

Mike Richards looks at the SDRuno software for use with the RSP from SDRplay, which can also be used with low-cost DVB-T USB SDR dongle receivers. In the process, he guides you through installation, functionality, the memory system and using virtual receivers.

**T**he Radio Spectrum Processor hardware from SDRplay forms the basis of a very powerful, budget priced SDR receiver that provides significantly better performance than the cheap RTL DVB-T based solutions. As with all SDR hardware solutions, it is the associated software that provides the interface that we use to access the receiver. For the past few years, as consumer SDR has developed, we've enjoyed a period where hardware and software developers have worked together to give the user a wide choice of hardware/software combinations using essentially free software. Of the free SDR software systems, two have dominated and they are SDR Console and SDR#. However, this changed when the SDR# team developed and released their own hardware in the form of the Airspy. Prior to that, SDR# provided support for many hardware platforms (including the RSP from SDRplay) through the ExtIO interface. However, that support was withdrawn following the release of the Airspy. From a hardware developer's viewpoint, that exposed a vulnerability that could leave them high and dry if the third party software support changed.

The SDRplay team were keen to protect their customers and ensure there was a continuing market for their hardware. In April 2016, the solution came to light when they purchased the rights to the SDRplay compatible version of the Studio1 SDR software. The

# SDRuno Software

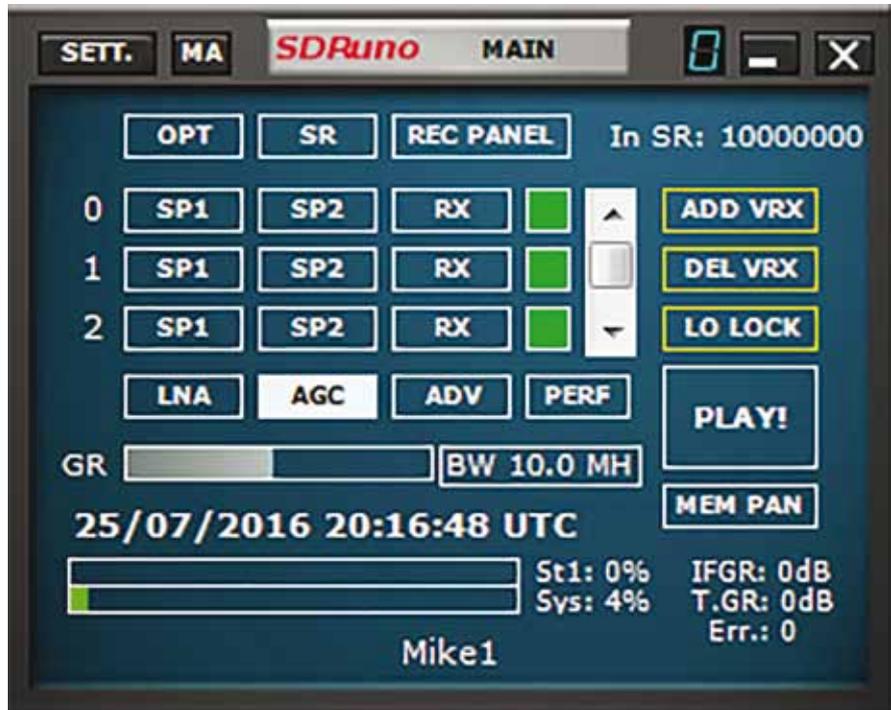


Fig. 1: SDRuno's small MAIN control panel.

SDRuno package reviewed here is the result of that partnership and provides RSP users with a sophisticated and well developed SDR software package that has been optimised for the RSP. Whereas Studio1 is paid for software, SDRuno is available free of charge for all RSP users, both existing and new. The SDRplay team has also opened up SDRuno so that it will work with other SDR hardware that uses the ExtIO interface. There are a few functional restrictions and the sample rate with other hardware is restricted to 1MSPS. Nevertheless, it is free to use and ideal for DVB-T users to try the software before upgrading to the RSP.

## Getting Started

The SDRuno software is available for free download from the SDRplay website.

[www.sdrplay.com](http://www.sdrplay.com)

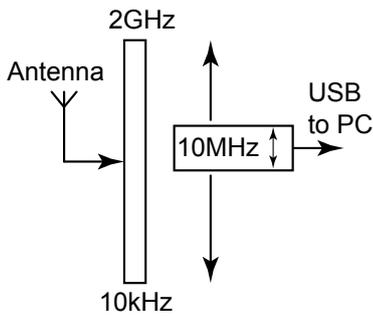
When you run the downloaded file, you are taken through a menu driven install process that includes installation of the hardware drivers required to support the RSP. Following installation of the software, I connected my RSP to a USB port, the hardware

was recognised by Windows and the appropriate drivers loaded without issue.

