

SWLIST & HAMLIB INTEGRATION



Preface

Greetings and thank you for selecting SWList, your premier iOS app for short wave listening. This guide is here to walk you through some features of SWList, with a special focus on its standout capability: the ability to control your radio receiver via local network.

SWList isn't just an app; it's a tool designed to enhance your radio listening experience. At its core, SWList employs the hamlib library, widely embraced in the ham world for its versatility in interfacing with various radio receivers. This integration enables SWList to connect seamlessly with an extensive range of receivers, providing users with unprecedented control.

SWList introduces two distinct modes to empower users with flexible control options: Auto Mode and On Request Mode.

Auto Mode: In this mode, SWList actively monitors the receiver's frequency every 2 seconds. If a change is detected, SWList dynamically updates and displays the corresponding radio stations, allowing users to effortlessly explore the latest frequencies.

On Request Mode: For those who prefer a more hands-on approach, On Request Mode enables users to manually pick up the current frequency on the receiver by simply pressing a button. This grants users precise control over their radio tuning experience.

SWList's intuitive interface ensures a seamless experience. Tapping on a frequency present in the result list in the SWList display instantly tunes your receiver to the corresponding frequency, offering a streamlined and responsive interaction.

This guide will drive you through the step-by-step processes of setting up hamlib on your pc and connecting with your receiver, ensuring that you make the most of SWList's unparalleled radio control features.

Thank you for choosing SWList – where hamlib integration meets user-friendly control, providing you with the ultimate radio receiver management experience. Get ready to redefine your radio listening journey!



Hamlib

Hamlib is an open-source software library that provides a standardized interface for controlling amateur radio equipment. It allows developers to write applications that can communicate with and control a wide range of radio transceivers and receivers, supporting various models and manufacturers. Hamlib facilitates the development of radio control software, enabling seamless integration with different radio hardware through a common API (Application Programming Interface).

Hamlib is developed by a community of volunteers and contributors in the amateur radio community. It is an open-source project, which means that the source code is freely available, and users are typically free to modify, distribute, and use the software within the terms of its open-source license. The collaborative nature of open-source projects allows individuals from around the world to contribute to the development and improvement of hamlib.

Here some resources to learn more about hamlib

<https://sourceforge.net/projects/hamlib/>

<https://hamlib.github.io/>

<https://github.com/Hamlib/Hamlib/wiki>

<https://github.com/Hamlib/Hamlib/wiki/Documentation>

Install

Follow the link : <https://hamlib.github.io/>

Then open the Current stable release (at the moment is the 4.5.5) where you can download the installer files for your operative system.

Windows :

[hamlib-w32-4.5.5.exe](#)

[hamlib-w64-4.5.5.exe](#)

Linux :

[hamlib-4.5.5.tar.gz](#)

MacOS :

The easiest way is to use/install home-brew and use the command : brew install hamlib
<https://formulae.brew.sh/formula/hamlib>

Test the set up locally

Here we describe a very common case and we suppose your system is a Windows pc, the receiver is connected to the pc via serial cable / usb at COM3 and your receiver is turned on and tuned on the frequency of 8.939 khz.

Step 1

You install the ham library following the instructions of the hamlib web site using the installer :

[hamlib-w64-4.5.5.exe](#)

Step 2

Connect physically your radio receiver to the computer (via serial port / usb / lan) and individuate the name of the serial port (COM3 ... COMx)

Step 3

Now you have to check if your receiver is included in the list of supported radios.
Follow this link to consult the list:

<https://github.com/Hamlib/Hamlib/wiki/Supported-Radios>

Once you have found your receiver take note of the number associated to it.
This number will be needed for the next steps.

Step 4

Locate the where are the ham lib executables.

In a Windows 10 pc you'll find them here : C:\Program Files\hamlib-w64-4.5.5\bin

SWLIST 1.4

Open the command prompt :

- Select the Start Menu in the taskbar or press the Windows key
- Type cmd
- Select Command prompt

In the black window type the command to change the working directory:

```
cd C:\Program Files\hamlib-w64-4.5.5\bin
```

and press enter

Then type :

```
rigctl -m XXXX -r COM3
```

Where XXXX is the number you have found in step 3 and COM3 is the serial port of your pc where is connected your receiver

Will be prompted

Rig command :

Type the command **f** and press enter

If everything is working fine you will see appear the frequency currently tuned on you receiver. Something like this :

Rig command : f

Frequency: 8939000

At this point we know that everything is set correctly and it is working, at least locally.

