

sBITX Toolbox Guide

v.3.0

Table of Contents

Introduction.....	1
Prerequisites.....	1
Installation.....	2
sBITX Toolbox	2
Applications	5
sBITX Manager	6
sBITX Editor.....	8
CW Detector	9
sBITX Scanner.....	9
sBITX EZ Data Utility.....	11
sBITX Backlight Control.....	11
sBITX Screen Resizer.....	12
sBITX Screen Scaler.....	13
sBITX USB Headset.....	13
sBITX Voice Keyer	15
sBITX Macro Manager.....	17
sBITX Log Exporter	18
sBITX Time Keeper.....	19
Update Manager.....	21
V3 PowerCal.....	22
Advanced Troubleshooting.....	23

Introduction

Welcome to the sBITX Toolbox Guide! This guide is here to introduce you to a set of applications created by JJ (W9JES), designed to improve your experience using the sBitx transceiver.

This guide is designed to help you from installation, using, updating, and removing the Toolbox. Each section is broken out into categories to quickly find what you need. You are encouraged to read the entire guide if you are new to the sBitx platform, Linux, and my applications.

Prerequisites

- An internet connection configured and active on your sBitx transceiver

- USB keyboard/mouse, virtual keyboard, SSH, or VNC for initial installation
- USB keyboard/mouse, virtual keyboard, or VNC required for entering text in some applications.
- sBitx 64 bit software version 3.025 or higher
- Python version 3 (if you are using a custom image or OS). The script will check and install Python if it is not present.
- Minimum OS: Raspberry Pi (Raspbian) version 12 “Bookworm”
 - Run **cat /etc/os-release** from terminal for verification

Installation

1. Open the terminal application on your sBitx



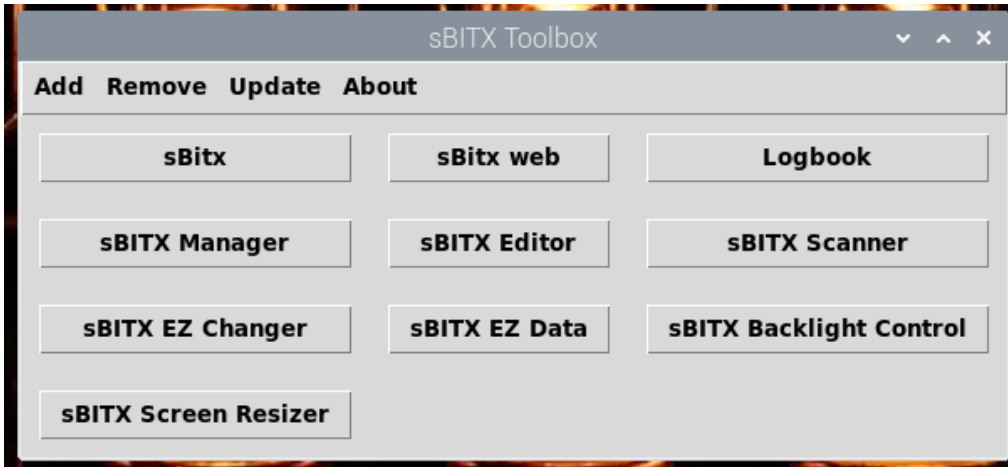
- 2.
3. Type: **cd \$HOME && git clone https://github.com/drexjj/sBITX-toolbox64.git && cd sBITX-toolbox64 && chmod +x *.sh && ./installer.sh**
4. Your screen should look like this when complete.

```
pi@sbitx:~ $ cd $HOME && git clone https://github.com/drexjj/sBITX-toolbox64.git
&& cd sBITX-toolbox64 && chmod +x *.sh && ./installer.sh
Cloning into 'sBITX-toolbox64'...
remote: Enumerating objects: 191, done.
remote: Counting objects: 100% (149/149), done.
remote: Compressing objects: 100% (100/100), done.
remote: Total 191 (delta 79), reused 84 (delta 45), pack-reused 42
Receiving objects: 100% (191/191), 38.43 MiB | 21.35 MiB/s, done.
Resolving deltas: 100% (88/88), done.
sBITX-Launcher setup completed successfully! The applications have been added to
the Pi Menu
pi@sbitx:~/sBITX-toolbox64 $
```

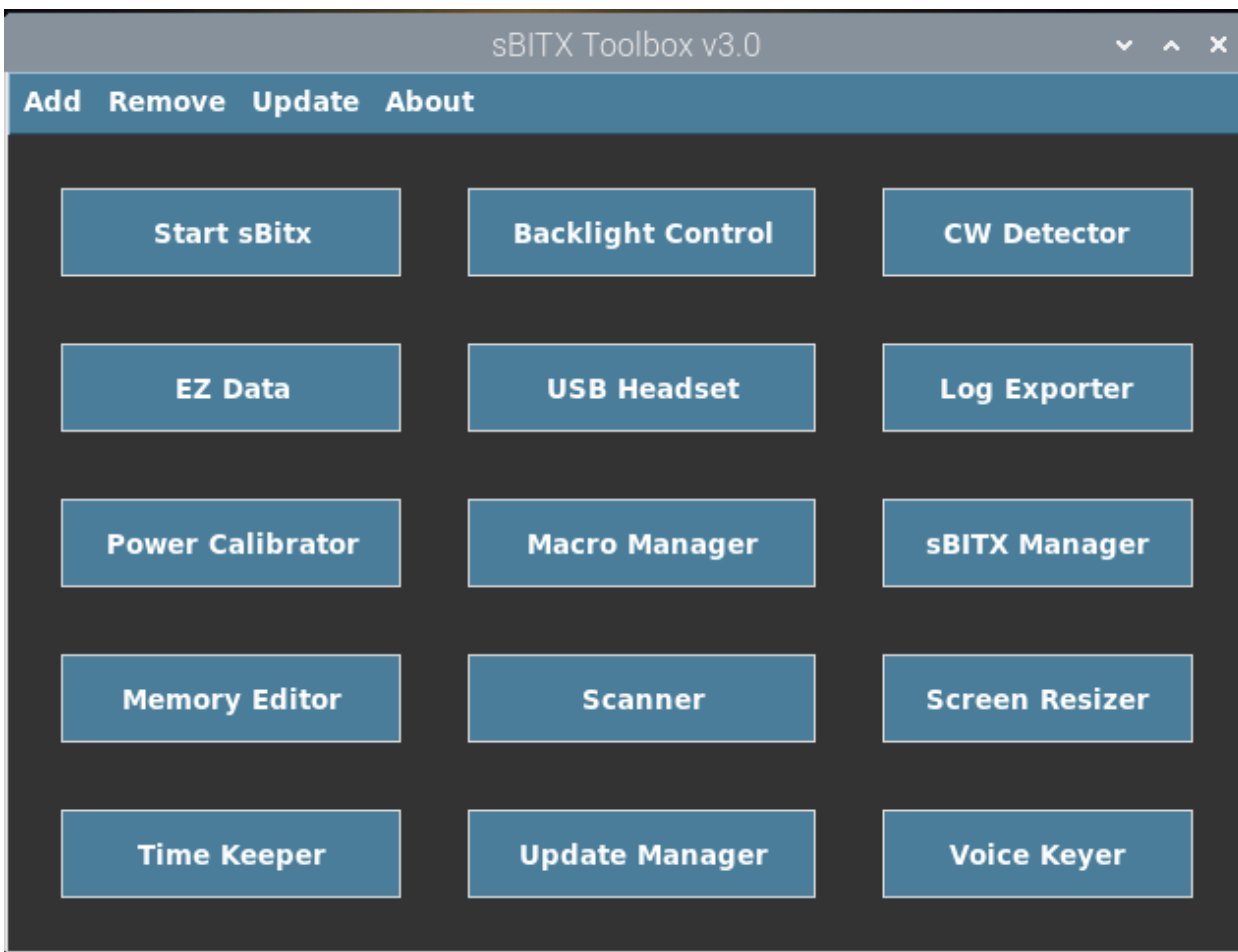
5. Wait for the installation to complete then close the terminal.
6. Verify the icons have been added to the desktop and Pi menu.



