1.0.40:

Move power, ALC, SWR and CMP bar graphs up

Show error messages for codes received from the K4

We display them in a pop-up window.

Permit connecting to the K4 via a button

Rename cyclebutton to togglebutton

Remove cyclebutton altogether as well as the updown option.

v1.0.39:

Avoid subscribing to set-only commands

Since they will never respond, there is no value in subscribing to responses for them.



v1.0.38:  
 Fix incrementing/decrementing frequency  
 It was failing when the existing frequency was not a multiple of the step size.

v1.0.37:  
 k4macro.yaml: Fix SWR bar graph  
  
 Remove atu\_trainer\_kx4ui.yaml, ctrl\_panel\_kx4ui.yaml  
 They have some good info and may be resurrected later, but for now we're not keeping them up-to-date.

v1.0.36:  
 Fix and recalibrate bar graphs  
  
 Add ATU extended tune

v1.0.35:  
 Rename all eval'd options  
 For example:  
   responsevalue is now evalresponse  
   cmdvalue is now evalcmd  
  
 Many changes (Sorry)  
 This change add bargraphs for power out, ALC, SWR and compression.  
 Bargraph labels now use a fixed-width font by default.  
 By default 1-point of leading is displayed before groups with labels.  
 k4macro.yaml: It also rearranges several displayed widgets  
 Miscellaneous bug fixes  
  
 k4\_macro.yaml: Add group marker comments  
 Add #{ and #} markers at the beginning and end of each group.  
 These help me find the end of a large group.

v1.0.34:  
 Display active VFO and transmitting VFO  
 Color the VFO frequency (currently pink) of the active VFO. While transmitting, color the VFO with the transmit frequency (currently red).  
  
 Clean-up the display a bit

v1.0.33:  
 radiobutton: fix bug referencing text  
  
 nk8bo\_k4macro.yaml: Add missing attenuator values

v1.0.32:  
 k4macro.yaml: Add goodies from NK80  
  
 nk8o\_k4macro.yaml: Include latest changes from NK80  
  
 Fix unexpected bandwidth and attenuator values  
  
 Permit set\_only and get\_only cmds if multiple cmds  
 For radio, dropdown, and cycle buttons, if a set\_only or get\_only

command is specified, it is now permitted as long as it is not the only command or response.

k4macro.yaml: Spelling change: Atten -> Attenuation

v1.0.31:

Add nk80\_k4macro.yaml

Example config file from Charles, NK80.

k4macro.yaml: Various updates

config: Make togglebutton a synonym for cyclebutton

cyclebutton: Set wraparound default to true

Move cyclebutton 'justify' from selections

Move 'justify'; option from selections to cyclebutton

Enable the width of the VFO to be specified

Auto-set S-meter bar graph length

Set the S-meter bar graph according to a percentage of the length of the text below the bar graph. The S-meter bar graph will match the text regardless of the point size of the text or the screen resolution.

Add check for group without "contains:" in yaml

v1.0.30:

Use dropdown buttons to replace cycle buttons

Add dropdown buttons via a combobox

v1.0.29:

Catch exceptions when calling eval

If an exception occurs during an eval, print additional information.

Change vfo INDY button text to UNLINKED

v1.0.28:

Make VFO frequency display larger

Calculate cycle selection width based on siblings

Set cycle button selection widths on the maximum width of all siblings within the cycle buttons group.

Use config preprocessor to organize YAML

Simplify config file with a preprocessor

Implement a simple preprocessor to simplify the organization of the YAML config file.

Fix range of the sidetone slider

The K4 Programmer's Manual says the range should be 0-100 in K4 mode, and only 0-60 in K3 mode. But it appears to be 0-60 in K4 mode as well.

Remove blinking from Tx Test on button

Make it red instead.

Change flashrate option to 'blink'

Make all widgets black on white

We can add color back in sparingly as needed.

Reduce VFO step width to 6 characters

Now that we don't offer a step width of 100 KHz, we no longer need a width of 7 characters to accomodate it. Six characters suffices.

