SDR

xSDR6000 for macOS

Introduction

xSDR6000 was written by Douglas Adams (K3TZR, douglas.adams@me.com) to allow a Mac user to operate a FlexRadio (TM) 6000 series radio from a Mac. It runs on macOS 10.11 and higher.

The source code can be found at:

https://github.com/K3TZR/xSDR6000

An executable is contained in the *Release* on that same repository, look for the "Current Release" unless you have a special interest in an earlier release.

https://github.com/K3TZR/xSDR6000/releases

Installation

Install xSDR6000 by simply copying the application bundle from the *Release* to a place of your choice (typically the "Applications" folder). Nothing else to install.

Using xSDR6000

xSDR6000 looks and works in much the same way that SmartSDR (TM) does. This is not a complete manual of its operation, unfortunately I don't have the time or resources to produce such a manual. For the most part, its operation is fairly obvious. I will attempt to document the less obvious parts here.

This is a work in progress, feel free to contact me (douglas.adams@me.com) with suggestions, bug reports, etc. Please be patient, this is a one man operation.

ScrollWheel / Trackpad / Mouse usage

Action	Where	Result
Wheel Up / Trackpad Up	Not in Slice	Active Slice frequency up/ down
	In Slice	Filter Width increase / decrease
Left Click	Not in Slice	Move active Slice
	In Slice	Make Slice active
Left Drag (left <-> right)	Not in Slice	Move Panadapter center
	In Slice	Move Slice
	Over Frequency Legend	Panadapter bandwidth increase / decrease
Left Drag (up <-> down)	Not in Slice	n/a
	In Slice	Filter Width increase / decrease
	In Db Legend (top half)	Increase / decrease upper Db level
	In Db Legend (bottom half)	Increase / decrease lower Db level
Right Click	Not in Slice, not in Tnf	Menu - Create Slice, Create Tnf
	In Slice	Menu - Remove Slice, Create Tnf
	In Tnf	Menu - Create Slice, Remove Tnf, Tnf Settings
	In Tnf & Slice	Menu - Remove Slice, Remove Tnf, Tnf Settings

Shortcut Keys

Key	Action	Menu
n/a	Enable / Disable SmartLInk	Radio->SmartLink enabled
⌘,	Show Preferences	xSDR6000->Preferences
⌘⇧P	Show Profiles	xSDR6000->Profiles
⌘⇧L	Show Log Viewer	xSDR6000->Log Viewer
⌘P	New Panadapter	Radio->New Pan
⌘T	Enable Tnfs toggle	Radio->Tnf On/Off
⌘M	Show Markers toggle	Radio->Markers On/Off

Key	Action	Menu
⌘S	Show Side view toggle	Radio->Side View On/Off
⌘A	Enable Mac Audio toggle	
⌘N	Cycle through the Slices on a Pan	Radio->Next Slice
⌥	When in Flag->XRIT, Option key toggles between "Tuning Step" and "Split Step"	
⇧	Holding Shift key down while scrolling makes tuning step = 100 Hz	
⌥	Holding Option key down while scrolling makes tuning step = 10 Hz	
⇧⌥	Holding Shift & Option keys down while scrolling makes tuning step = 1 Hz	

Unimplemented features

At present, there are many unimplemented features. SmartSDR is a large and complex piece of software with hundreds of features. I have tried to address the most basic features first. I'm always interested to learn what features would be most useful to anyone using this software. Please let me know what you would like me to address next.

- Export / Import profiles
- FlexControl support
- Xvtr preferences page
- Gps preference page
- TX Band Settings preference page
- CWX user interface
- TX audio from the Mac
- Spots
- USB cables (partially implemented)
- Amplifier functionality (partially implemented)
- Software interlocks
- The "OPT" flag tab when FM mode selected
- Waveforms (partially implemented)
- Focus helper
- Network health monitoring
- Initial SmartLink setup (account creation, port settings)
- 2-Meter support for Flex-6700
- RTTY frequency lines

- Many more that I haven't yet discovered in SmartSDR

Known issues

Credits

xSDR6000 is built on top of my xAPI framework (https://github.com/DougPA/xLib6000).

Thanks to Mario Illgen (DL3LSM) for many hours of testing and for writing portions of the xAPI framework related to DAX streams. Please see his DAX and CAT apps (http://dl3lsm.blogspot.de).

Thanks to Robbie Hanson, Deusty LLC and the Apple development community for the CocoaAsyncSocket framework (https://github.com/robbiehanson/CocoaAsyncSocket).

Thanks to Radek Pietruszewski for SwiftyUserDefaults (https://github.com/radex).

Thanks to the Opus organization (https://opus-codec.org/downloads/) for the code to produce OpusOSX.framework.

Thanks to FlexRadio for the hardware and opening up the API and the FlexLib sources for 3rd party developers (http://www.flexradio.com).

License and Disclaimer

The MIT License (MIT)

5/29/2020