sBITX Toolbox

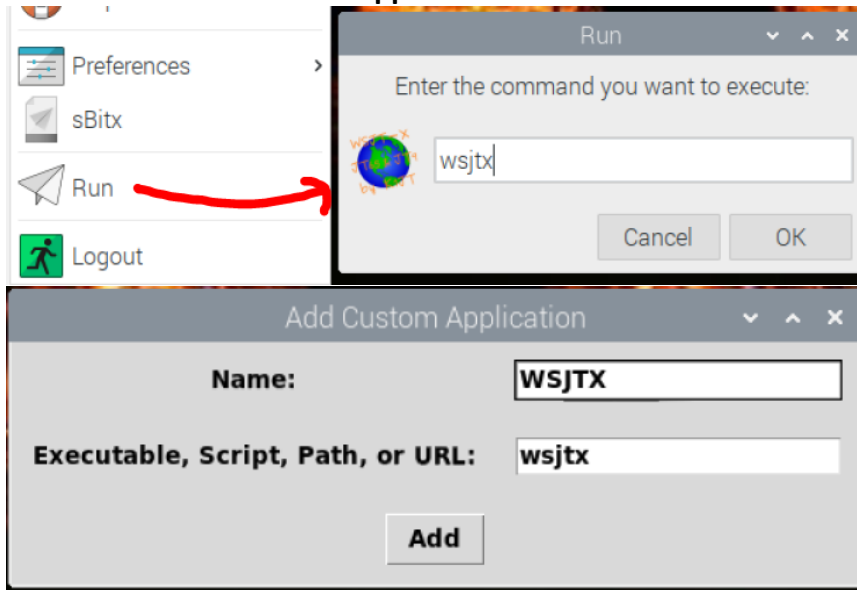


Using the sBITX Toolbox is fairly straightforward. The top menu contains commands and options to **add** or **remove** application buttons to the main area. Add or remove apps by selecting them from the dropdown menu. You can change the order with the add/remove buttons. Press a button to open the app of your choice.

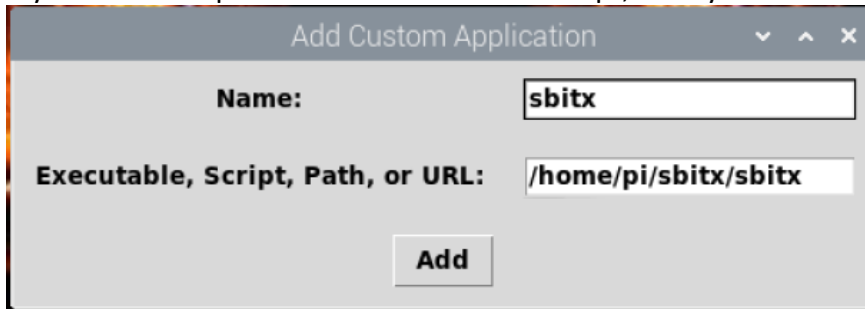


You can also add custom buttons such as URLs, executable binaries or scripts. Click **Custom Application**. The format for a URL is **<https://mywebsite.zzz>** or **<http://mywebsite.zzz/mypage>**. For example, to add your QRZ profile, type **<https://qrz.com/db/CALLSIGN>**. To add an executable binary or script application like WSJTJX, you can either try one of these options.

1. Click Run from the Pi menu and type the executable name to test. If it works, then add this same format in the **Add Custom Application** box as shown.

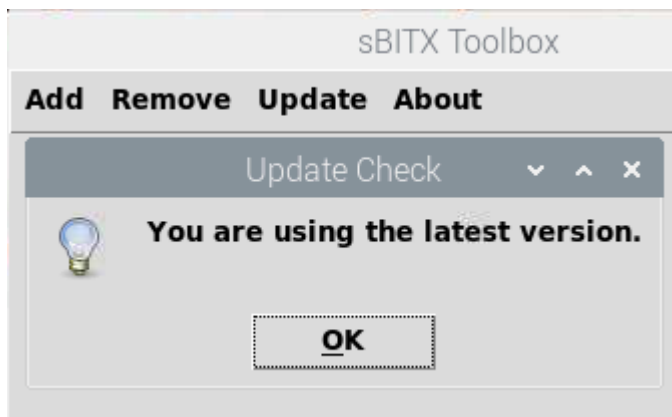


2. If you know the path of the executable or script, then you can enter it as shown.



3. Some executable apps or scripts may not be compatible.

Periodic updates to the sBITX Toolbox applications and features may be added over time. Press the **Update** button to check for updates and install them. It is best to visit https://github.com/drexji/sBITX-toolbox64/blob/main/release_notes.txt to see what has changed before updating your apps.



You can check the currently installed version from the **About** menu.

Applications

sBitx

A shortcut to open the sbitx application.

sBitx web

A shortcut to open the web version of sbitx.

Logbook

A shortcut to open the web-based, sBitx logbook.

sBitx Manager

An app that performs auxiliary, “rig control” of your sBitx.

sBitx Editor

A frequency editor application for sBITX Manager.

sBITX Scanner

An app that adds frequency range scanning functionality to your sBitx

sBITX EZ Data

An application for saving/restoring the hardware and software configuration data via USB drive.

sBITX Backlight Control

An application to conveniently control the screen brightness of the built-in display on your sBITX.

sBITX Screen Scaler

A handy tool for adjusting screen resolution, allowing apps with larger GUI windows to fit entirely on the built-in display.

sBITX USB Headset

An application that allows for a USB headset, headphones, or microphone to be used with your sBitx.

sBITX Voice Keyer

An application that allows you to record, store, and transmit prerecorded voice messages commonly used for contests, DXing, POTA activations.etc with the press of a button.

sBITX Macro Manager

An application deigned to edit, create, and manage macros on the sBITX transceiver.

sBITX Log Exporter

An application to select and download logs from the sBitx database and export them to an ADIF format to be used with other logging applications.

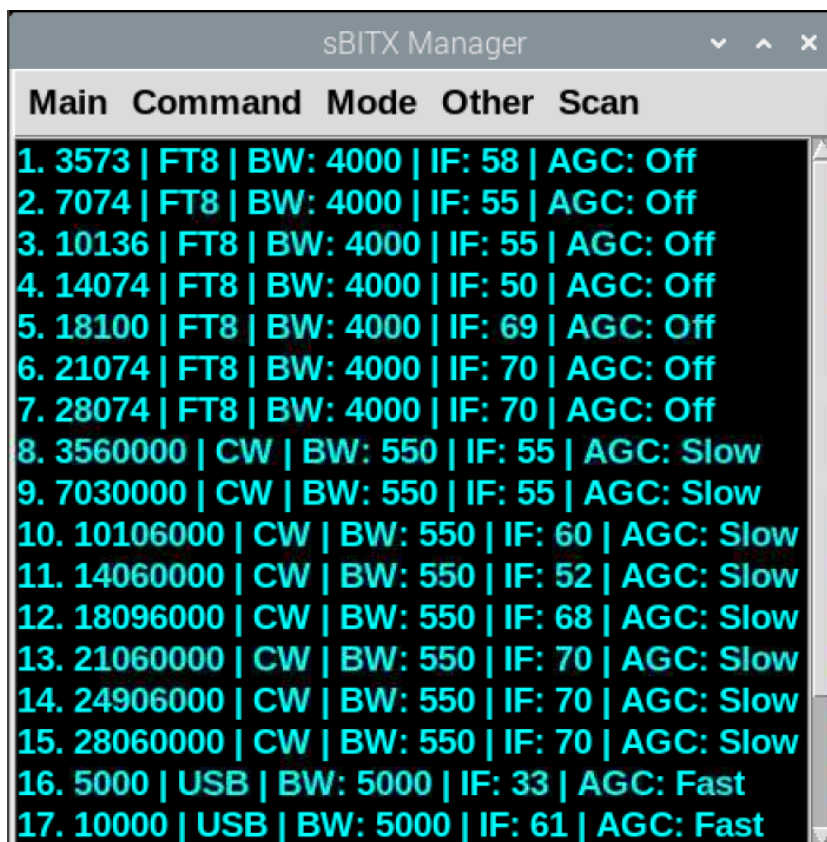
sBITX Time Keeper

A utility that syncs the real-time clock (RTC) module inside your sBitx transceiver to the system clock or accurate time source. Useful for off-network communications and data modes.

