

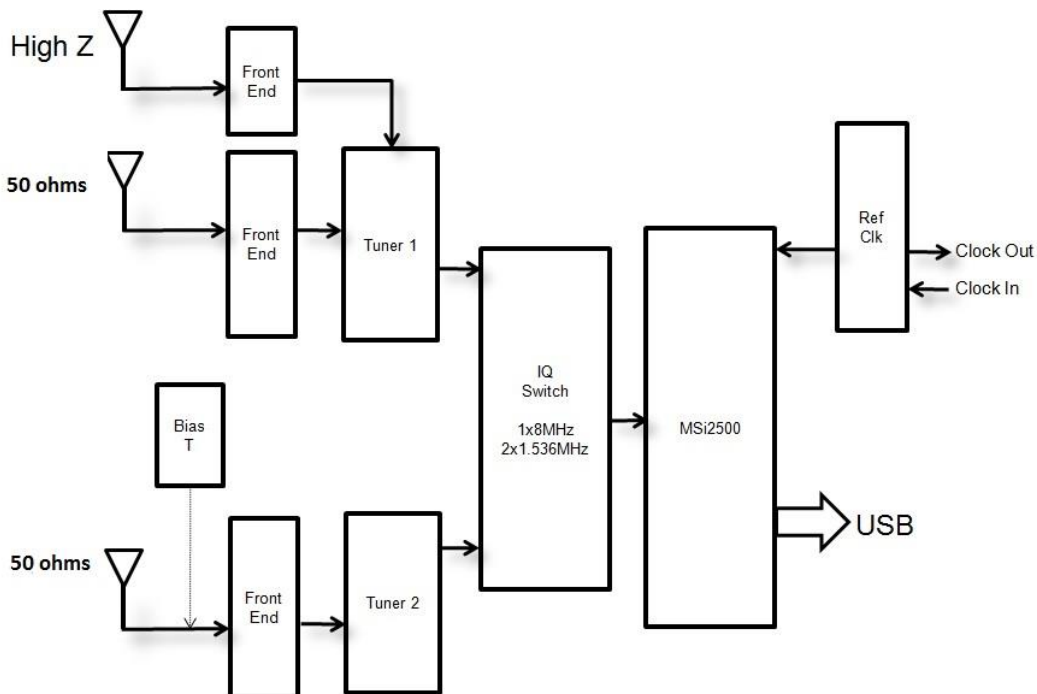
Introduction to the RSPduo

Dual tuner 14-bit 1kHz – 2GHz SDR

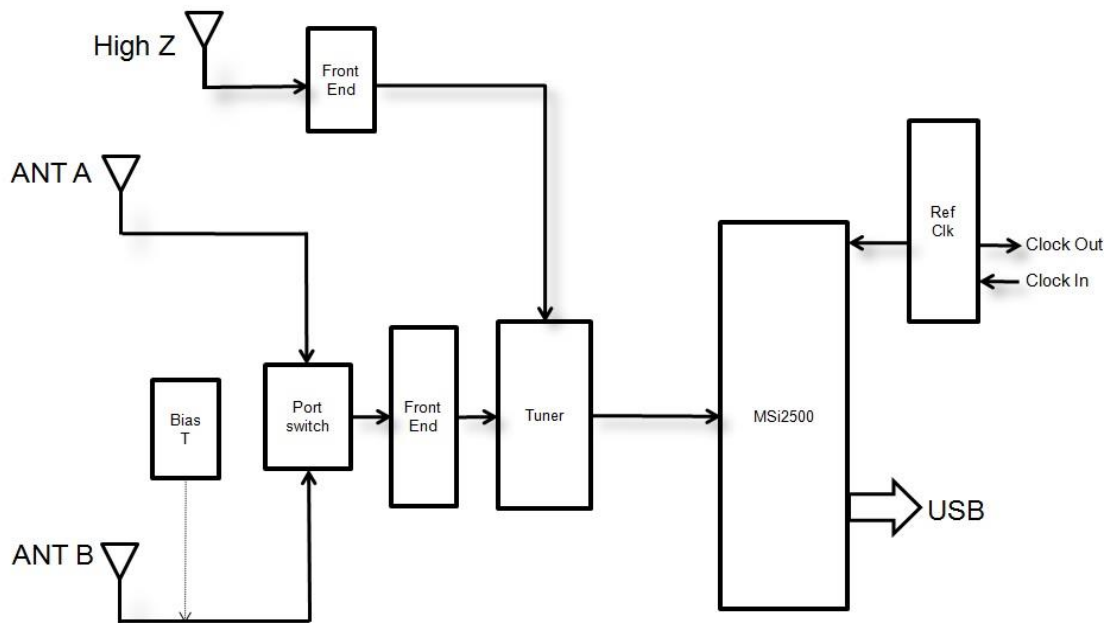
The RSPduo is a radical new product that will shortly be introduced as part of the RSP line of SDR radios from SDRplay. Architecturally, it is different from any previous RSP in that it features dual independent tuners, both piped through a single USB 2.0 interface.