Step 5

Press Ctrl-c to exit from the rigctl command

Test the set up in a local network

After the previous test you are confident that everything works, but you have to test if it will work also in the local network (LAN).

Step 1

In the command prompt window and still in the working directory

C:\Program Files\hamlib-w64-4.5.5\bin

Type the command

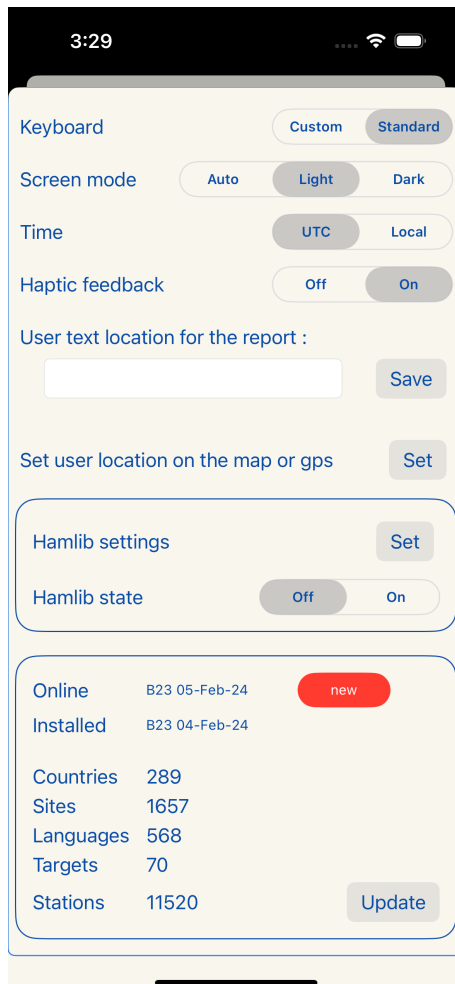
`rigctld -m XXXX -r COM3 -vvvvv`

Where XXXX is the number you have found in step 3 and COM3 is the serial port of your pc where is connected your receiver

The parameter -vvvvv set the rigctld in verbose mode so you can have visible feedback of your commands

Step 2

Start SWList and navigate to the settings panel and locate the hamlib section. Push the button "Set" and it will display the sheet with the hamlib settings



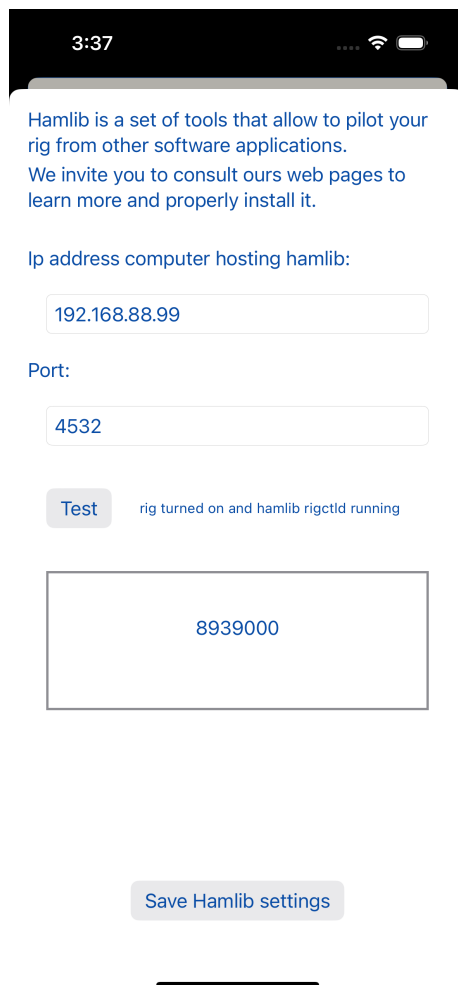
SWLIST 1.4

You must enter the ip address of the pc hosting the radio receiver and the communication port (the default is the 4532).