V3 PowerCal

A calibration utility to adjust the power output per band.

sBITX Manager



Using sBITX Manager is much like using other rig control software even with the limitations of the sBitx transceiver. The key difference is that this application uses the telnet protocol to establish connection and send or receive information.

Features:

- Provides a user-friendly interface for interacting with the sBITX transceiver.
- Memory management allows for adding, editing, and removal of frequencies.
- Scanning function that scans through the list of stored frequencies at a customized interval.
- Send commands specifying details such as Frequency, VFO, Step, Mode, Bandwidth, IF, AGC, Audio, and more.
- Decodes messages in FT8 and CW modes.
- Visual adjustments such as text size and dark mode for enhanced readability.

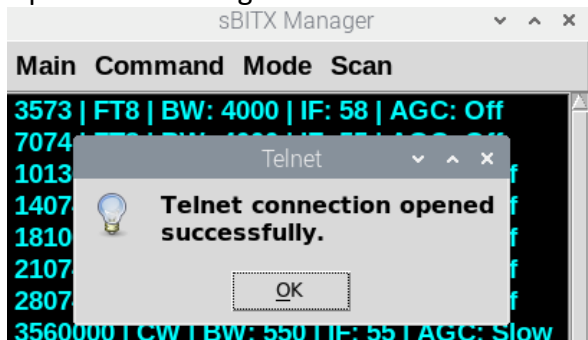
The interface is an attempt at a “ham’s don’t read manuals” layout and closely resembles the user interface text of the sBITX. The menu bar contains most of the commands that require your input. You can select commands from the dropdown menu and either select or type a command. The sBitx will respond to the command you choose and set the desired value. The main window consists of the frequency memory bank where you can press to set the sBitx frequency.

The memory bank for this app is stored in a file called **sbmanager.json** in the /home/pi/sBITX-toolbox/data directory. A prepopulated frequency list is provided or you can start your own list by simply deleting the file and reopening sBITX Manager, then add a frequency to the memory bank.

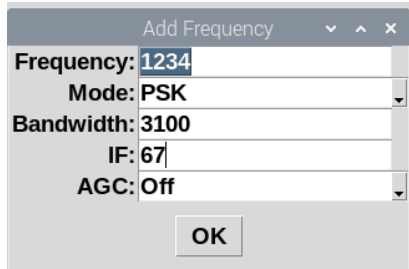
A word of caution: The sBitx transceiver can only allow one active telnet session at a time. That means you cannot use sBITX Manager and sBITX Scanner simultaneously.

Steps:

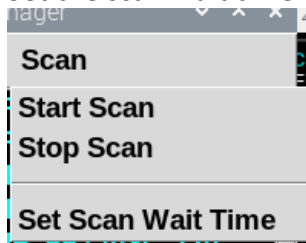
1. Ensure that the sBITX transceiver application is open and functional
2. Open sBITX Manager and click **Connect** from the **main** menu. You should see a popup message



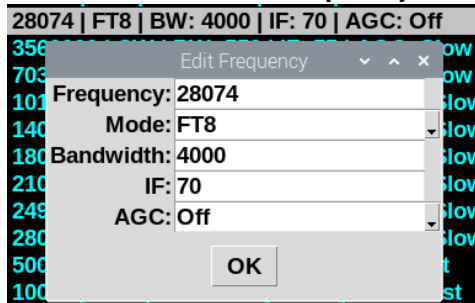
3. If you receive an error and/or the commands and stored frequencies don’t work, then close the sBitx app, close sBITX Manager. Start at step 1 and try again.
4. Enter the desired command or add/edit a frequency in the memory bank.



5. The scan function selects the first memory in the bank and scans “downward” in the memory bank list. Set the scan wait time to pause between memory bank changes.



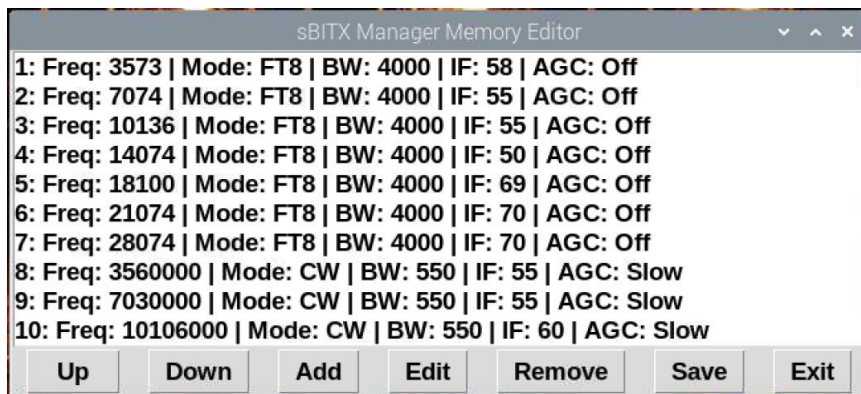
6. Editing or removing frequencies is easy. Select the frequency from the memory bank and go to the **Command menu → Frequency → edit/remove**. Make your changes and press OK .



7. When you are done using the app, select **Disconnect** or **exit**. The app will close, however, the telnet session may still be “active” in the sBitx app (there is a bug on the sbitx side). I recommend you restart the sbitx app when finished with a “telnet” session.

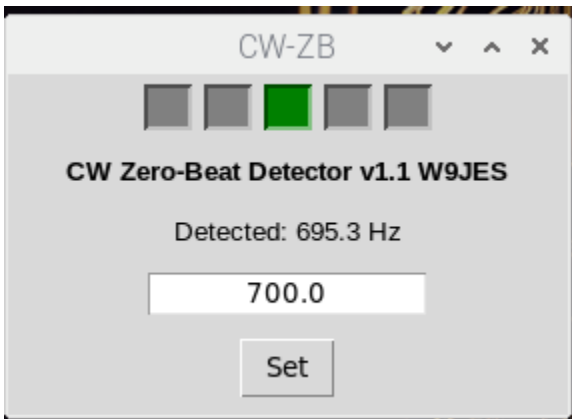
A standalone version of this app is available for Windows 10/11 as well. <https://github.com/drexjj/sBITX-toolbox/releases> You can use this app on a computer that is connected to the same network as your sBitx to perform remote control functions. It is a great companion with the web version sBitx too.

sBITX Editor



sBITX Editor is a companion to sBITX Manager and allows you to change the order of, add, edit, or remove frequencies in the memory bank. Select a frequency to change the order or edit the settings. Click **Save Order** when finished and restart sBITX Manager to see your changes.