Superficially the RSPduo will look virtually identical to the RSP2pro and will be able to operate in a very similar way, but it also allows a completely new and unique set of usage scenarios.

The Basic block diagram of the RSPduo is:



In contrast the RSP2 block diagram is shown below:

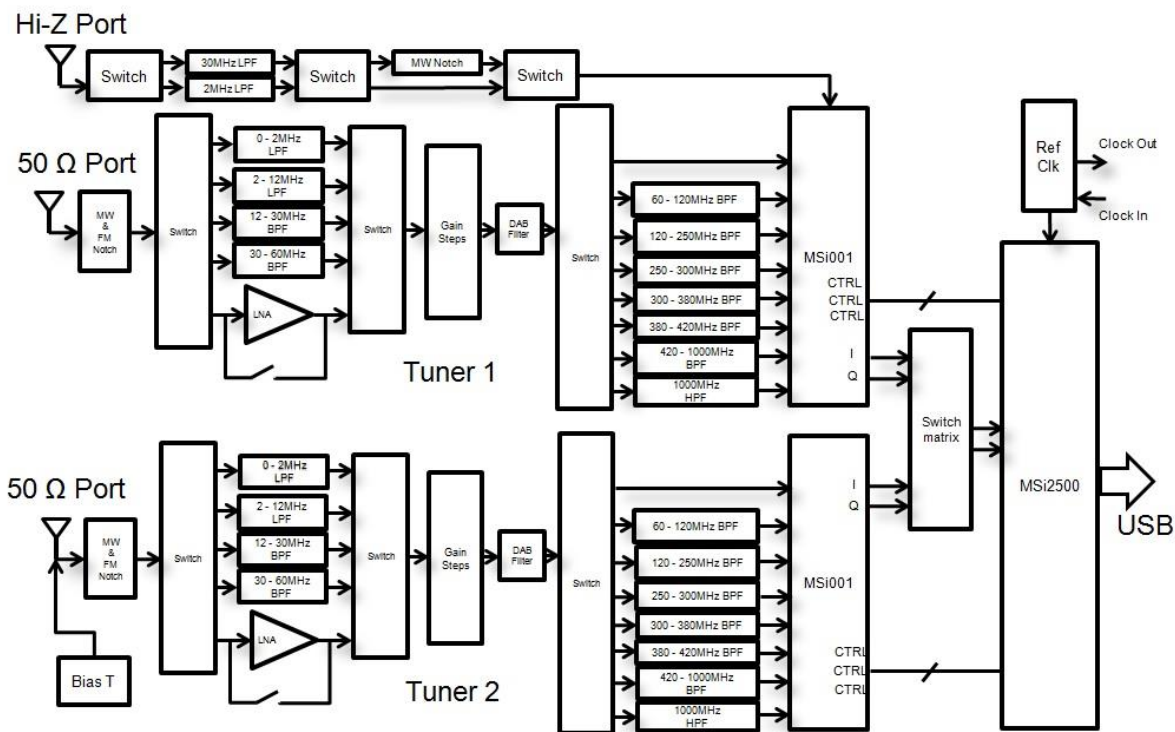


As can be seen, the principle difference is that the RSPduo contains two tuners as opposed to the RSP2 which has a single tuner and switched antenna ports.