On launch, new users might be surprised by SDRuno's minimal display, with just the small MAIN control panel active (Fig. 1). Despite its small size, this panel is the central control for each SDRuno instance and allows you to change a variety of options and add extra virtual receivers. One aspect of SDRuno that is different from other SDR systems is its use of undocked windows in the user interface. This might seem a bit unusual at first but it does give you the flexibility to create your own screen layouts; more on this later.

## Virtual Receivers

The use of virtual receivers is common practice in many modern SDR packages but it is often misunderstood so I'll try to add some clarification here. All SDR receivers employ a broadly two-stage process to get from the wide bandwidth of signals supplied by the antenna to a narrower band of frequencies used for signal processing and demodulation. This first stage can be likened to the RF amplifier and first mixer in a traditional superhet receiver. In receivers such as

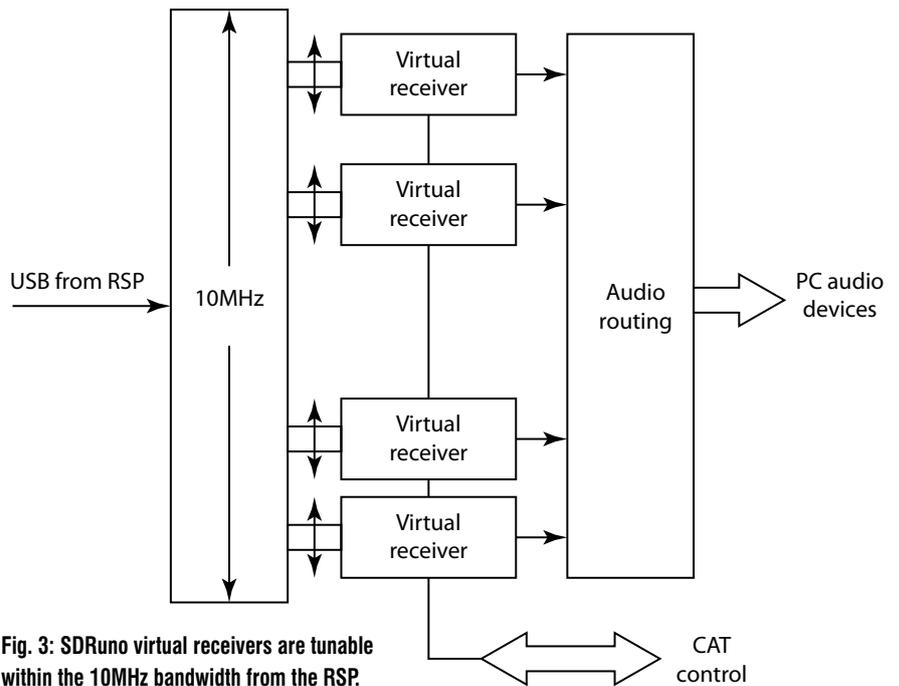


**Fig. 2:** An illustration showing how the RSP selects a slice of up to 10MHz from the available spectrum.

the RSP, this first stage employs analogue technology to downconvert the incoming band of signals to a narrower band or slice of the original bandwidth. For example, the RSP can receive signals from 10kHz through to 2GHz. However, the SDR hardware is used to extract a slice of frequencies that can be up to 10MHz wide. This slice can be placed anywhere in the 10kHz to 2GHz range by electronically tuning the local oscillator in the RSP (Fig. 2).

The next stage in the process is to digitise the slice so that it can be passed to the PC for processing with SDR software. In the RSP and many other receivers in this class, the digitised slice is passed to the PC via the USB port.