CW Detector



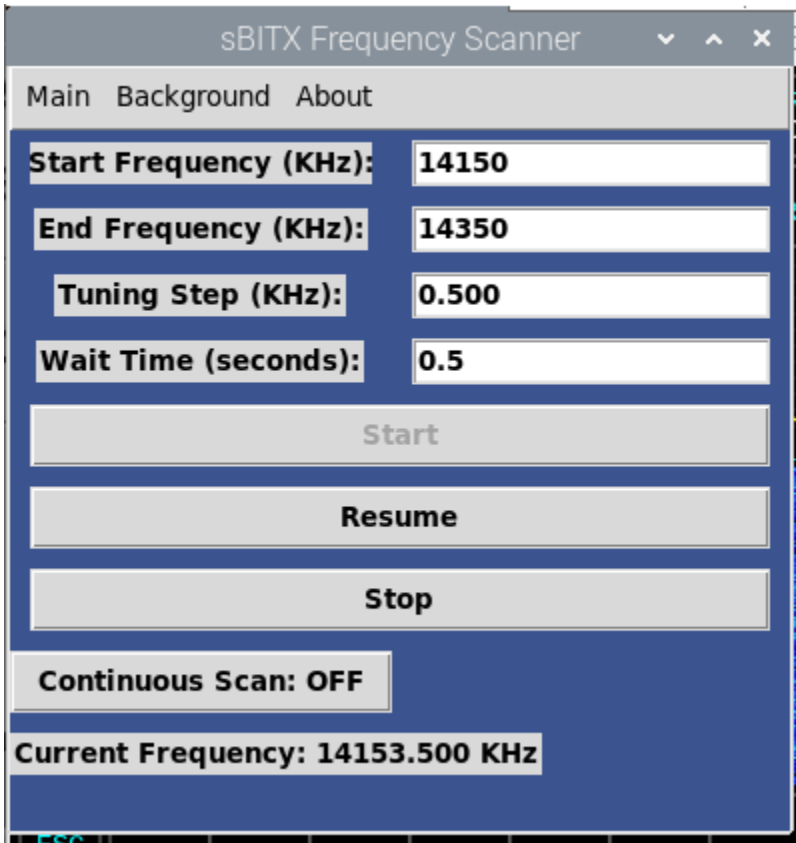
A visual aid designed to help tune into an active CW signal, enabling the user to "zero-beat" with the sender's frequency. It is commonly used to precisely align the tone with the sender's for effective communication during a QSO.

Set your desired RX tone (usually the same as your TX tone) and press Set. The bars will light up red, yellow or green as you tune to a CW signal. Align the LEDs where mostly green is showing up while tuning.

- The far left LED lights up red when the detected frequency is 20 to 50 Hz below the target tone.
- The left middle LED lights up yellow when the detected frequency is 6 to 20 Hz below the target tone.
- The middle LED lights up green when the detected tone is within ± 5 Hz of the target tone.
- The right middle LED lights up yellow when the detected frequency is 6 to 20 Hz below the target tone.
- The far right LED lights up red when the detected frequency is 20 to 50 Hz below the target tone.

The text box is used to set the target tone. Default is 700 Hz unless you specify a tone value. Clicking or pressing the barograph will shrink the display for easier viewing.

sBITX Scanner



sBITX Scanner turns your sBitx into an automatic scanner to find open or active frequencies.

Features:

- Choose the frequency range, tuning step, and wait time
- Bidirectional scanning capabilities
- Continuous loop scanning
- Current frequency display in the app
-

A word of caution: The sBitx transceiver can only allow one active telnet session at a time. That means you cannot use sBITX Manager and sBITX Scanner simultaneously.

Steps:

1. Ensure that the sBITX application is open and functional
2. Open sBITX Scanner and click **Connect** from the **main** menu. You should see a popup message



3. If you receive an error and/or the commands and stored frequencies don't work, then close the sBitx transceiver app, close sBITX Scanner. Start at step 1 and try again.
4. Select a start and end frequency then select the tuning step and wait time between frequency changes.

5. Press start and watch the current frequency increment. You will hear the band activity through the speaker. If you find a frequency that you want to stop at, press pause or stop.
6. When you are done using the app, select **Disconnect** or **exit**. The app will close, however, the telnet session may still be “active” in the sbitx app (there is a bug on the sbitx side). I recommend you restart the sbitx app when finished with a “telnet” session.

A standalone version of this app is available for Windows 10/11 as well. <https://github.com/drexij/sBITX-toolbox/releases> You can use this app on a computer that is connected to the same network as your sBitx to perform remote control functions. It is a great companion with the web version sBitx too.

sBITX EZ Data Utility

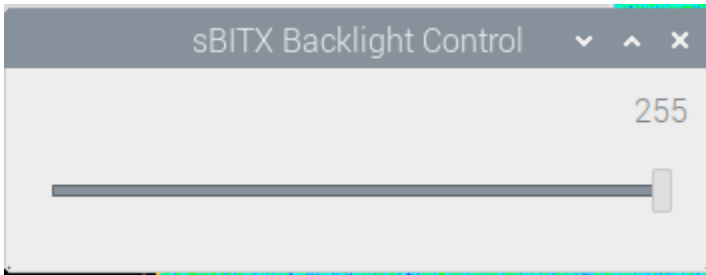


This app allows you to save, restore, and export backups of your calibrated hardware files located in **/home/pi/sbitx/data** and **/home/pi/sbitx/web** folders.

Steps:

1. Insert a USB drive into an open slot of your sBitx, then open **sBITX EZdata**.
2. Click **Refresh** and select the appropriate drive from the list.
3. Click "**Backup Files To The Selected USB Drive**".
4. A new folder will be created on the USB drive called **sBitx_Data_backup**.
5. Inside that folder will be your backup files with the date and time of the backup.
6. You can select a date to restore the data folder back to the sBitx transceiver.

sBITX Backlight Control



A convenient way to control the backlight level (brightness) of the sBitx display for eye comfort and reduced power draw.

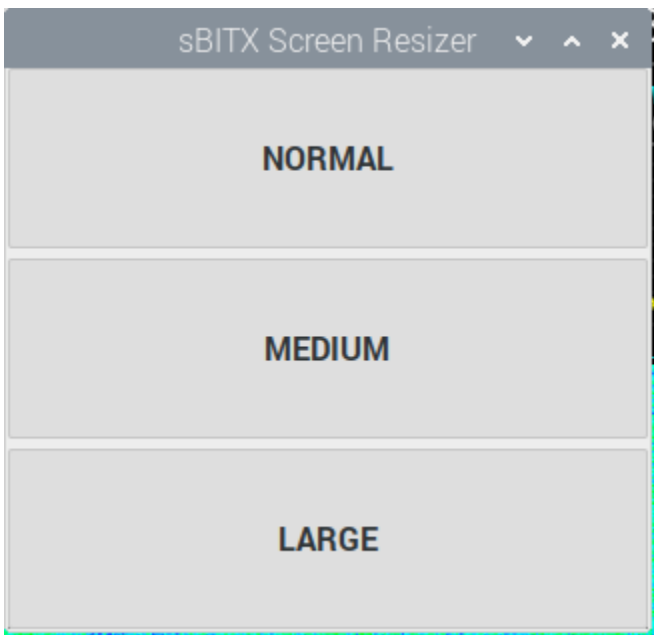
A touchscreen friendly slider control allows you to change the brightness.

The backlight level setting is stored on the system and will remain enabled after a reboot.

The minimum brightness level is set to 15 which is barely readable in low light settings to consume less power and extend life of the display.