To verify the settings and the local network just press the button Test.

In the rectangle below the button should appear the frequency currently tuned in the radio receiver.

In alternative you could use the command **telnet** to communicate with the rigctld demon via local network.



The screenshot shows a mobile application interface with a black status bar at the top displaying the time 3:37, signal strength, Wi-Fi, and battery icons. The main content area has a white background and contains the following elements:

- Introductory text: "Hamlib is a set of tools that allow to pilot your rig from other software applications. We invite you to consult ours web pages to learn more and properly install it."
- Label: "Ip address computer hosting hamlib:"
- Input field: A rounded rectangle containing the IP address "192.168.88.99".
- Label: "Port:"
- Input field: A rounded rectangle containing the port number "4532".
- Button: A rounded rectangle labeled "Test" with a tooltip that says "rig turned on and hamlib rigctld running".
- Output field: A large rounded rectangle containing the frequency "8939000".
- Button: A rounded rectangle labeled "Save Hamlib settings".

At this point we know that the everything is set correctly and it is working.

Starting hamlib at the Windows boot

It is possible to launch hamlib automatically at the Windows boot, just follow the following instructions.

Step 1

Create a simple text file with notebook and write the following text :

```
@echo off  
start /min "" C:"Program Files"\hamlib-w64-4.5.5\bin\rigctld -m XXXX -r COM3
```

Where XXXX is the number that identifies your radio receiver.

Save the file with a meaningful name like "start_hamlib.bat" : it's mandatory the suffix .bat

Step 2

Pressing the Windows key and R key to show up the "Run" window and type **shell:startup**.

It will open the Startup folder where you move the precedently saved file "start_hamlib.bat".

Since now every time Windows will boot also rigctld will be running.

Rigctld is a demon that can make a bridge between the local network and the serial port connected to your receiver.

Using hamlib with an SDR software

It is also possible to use hamlib with an SDR software at the condition it emulates an existent radio protocol. As example her- the instructions to connect hamlib with the well-known application SDRUno.

Before to start a little explanation.

SDRUno, being a software application, doesn't directly connect to physical serial ports. To enable communication between SDRUno and other software that uses serial communication (like hamlib), a virtual COM port is used as an intermediary.

A null-modem emulator is a software tool that emulates the behavior of a null-modem cable, which is traditionally used to connect two serial devices (like computers or devices with serial ports) for communication.

To create a virtual COM pairs it is possible to use a null-modem emulator called com0com.

Instead of using physical serial ports and cables, com0com creates virtual COM port pairs. These are pairs of virtual serial ports that are connected to each other as if they were physical ports with a null-modem cable in between.

For example, if you create a pair COM4 <-> COM5, data sent to COM4 will be received by COM5, and vice versa.

In the example of COM4 <-> COM5:

- SDRUno communicates with the virtual COM port COM4.
- COM4 is part of the virtual COM port pair created by com0com.
- COM5, the other end of the pair, is connected to hamlib.
- Therefore, SDRUno <-> COM4 <-> COM5 <-> hamlib forms a complete communication chain.

In summary, the null-modem emulator (com0com) is used to create virtual COM port pairs, allowing SDRUno to connect to a virtual COM port, which is then linked to hamlib through another virtual COM port, establishing a communication path for controlling amateur radio equipment.

Let's go

Step 1

Download com0com at this link :

<https://sourceforge.net/projects/com0com/>

Run the setup and create a virtual COM pairs. COM4 and COM5 (the numbers of the port can be different because it depends on your hardware).

Take note of the name of the COMs created.

Step 2

Open SDRUno and click on the SETT. Button of the SDRUno RX CONTROL



Select the CAT tab and set the COM and click on the click box ENABLE & CONNECT

Step 3

As you have seen in the previous sections dedicated to hamlib you have to let know to hamlib how to connect to your radio that in this case is the SDRUno software.

The rigctld command will be the following :

```
rigctld -m 2051 -r COM5
```

Where 2051 is the number that identifies the SDRUno sdr application.
COM5 is the other end of the virtual COM pairs COM4<->COM5