Once the digitised slice is in the PC, the SDR software running on the PC will use a process called undersampling or decimation to extract a smaller digital slice for further filtering and demodulation. This smaller slice can then be digitally tuned anywhere across the incoming digital slice (up to 10MHz) from the hardware



**Fig. 3:** SDRuno virtual receivers are tunable within the 10MHz bandwidth from the RSP.

(Fig. 3). This process will also handle the demodulation of the signal along with the final filtering and any special features such as noise reduction, notch filters and so on. You can probably see that the software running on the PC is doing all the tasks that you would typically expect to find in an analogue receiver. However, the entire operation is carried out in software and for that reason, it's called a virtual receiver. Because the virtual receiver exists only in software, there's nothing to stop you having more than one. In fact, you can have as many virtual receivers as your processor and SDR software can handle. Other than processing power, the limitation is that the virtual receivers can only tune across the slice being provided

by the hardware, which is just under 10MHz wide in the case of the RSP. Much the same process is used for direct digital sampling SDR receivers, except the very first stage is handled digitally.

### SDRuno Functions

The new SDRuno has two primary functions when used with the RSP. The first is to control the RSP hardware and in this case, it sets the local oscillator frequency and the sample rate that determines the bandwidth of the digital slice sent to the PC. In addition, SDRuno controls the RSP's LNA and the other digital gain reduction settings of the MIRICS' front-end hardware.

The second function of SDRuno is



**Fig. 4:** An SDRuno virtual receiver panel.

to create and manage multiple virtual receivers. When you start SDRuno, you will see the MAIN control panel that is the start point for using SDRuno (Fig. 1). In order to begin using the receiver, you need to press the PLAY! button. Although not visible in the default settings, the MAIN control panel also starts virtual receiver 00 and this can be seen by clicking the RX button, which will reveal the RX CONTROL window that contains lots of familiar controls (Fig. 4). Also associated with receiver 00 is a spectrum display that's accessed by pressing SP1 (Fig. 5). This displays the full spectrum that's being sent from the RSP hardware. This can be up to 10MHz wide and is dependent on the sample rate (SR) setting in the MAIN control panel.

### Using SDR Virtual Receivers

With the receiver panel enabled, the demodulation is set using the buttons immediately below the tuning display. Below this, again you will see the secondary mode selections for FM. As with all SDRs, the SDRuno software includes some excellent filtering that can be easily customised. To the centre right of the RX CONTROL panel, there is a group of filter buttons that change value according to the selected demodulation mode. These default filter widths were well chosen and provided a very quick way to access the most useful values. For more precise control of the receive filters, SDRuno includes a narrow band auxiliary spectrum display where you can drag the filter width using your mouse. To reveal this display, you need to click the appropriate SP2 button in the MAIN panel (Fig. 6). The auxiliary display is similar to the main spectrum window except it provides access to the downsampled spectrum of the selected demodulator. The width of this spectrum will vary according to the demodulation mode but for narrow band modes such as SSB, it was 12kHz wide. In this spectrum window, you can use your mouse pointer to move both the LF and HF sides of the filter independently. This provides an excellent range of adjustment so that you can maximise the readability of the signal under just about any conditions. To supplement the bandwidth filtering, SDRuno includes four notch filters per receiver that can be used to notch out any interfering carriers. There is also a notch lock facility that causes the notch filters to track interfering carriers. This can be particularly useful when you have a drifting

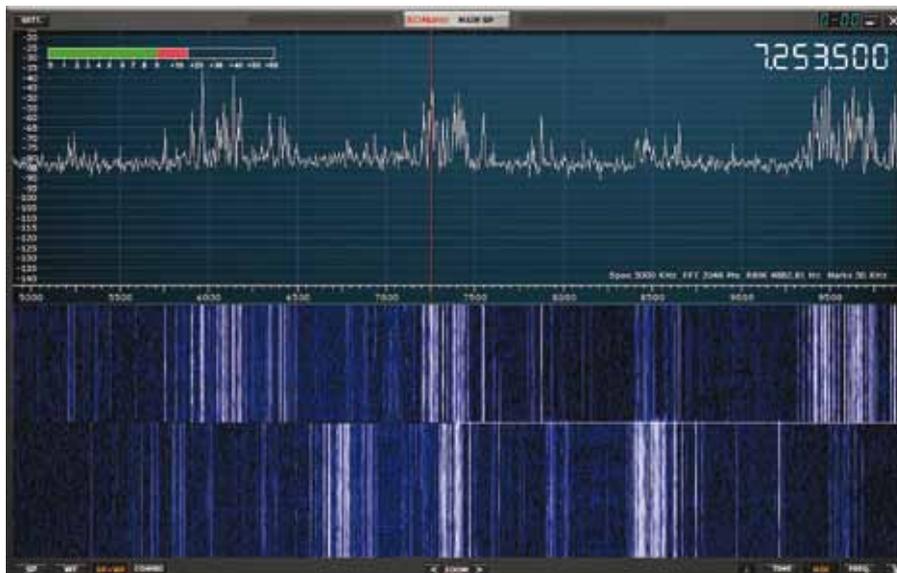


Fig. 5: The SDRuno MAIN SP display showing the incoming IQ spectrum.