Simply drag the slider control to your preferences.

sBITX Screen Resizer



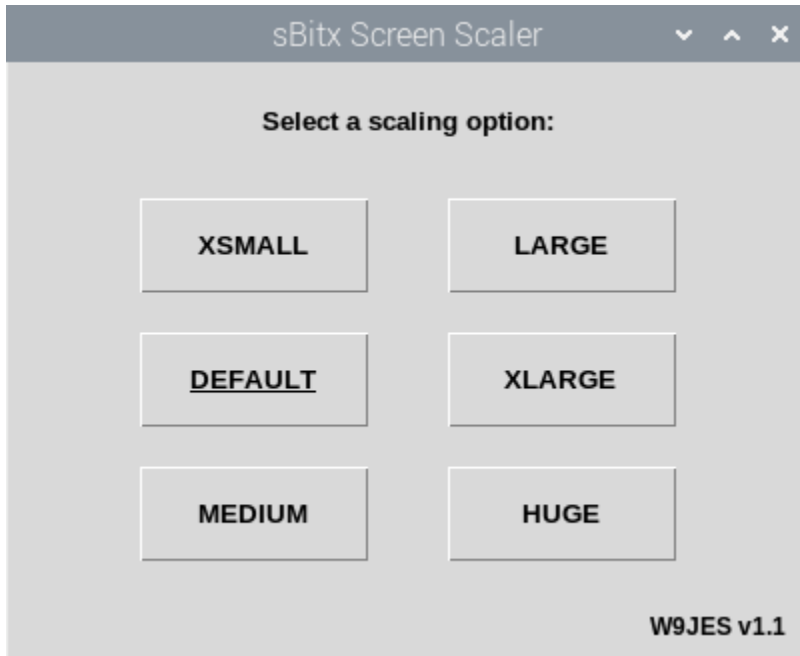
An application that allows you to easily change the screen size of your sBitx desktop equipped with the 7 inch Raspberry Pi display.

The app has three settings which change the resolution as follows:

- Normal - The default, sBitx resolution of 800 X 400
- Medium - 1024 x 768 resolution for running apps like WSJT-X with screen panning enabled.
- Large - 1820 x 1024 resolution for running multiple apps via VNC with screen panning enabled.

The transceiver will default back to the normal screen size on reboot.

sBITX Screen Scaler



An application that allows you to temporarily change the screen size of your sBitx desktop equipped with the 7 inch Raspberry Pi display. This is useful for apps with larger GUI windows to fit entirely on the built-in display such as WSJTX

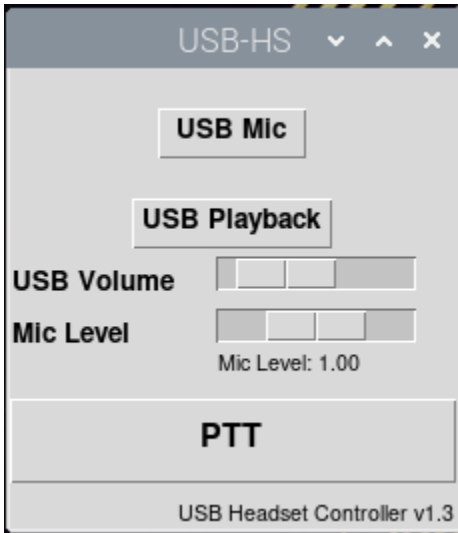
The app has six buttons which change the scaling resolution as follows:

- Default - The native, scaling resolution of the built-in display
- XSMALL – Useful when using the Raspberry Pi Touch Display 2 only to work with the 800X480 screen size of the sBitx app for touchscreen use.
- MEDIUM to HUGE - Increases scaling of 1.28 to 1.9

Important Note: Running the huge or xlarge option can cause slowness due to GPU rendering. It also results in the text and graphics to be smaller and harder to read. It is recommended to use an external mouse.

The transceiver will default back to the normal screen size on reboot.

sBITX USB Headset



sBITX USB Headset is an application to control a USB headset, USB headphones, or USB microphone with your transceiver.

Instructions:

1. Plug in your USB audio device
2. Power on the sBitx
3. Run the sBitx transceiver app and set mode to **LSB** or **USB**
4. Open the sBITX USB Headset app
5. Press the Enable USB Mic button to enable microphone audio and/or press the USB Audio ON/OFF button to route receiver audio to the USB headphones.
6. Adjust the slider controls for your desired levels

Transmitting:

The PTT button can be controlled by pressing it momentarily to place sBitx into TX mode and again to switch back to RX mode.

You can also press the spacebar key on an externally connected keyboard to control the PTT function as well. To enable the spacebar function, simply toggle the USB Audio On/OFF button then ensure the sBITX USB Headset Controller window is selected on top of sBitx app. Then press the spacebar to enable TX mode or RX mode.

The mic level can only be adjusted when the sBitx is not in TX mode.

If you are changing to FT8 or digital modes, press the Disable USB Mic button.

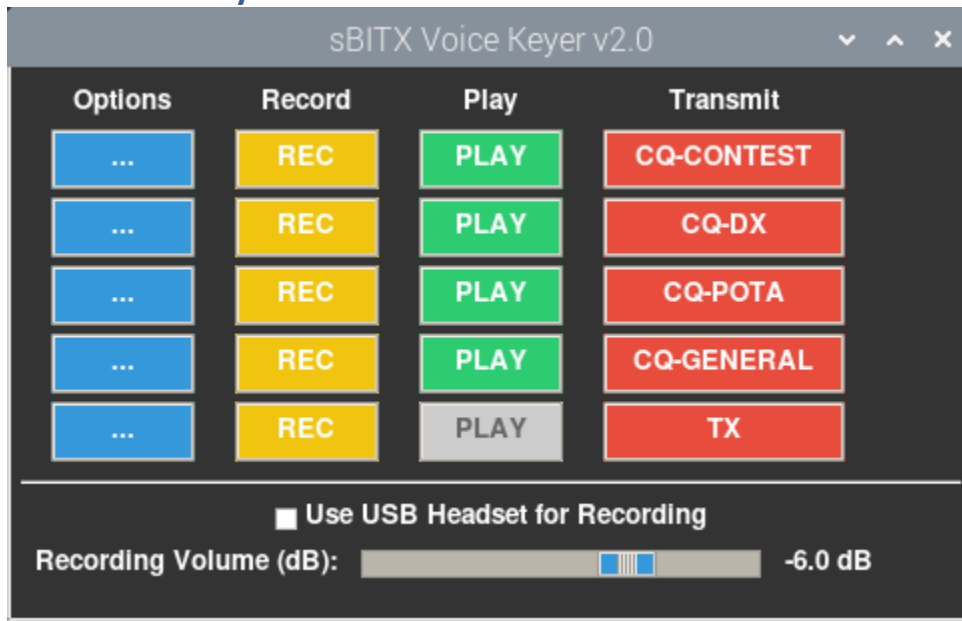
Listening:

Press the USB Audio On/OFF button to listen to audio on the USB headphones or USB speaker. The volume control is independent of that coming out of the built-in speaker of the sBitx so you may want to turn that volume down when using the headphones.

NOTE: There is a slight delay between transmit audio and receive audio due to the loopback conversion process.

Please see the advanced troubleshooting section for USB audio device compatibility and troubleshooting.

sBITX Voice Keyer



An application that allows you to record, store, and transmit prerecorded voice messages commonly used in contests, DXing, POTA activations, and more.

The interface consists of four columns labeled: **Options**, **REC**, **PLAY**, and **TX**. Each row corresponds to a different voice keying slot, allowing you to manage and transmit multiple voice recordings.

Usage:

1. File Management:

- **Create New File:** Click the '...' button corresponding to the desired slot and select '**Create**'. Choose a filename and save location for the new WAV file.
- **Open Existing File:** Click the '...' button and select '**Open**' to choose an existing WAV file from your system.
- **Edit Filename:** Click the '...' button and choose '**Edit Name**' to rename the selected file.
- **Save As:** Click the '...' button and select '**Save As**' to save the file under a new name and/or location.

2. Recording:

- Click the '**REC**' button corresponding to the slot where you want to record. This starts recording audio. Click again to stop recording.
- Note: Recording is disabled if sBITX is running.

3. Playback:

- Click the '**PLAY**' button to play the audio file through the sBitx speaker in the corresponding slot. Click again to stop playback.

- Playback is disabled during active transmission or when sBITX software is running.

4. Transmitting:

- Click the '**filename**' button column to start transmitting the audio file in the corresponding slot.
5. Click the checkbox to use a USB headset for recording.
 6. Adjust the recording volume slider to set the appropriate gain during the REC function.

Troubleshooting:

Playback or Recording Issues:

- Ensure your audio hardware is properly configured and that the correct device is selected in the system or application settings.