Remove auto-wrapping button text

Auto-wrapping button text messed up multi-line button text, so this commit removes it.

When debugging, avoid printing auto received msgs

Certain messages are automatically received continuously from the K4. Currently that includes the S-Meter messages. Now, we'll only print those if the debug level is 3 or higher. This permits us to more easily monitor the other send/receive messages.

Rename options that are eval()ed

If they are passed to eval() rename the optios. This includes:

cmd -> cmdvalue

response -> responsevalue

increment -> incrementvalue

k4macro.yaml: Make some text changes

Thanks to Charles, NK80.

v1.0.27:

Latest yaml changes using more cycle buttons

Add support for Notebook (tabs)

Add a bargraph, with S-Meter example

Add a getvalue widget

The getvalue widget is used to report a value that is automatically updated by the K4. For example, an S-Meter reading.

Apparently, the K4 can return a null mode

Some have reported receiving a response of MD; to a command MD; instead of MDn; as documented. It appears that the VT command can be sent with a null mode, rather than the documented required mode, so we'll just go with

it.

Oops, fix partial cmd at the end of a read buffer

Fix silly error in commit c77bfb8e4bfe6331c8ce8e71ab27afa3cf553462

v1.0.26:

Handle a partial cmd at the end of a read buffer

When a buffer is received that does not end in ';', a partial command is at the end of the buffer, and must be prepended to the following buffer that is received. The logic handling this was wrong. I hope it is correct now.

Thanks to Dick, N4UN, for reporting the bug.

Make a few more buttons into toggle buttons

Thanks to Charles, NK80.

Make ATU bypass/auto buttons into a toggle button

Add ability to cycle cyclebuttons up and down

They cycle up if the button is clicked in its upper half and they cycle down if the button is clicked in its lower half.

Change cyclebutton wrap to wraparound

It now matches the (brief) documentation.

v1.0.25:

Implement the cyclebutton widget

This is a button that cycles between several button selections. Each selection can have its own command and text or image. As a special case, when there are two selection it is a toggle button that alternates between the two states.

Change K4 error return to a warning unless debug

Unless debugging is turned on, if the K4 returns an error response, we'll just print a warning message. If debugging, it will continue to produce an exception. We haven't seen any of these errors, I just want to be able to note them if/when they occur.

v1.0.24:

Make vfo step combox box read-only

This prevents the user from inputting invalid values.

Fix checking for an error response from the K4

We now will accep responses consisting of the command base.

Fix sidetone slider

v1.0.23:

Workaround a K4 bug dealing with 100 KHz vfo steps

At least on the current version Production V36 and Beta V37.1, On startup, regardless of the VFO A step size, it reports a step size for VFO A of 100 KHz. The step size for VFO B is reported correctly.

Also, if you try to select 100 KHz steps, it shows 100 Hz steps but reports via the API that it has 100 KHz steps.

We disallow seting 100 KHz step size via the API and if we receive a step size of 100 KHz, we force it to be 100 Hz, matching what is on the K4 display.

Request the K4 to send updates immediately

Previously, updates were sent every 60 mS.

Remove a harmless exception

The condition triggering it is harmless and probably caused by a difficult-to-avoid race condition.

Bugfix: Handling receiving vfo step size was broken

Change sidetone level to a slider

Only store response after base in k4\_parameters

Instead of storing the cmd/response base, we just store the portion of the command/response after the base characters. For example, for the response "MD1", where we used to store the entire response, we now just store "1".

Add list of all command bases

For future possible use.

Check for error response from K4

Raise an exception if the K4 responds that a bad command was sent.

k4macro.yaml: Organize into two columns

This removes the need to specify as many row and column placements.

