

# MFJ-1234 *RigPi* Station Server

Reviewed by Pascal Villeneuve, VA2PV  
va2pv@arrl.net

The MFJ-1234 *RigPi* Station Server is a standalone, mini PC based on the popular Raspberry Pi. It's also a web server, a bidirectional audio server, an audio interface, an electronic keyer, and an amateur station computer. It is advertised as a computer system that controls your station and handles on-the-air activities, and it allows multiple users and radios to interact simultaneously using any internet browser.

You can operate your transceiver remotely over your home network or from anywhere via the internet. You can operate digital modes, such as FT8 and RTTY, using preinstalled software, and then log your contacts and upload them to your Logbook of The World (LoTW) account. You can browse the internet, send and receive emails, or look up call signs online.

MFJ packaged the hardware and software in one box to get the most out of the excellent Raspberry Pi software called *RigPi*. This combination makes it possible for almost any computer-controllable radio to be operated remotely using any web browser on any mobile phone, tablet, laptop, or desktop computer.

With the MFJ-1234, you can control more than 200 radios and 30 rotators supported by the Hamlib library. You can update the software when new radios and rotators are added, and you can customize CAT (computer-aided transceiver) radio control commands for your transceiver.

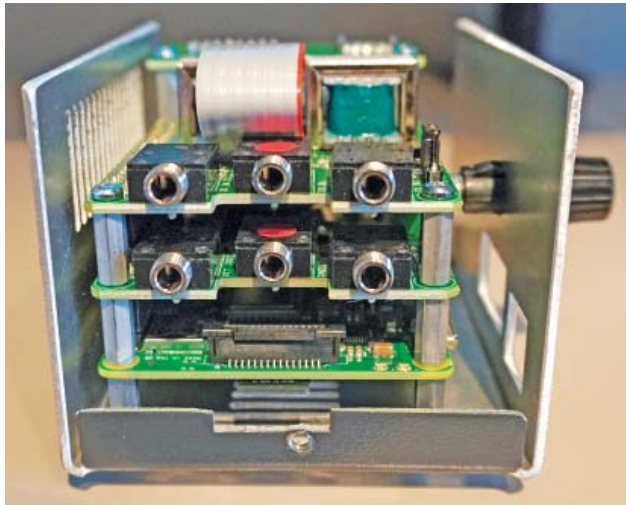


## Bottom Line

The MFJ-1234 *RigPi* Station Server offers a complete remote station solution and also works as a station computer. Although it is easiest to set up with newer transceivers that have CAT and audio features available over a single USB connection, it can be used with older radios with separate connections as well.

## The Software Packages

The *RigPi* Station Server was first demonstrated by Howard Nurse, W6HN (the *RigPi* software author), at the MFJ booth at the 2019 Dayton Hamvention. The *RigPi* software is open source and can be modified. When you buy the MFJ-1234, all the needed software is installed and just a few settings are required to be operational.



**Figure 2** — The inside of the MFJ-1234. The Raspberry Pi board is at the bottom, with *RigPi* keyer and audio boards above.

In the **HAMRADIO** menu, you will find software including *flarq*, *fldigi*, *flrig* ([www.w1hkj.com](http://www.w1hkj.com)); *js8call* ([js8call.com](http://js8call.com)); *TrustedQSL* (for LoTW, [www.arrl.org/lotw](http://www.arrl.org/lotw)), and *WSJT-X* ([www.physics.princeton.edu/pulsar/K1JT/](http://www.physics.princeton.edu/pulsar/K1JT/)). On the desktop, there is a shortcut for *RigPi*, and an application for auto starting the *Mumble* VoIP server when *RigPi* boots.

Another application, *USB ID*, allows *RigPi* to identify your radio's USB port on the MFJ-1234. On a Raspberry Pi, finding the correct USB port is not simple — your port will look like this: `/dev/ttyUSB0`. The USB device can be mapped to a name, and *USB ID* does it for you.

The Raspberry Pi platform reliability depends heavily upon the quality of the software image and the quality of the SD card it's on. With the MFJ-1234, the software image is secure because the SD card is inaccessible without removing the cover, and there's no chance of removing the card by accident.

After using the MFJ-1234 for several months, I have found the software to be very reliable. The hardware

and software integration are very well done. For more information about *RigPi*, visit [rigpi.net](http://rigpi.net).

## The Hardware

The MFJ-1234 hardware is shown in Figure 2. The bottom board is the Raspberry Pi 3 Model B+. In the middle is the *RigPi* keyer, and at the top is the *RigPi* audio board that also includes an I/Q input for SDR radios. If you have a recent transceiver with a USB port that can be used for audio and CAT control, just connect the radio to any of the USB ports on the MFJ-1234 to be fully operational without additional cables.

If your transceiver doesn't have a USB port that can be used for the audio interface, you can use separate cables to connect to an accessory (ACC) port on your transceiver. On the left, Figure 3 shows the 3.5-millimeter jacks for the inputs and outputs for the keyer and audio boards. On the right, Figure 3 shows the other side of the unit with the Raspberry Pi ethernet and USB ports. There's also an RJ-45 jack that you can use for the audio interface.

To power the unit, you will have to provide an external 5 V power supply with a micro USB cable. The MFJ-1234 requires a 2.5 A supply, more than many typical USB power cubes. MFJ offers an optional MFJ-1305RP supply intended to power this device.

## Setting Up the Device

The day I received the unit, I was very excited and wanted to test it right away. An hour later, I was controlling my Icom IC-7300 with my iPhone. I only needed to run a USB cable between the radio and MFJ-1234.