Connection Errors:

- Verify that the sBITX transceiver software is running if you encounter connection issues.

File Not Found:

- Check the file paths and filenames to ensure they are correct if playback or transmission errors occur regarding missing files.

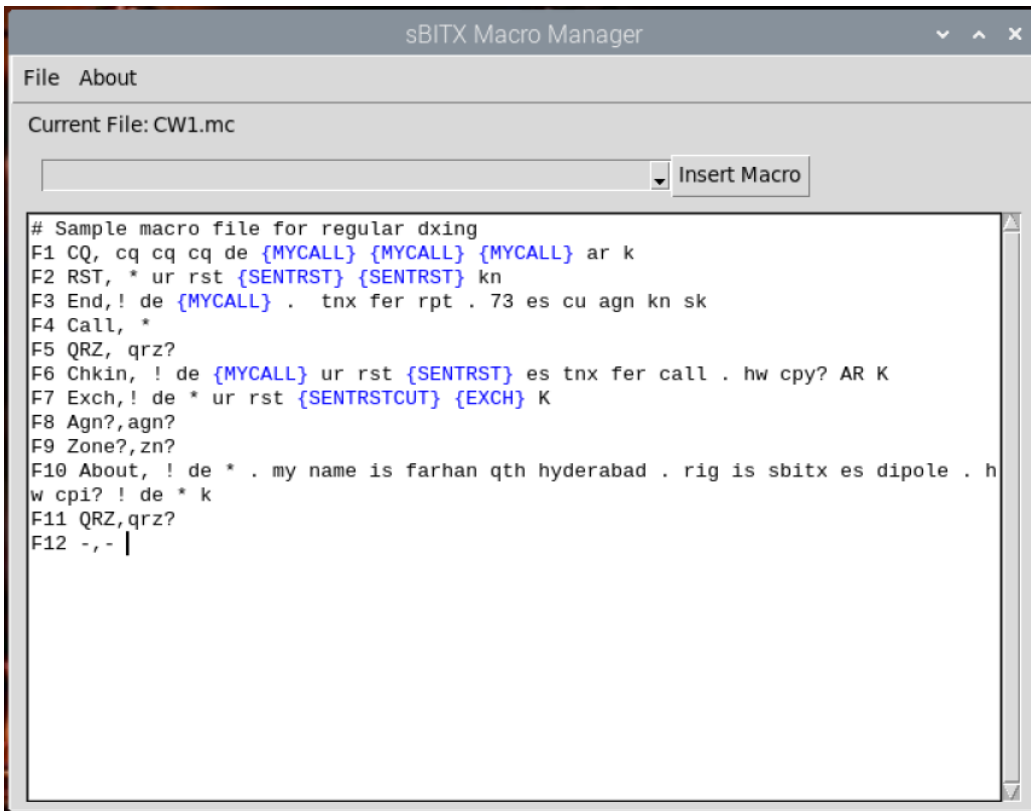
Tips:

You can use Audacity on your sBitx to modify or enhance the wav files for better results. To install Audacity on your sBitx, open terminal and type **sudo apt install audacity** It will show up in your application menu



The Audacity reference manual is at <https://manual.audacityteam.org/> and YouTube tutorial video at <https://www.youtube.com/watch?v=vlzOb4OLj94>

sBITX Macro Manager



Creating a New Macro File:

1. Click on "**File**" in the menu bar and select "**New**". This will open a template in the text area where you can start adding your macros.
2. Use the provided template and modify its contents.

Opening an Existing Macro File:

1. Click on "**File**" in the menu bar and select "**Open**".
2. The default location where sBITX macro files are stored is displayed, select the desired file, and click "**Open**". The contents of the file will be loaded into the text area for editing.

Saving Macro Files:

- **Save:** To save your changes to the currently opened file, click "**File**" → "**Save**".
- **Save As:** To save your macro file with a new name or location, click "File" → "Save As...", enter the file name, and click "**Save**".

Inserting Macros:

1. In the macro frame, use the dropdown menu to select the macro you want to insert into your file. The dropdown shows both the macro code and a brief description for each.
2. Click the "**Insert Macro**" button. The selected macro will be inserted into the text area at the current cursor position.

Editing Macros:

- Type directly into the text area to add or modify the text and macros in your file.

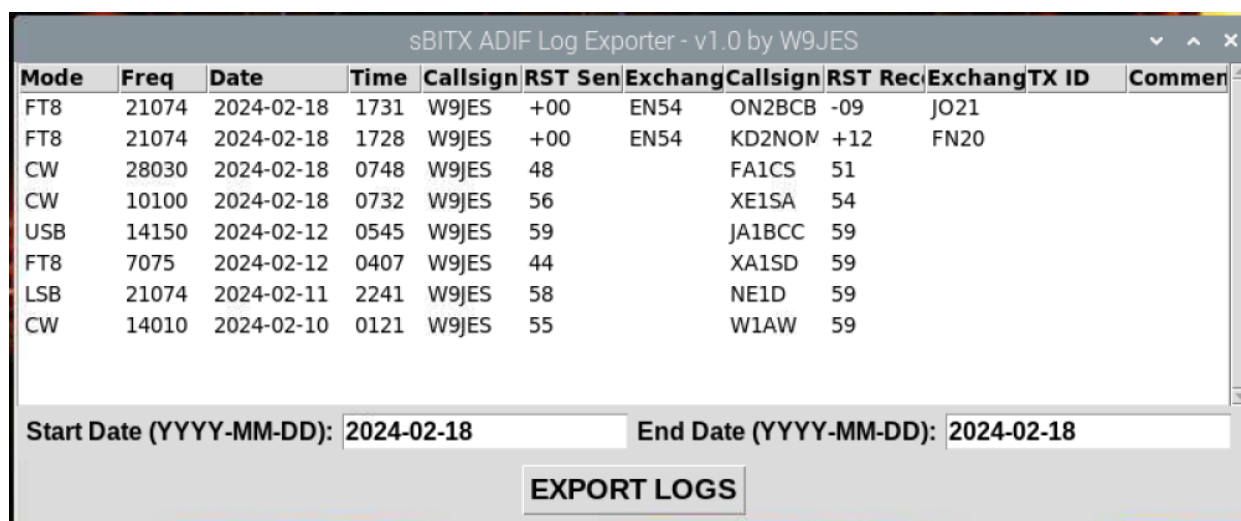
Highlighting Macros:

- As you type or after inserting macros, the application will automatically highlight valid macro codes in blue for easy identification.

Tips for Effective Use

- Familiarize yourself with the available macros and their purposes to make the most out of your macro files.
- Test the macros on low power into a dummy load or on an unused frequency to ensure desired functionality.
- You can load the macro into your sBitx by using sBITX Manager or by typing \macro "macro name" in the text field on the sBitx application.
- Use the keyboard from sBITX Toolbox or connect and external keyboard to enter text.

sBITX Log Exporter



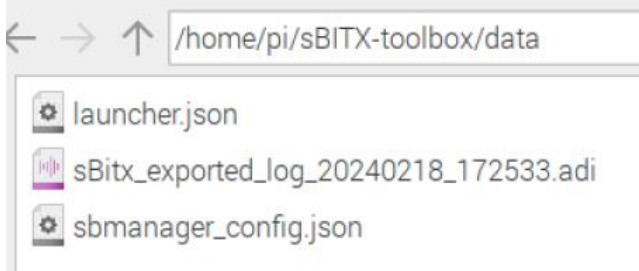
sBITX Log Exporter is an application that allows you to select and export logs from the built-in sBitx log database to use them in other logging applications. The file format is standard ADI and can be imported into various online and software based loggers.

You will need to use the onscreen keyboard from the toolbox or an external keyboard to enter the date. Select the date range you would like to export to the file and click **EXPORT LOGS**. A popup will display the filename and the number of entries added to the file.