Change volume control to be a slider

k4macro.yaml: Fix 5% volume value

Cache get-only commands as well as normal cmds

Make k4macro.yaml display more vertically compact

Add back the yaml comments re the vfo settings

v1.0.22:

Add latest mods to NK80's k4macro.yaml file

Bugfix: Set vfo step size properly on startup

v1.0.21:

Change vfo step selector to a combobox

The vfo selector defaults to combobox, a spinbox can be selected by adding 'step\_spinbox: true' to the vfo yaml entry.

New k4macro.yaml changes form NK80

Improve vfo direct frequency entry method

Oops, the previous commit was too lenient, IMHO

Raise an exception when trying to add a duplicate poll request.

Make poll add/remove functions more tolerant

Add button flashing capability to radio buttons

v1.0.20:

Speed up displaying of multiple pages

vfo fixes

Handle commands with special "get format"

v1.0.19:

Fix multiple page support

Now the directives 'show\_page' and 'show\_previous\_page' now work, although page display is not fast.

Remove unused variables

Synchronize VFO step size with K4

Maintain state only for stateful commands

Do not maintain the K4 parameter state for non-parameter commands. These are the commands that are set-only or get-only. Also, disallow set-only and get-only commands in radio buttons, because radio buttons need to match the K4 state.

v1.0.18:

Unconditionally send commands

We were being smart and assumed that all commands set K4 state. This is true for most commands, but not all. In particular, the SW commands modify the K4 state, but they don't directly represent state. Now, we'll send commands even if we recently sent the same command. In the future, we could distinguish between those commands that represent state and those that don't, but it likely isn't worth it.

v1.0.17:

Update example\_config\_files/k4macro.yaml  
Incorporate the latest changes from NK80.

Always request parameter status when subscribing

When subscribing to a K4 parameter, unconditionally request the parameter's current status.

Rework polling mechanism

Add capability of having poll callbacks with differing intervals. Right now we send vfo updates after 50 mS from the last change and poll receiving from the K4 every 1 mS.

Remove server.base\_cache

It is an unnecessary kludge

v1.0.16:

Internally maintain the K4's parameter status

This simplifies the handling of radio buttons that send multiple commands and responses, permitting some radio button bug fixes.

v1.0.15:

Implement a slider widget

An example using it to set/get the K4 power output has been added to example\_config\_files/k4macro.yaml

k4macro.yaml: incorporate changes from NK80

Fix debug msg on received bytes vs string

Print debug messages to stderr

Do not decode responses from non-K4 servers

Add two config files created by KX4UI

Clarify a message when the K4 has disconnected

v1.0.14:

Gracefully handle server (k4) shutdown

v1.0.13:

Rework radio button response handling

Now gracefully handles radio buttons that do not include all possible values for the particular parameters controlled by the radio buttons.

Rename k4macro.yaml.nk8o to k4macro.yaml

Keep k4macro.yaml.nk8o as the only current example config file and rename it k4macro.yaml.

Simplify specifying server startup commands

This simplifies the implementation as well as the specification of startup commands.

Remove no longer needed comments from config

README: Add request for problem reports

v1.0.12:

Add ability to send commands to K4 at startup

Cleanup the code a bit

Bugfix: Button cmd sending was broken, now fixed

v1.0.11:

v1.0.10:

NK80's latest yaml

Permit multiple callbacks on same radio response

v1.0.9:

v1.0.8:

v1.0.7:

Raise default button font size to 10

This makes the smaller controls easier to see and to click.

We leave the default label size at 8 as they don't need to be as prominent as the controls, IMHO.

Close the program if a server is not found

v1.0.6:

v1.0.5:

Fix bug to handle text that is an integer

Text in the yaml file that happened to be an integer was being treated as an int rather than a string. Force it to be a string by using the str() function.

v1.0.4:

v1.0.3:

Remove debug msgs for delay between servers

Add email address to README

v1.0.2:

Specify multiple cmds to servers via a list

There was a problem when specifying multiple commands to multiple servers via a dict. Repeated commands to the same server would replace previous commands to that server in the dict. We now specify a list of dicts to alleviate that problem.

v1.0.1:

Version 1.0.0

First release to github