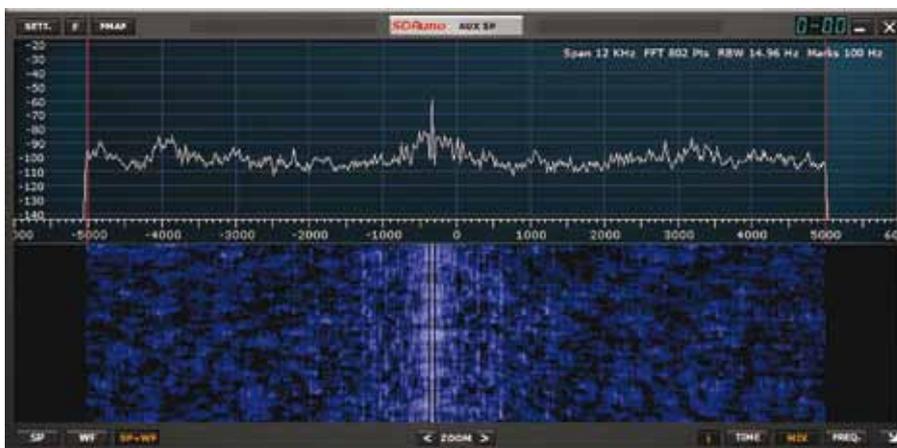


Fig. 6: The SDRuno AUX SP showing the demodulator spectrum display.

carrier causing interference.

Simple control of the receiver's AGC is done via the Off, Fast, Medium and Slow buttons on the receiver panel. However, more detailed gain control facilities are available via the Settings menu on the RX CONTROL and MAIN panels. Just a word of warning here because the RSP gain settings have been optimised for the best compromise of sensitivity and dynamic range. As a result, it is easy to compromise the overall performance if these controls are not adjusted with care.

SDRuno has a wide range of options for frequency control. For those with an interest in the amateur bands, there are band presets on the right-hand side of the RX CONTROL panel. These cover the main HF bands and 2m. This same panel doubles as a direct frequency entry keypad. To use it, I clicked on the frequency display and enter the desired frequency in kHz. I could also enter frequencies using the number keys on the PC keyboard. I found that fine tuning was best handled using the mouse wheel

which increased/decreased the tuning by the default steps for the selected mode/frequency. For different fine tuning rates, this was easily changed by right-clicking on the frequency display and adjusting the tuning steps from 1Hz through to 100kHz. It was also possible to change frequency by hovering the mouse pointer over the main frequency display. When I did this, a highlight bar appeared to show the selected digit and rolling the mouse wheel caused that digit to increase/decrease. I found this to be a good way to make large frequency changes. In addition to the keyboard and mouse driven tuning, SDRuno provides full support for the Tmate hardware control accessories. I didn't have any Tmate kit to hand but it is reported to work well.

For those that might want to link an RSP to other receivers or transmitters, SDRuno has some excellent CAT support via the OmniRig library. SDRuno can be configured either to control external hardware or vice-versa. This can be really helpful if you want to integrate your

RSP with other receiving equipment in the shack.

As you would expect from any modern SDR, SDRuno supports the recording and playback of IQ files and you can even run multiple virtual receivers during playback. This is an often overlooked but very powerful feature of modern SDR software.

One very powerful feature of SDRuno virtual receivers is their audio routing. When you start getting to grips with the use of multiple receivers you need to decide what to do with their audio outputs. In some cases, you will want to hear the audio through the PC's speakers but how do you know which receiver is producing the audio? One simple technique is to use the configurable stereo output of the SDRuno to position them. For example, the audio from one virtual receiver could be directed to the left channel, another to the right and a third to both left and right. That way you can use the stereo soundstage to identify the signals. If like me, you have an interest in data signals, you will also want some outputs sent to a virtual audio cable that's linked to one or more decoders. All this is easily achieved in SDRuno by utilising the OUT tab in the settings panel of each receiver. The software handles all the audio channel mixing and you can send any or all receiver outputs to any of the output devices on your computer!