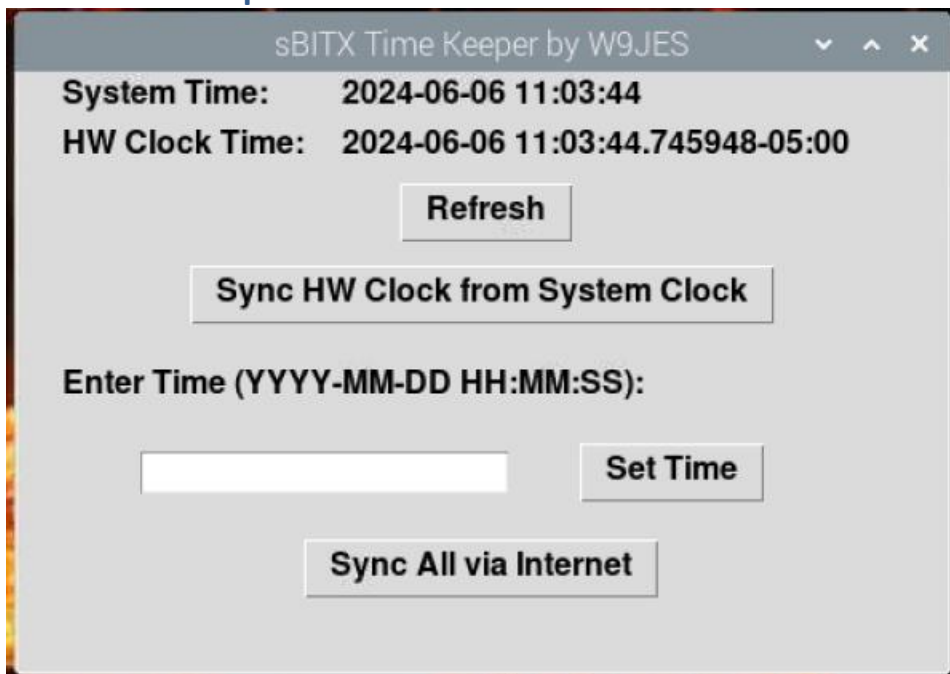


: 2024-02-18 End Date (YYYY-MM-DD): 2024-02-18

The logs are stored in **/home/pi/sBITX-toolbox/data** and written with the date and time of export. You can use the web browser, USB storage drive, SSH, or other means to transfer the adi file to your preferred logger.



sBITX Time Keeper



Time Keeper is designed to leverage the real-time clock (RTC) module within your sBitx, ensuring precise timekeeping for the sBitx application and third party applications. This utility enables you to set the RTC using an external time source and update the system clock, which is essential for digital modes and logging. It is particularly beneficial for POTA operations or for storing your sBitx for long periods while it is powered down and disconnected from the internet.

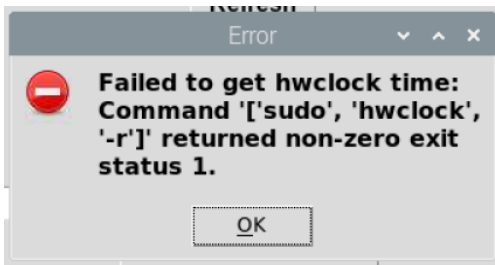
The **Refresh** button can be used to get the current clock and RTC time.

Pressing **Sync HW Clock from System Clock** writes the RTC from the system clock.

Sync All via Internet obtains the current time from a NTP source (internet) and updates the RTC and system clocks.

You can also manually enter the time and press **Set Time** to update both clocks.

To obtain full functionality of this utility, you must run the **rtc_enable.sh** script located in the **/home/pi/sBITX-toolbox/scripts** folder from terminal and reboot your sBitx if you receive an error like this.



Note The **rtc_enable.sh** script only needs to be run once to use the Time Keeper utility.

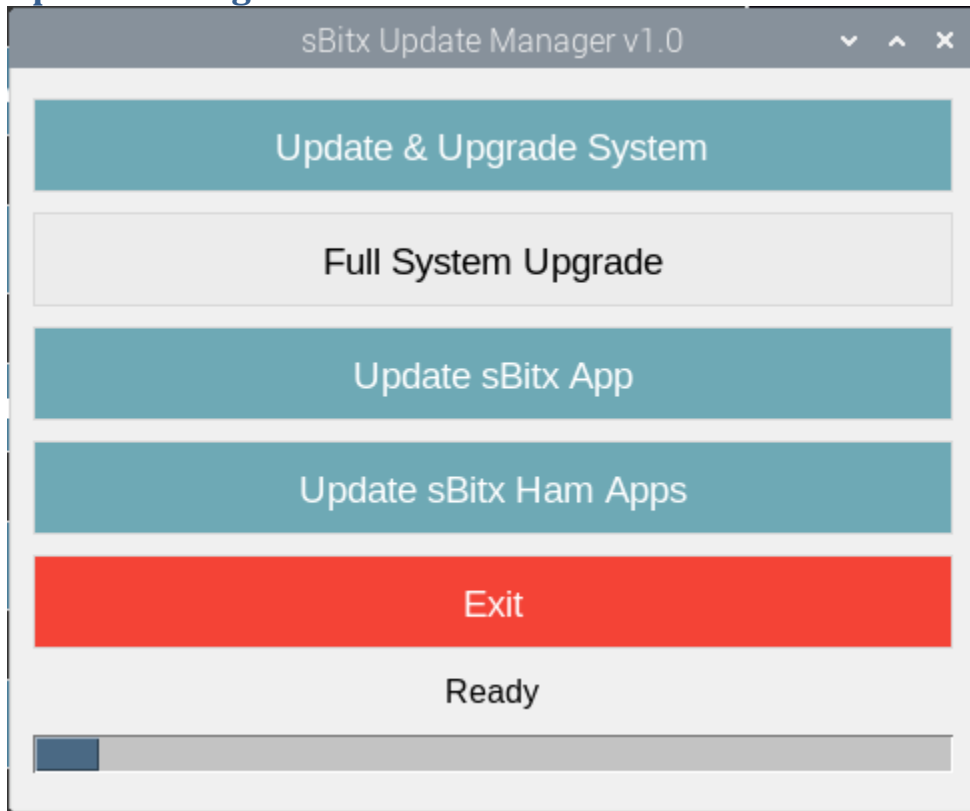
1. Open the terminal application on your sBitx



- 2.
3. Type: **cd \$HOME**
4. Type: **cd /home/pi/sBITX-toolbox/scripts** and press enter
5. Type: **./rtc_enable.sh** and press enter
6. Reboot your sBitx and run the Time Keeper Utility from sBitx Toolbox.

The easiest way to sync the RTC and system time is to connect your sBitx to the internet via Ethernet or WiFi and press **Sync All via Internet**.

Update Manager



A convenient way to update yours system or application with the click of a button.

An internet connection is required for this app to work. It is best to reboot the system after updates have been performed.

