RSP TCP Server User Guide

V1.1 September 2021

Overview

This brief document describes the installation and operation of the RSP TCP Server. This software is a fork of F4FHH's version of the RTL TCP Server. This software works with all RSPs (RSP1, RSP1A, RSP2, RSP2pro, RSPduo in single tuner mode).

When the server is run in standard mode, it will be compatible with the RTL TCP protocol. When the server is run in extended mode, all of the features of the RSPs become available and the full ADC bit rate up to 14bits is supported. The extended mode requires the use of a compatible client (such as the ExtIO plugin you can download from our Downloads page, along with the RSP TCP server software)

Downloads Page: https://www.sdrplay.com/downloads

Changes: 1.1 remove reference to API 2

1. Purpose

The purpose of providing this software is to show how the RSP IQ stream can be compatible with existing RTL based software and streaming methods. This software also has an extended mode to fully support all of the RSPs capabilities and 16bit IQ stream values.

2. Recommendations

We have provided the software in both binary (for Windows, Mac and RPi platforms) and in source form. Our recommendation is to use the binaries for the platforms provided, but then build from source if you need to.

3. Released Contents

Some brief descriptions of what is contained in this release.

3.1 Binaries for the Server

For Windows, Mac and Raspberry Pi you can get the latest server binaries from our downloads page (https://www.sdrplay.com/downloads)

PLEASE NOTE: The API 3.07 or above is required to be installed BEFORE these binaries are used.

3.2 Source code for the Server

Our GitHub repository contains the source code to build on any platform supported by the API: https://github.com/SDRplay/RSPTCPServer

PLEASE NOTE: The API 3.07 or above is required to be installed BEFORE the source code can be built.

3.3 ExtIO Plugin

We have also released an ExtIO plugin that can be used in conjunction with the extended mode of the server to fully exploit all of the functionality of the RSPs

The Windows binary can be found on our Downloads page (https://www.sdrplay.com/downloads)
The source code can be found on our GitHub repository (https://github.com/SDRplay/ExtIO_RSP_TCP)

The plugin will need to be put inside the application installation directory before starting the application. Please note: If using SDRuno (ExtIO version) then the plugin needs to be put into your My Documents folder.

4. Acknowledgements

SDRplay acknowledges the efforts made by the authors of the relevant previous builds and provides these binaries and source code as free software for the RSP community.

All software remains the property of the respective authors.

5. Starting with the Server

If you are using binaries, skip to 5.2

5.1 Building from Source Code

Once the API is downloaded and installed, follow these steps to build and install the server software. If you are building on Windows, use the Visual Studio solution in the GitHub repository, otherwise…

```
git clone https://github.com/SDRplay/RSPTCPServer.git
cd RSPTCPServer
mkdir build
cd build
cmake ..
make
sudo make install
```

5.2 Run the Server Executable

```
rsp_tcp <command line options>
```

```
Usage: [-a listen address (default: 127.0.0.1)]
       [-p listen port (default: 1234)]
       [-d RSP device to use (default: 1, first found)]
       [-P Antenna Port select* (0/1/2, default: 0, Port A)]
       [-T Bias-T enable* (default: disabled)]
       [-R Refclk output enable* (default: disabled)]
       [-f frequency to tune to [Hz]]
       [-s samplerate in Hz (default: 2048000 Hz)]
       [-n max number of linked list buffers to keep (default: 500)]
       [-v Verbose output (debug) enable (default: disabled)]
       [-E RSP extended mode enable (default: rtl_tcp compatible mode)]
       [-A AM notch enable (default: disabled)]
       [-B Broadcast notch enable (default: disabled)]
       [-D DAB notch enable (default: disabled)]
       [-F RF notch enable (default: disabled)]
       [-b Sample bit-depth (8/16 default: 8)]
```

Please note that the -E option enables the new extended mode. In this mode, all of the functionality will be available from the client, however this will require a compatible client such as the ExtIO plugin we have provided with this release. Some RSP functionality can be achieved in "RTL compatible" mode by using the command line options to enable RSP functions such as ports, hardware notches and Bias-T

5.3 Examples

1. Running the RSP TCP Server in extended mode on a remote PC whose IP address is 192.168.0.43, this would then be connected to by a compatible client using 192.168.0.43:1234 as the server address.

```
rsp_tcp -E -a 192.168.0.43
```

2. Running the RSP TCP Server in RTL mode AND running the client software on the SAME machine with the DAB notch and Bias-T enabled. The client software would use 127.0.0.1:1234 as the server address.

```
rsp_tcp -D -T
```

For more information contact https://www.sdrplay.com/contact

6. Legal Information

SDRplay modules use a Mirics chipset and software. The information supplied hereunder is provided to you by SDRplay under license from Mirics. Mirics hereby grants you a perpetual, worldwide, royalty free license to use the information herein for the purpose of designing software that utilizes SDRplay modules, under the following conditions:

There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Mirics reserves the right to make changes without further notice to any of its products. Mirics makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Mirics assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Typical parameters that may be provided in Mirics data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters must be validated for each customer application by the buyer's technical experts. SDRplay and Mirics products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Mirics product could create a situation where personal injury or death may occur. Should Buyer purchase or use SDRplay or Mirics products for any such unintended or unauthorized application, Buyer shall indemnify and hold both SDRplay and Mirics and their officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that either SDRplay or Mirics were negligent regarding the design or manufacture of the part. Mirics FlexiRF™, Mirics FlexiTV™ and Mirics™ are trademarks of Mirics .

SDRplay is the trading name of SDRplay Limited a company registered in England # 09035244.
Mirics is the trading name of Mirics Limited a company registered in England # 05046393