The MSi2500 contains dual ADCs. This allows the sampling of I/Q analog baseband signals when the tuner is used in Zero IF mode. However, the tuner can also operate in a Low IF mode, where a single ADC is used to sample the output from the tuner. This opens up the possibility of having two independent tuners operating simultaneously, but this is **ONLY POSSIBLE IF BOTH TUNERS ARE OPERATING IN LOW IF MODE**. In Low IF mode, the maximum flat pass-band bandwidth of the IF filters in the tuner is 1.536 MHz, whereas in ZIF mode, it is possible to widen these filters to 8 MHz.

This is the approach used in the RSPduo. The receiver can either operate each tuner individually (one at a time) in zero IF mode with a displayed bandwidth up to 10 MHz or both tuners simultaneously with a maximum displayed bandwidth of 2 MHz.

A more detailed block diagram of the RSPduo is shown below:



As can be seen from the diagram above, Tuner 1 contains both a 50Ω port and a Hi-Z port. Unlike the RSP2, the 50Ω ports on both tuners now operate down to 1 kHz and deliver significantly improved dynamic range when compared to the RSP2. In addition, like the RSP1A, there is a 0-2 MHz low pass filter on both 50Ω ports, which helps reduce the number of spurious mixing products that appear at LF and MW. Unlike the RSP2, the Hi-Z port on the RSPduo now incorporates both a MW notch filter and a 0-2 MHz low pass filter, which should help reduce overload from strong MW AM signals and spurious mixing products from HF. The Hi-Z port should deliver the best performance for LF, MW operation and ought to be far superior to that on the RSP2. The 50Ω ports on Tuner 1 and 2 deliver very good performance from LF to 2 GHz, but deliver the best performance for HF and above due to the extra preselection filters which are not available on the Hi-Z port above 2 MHz. The 50Ω ports on Tuners 1 and 2 should deliver similar performance to the RSP1A, but with improvements in both the MW and VHF FM notch filter rejection and also a reduction in spurious mixing products from UHF appearing in Band 3 (120 – 250 MHz). The reference clock circuitry has been significantly enhanced and can now interface to a GPSDO delivering a 0-3v output without any interface circuit. It is capable of operating as a clock master to other RSPduos or RSP2s and will also operate as a clock slave when using a RSP2 as a clock master. In tests, we find that we can ‘hot swap’ between an internal and external clock without hanging the MSi2500, which is not possible on the RSP2.

Finally, like the RSP1A, the RSPduo also features 14bit ADC performance for sample rates of 6 MHz or below.

The simultaneous use of two independent tuners (albeit with reduced bandwidth) makes certain key usage scenarios possible:

1. Simultaneous monitoring of two widely spaced bands – e.g. 40m and 2m
2. Mixing and matching applications simultaneously – e.g. ADS-B and ATC scanning
3. Phase and time coherent demodulation of two receivers

Scenario 3 is very difficult to achieve with two separate USB devices because of the uncertainty of USB latency. So whilst it is possible to phase lock the clocks of multiple RSP2s, the uncertainty of USB latency meant that the timing of the two units would not be aligned when it came to demodulation. The only way to overcome this was to apply timing correlation in software which requires a 'training sequence' to be simultaneously applied to both devices. The RSPduo overcomes this limitation because all traffic goes through a single USB interface, it opens up the possibility of the development of various types of diversity demodulation such as: spatial, frequency and polarization which can bring huge benefits in terms of improved performance.

Operating with two tuners simultaneously

Whilst the tuners can be controlled completely independently in terms of gain and frequency, there is one common factor that cannot (indeed should not) be separated and that is the ADC sample rate.

Because of this, whichever tuner is set up first will dictate the sample rate of the second receive path. To make it clear, we designate the first receiver to be set up as the 'Master' and the second one the 'Slave'.