- **Update & Upgrade System**
 - What it does:
 - Updates all installed packages to their latest available versions without removing or installing new dependencies unless absolutely necessary.
 - Behavior:
 - It only upgrades packages that can be updated without affecting the system's dependency structure.
 - If a package upgrade requires installing new dependencies or removing existing ones, apt upgrade will skip updating that package and leave it at its current version.
 - Safety:
 - More conservative and less likely to break your system because it avoids major changes.
 - Use case:
 - Good for routine updates where you want to keep the system stable and avoid surprises.
 - Equivalent of “sudo apt update && sudo apt upgrade -y” from terminal
- **Full System Upgrade**
 - What it does:

- Updates all installed packages to their latest versions and resolves dependency issues by installing new packages or removing conflicting ones as needed.
- Behavior:
 - More aggressive than apt upgrade. It will attempt to complete the upgrade process even if it means adding new dependencies or removing obsolete packages.
 - This can result in a more significant system change, potentially removing packages you might still want.
- Safety:
 - Riskier because it might uninstall packages or make broader changes to your system.
- Use case:
 - Useful when you want to fully update the system, especially after a major release upgrade or when dependency conflicts need to be resolved.
 - Equivalent of running “sudo apt update && sudo apt full-upgrade -y” from terminal
- **Update sBitx App**
 - Performs an update to your 64 bit sBitx app
 - Equivalent of running “cd sbitx && ./update” from terminal
- **Update sBitx Ham Apps**
 - Performs an update to the 64 bit sBitx Ham Apps repository
 - Equivalent of running “cd sbitx-ham-apps && ./update” from terminal

V3 PowerCal

V3 Powercal Editor		
Band	Range	Scale
80M	3500000 - 4000000	0.0042
60M	5250000 - 5450000	0.0033
40M	7000000 - 7300000	0.0035
30M	10000000 - 10200000	0.006
20M	14000000 - 14350000	0.006
17M	18000000 - 18200000	0.004
15M	21000000 - 21450000	0.004
12M	24800000 - 25000000	0.006
10M	28000000 - 32000000	0.006
<div> <div>Save</div> <div>Start sBitx</div> <div>Stop sBitx</div> <div>W9JES v1.2</div> </div>		

See <https://github.com/drexjj/sbitx/wiki/sBitx-Hardware-Calibration> for information on this utility.

Advanced Troubleshooting

Reinstallation & Removal

If you encounter errors performing the update or sBITX Toolbox becomes unstable, then try these steps.

Option A (Force an update):



1. Open the **terminal** application on your sBitx
2. Type: **mousepad /home/pi/sBITX-toolbox/VERSION.txt** and press enter.
3. Change the version to 1.0 then press **File→Save** and close the program.
4. Close the Terminal.
5. Open **sBITX Launcher** and try the update again.

Option B (Force a reinstall of the local application):



1. Open the **terminal** application on your sBitx
2. Type: **/home/pi/sBITX-toolbox/installer.sh** and wait for the script to complete
3. Close **Terminal**.
4. Open **sBITX Launcher** and try the apps or run an update.

Option C (Reinstall the sBITX Toolbox application):

Word of Caution: This option will remove your launcher configuration and customizations in the Toolbox data folder! Perform a backup of the **/home/pi/sBitx-toolbox/data** folder (optional) before using this option.



1. Open the **terminal** application on your sBitx
2. Type: **/home/pi/sBITX-toolbox/uninstaller.sh** and wait for the script to complete
3. Type: **cd /home/pi**
4. Type: **cd \$HOME && git clone https://github.com/drexjj/sBITX-toolbox64.git && cd sBITX-toolbox64 && chmod +x *.sh && ./installer.sh**
5. Wait for the script to complete
6. (Optional) Restore the launcher.json and sbmanager_config.json files
7. Run **sBITX Launcher**

Option D (Remove any version of sBITX Toolbox manually)



1. Open the **terminal** application on your sBitx
2. Type: **cd \$HOME**
3. Type: **sudo rm -rf /home/pi/sBITX-toolbox**
4. Type: **sudo rm /usr/share/applications/sBITX_launcher.desktop**
5. Type: **sudo rm /home/pi/Desktop/sBITX_launcher.desktop**

6. For version 2.0 and above only – continue to step 7 and 8
7. Type: **sudo rm /usr/share/applications/sb_launcher.desktop**
8. Type: **sudo rm /home/pi/Desktop/sb_launcher.desktop**

Option E (Manual update to sBITX Toolbox Application):



1. Open the **terminal** application on your sBitx
2. Type: **/home/pi/sBITX-toolbox/update.sh** and wait for the script to complete

USB Audio

Prerequisites:

Ensure that your USB audio device is compatible with Linux. It must be a plug & play or driverless device such as those used with Chromebooks or other Linux operating systems.

USB headsets compatible with the Raspberry Pi 4 (or Raspberry Pi devices in general) typically rely on standard audio class drivers provided by most Linux distributions, including Raspberry Pi OS. This means a wide range of USB headsets should work out of the box, but specific models aren't always listed as "compatible" due to the vast number of products available and the general nature of USB audio support in Linux.

Considerations for Compatibility:

Standard USB Audio Class: Look for headsets that comply with the standard USB Audio Class 1 or 2. These should work without needing special drivers.

Power Requirements: Ensure the headset doesn't require more power than the USB port can provide. The Raspberry Pi 4 can provide a decent amount of power over USB, but very power-hungry devices might still have issues.

Vendor Support: Some manufacturers are better than others when it comes to supporting Linux or providing information about compatibility.

Here's a list of considerations and some brands/models known for good compatibility based on user reports and general Linux compatibility. However, always check the latest user reviews or manufacturer FAQs for Raspberry Pi compatibility mentions, as firmware updates can change compatibility status.

Audio-Technica:

- Some of their USB headsets and even some USB microphones that include headphone jacks are known to be compatible.

Logitech USB Headsets:

- Logitech H390
- Logitech H540
- Logitech H600 (Note: The H600 is wireless but uses a USB receiver, and generally works well with Linux)

Plantronics (Poly) USB Headsets:

- Plantronics Blackwire 3210
- Plantronics Blackwire 5220

Jabra USB Headsets:

- Jabra Evolve 20
- Jabra Evolve 30 ii
- Jabra Evolve 40

Sennheiser (Now EPOS | Sennheiser) USB Headsets:

- SC 60 USB ML
- SC 165 USB

Microsoft USB Headsets:

- LifeChat LX-3000

Sabrent

- USB External Stereo 3D Sound Adapter

EKSA E1000 USB Headset

A full list of compatible USB devices is available at: https://www.alsa-project.org/wiki/Matrix:Main#Sound_Card_List

General Advice:

1. Research: Before purchasing, look for recent reviews or questions regarding the specific model and Raspberry Pi or Linux compatibility.

2. Plug and Play: Most compatible headsets should work as soon as you plug them in.

Keep in mind that while a headset may be compatible, specific features (like inline controls) might not work as expected on all systems, depending on how the device implements those features.

Troubleshooting USB audio:

Your sBitx may or may not have SOX installed. It is required to use the headset and/or headphones.

Please run this from terminal to ensure that you have SOX installed

pi@sbitx:~ \$ sudo apt-get install sox

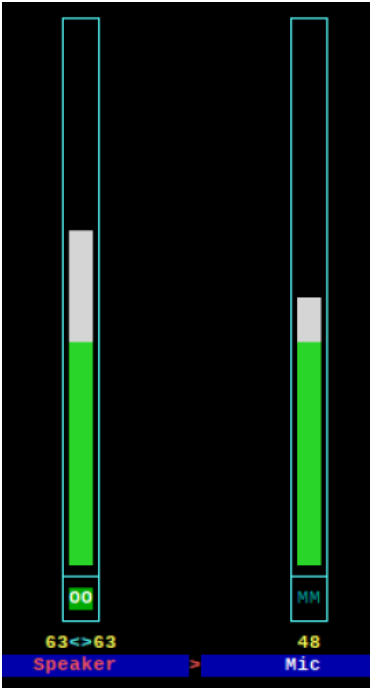
Your USB audio device may not be connected or compatible with sBitx. Run these commands from terminal to see if there is a device listed for card 5. If not, then reboot and try again following the previous instructions.

You can also check to see if the USB audio device is compatible in the USB compatibility section.

Mixer Verification

alsamixer -c 5 then press **F5**

You should see a PCM or Speaker control and a Mic control if you have a microphone as shown. If these are not shown, then Linux drivers may be needed for the device.



USB Microphone:

arecord -l

card 5: Device [USB Audio Device], device 0: USB Audio [USB Audio]

USB Headphones:

aplay -l

card 5: Device [USB Audio Device], device 0: USB Audio [USB Audio]

Mic or PTT isn't working?

Ensure that you do not have an external application open such as Fldigi, WSJTX, etc. These applications must be closed before using the USB microphone because it used the same audio loopback and control methods.