I connected one of my station computer monitors directly to the MFJ-1234 using an HDMI cable, along with a wireless mouse and keyboard. There is no power switch on the Raspberry Pi. It will start automatically when you connect power, and it will boot directly in a Windows-type interface (*Raspbian*).

Using the mouse and keyboard, I configured my Wi-Fi network (it's best to use a wired ethernet cable). After you are connected to your home network, you can do all the settings remotely or continue directly on the device.



**Figure 3** — The MFJ-1234 inputs and outputs.



**Figure 4** — The MFJ-1234 *RigPi* interface on the Raspberry Pi with customized macros for my transceiver and operating preferences.

With the MFJ-1234 connected to my home network, I was able to connect to the *RigPi* server by simply typing the *RigPi* IP address in my cell phone's internet browser. I entered the default user and password (from the MFJ-1234 manual) and I was in. Then I selected my radio in the **SETTINGS** tab and was able to power it on.

I noticed there was no audio and found out that another server was running on the Raspberry Pi, *Mumble* (for VoIP applications). I went to Apple's App Store on my iPhone and downloaded the *Mumble* app. After it was installed, I chose a unique username and used the default password and connected to it. There was still no audio, so I went into the *Mumble* settings on the *RigPi* computer, selected the USB audio connections (the *RigPi* automatically detected the sound card interface built into the IC-7300), and it worked. For Android devices, I used the *Plumble* VoIP application available in the Google Play Store.

To access the *RigPi* server from the internet (outside your home network), you will need to do some port forwarding in your home router. Then you need to use your public IP address to connect from an outside network.

Keep in mind that even if the unit is preconfigured, you will need some computer and networking skills in order to make it work. It's not exactly plug-and-play, but it's

pretty close if you know what you're doing. A good online support forum is available, as discussed later.

### On-Air Operation

The *RigPi* interface is pretty simple to use. Load the web interface on your browser, log in, and turn on the radio using a macro (the **PWR ON** button). Click on **CONNECT RADIO**, open the *Mumble* app on your remote device, and connect to the audio server.

Now that you are connected, you can use the on-screen VFO to change frequencies and the on-screen PTT to transmit while talking on your phone. You need to test everything while you're in your station to ensure that everything works fine and that you will not damage anything. At one point, I was operating remotely from another room in my house, and when I got back into the station, I noticed that my transceiver was stuck in transmit. Fortunately, I had not been transmitting for very long. It was not because of a malfunction, but because I had touched the **T/R** button on my iPhone screen. As soon as you remove your finger from the PTT button, the radio stops transmitting, but the **T/R** button locks the PTT until you touch it again. To be safe, I deleted the **T/R** macro from my interface screen, as shown in Figure 4.

Another thing to consider is your country's regulations about remote operations. In Canada, you need an Advanced license to operate a remote station.

I wanted to set up the system to be able to monitor the IC-7300's SWR meter while transmitting. By default, I couldn't see my SWR when I was transmitting. This worried me because my antenna switch is manual, so I need to set everything correctly before I leave the station if I want to operate remotely. Fortunately, using the online *RigPi* support forum, I was able to learn how to set up SWR monitoring for my transceiver using the **ADVANCED RADIO** settings. I had to go through a calibration process, but it works.

The software package includes everything you need to operate using FT8 and other digital modes. After selecting the correct audio interface for your radio in the *WSJT-X* software, everything works going through the *RigPi* radio control interface, and the application looks and works just like it does on a locally connected desktop computer. If you encounter audio connection problems, confirm that the correct audio device is selected by right-clicking the speaker icon on the Raspberry Pi desktop along with checking the audio device settings in *WSJT-X*.

I was able to operate remotely using both Android and iOS mobile devices (see Figure 5). You can also use any desktop or laptop computer with a web browser. Another cool thing about the software image is that it includes a VNC (virtual network computing) server. This allows you to connect remotely to the Raspberry Pi interface as if you were connected directly to the device.

## Support

There's a well-organized community for the *RigPi* users. You can find help and a lot of useful information on the *RigPi* forum ([groups.io/g/RigPi](https://groups.io/g/RigPi)). *RigPi* developer Howard Nurse, W6HN, is active on the forum, and he's answered many questions.

Here's an example of support I got from the group. When you select your transceiver in the **SETTINGS** tab of the *RigPi* interface, most macros will work, but many radio functions are not included in the default



Figure 5 — The *RigPi* web interface running on my iPhone XS MAX.

setup. So, I went on the support forum and got a list of macros for my radio, and I customized my interface (see Figure 4).

## Conclusion

I've had a lot of fun with the MFJ-1234. I can now monitor my favorite HF frequencies from any room in my home, and from anywhere away from home, as long as I have internet access. I can also transmit if I want to, instead of just listening.

I had never tried FT8 before I got the MFJ-1234. It works so easily that I'm now addicted, and I made quite a few DX contacts in just a few days using this mini station computer. I imported my Logbook of The World certificate and uploaded my *WSJT-X* log (.adi) to LoTW using the *TQSL* application on the MFJ-1234 as well.

The MFJ-1234 is a very good addition to my station. It's a compact, multifunctional PC that combines practicality with many fun features. Because the software is open

source, I expect even more functionality will be available in the future when it becomes available on [www.github.com](https://www.github.com).

**Manufacturer:** MFJ Enterprises, 300 Industrial Park Rd., Starkville, MS 39759; [www.mfjenterprises.com](https://www.mfjenterprises.com). Price: \$319.95; MFJ-1305RP power supply, \$24.95. *RigPi*™ is a trademark of Howard Nurse, W6HN. More information is available at [rigpi.net](https://rigpi.net).



Visit <https://youtu.be/Hoc4213UOmE> to see our review of the MFJ-1234 RigPi Station Server on YouTube.