## Receiver Workspace

While having the flexibility to create your own SDR screen layout is very attractive; you really need a simple way to recreate these layouts. SDRuno handles this very neatly with its workspace memory. At the bottom of the MAIN control panel is the workspace indicator and a single click on the name will bring up a list of 10 user workspace memories. To store your current workspace, you simply click on the workspace name with the Ctrl button depressed. You could also rename the workspace to something more meaningful by right-clicking on the name.

## Memory System

Handling memories in an SDR with multiple virtual receivers and potentially multiple instances could very easily get complicated. However, SDRuno has adopted a relatively simple approach. Each memory set is specific to a particular program instance. That means the memory channels are available to all the virtual receivers in the current instance. If you run a second or more instances, you will have other associated memory channels.

The memory system employs what are called memory bank files. These are spreadsheet files in CSV format, each of which holds one memory bank. These bank files can contain any number of frequencies and the general idea is to

use them to group frequencies from a common listening mode. For example, I put all my common amateur radio frequencies in one bank, while another was all airband and a third was all marine band channels. The use of the CSV format also makes it relatively easy to convert frequency lists from other sources to work with SDRuno. To access the SDRuno memory banks from a given receiver, you simply click the MCTR button for that receiver. If you then want to connect another virtual receiver to memory, you simply click that receiver's MCTR button and it will automatically take over the memory bank connection. I found this a very simple yet practical system for managing memories across several virtual receivers.

## RDS Support

Those who enjoy VHF broadcast DXing will be pleased to hear that SDRuno has excellent support for receiving the RDS. RDS provides a very convenient, electronic, method of identifying broadcast stations so you don't have to wait for a DJ or station announcer. The RDS system sends its information using four blocks of data. In a standard decoder, nothing will be displayed until a full set of blocks have been received in the correct order. When the signal level is very weak, you can have a situation where all of the blocks are being

Fig. 7: The numbers displayed when two instances of SDRuno are running.





Fig. 8: SDRruno with two instances and 8 receivers running on an Intel Core i5 quad-core PC.

received but not necessarily in the right order – there’s a joke there! To speed the identification process, SDRruno has an RDS DX mode where it uses the part received block to obtain a partial synchronisation and attempts to recover the vital Programme Information data.

### Multiple Instances

In addition to the flexibility offered through multiple receivers, SDRruno can also operate with multiple instances. By that, I mean that while SDRruno is running, you can go to the Windows menu and start another instance of SDRruno. Clearly, it’s not practical to have two versions of the software running and attempting to control the same hardware. Nevertheless, it does allow you to connect a second RSP and control it with the second instance of SDRruno. With the current low-price of the RSP hardware, this is a feasible option and would enable the simultaneous monitoring of two frequency bands anywhere in the 10kHz to 2GHz spectrum. One point to watch when using multiple RSP instances is the USB bandwidth. If you’re running both RSPs with the wide 10MHz sample rate, then you will need to connect the two RSPs to different USB hubs to avoid congestion. Helpfully, the software provides a warning

if you try to exceed the hub bandwidth. Another useful feature is the ability to direct the demodulated audio from receivers in different instances to the same audio device.

Because the MAIN and RX CONTROL panels all look the same, SDRruno employs a simple numbering system to make life in a multi-instance environment a bit easier to manage. If you look at Fig. 7, you will see that both MAIN panels have a number displayed just to the right of the SDRruno title, this is the instance number and it counts from zero. Therefore, in this example, I have two instances running and they are shown as 0 and 1. If you now look at the associated virtual receiver panels, you will see that one is numbered 1-00 and the other is 0-00, where the first digit is the instance number and the second digit pair shows the virtual receiver number. This simple numbering system makes it easy to keep track of the control panels and receivers in a multi-instance setup.

Fig. 8 shows SDRruno with two instances and eight receivers running on an Intel Core i5 quad-core PC.

## summary

SDRruno is a well developed and sophisticated SDR application that is ideally suited for the RSP from SDRplay. The use of undocked windows within the interface is unconventional but easy to master. You are then free to use SDRruno’s workspace store to build and save custom layouts to suit different monitoring roles. The easy linking with other hardware and external controllers is another big plus point that will appeal to many users. Given the excellent value of the RSP from SDRplay, the free availability of the SDRruno software makes this a compelling combination. If you have an RSP, then you really ought to give this a try. SDRruno is a free Windows download and is available from the SDRplay website.

[www.sdrplay.com](http://www.sdrplay.com)