In low IF mode, there will only be two sample rates available:

1. 6 MHz – This will deliver 14 bit ADC resolution, but with slightly inferior anti-aliasing performance at the widest bandwidth
2. 8 MHz – This will deliver 12 bit ADC resolution, but with slightly better anti-aliasing performance at the widest bandwidth

6 MHz operation should result in a slightly lower CPU load.

A new API concept

The problem with the existing API concept is that when starting the application with embedded API, it is simply not possible to determine in advance what is available in terms of receivers. We have therefore completely changed the concept behind the API.

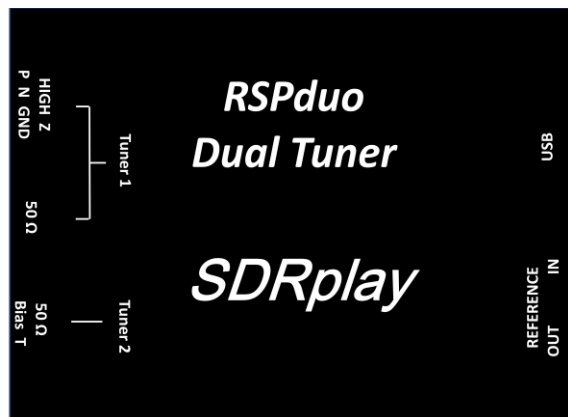
With the current API approach, if one application has already set up a SDRduo receiver as the ADC 'Master' and another application attempts to set up the second receiver in a way that is not possible, then the application will error and (most probably) crash out. Instead of the API either being embedded within the application (such as with SDRuno) or a separate DLL (as with HDSDR and SDR Console), it is therefore necessary to operate the API as a background 'service' within Windows and a 'daemon' under Linux and Mac OS. In this way, the service constantly monitors what is available in terms of receiver hardware and communicates this to the application at startup. That way, it is possible to avoid the application attempting to configure a tuner on startup into a mode that is simply not possible because another tuner is already designated as the 'Master' by a different application.

The matrix of available hardware options (for a single RSPduo) to the various applications will be as follows:

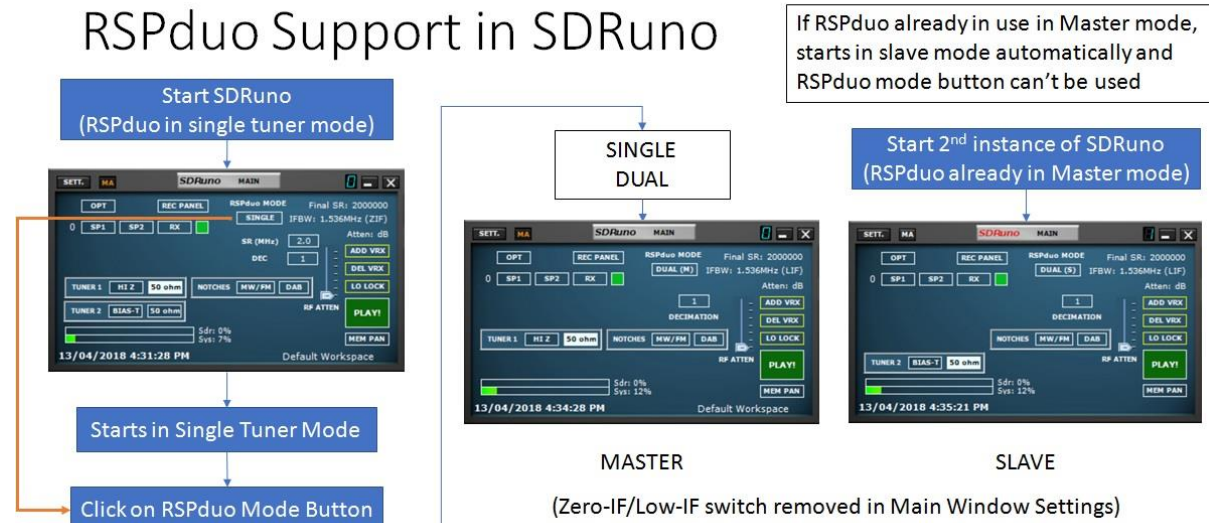
Operating mode	Tuner 1 Available RF ports	Tuner 2 Available RF ports	Master sample rate	Slave Sample rate	Tuner 1 IF Bandwidth	Tuner 2 IF Bandwidth	Tuner 1 frequency range	Tuner 2 frequency range
Single receiver (Tuner 1)	50 Ω /Hi-Z	Not available	2-10 MHz	N/A	200 kHz – 8 MHz	N/A	1 kHz – 2 GHz	N/A
Single receiver (Tuner 2)	Not available	50 Ω with Bias T	2-10 MHz	N/A	N/A	200 kHz – 8 MHz	N/A	1 kHz – 2 GHz
Dual tuner (tuner 1 master)	50 Ω /Hi-Z	50 Ω with Bias T	6/8 MHz	Determined by Tuner 1	200 kHz – 1.536 MHz	200 kHz – 1.536 MHz	1 kHz – 2 GHz	1 kHz – 2 GHz
Dual tuner (tuner 2 master)	50 Ω /Hi-Z	50 Ω with Bias T	6/8 MHz	Determined by Tuner 2	200 kHz – 1.536 MHz	200 kHz – 1.536 MHz	1 kHz – 2 GHz	1 kHz – 2 GHz

