|  |  |
| --- | --- |
| **Troubleshooting** |  |

**Radio won't connect**

When you try to connect a radio by using the Connect Radio button and the connection fails, take a close look at the error message to get clues. The normal failure is "Connection Refused" with a suggestion to check settings.

First verify that the radio is on and the USB cable associated with that radio is plugged in to RSS. Check the Port option in SETTINGS>Radio to make sure the radio com port is in the drop-down list.

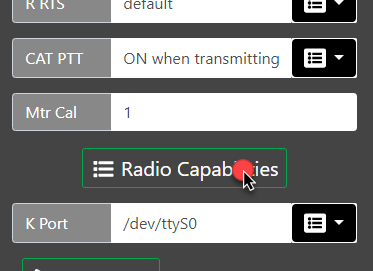
Try rebooting RSS from SETTINGS>System>Reboot RigPi to see if that fixes it.

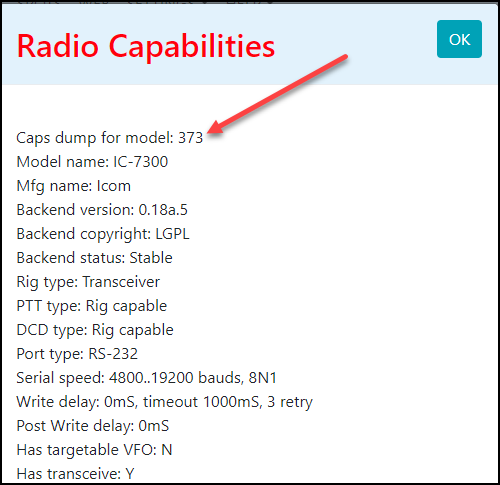
The Connect Radio and Disconnect Radio buttons on the Tuner window require the radio to be set up in SETTINGS>Radio first. If it hasn't been set up, RSS won't be able to connect. (RigPi is initially configured with a "dummy" radio simulation so you can use Connect Radio with no further settings required.)

If you have more than one USB device connected to RSS (several radios, rotor controls or an external WinKeyer), the port assigned to each cable can change when you reboot. What was /dev/ttyUSB0 connected to radio A may now be connected to radio B, and radio A is /dev/ttyUSB1. In SETTINGS>Radio>Port you can select another port to see if the radio will connect.

RSS has a utility program that will prevent port changes from happening in the future. See the Help topic Other Programs>[USB ID](https://rigpi.net/help/usb-id.html) for more information on USB ID.

Next try starting the rig connection manually. This requires the use of the Raspberry Pi Desktop Terminal. The manual connection uses the Hamlib model number for your radio. In SETTINGS>Advanced Radio click the Radio Capabilities button.





At the top you will see 'Caps dump for model: xxx,' where xxx is the model number. Make note of this number.

To clear any radio control remnants, reboot RSS from the System Settings window.

Open Terminal by double-clicking the black prompt icon in the Raspberry Pi Task Bar and type the following (373 happens to be the model number for the Icom IC-7300):

 rigctl -m 373 -r /dev/ttyUSB0

This command assumes you have made no changes to the default settings in the radio.  If you have changed Baud rate to 115,200, use this command:

 rigctl -m 373 -r /dev/ttyUSB0 -s 115200

The resultant report should give more clues, which might simply be "Timed out." More diagnostic information can be obtained by using the "v" parameter such as below:

 rigctl -m 214 -r /dev/ttyUSB0 -vvvvv

If nothing is learned, we need to check that the port is correct. Get a list of available ports by typing (lower-case L):

 ls /dev/ttyU\*

If more than one USB port is listed, try the rigctl commands with the other ports.

Once you are able to get rigctl started, a prompt "Rig command:" is shown. Type f followed by Enter and see if the reported frequency is correct. Type Ctrl+C to exit and stop the rigctl process.

If you are able to connect to and control the radio, we can start the RigPi using a special test mode. Starting the test requires that a file in RigPi be edited with a text editor. This can be done from RSS Terminal using the Nano text editor with these commands:

 cd /var/www/html

 sudo nano RigDo.php

Near the top you will see a line that reads "$tUser=$argv[1];" which we need to modify. Use the cursor arrow keys to move the insertion point to the front of $argv, then enter the characters 1;//.