For the avoidance of doubt, when the service based API operating correctly, when operating a single instance of SDRuno, the RSPduo will operate in exactly the same way as the RSP2 if single tuner mode is selected from the main panel. Selecting either the 50 Ω or the Hi-Z port, associate with Tuner 1 will enable Tuner 1 and selecting the 50 Ω associated with Tuner 2, will enable Tuner 2. If dual tuner mode is selected from within main panel of the first instance of SDRuno, then a second instance of SDRuno will automatically designate the remaining tuner as the 'Slave' tuner.

The case graphic is shown below:



The following diagram illustrates how SDRuno will work with the RSPduo:



When first starting SDRuno, if more than one RSP is plugging into the PC, a device selection panel will appear allowing the user to select which RSP they wish to use.

Assuming no other application is already using one of the tuners in the RSPduo, the RSPduo will always start in single tuner mode where all device ports will be displayed on the main panel as shown in the diagram above.

In single tuner mode, the RSPduo will operate in a very similar way to the RSP2 and can be configured for ZIF mode with sample rates up to 10 MHz or LIF mode with sample rates of up to 2MHz. Simply selecting a particular RF port will enable the relevant tuner.

To switch to dual tuner mode first decide which tuner you wish to designate as the ADC clock 'Master' and select the appropriate RF port for that tuner.

Secondly switch the device from single tuner to dual tuner mode using the mode selection menu on the Main panel. Whichever tuner is operating when the device switched from single tuner mode to dual tuner mode will become the Master tuner.

Finally open a new instance of SDRuno. If you have more than one RSP plugged in, the device selection panel will appear and on the device list with will see:- RSPduo: [Serial Number] (Slave) along with whatever other RSPs you have connected.

Once you have individual instances of SDRuno running for both the Master and the Slave, you will first need to start the stream for the Master (press PLAY) before you can start the stream for the Slave. This is necessary to ensure that the ADC is configured for the correct sample rate, but once done, it is possible to arbitrarily start or stop the stream on either the Master or the Slave without affecting the other instance.

An example of dual tuner operation is shown below:



You will be able to save the workspace for each instance.

Other things to note are:

1. It is now possible to perform band framing in LIF mode.
2. The CPU load has been lowered due to more efficient data handling within the API
3. Frequency calibration can only be performed in the Master device in dual tuner mode. The RSPduo only has one reference clock
4. The 50Ω ports work well down to 1 kHz (unless the Bias T is enabled on Tuner 2) and are the preferred ports for use at HF and onwards up to 2GHz
5. The Hi-Z port is now an auxiliary port. It will give slightly worse performance above 2 MHz than the 50Ω ports (it will tend show more spurious responses as a result of higher order mixing products as it does not have the same level of RF pre-selection as the 50Ω ports), but because it has a lower noise floor it is likely to give slightly better performance below 2MHz.