Slightly further down, after "}else{," you need to change the variable $tUsername='admin'; so that is shows your username.

Type Ctrl+O to save the file and Ctrl+X to exit Nano.

Start hamlibDo by typing this line:

 php hamlibDo.php

See if there are any error reported. If there don't appear to be any errors, open a RigPi browser and see if you can control the radio.

The report shown in Terminal includes the command to start rigctld. Try entering the complete line in Terminal and press Enter. See if there are any error reported.

If none of these steps gets the radio connected, it is time to check the USB cable and radio for problems. This can be done running a control program on another computer by moving the USB cable to that computer.

For a helpful guide providing additional information, here is a link to the Hamlib FAQ:

[Hamlib FAQ](https://github.com/Hamlib/Hamlib/wiki/FAQ)

**Radio Control Logging**

The Test Radio function in Advanced Radio settings also starts a log for recording radio status and commands.  You will find this log in /var/log/rigpi\_radio.log.  It is cleared each time you use the Test Radio function so only the most recent session is kept.   You can use the edit program nano from the Raspbberry Pi terminal to view the log.

 sudo nano /var/log/rigpi-radio.log

**Rotor Control Logging**

The Test Radio function in Advanced Radio settings also starts a log for recording rotor status and commands.  You will find this log in /var/log/rigpi\_rotor.log.  It is cleared each time you use the Test Radio function so only the most recent session is kept.   You can use the edit program nano from the Raspbberry Pi terminal to view the log.

 sudo nano /var/log/rigpi-rotor.log

**User Access Logging**

A log of all users accessing RigPi can be found here:

 sudo nano /var/log/rigpi-access.log

**No RigPi Audio**

RigPi uses the RigPi Audio board for VoIP, digital modes, or even for a spectrum analyzer (panadapter). If RigPi Audio doesn't appear to be working, there are a number of tests we can use to isolate the problem.

RigPi Audio can receive/send audio through the RJ-45 connector on the USB end of the RigPi unit or from the 3.5 mm jack end. We will use the 3.5 mm jack end. If using the RJ-45 connection, remove that cable for now.

RigPi Audio uses a Linux device driver named snd\_rpi\_proto. All drivers and settings required for RigPi Audio are initially set and do not require changes.

Right-click the Speaker icon at the top of RSS Desktop to make sure snd\_rpi\_proto is shown as the selected device. If it isn't in the list, try rebooting RSS from the RigPi SETTINGS>System menu. If it fails to appear, the RigPi Audio board may have a problem.

To look for errors on booting, open RSS Terminal and type the following:

 dmesg | grep audio

You will see a list of possible errors. The last line should say

[  4.xxxxx] snd-rpi-proto soc:sound: wm8731-hifi <-> 3f203000.i2s mapping ok

which says the board is connected. If not, the RigPi Audio board probably needs repair.

Next we will try some audio tests using Linux programs through RSS Terminal.

Plug in a high impedance set of ear buds to the TX OUT connector. Type the following command (c2 says 2 channels):

 speaker-test -c2

You should hear pink noise in the left ear bud, then silence, followed by pink noise.

Here is a tone test:

 speaker-test -t sine -f 2000 -c2

You will hear a 2000 Hz tone in the left ear bud.

Finally, a voice test:

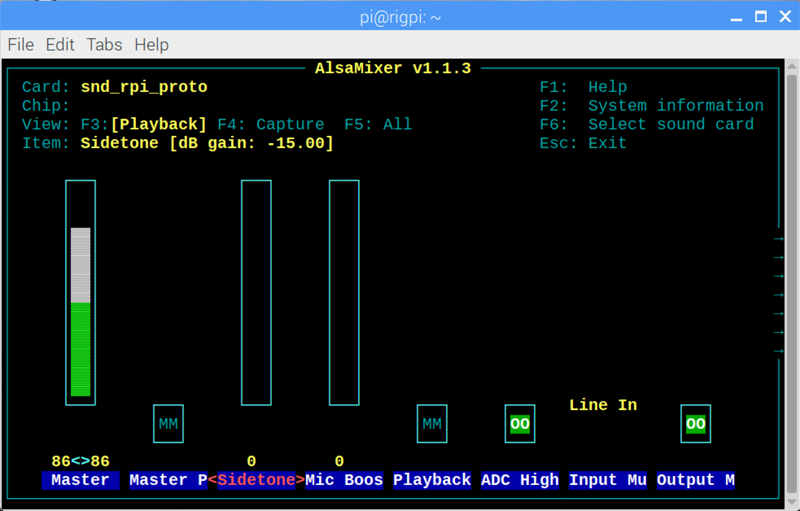
 speaker-test -t wave -c2

You will hear a woman's voice in the left ear bud.

If any of these test fail, we can use the Linux program named Alsamixer from RSS Terminal to check the settings. Type

 alsamixer

The following settings are used for RigPi Audio:



|  |  |
| --- | --- |
| Master Gain | 86<>86 (adjust to suit) |
| Master Playback | OFF (MM) |
| Sidetone | 0 |
| Mic Boost | 0 |
| Playback Deemphasis | OFF (MM) |
| ADC High Pass Filter | ON (OO) |
| Input Mux | Line In |
| Output Mixer HiFi | ON (OO) |



|  |  |
| --- | --- |
| Line | L R Capture |
| Mic Boost | 0 |
| Capture | 18<>18 (adjust to suit) |

**Can't connect to Mumble server**

Mumble server starts automatically when RigPi boots. First we can make sure Mumble server is actually running, then we'll check the Mumble log.

To confirm the Mumble is running:

In Terminal, type:

 ps -ef | grep mumble-server

The command should show:

 mumble-+  582   1 0 12:22 ?    00:00:00 /usr/sbin/murmurd -ini /etc/mumble-server.ini

The results confirm that the murmur (mumble server) daemon is running, using the mumble-server.ini initialization file.

To make sure we can connect to the default Mumble port, type:

 nc -vz rigpi2.local 64738

You should see this response:

 Connection to rigp2i.local 64738 port [tcp/\*] succeeded!

If these tests are successful, we can check the Mumble server log. Type:

 sudo cat /var/log/mumble-server/mumble-server.log

Here is the response:

...

<W>2019-01-23 12:22:25.792 Initializing settings from /etc/mumble-server.ini (basepath /etc)

<W>2019-01-23 12:22:25.815 Meta: TLS cipher preference is "ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:AES256-SHA:AES128-SHA"

<W>2019-01-23 12:22:25.815 OpenSSL: OpenSSL 1.1.0j 20 Nov 2018

<C>2019-01-23 12:22:25.850 Successfully switched to uid 116

<W>2019-01-23 12:22:28.784 ServerDB: Opened SQLite database /var/lib/mumble-server/mumble-server.sqlite

<W>2019-01-23 12:22:28.814 Resource limits were 0 0

<W>2019-01-23 12:22:28.814 Successfully dropped capabilities

<W>2019-01-23 12:22:33.835 OSInfo: Failed to execute lsb\_release

<W>2019-01-23 12:22:37.123 Murmur 1.2.18 (1.2.18-1) running on X11: Linux 4.14.79-v7+: Booting servers

<W>2019-01-23 12:22:45.343 1 => Server listening on [::]:64738

<W>2019-01-23 12:23:32.814 1 => Announcing server via bonjour

<W>2019-01-23 12:23:33.002 1 => Registration needs nonempty 'registername', 'registerpassword' and 'registerurl', must have an empty 'password' and allowed pings.

<W>2019-01-23 12:23:33.003 Object::connect: No such slot MurmurDBus::userTextMessage(const User \*, const TextMessage &)

If connection attempts have been made they are included at the end of the log. Check these line for an indication of why the connection was not made.

Mumble gives access to more settings through an account named SuperUser. Try logging in using that username with password 7388.

Finally, visit the Mumble Help pages to additional help. Here is a link to the instructions for logging into a Mumble server:

[Mumble Help](https://www.mumble.com/support/how-to-connect-to-a-mumble-server.php)

**RigPi Audio Input Level Too Low**

RigPi audio can be high level line-input of up to 2-volts peak-to-peak, or low level microphone input (200-mv peak-to-peak). In addition, the input levels can be adjusted. These settings are done through Alsamixer. See the Alsamixer topic in Other Programs.